

I Всеукраїнська науково-практична конференція

# **«Системні науки та інформатика»**

**Збірник доповідей**

22–29 листопада 2022 року

Київ, Україна

**Системні науки та інформатика:** збірник доповідей I науково-практичної конференції «Системні науки та інформатика», 22–29 листопада 2022 року, Київ. – К., НН ІПСА КПІ ім. Ігоря Сікорського, 2022. – 490 с.

© Навчально-науковий Інститут прикладного системного аналізу КПІ ім. Ігоря Сікорського, 2022.

Співголови програмного комітету конференції:

Касьянов П.О.	Романенко В.Д.	Панкратова Н.Д.
---------------	----------------	-----------------

Члени програмного комітету конференції:

Бідюк П.І.	Кисельов Г.Д.	Положаєнко С.А.
Гавриленко В.В.	Корабльов М.М.	Савченко І.О.
Гожий О.П.	Литвиненко В.І.	Синєглазов В.М.
Данилов В.Я.	Мухін В.Є.	Тимошук О.Л.
Єфремов К.В.	Петренко А.І.	Цюцюра С.В.
Калініна І.О.	Пишногрєв І.О.	Чумаченко О.І.

Організаційний комітет конференції:

Пишногрєв І.О. – голова	Савченко І.О.
Левенчук Л.Б.	Кисельов Г.Д.

Верстка збірника: Савченко І.О.

# ЗМІСТ

## Секція 01

### Системний аналіз і управління

<b>Порівняльний аналіз моделей генерації рекомендацій на основі методів машинного навчання на графах</b>	<b>9</b>
<i>Андросов Д.В., Недашківська Н.І.</i>	
<b>Дифузійні рівняння з дробовими похідними: чисельні методи розв'язування задачі Коші та застосування</b>	<b>14</b>
<i>Бондаренко В.Г., Собко Т.О., Подколзін Г.Б.</i>	
<b>Оцінка ADME-Tox властивостей молекул</b>	<b>21</b>
<i>Возняк В.З., Островський З.Ю., Недашківська Н.І.</i>	
<b>Методи нестатичного прогнозування для відновлення зображень</b>	<b>30</b>
<i>Клименко І.О., Подколзін Г.Б., Тимощук О.Л.</i>	
<b>Побудова точних довірчих інтервалів для параметрів деяких розподілів та порівняння з наближеними</b>	<b>36</b>
<i>Корабльов М.М., Каніовська І.Ю.</i>	
<b>Побудова S-кривих і паттернів S-кривих для швидкого аналізу дифузії тенденцій та еволюції інновацій нелінійних систем</b>	<b>40</b>
<i>Лопатін О.К.</i>	
<b>Алгоритми надання рекомендацій на основі наборів даних транзакцій</b>	<b>49</b>
<i>Недашківська Н.І., Деньгуб Д., Лавріненко В.</i>	
<b>Метод рефінансування в управлінні кредитним ризиком іпотечного портфеля комерційного банку</b>	<b>57</b>
<i>Півень О.К., Стулей В.А.</i>	
<b>Системний аналіз фінансових ризиків</b>	<b>62</b>
<i>Саркісян А.О., Бідюк П.І.</i>	
<b>Моделювання найкращої ціни мобільного додатку за поведінкою користувача</b>	<b>71</b>
<i>Семиволос О.Р., Каніовська І.Ю.</i>	
<b>Генерація зображень емоцій людини з використанням генеративних змагальних мереж</b>	<b>74</b>
<i>Столяр М.В., Каніовська І.Ю.</i>	
<b>Системний підхід до процесу керування доступами в об'єктах критичної інфраструктури в умовах воєнного стану</b>	<b>78</b>
<i>Тимощук О.Л., Пастушок О.О.</i>	
<b>Аналіз гетероскедастичних процесів та фінансового ризику</b>	<b>82</b>
<i>Федоренко А.П., Бідюк П.І.</i>	
<b>Спеціальні функції матричного аргументу</b>	<b>90</b>
<i>Шутяк Д.О., Подколзін Г.Б.</i>	

---

## Секція 02

### Системний аналіз фінансового ринку

---

<b>Модифікований метод «актор-критик» для фондового ринку як частина ситуаційно-аналітичної платформи</b>	<b>96</b>
<i>Батейко Е.А., Касьянов П.О.</i>	
<b>Моделі та методи організації розподілених баз даних з динамічною структурою</b>	<b>102</b>
<i>Васильченко І.В., Мухін В.Є.</i>	
<b>Оцінювання ринкових ризиків та порівняння різних методів</b>	<b>107</b>
<i>Воловоденко Т.О., Тимошук О.Л., Бідюк П.І.</i>	
<b>Системний аналіз індексу акцій з використанням штучних нейронних мереж</b>	<b>111</b>
<i>Данилов В.Я., Деменкова В.В.</i>	
<b>Системний підхід у навчанні математиці</b>	<b>119</b>
<i>Дроздова А.А., Данилов В.Я.</i>	
<b>Прогнозування операційних ризиків підприємства (економетричний аналіз)</b>	<b>124</b>
<i>Д'яченко А.С., Бідюк П.І.</i>	
<b>Аналіз ринкового ризику ймовірно-статистичними методами</b>	<b>130</b>
<i>Затірка В.В., Бідюк П.І.</i>	
<b>Моделювання системи прогнозування ефективності рекламних носіїв з використанням машинного навчання</b>	<b>135</b>
<i>Каніовська І.Ю., Бойко П.О.</i>	
<b>Аналіз розповсюдження COVID-19 на території України з використанням модифікованої SEIRD-моделі</b>	<b>140</b>
<i>Клименко А.І., Подколзін Г.Б.</i>	
<b>Застосування фрактальних моделей необмеженої варіації до аналізу часових рядів</b>	<b>147</b>
<i>Кравченко А.А., Бондаренко В.Г.</i>	
<b>Порівняльний аналіз методів машинного навчання для прогнозування показників смертності від забруднення повітря</b>	<b>152</b>
<i>Кулина А.П., Данилов В.Я.</i>	
<b>Методи оцінки і прогнозування фінансових ризиків для банківського сектора</b>	<b>157</b>
<i>Мельник І.А., Гуськова В.Г.</i>	
<b>Аналіз процесів та прийняття рішень на ринку криптовалют</b>	<b>162</b>
<i>Мельников А.А., Бідюк П.І.</i>	
<b>Моделі і прогнози волатильності фінансових процесів з використанням ймовірно-статистичних методів</b>	<b>170</b>
<i>Морозов Р.Д., Бідюк П.І.</i>	
<b>Симуляція Монте Карло систем керування запасами</b>	<b>175</b>
<i>Наконечна В.А., Тимошук О.Л.</i>	
<b>Інтегрування пуассонівських випадкових мір та знаходження їх асимптотик та оцінка параметрів пуассонівських випадкових полів</b>	<b>181</b>
<i>Пилипенко А.Ю., Горбенко Н.О.</i>	

---



<b>Дослідження стохастичних систем за частковим спостереженням</b>	<b>186</b>
<i>Пилипенко А.Ю., Ковальчук О.О.</i>	
<b>Виявлення шахрайських операцій методами машинного навчання</b>	<b>192</b>
<i>Позняк Д.І., Кузнєцова Н.В.</i>	
<b>Система підтримки прийняття рішень для аналізу розвитку фінансових процесів</b>	<b>198</b>
<i>Радіо О.В., Бідюк П.І.</i>	
<b>Методи аналізу та прогнозування стану підприємства за допомогою регресійних моделей</b>	<b>204</b>
<i>Сандига Я.С., Бідюк П.І.</i>	
<b>Інтелектуальна система прогнозування нелінійних нестационарних процесів в економіці</b>	<b>209</b>
<i>Селін Ю.М., Жук В.М.</i>	
<b>Теоретико-ігровий підхід для трейдингових стратегій на крипторинку</b>	<b>214</b>
<i>Семіконь Я.В.</i>	
<b>Системний підхід до аналізу кредитного та ринкового ризиків</b>	<b>220</b>
<i>Смиковська Д.В., Бідюк П.І.</i>	
<b>Методи побудови прогнозу фінансових даних. Аналіз та порівняння</b>	<b>226</b>
<i>Собкович В.С., Тимощук О.Л.</i>	
<b>Системний підхід до аналізу фінансових ризиків</b>	<b>231</b>
<i>Танчук М.</i>	
<b>Прогнозування індексу акцій на основі хмарних обчислювальних сервісів</b>	<b>237</b>
<i>Тараненко О.С., Бідюк П.І.</i>	
<b>Інструменти аналізу проектних ризиків</b>	<b>242</b>
<i>Тимощук О.Л., Гоголя О.О.</i>	
<b>Застосування теорії ігор при банкрутстві організації</b>	<b>246</b>
<i>Тіщенко А.Є., Барановська Л.В.</i>	
<b>Методи і механізми управління і підтримки комплексом управління складом</b>	<b>251</b>
<i>Торліна Н.М., Мухін В.Є.</i>	
<b>Система підтримки прийняття рішень для короткострокового прогнозування нелінійних нестационарних процесів в економіці та фінансах</b>	<b>257</b>
<i>Харченко Р.А., Бідюк П.І.</i>	
<b>Аналіз впливу військового конфлікту на соціально-економічні процеси в Україні</b>	<b>264</b>
<i>Худіков П.В., Бідюк П.І.</i>	
<b>Системний підхід щодо проектування архітектури програмного забезпечення</b>	<b>269</b>
<i>Шахворостова В.Д., Тимощук О.Л.</i>	
<b>Моделювання залежності факторів ризику за допомогою копул</b>	<b>273</b>
<i>Шепель І.О., Бідюк П.І.</i>	
<b>Стохастична модель в біології та вірусології</b>	<b>278</b>
<i>Яблуновський О.В., Мальцев А.Ю.</i>	
<b>Методи аналізу та прогнозування фінансового стану корпорації на прикладі компанії “Philip Morris International”</b>	<b>283</b>
<i>Ярмола А.О., Зайченко О.Ю.</i>	

---

<b>Порівняльний аналіз методів короткострокового прогнозування</b>	<b>291</b>
<i>Ярошенко В.О., Бідюк П.І.</i>	

---

### **Секція 03**

Інтелектуальні сервіс-орієнтовані розподілені обчислювання

---

<b>Застосування мікросервісного та мікро-фронтендного підходів для задачі створення віртуального кабінету лікаря та пацієнта</b>	<b>297</b>
<i>Безносик О.Ю., Стефура О.Я.</i>	
<b>Підходи до оптимізації запитів до реляційних баз даних з використанням машинного навчання</b>	<b>304</b>
<i>Булах Б.В., Загородній Д.О.</i>	
<b>Моделі нечіткого прогнозування в вимірюваннях</b>	<b>311</b>
<i>Грищенко О.Ю., Рогоза В.С.</i>	
<b>Комбінована архітектура для мобільних застосунків</b>	<b>315</b>
<i>Зарічний Я.С., Гіоргізова-Гай В.Ш., Яременко В.С.</i>	
<b>Представлення інтероперабельних сервісів для роботи з медичними даними в мікросервісній архітектурі</b>	<b>323</b>
<i>Кандель К.В., Кулик В.О., Мельник О.Р., Письменний І.О.</i>	
<b>Методи розпізнавання текстів та пошуку ключових слів для автоматичного реферування текстів</b>	<b>331</b>
<i>Кузнєцов О. А., Кисельов Г.Д.</i>	
<b>Підтримка прийняття рішень за допомогою засобів Decision Intelligence</b>	<b>336</b>
<i>Макаров І.В., Кисельов Г.Д.</i>	
<b>Оцінювання і прогнозування стану пацієнтів та поліпшення ефективності вимірних даних та підтримки прийняття рішень при аналізі неточних та неповних вимірних даних</b>	<b>342</b>
<i>Медвідь В.А., Харченко К.В.</i>	
<b>Перевірка ефективності використання машинного навчання у автоматизації регресивного тестування</b>	<b>349</b>
<i>Михайловин Р.Г., Булах Б.В.</i>	
<b>Система моніторингу в електронній медичній системі</b>	<b>355</b>
<i>Музика О.А., Булах Б.В.</i>	
<b>Організація збору діагностичних показників в системах крайових обчисленнях</b>	<b>360</b>
<i>Охота Д.Л., Булах Б.В.</i>	
<b>Вирішення задачі графемно-фонемного перетворення текстів</b>	<b>368</b>
<i>Самвелян А.Р., Кисельов Г.Д.</i>	
<b>Нейронна мережа для діагностування захворювань системи дихання на базі сигналів ЕКГ</b>	<b>371</b>
<i>Скріпченко М.О., Кислий Р.В.</i>	
<b>Алгоритм розрахунку точок для захвату об'єкта маніпулятором на основі апроксимації 3D моделі об'єкта просторовими примітивами</b>	<b>375</b>
<i>Харченко К.В., Вовк І.С.</i>	
<b>Розробка додатку для моделювання та аналізу електричних кіл</b>	<b>377</b>
<i>Харченко К.В., Тараненко А.В.</i>	

---

---

## Секція 04

### Системи і методи штучного інтелекту

---

<b>Система відстеження погляду в реальному часі для взаємодії з комп'ютером на основі методів глибокого навчання</b>	<b>382</b>
<i>Гончарук О.П., Недашківська Н.І.</i>	
<b>Визначення емоційного забарвлення голосу за допомогою технік Deep Learning у реальному часі</b>	<b>387</b>
<i>Загарницький Д.В., Шаповал Н.В.</i>	
<b>Прогноз траєкторії учасників дорожнього руху з використанням кількох камер</b>	<b>393</b>
<i>Загній Є.В., Данилов В.Я.</i>	
<b>Модель для розпізнавання військової техніки та військовослужбовців із камери БПЛА в реальному часі для систем з обмеженими обчислювальними ресурсами</b>	<b>397</b>
<i>Іванов С.І., Шубенкова І.А.</i>	
<b>Рекомендаційна система на основі оглядів користувачів і продуктів</b>	<b>404</b>
<i>Кравченко О.В., Недашківська Н.І.</i>	
<b>Дослідження гібридних мереж глибокого навчання в задачах прогнозування у фінансовій сфері</b>	<b>410</b>
<i>Кузьменко О.В., Зайченко Ю.П.</i>	
<b>Розпізнавання пожежі за допомогою згорткових нейронних мереж</b>	<b>416</b>
<i>Маркін І.Д., Шаповал Н.В.</i>	
<b>Методи прогнозування індексу акцій на основі механізмів штучного інтелекту</b>	<b>421</b>
<i>Міщенко Д.В., Мухін В.Є.</i>	
<b>Зоровий трансформер для задачі класифікації раку шкіри</b>	<b>427</b>
<i>Нікітін В.О., Шаповал Н.В.</i>	
<b>Алгоритми сортування з використанням нейронних мереж</b>	<b>431</b>
<i>Подколзін Г.Б., Павлюк В.</i>	
<b>Використання нейронних мереж та Gradient Boosting для оптимізації логістичних витрат</b>	<b>436</b>
<i>Скоробогатов С.Ю., Зайченко О.Ю.</i>	
<b>Методи багатofакторної автентифікації до веб додатків за допомогою штучного інтелекту та технології блокчейн</b>	<b>441</b>
<i>Славінский В.О., Гуськова В.Г.</i>	
<b>Агентна архітектура механізмів консенсусу в блокчейні</b>	<b>447</b>
<i>Тимощук О.Л., Лисов Б.С.</i>	
<b>Виявлення аномалій методами інтелектуального аналізу даних</b>	<b>452</b>
<i>Хомич О.Р., Недашківська Н.І.</i>	
<b>Класифікація вакансій відповідно до класифікатора професій методами штучного інтелекту</b>	<b>456</b>
<i>Цимбал Ю.О., Недашківська Н.І.</i>	
<b>Ідентифікація психічних захворювань методами обробки природної мови на основі дописів користувачів у соцмережах</b>	<b>463</b>
<i>Цупрун І.Ю., Стусь О.В.</i>	

---

---

<b>Гібридна нейронна мережа для виявлення і відстеження людини у відеопотоці</b>	<b>469</b>
<i>Шаповал Н.В., Русакова Л.О.</i>	
<b>Згорткова нейронна мережа для задачі класифікації захворювання листя рослин</b>	<b>474</b>
<i>Шевченко В.В., Шаповал Н.В.</i>	
<b>Розробка моделей штучного інтелекту для аналізу та формування інвестиційного портфелю</b>	<b>476</b>
<i>Шевчук О.С., Кузнєцова Н.В.</i>	
<b>Making decisions in the service business sphere using artificial intelligence</b>	<b>482</b>
<i>I.A. Shubenkova., M.S. Marenych</i>	
<b>Використання моделі BI-LSTM для підвищення точності прогнозування курсу криптовалют</b>	<b>487</b>
<i>Яцько Я.В., Терентьєв О.М.</i>	

---

# ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ ГЕНЕРАЦІЇ РЕКОМЕНДАЦІЙ НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ НА ГРАФАХ

Андросов Д.В.<sup>1</sup>, Недашківська Н.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> androsovdmitry80@gmail.com, <sup>2</sup> n.nedashkivska@gmail.com [0000-0002-8277-3095]

**Проблема видачі релевантних рекомендацій завжди була та є ключовою для клієнт-орієнтованих бізнесів. Протягом останніх років для вирішення даної проблеми все активніше використовують методи статистичного навчання. Такий підхід полягає у видачі рекомендацій заснованих на історії взаємодії користувача з бізнесом. У роботі створено модель глибинного навчання, що використовує апарат графових нейронних мереж для вирішення задачі видачі релевантних рекомендацій. Отримано оцінки точності різних алгоритмів машинного навчання для видачі рекомендацій.**

**Ключові слова:** рекомендаційна система, графова нейронна мережа, змінні найменші квадрати, динамічні графи, вектори вкладень.

## 1. ВСТУП

Термін «рекомендаційна система» хоч і є інтуїтивно зрозумілим, проте для подальшої роботи необхідно його формалізувати. Посилаючись на [1], пропонується формалізувати дане поняття наступним чином: рекомендаційна система – комплекс статистичних моделей, що відбирає серед контенту лише найбільш релевантні позиції для користувача, опираючись на профіль користувача.

Рекомендаційні системи (РС) за ступенем персоналізації поділяються на:

1. Неперсональні – найпростіші системи, що фільтрують контент в пошуку найбільш релевантного у середньому (себто популярного).
2. Напівперсоналізовані РС – оперують не інтересами конкретного суб'єкту, а орієнтуються на групу, до якої цей суб'єкт належить, для генерації рекомендацій.
3. Персоналізовані РС – рекомендують те, що релевантно кінцевому користувачу.

В залежності від типу поставленої задачі використовують той, чи інший вид РС.

В контексті даного дослідження поняття РС ототожнюється з поняттям персоналізованих РС. Персоналізовані РС є найпопулярнішим рішенням для бізнесів найрізноманітнішого масштабу та спеціалізації, оскільки вирішує дві ключові задачі генерації рекомендацій. Насамперед, для бізнесу видача постійно релевантного контенту користувачу означає лояльність користувача та збільшує прибуток у довгостроковій перспективі. Також потужна рекомендаційна система завжди буде надавати користувачу лише релевантні пропозиції та стимулювати його спробувати новий контент.

## 2. ПРОБЛЕМА ІНТЕРПРЕТАЦІЇ ДАНИХ У СИСТЕМАХ БЕЗ ЗВОРОТНЬОГО ЗВ'ЯЗКУ

При вирішенні задач видачі рекомендацій перед архітектором системи видачі РС постає низка проблем. Першою з них є оцінка контенту користувачем: що з цим контентом ще не взаємодівав. Тут і надалі для оцінки контенту пропонується використовувати поняття «рейтинг».

Рейтинг контенту може задаватися явно, через зворотній зв'язок, наприклад, на деяких онлайн маркетплейсах є можливість оцінити товар за N-бальною шкалою; або неявно, що значно ускладнює процес проектування системи, оскільки відсутній зворотній зв'язок від користувача, а є лише факт його взаємодії.

Для обох випадків одним з найпопулярніших рішень є застосування алгоритмів колаборативної фільтрації. Колаборативна фільтрація (КФ) – особливе сімейство алгоритмів РС, котрі використовують інформацію не тільки про контент, що рекомендується користувачу, а й інформацію взаємодії всіх користувачів з контентом. Основна ідея алгоритмів КФ полягає у неявному групуванні користувачів відносно контенту, з яким вони взаємодіяли з метою визначення рейтингів контенту, з котрим певний користувач з групи ще не взаємодівав.

Алгоритми КФ накладають додаткові умови та обмеження на вхідні дані – вони мають бути представлені у вигляді матриці рейтингів  $R = r_{ij} \quad i=1,N, j=1,M$ , де  $N$  – кількість користувачів,  $M$  – кількість контенту (товару). При цьому  $r_{ij} \in \mathbb{R}$  у випадку явних рейтингів та  $r_{ij} \in \{0,1\}$  у випадку наявності лише неявних рейтингів.

При такому підході кожного користувача  $u_i$  можна представити як вектор відомих рейтингів товарів (контенту). Тоді для пошуку схожих користувачів використовують різні метрики відстаней, наприклад, косинусну подібність, або коефіцієнт Отіаї [1]:

$$\cos u_i, u_k = \frac{\sum_{j=1}^M r_{ij} r_{kj}}{\sqrt{\sum_{j=1}^M r_{ij}^2} \sqrt{\sum_{j=1}^M r_{kj}^2}}$$

Тоді рейтинг невідомою товару розраховується зваженою сумою:

$$r_{ij} = \sum_{p \in J} r_{pj} * \cos u_p, u_i$$

Більш популярні наразі є машини факторизації, наприклад алгоритм змінних найменших квадратів, описаний у [2]. Суть машини факторизації полягає у представленні матриці рейтингів  $R$  у якості добутку матриці користувачів  $U$  та контенту  $I$ . У методі змінних найменших квадратів пропонується знаходити розв'язок даної задачі шляхом знаходження мінімуму суми квадратів похибки:

$$\min_{W,V} \sum_{i,j : r_{ij} > 0} (r_{ij} - w_i v_j^T)^2$$

Іншим популярним алгоритмом факторизації є алгоритм байєсівського персоналізованого ранжування. Його суть полягає в знаходженні таких параметрів матриць  $U$  та  $I$  що гарантували б існування відношення порядку для кожного з користувачів  $u_i$ , тобто:

$$\max_{\theta, u \in U} p \theta \succ_u$$

де  $\theta = [U, I]$ .

У [3] показано що задача вище еквівалентна наступній:

$$\max_{\theta} \ln \sigma \sum_{i \in U, (j,k) \in \mathcal{I}_u} r_{ij} - r_{ik} - \frac{1}{2\sigma^2} \theta^2$$

Дану задачу вже пропонується вирішити будь-яким градієнтним методом оптимізації, наприклад покоординатного спуску.

Методи колаборативної фільтрації є досить потужними та простими, що робить їх незамінним інструментом при проектуванні MVP версій рекомендаційних систем. Проте в силу обмеження на представлення вхідних даних, що унеможливорює використання додаткових даних з профілю користувача, а також, як зазначено в [4], неможливості використання вхідних даних без прив'язки до самої моделі, алгоритми КФ не можуть досягнути бажаної точності та інтерпретованості.

### 3. МАШИННЕ НАВЧАННЯ НА ГРАФАХ

Для можливості аналізу даних про взаємодію користувачів та товарів, відслідковування еволюції системи цих взаємодій запропоновано представити дані у вигляді динамічного графу.

Динамічним графом пропонується називати граф, що задається четвіркою  $G^{(t)}(V^{(t)}, E^{(t)}, w^{(t)}, f)$ , де  $V$  – вершини графу,  $E$  – ребра графу,  $w$  – ваги ребер та  $f: E \rightarrow w$ .

При цьому в якості атрибутів вершин в контексті даної задачі можна використовувати дані з профілю користувача, а ваги ребер є рейтингами товарів. Таким чином, матриця суміжності графу й буде матрицею рейтингів, що необхідна для побудови моделей КФ. Таким чином, графи – інтуїтивно зрозумілий та простий у інтерпретації спосіб представлення даних для РС.

Для обробки даних на графах пропонується застосувати моделі глибинного навчання на основі механізму уваги, що запропоновані у [5]. Мотивація використовувати даний підхід пояснюється ефективністю механізму уваги при обробці динамічних даних (послідовностей), та популярністю таких мереж у якості основи для рекомендаційних систем ([6], [7], [8]).

Кодування графових даних здійснюють з використанням методів вкладень графу, наприклад алгоритмом Node2Vec, описаним в [9]. Даний алгоритм отримує низку послідовностей відвіданих вершин, що отримані алгоритмом випадкового блукання по вхідному графу взаємодій, після чого застосовується алгоритм Word2Vec [10] для генерації числових векторів малого розміру.

### 4. ПОСТАНОВКА ТА АНАЛІЗ ЕКСПЕРИМЕНТУ

В рамках дослідження було прийнято рішення побудувати моделі на основі алгоритмів послідовних найменших квадратів (ALS), байєсівського персоналізованого ранжування (BPR), довгої короткострокової пам'яті (LSTM) та в подальшому запропонувати модель генерації рекомендацій на основі динамічних вкладень графів та механізму уваги (Attention network) для порівняння та вибору найкращого методу по метрикам точності та ранжування.

У якості даних був обраний набір даних e-commerce платформи, що задіяна у роздрібній торгівлі побутовою електронікою (смартфони, ноутбуки та ПК, телевізори, тощо). Причому структура набору даних однаково підходить для тестування моделей видачі рекомендацій на основі послідовностей виконаних дій або транзакцій користувача та в загальному, зокрема так як дану задачу вирішують методи колаборативної фільтрації.

Точні наявні поля набору даних наведені в Таблиці 1. Як видно, для задачі рекомендації наступного товару слід згрупувати дані по ідентифікатору сесії та подавати на вхід до мережі послідовність товарів, над якими виконувалася взаємодія користувача. Для вирішення задачі



рекомендацій у загальному, додаткові маніпуляції та групування даних не є необхідними операціями.

Таблиця 1. Структура набору даних

Назва поля	Тип поля	Опис поля
user_id	Цілочисельний додатній	Ідентифікатор користувача в БД
product_id	Цілочисельний додатній	Ідентифікатор товару в БД
event_type	Символьний	Тип взаємодії (перегляд, покупка, додавання до кошику)
session_id	Base64	Ідентифікатор сесії користувача
category_id	Цілочисельний додатній	Ідентифікатор категорії товару

Спочатку пропонується проаналізувати задачу видачі рекомендацій наступної пропозиції.

Архітектура рекомендаційної системи однакова та складається з двох модулів:

1. Модуль-запит (query model). Даний модуль виконує задачу генерації вкладення наступного товару на основі раніше переглянутих.

2. Модуль-кандидат (candidate model). Даний модуль розраховує вкладення всіх наявних товарів, та обирає по метриці відстані (наприклад, по Евклідовій) найближчі товари до вкладення, згенерованого модулем-запитом.

У якості модулів-запитів для задачі прогнозування наступного товару для користувача були обрані наступні моделі:

1. Модель генерації рекомендацій на основі графових мереж з поєднанням механізму self-attention та багатошарового перцептронну (DGE-ANN).

2. Модель генерації рекомендацій на основі графових мереж з поєднанням механізму self-attention та глибокої перехресної мережі (DGE-ANN-DCN).

Для оцінки результатів у якості референтної моделі пропонується застосувати відому модель видачі рекомендацій на основі рекурентних нейронних мереж, а саме на основі мережі довгої короткострокової пам'яті (LSTM) [1].

Навчання мережі пропонується здійснювати з використанням функції перехресної ентропії.

Задачу видачі рекомендацій у загальному пропонується вирішувати з використанням двох референтних моделей: машини матричної факторизації на основі алгоритму байєсівського персоналізованого ранжування (BPR) [3, 4] та машини змінних найменших квадратів (ALS) [2]. Запропоновано модель кандидат – це модель генерації рекомендацій на основі графових мереж з використанням архітектури глибокої перехресної мережі.

У якості моделі конструювання графу знань була заздалегідь обрана модель Node2Vec.

Результати навчання для задачі рекомендації наступної пропозиції приведені в Табл. 2. Результати для задачі рекомендацій у загальному наведено в Табл. 3.

Таблиця 2. Оцінки точності моделей DGE-ANN, DGE-ANN-DCN і LSTM за множиною критеріїв

Тип	MAP@1	MAP@10	MAP@100	NDCG@1	NDCG@10	NDCG@100
DGE-ANN	0.0800	0.1360	0.1449	0.0799	0.1682	0.2188
DGE-ANN-DCN	0.1018	0.1794	0.1876	0.0876	0.2199	0.2635
LSTM	0.0383	0.0552	0.0601	0.0384	0.0701	0.0986

Таблиця 3. Оцінки точності моделей ALS, BPR і DGE-DCN за множиною критеріїв

Тип	MAP@1	MAP@10	MAP@100	NDCG@1	NDCG@10	NDCG@100
ALS	0.0103	0.0144	0.0157	0.0799	0.0182	0.0253
BPR	0.0201	0.0236	0.0063	0.0662	0.0292	0.0095
DGE-DCN	0.1457	0.1750	0.1771	0.1459	0.1878	0.1995

Як видно з таблиці 2, найкраща модель за всіма параметрами – пропонується модель DGE-ANN-DCN, тобто модель генерації рекомендацій на основі графових мереж з поєднанням механізму self-attention та глибокої перехресної мережі.

В свою чергу у Таблиці 3 наведені результати навчання моделей при розв’язанні задачі видачі рекомендацій без прив’язки до послідовності дій.

З таблиці 3 видно, що пропонується модель DGE-DCN на порядок краща за всіма метриками, ніж моделі колаборативної фільтрації, оскільки останні обмежені за своїм функціоналом та використовують лише дві характеристики – ідентифікатор користувача та товару.

## 5. ВИСНОВКИ

Експериментально встановлено, що моделі глибокого навчання на основі графових нейронних мереж мають кращі показники метрик ранжування в порівнянні з іншими алгоритмами, які такі дані не використовують. Результати дослідження довели результативність практичного застосування моделей глибокого навчання на основі графових нейронних мереж при вирішенні практичних задач генерації рекомендацій.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Фальк К. Рекомендательные системы на практике / пер. с англ. Д. М. Павлова. – Москва : ДМК Пресс, 2020. 448 с.
2. Collaborative filtering. Apache Spark Documentation URL: <https://spark.apache.org/docs/2.2.0/ml-collaborative-filtering.html> (дата звернення: 18.11.2022).
3. Collaborative Filtering for Implicit Feedback Datasets. URL: <http://yifanhu.net/PUB/cf.pdf> (дата звернення: 20.11.2022).
4. Bayesian Personalized Ranking from Implicit Feedback. URL: <https://arxiv.org/pdf/1205.2618> (дата звернення: 19.11.2022).
5. Attention Is All You Need. URL: <https://arxiv.org/abs/1706.03762> (дата звернення: 18.11.2022).
6. Sequential Recommender System based on Hierarchical Attention Network. URL: <https://www.ijcai.org/proceedings/2018/0546.pdf> (дата звернення: 19.11.2022).
7. Attention-Based Recommendation On Graphs. URL: <https://arxiv.org/pdf/2201.05499> (дата звернення: 19.11.2022).
8. Sequential Recommendation via Stochastic Self-Attention. URL: <https://arxiv.org/pdf/2201.06035> (дата звернення: 21.11.2022).
9. node2vec: Scalable Feature Learning for Networks. URL: <https://arxiv.org/abs/1607.00653> (дата звернення: 20.11.2022).
10. Efficient Estimation of Word Representations in Vector Space. URL: <https://arxiv.org/abs/1301.3781> (дата звернення: 20.11.2022).

# **ДИFUЗІЙНІ РІВНЯННЯ З ДРОБОВИМИ ПОХІДНИМИ: ЧИСЕЛЬНІ МЕТОДИ РОЗВ'ЯЗУВАННЯ ЗАДАЧІ КОШІ ТА ЗАСТОСУВАННЯ**

Бондаренко В.Г.<sup>1</sup>, Собко Т.О.<sup>2</sup>, Подколзін Г.Б.<sup>1</sup>

<sup>1,2</sup> Національний технічний університет України  
«Київський політехнічний інститут ім. Ігоря Сікорського», Київ, Україна

<sup>2</sup>sobko65@gmail.com

Дробові похідні мають досить широке застосування в різних галузях науки в тому числі стають в пригоді для опису процесу зміни в'язкоеластичного матеріалу під впливом зовнішніх факторів в залежності від часу, за допомогою них можна описати низку процесів, які досі описувались не точно. Дана робота допоможе читачу зрозуміти застосування, а також методи розв'язання задачі Коші з диференційними рівняннями дробового порядку, а саме похідними Капуто, а також будуть запропоновані для ознайомлення два чисельні методи розв'язання таких рівнянь та дробової похідної в цілому. Після ознайомлення читач зможе застосувати отримані знання для моделювання власних процесів і побачити, як поводить себе похідна дробового порядку в залежності від її значення та обраного методу розв'язання.

**Ключові слова:** диференційні рівняння, похідна Капуто, дифузія, в'язкоеластичність, моделювання, субдифузія, дробове числення.

## **1. ВСТУП**

Протягом останніх десятиліть прикладні науки звернули увагу на ДЧ і перевірили, що воно є потужним інструментом для опису багатьох природних і штучних явищ, як, наприклад, ефекти пам'яті дальньої дії, якими нехтує класичне обчислення цілих чисел. Це спонукало до величезного розвитку в теоретичному, чисельному та прикладному аспектах, а також появи безлічі моделей і нових пропозицій.

Підвищений інтерес до диференціальних рівнянь дробового порядку обумовлено їх фізичною інтерпретацією. Показано, що перехід до похідної дробового порядку за часом дозволяє враховувати ефекти пам'яті системи. Це дозволяє отримувати нові результати, більш глибоко осмислити відомі та створити адекватні моделі досліджуваних явищ [1–3].

Насправді кількість літератури, присвяченої ДЧ та її застосуванням, величезна, але читачі стикаються з високою неоднорідністю і, в деяких випадках, з оманливою та неточною інформацією. Ця стаття має намір заповнити цю прогалину та надати читачам зрозумілий і систематичний розгляд основних аспектів і застосувань ДЧ.

## **2. ЗАСТОСУВАННЯ ТА МЕТОДИ РОЗВ'ЯЗУВАННЯ ДИFUЗІЙНИХ РІВНЯНЬ З ДРОБОВИМИ ПОХІДНИМИ**

### **2.1 Аномальна дифузія**

Диференційні рівняння з дробовими похідними використовуються для опису різних варіантів аномальної дифузії по фракталах. Суб- і супердифузія входять в клас аномальних

дифузійних процесів. Аномальна дифузія виникає в таких областях, як турбулентна дифузія в плазмі, міграція вологи в ґрунті, рух радону і природного газу в геологічних середовищах, транспорт заряду в аморфних напівпровідниках, дифузія в пористих матеріалах, «ковзна» дифузія по твердій поверхні і т.д.

Субдифузія – це випадкове блукання, при якому швидкість росту кореня середнього квадратного відхилення частинок з плином часу не залишається постійною, як при звичайній дифузії, а монотонно зменшується. Цей процес реалізується, наприклад, коли дифузор захоплюється пастками (дефектами, абсорбційними або хімічно-активними центрами), переходить в бічні, тупикові шляхи й на деякий час або назавжди віддаляється від процесу міграції. Ефекти уповільнення можливі та при наявності в системі статичного розладу, динамічного розладу, в'язкоеластичності і т.д. При субдифузії наявність пасток призводить до розбіжності середньої тривалості очікування стрибків, завдяки чому набувають дискретний характер в просторі, відбувається уповільнення процесу перенесення і в кінцевому підсумку міграція перестає залежати від часу.

Типовим прикладом субдифузії є процес дифузії радону в фрактально-пористому середовищі, який протікає повільніше, ніж звичайний режим дифузії, тому що ґрунт має складну топологію каналів між порами. Окрім субдифузії (затримка дифузії, дифузія по фракталах), можлива і супердифузія (прискорена, фрактальна дифузія).

Супердифузія здійснюється зі швидкостями, що значно перевищують класичну дифузію. Цей режим спостерігається, якщо в системі є полегшені шляхи (наприклад, тріщини) або є процеси випадкової або спрямованої адвекції (захоплення дифузанта потоками рідини). Прикладами супердифузії є турбулентні середовища (газ, рідина, плазма, хаотичні структури). У супердифузії частка в дискретні моменти часу робить стрибки довільної довжини, що характеризуються розбіжним коренем середньоквадратичного відхилення; міграція прискорюється з часом і може стати нескінченно великою; послідовні положення блукаючої частинки утворюють кластерну структуру, що представляє фрактальну множину, розмірність якої пов'язана з показником потужності.

Кінетика субдифузії описується розв'язками диференційних рівнянь з дробовими похідними по часовій змінній. Рівняння для 2-го закону Фіка, засноване на дробовій похідній за часом, має вигляд:

$$\frac{d^\gamma C}{dt^\gamma} = D \frac{d^2 C}{dx^2}$$

Це інтегрорізничеве рівняння вирівнювання концентрації в субдифузії при  $0 < \gamma < 1$  описує випадковий процес, який сповільнюється в часі – дифузію над фракталами. Дробова похідна в часі виникає при обліку нелокальності з плином часу.

Кінетика супердифузії описується розв'язками диференційних рівнянь з дробовими похідними по координаті:

$$\frac{\partial C(x, t)}{\partial t} = D \frac{\partial^\alpha C(x, t)}{\partial x^\alpha}, \quad 1 \leq \alpha \leq 2$$

$$\frac{\partial^\alpha}{\partial x^\alpha} = \frac{1}{2} (1 + \beta) \frac{\partial^\alpha}{\partial x^\alpha} + \frac{1}{2} (1 - \beta) \frac{\partial^\alpha}{\partial (-x)^\alpha}, \quad -1 \leq \beta \leq 1,$$

де  $C(x, t)$  характеризує концентрацію дифузанта;  $\frac{\partial^\alpha}{\partial x^\alpha}$  – оператор диференціювання дробового порядку  $\alpha$  по  $x$ ;  $D > 0$  – константа розмірності  $L^\alpha T^{-1}$  (узагальнений коефіцієнт дифузії);  $\beta$  – коефіцієнт асиметрії, що характеризує напрямок перенесення речовини при  $\alpha \rightarrow 1$ ;  $x$  і  $t$  – просторова та часова змінні.

В загальному вигляді рівняння дифузії має вигляд:

$$\frac{\partial^\gamma C(x, t)}{\partial t^\gamma} = D_f D_x^\alpha C(x, t),$$

де  $D_t^\gamma$  – частинна похідна по часу, а частинна похідна по координаті  $D_x^\alpha = \frac{1}{2} (1 + \beta \frac{\partial^\alpha}{\partial x^\alpha} + \frac{1}{2} (1 - \beta) \frac{\partial^\alpha}{\partial (-x)^\alpha})$ ,  $0 < \gamma \leq 2$ ,  $1 \leq \alpha \leq 2$ ,  $-1 \leq \beta \leq 1$ ,  $D_f$  – константний коефіцієнт дифузії.

В наведених рівняннях  $\alpha$  – дробовий порядок диференціювання по просторовій координаті;  $\beta$  – коефіцієнт асиметрії (скошеності), що характеризується напрямком перенесення речовини при  $\alpha \rightarrow 1$ ;  $\alpha = 2$  – класична дифузія,  $1 < \alpha < 2$  – супердифузія;  $\gamma$  – дробовий порядок диференціювання по часовій змінній,  $\gamma = 1$  – класична дифузія,  $0 < \gamma < 1$  – субдифузія,  $1 < \gamma < 2$  – супердифузія;  $\gamma = 2$  – хвильове рівняння.

## 2.2 Метод кінцевих різниць

Розглянемо задачу Коші для рівняння дробового порядку такого виду

$$\partial_{0t}^\alpha u = f(u, t), \quad (1)$$

$$u(0) = u_0, \quad (2)$$

де

$$\partial_{0t}^\alpha u = D_{0t}^\alpha u - \frac{u(0)}{\Gamma(1-\alpha) t^\alpha} = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{u_\tau(\tau) d\tau}{t-\tau}^\alpha$$

це регуляризована дробова похідна порядку  $\alpha$ ,  $0 < \alpha < 1$ ,  $D_{0t}^\alpha u = \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_0^t \frac{u_\tau(\tau) d\tau}{t-\tau}^\alpha$

є дробовою похідною Рімана-Ліувілля порядку  $\alpha$ ,  $0 < \alpha < 1$ ,  $f(u, t)$  – задана неперервна функція від  $t, u$ . Будемо вважати, що функція  $f(u, t)$  визначена на деякому прямокутнику  $D = \{0 \leq t \leq T, u - u_0 \leq U\}$  і також задовольняє умову Ліпшиця за змінною  $u$ . Тоді, як відомо, задача Коші (1, 2) має єдиний розв'язок (див. [2]).

Дискретний аналог дробової похідної записується у такому вигляді:

$$\Delta_{0t_j}^\alpha u = \frac{1}{\Gamma(2-\alpha)} \sum_{s=1}^j (t_{j-s+1}^{1-\alpha} - t_{j-s}^{1-\alpha}) u_{t,s}$$

Розглянемо крайову задачу в такій області  $Q_T \equiv \{x, t : 0 < x < l, 0 < t \leq T\}$

$$\partial_{0t}^\alpha u = \frac{\partial}{\partial x} k(x, t) \frac{\partial u}{\partial x} - q(x, t) u + f(x, t), \quad (3)$$

$$u(0, t) = u(l, t) = 0,$$

$$u(x, 0) = u_0(x),$$

$$0 < c_1 \leq k(x, t) \leq c_2, \quad q(x, t) \geq 0,$$

Без доведення наведемо умову стійкості [4–6]

$$\tau^\alpha \leq \frac{2-2^{1-\alpha} h^2}{2c_2 \Gamma(2-\alpha) (1-\sigma)}. \quad (4)$$

Умова стійкості (4) при  $\alpha \rightarrow 1$  переходить у відому для чисельних методів умову

$$\tau \leq \frac{h^2}{2c_2 (1-\sigma)}.$$

### 2.3 Метод декомпозиції

Перед тим, як переходити до ознайомлення з даним методом знадобляться деякі допоміжні теоретичні відомості, тому спершу наведемо основні з них.

$$N h(t) = R s, u = \int_{-\infty}^{\infty} e^{-st} h ut dy, s, u \in -\infty, \infty, \quad (5)$$

де  $N$  це  $N$ -перетворення функції  $h(t)$ , а  $s, u$  – це змінні цього перетворення.

**Теорема.** Якщо  $k \in \mathbb{Z}^+$ , де  $k - 1 \leq v < k$ . Тоді  $N$ -перетворення похідної Капуто функції  $\varphi t$  може бути записане у вигляді:

$$N^+ {}^C D_t^v \varphi t = \frac{s^v}{u^v} R s, u - \sum_{m=0}^{k-1} \frac{s^{v-m+1}}{u^{v-m}} D^m \varphi t \Big|_{t=0}. \quad (6)$$

Розглянемо загальне нелінійне диференціальне рівняння дробового порядку:

$${}^C D_t^v y t + L y t + F y t = g t, \quad (7)$$

де  $t > 0$  і  $0 \leq v \leq 1$ , а також разом з цим початкова умова  $y 0 = y_0$ ,  ${}^C D_t^v y t$  – похідна Капуто для  $y t$ ,  $L$  – лінійний диференціальний оператор, а  $F$  вказує на нелінійну частину. Також  $g t$  – це неоднорідна частина і  $y_0$  – визначена неперервна функція.

Після застосування теореми до (7) можемо зробити висновок, що:

$$N^+ y t = \frac{u^v}{s^v} \sum_{k=0}^{n-1} \frac{s^{v-k+1}}{u^{v-k}} D^k y t \Big|_{t=0} + \frac{u^v}{s^v} N^+ g t - \frac{u^v}{s^v} N^+ L y t + F y t. \quad (8)$$

Підставляючи початкову умову у рівняння (8) і взявши обернене перетворення  $N^{-1}$ , отримуємо

$$y t = H t - N^{-1} \frac{u^v}{s^v} N^+ L y t + F y t, \quad (9)$$

де  $H t$  утворюється з неоднорідної частини разом з початковою умовою.

Тепер припустимо, що розв'язок  $y t$  існує в такому вигляді

$$y t = \sum_{n=0}^{\infty} y_n(t), \quad (10)$$

$$F y t = \sum_{n=0}^{\infty} A_n(t), \quad (11)$$

де поліноми  $A_n(t)$  від  $y_0, y_1, \dots, y_n$  обчислюються за формулою

$$A_n = \frac{1}{n!} \frac{d^n}{d\lambda^n} \mathbb{F} \sum_{i=0}^n \lambda^i y_i \Big|_{\lambda=0}, n = 0, 1, 2, \dots \quad (12)$$

Підставивши (11) в (9) отримуємо

$$\sum_{n=0}^{\infty} y_n(t) = H t - N^{-1} \frac{u^v}{s^v} N^+ L \sum_{n=0}^{\infty} y_n t + \sum_{n=0}^{\infty} A_n t. \quad (13)$$

За допомогою (13) можемо зробити висновок, що

$$y_{n+1} t = -N^{-1} \frac{u^v}{s^v} N^+ L y_n t + A_n t, \quad n \geq 1.$$

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Розглянемо задачу Коші:

$$\begin{aligned} \frac{\partial^\alpha}{\partial t^\alpha} \theta(x, t) &= \frac{\partial^2}{\partial x^2} \theta(x, t) + \theta(x, t), \\ 0 < \alpha \leq 1, -4 < x < 4, 0 \leq t \leq 1 & \\ \theta(x, 0) &= \cos \pi x. \end{aligned} \quad (14)$$

За допомогою методу кінцевих різниць та за попередніми обчисленнями отримуємо рекурентну формулу для обчислення розв'язку задачі Коші:

$$\begin{aligned} \theta_k^{n+1} = \theta_k^n + \frac{\Delta\tau \cdot \Gamma(2-\alpha)}{\tau_1^{1-\alpha} - \tau_0^{1-\alpha} h^2} (\theta_{k+1}^n + (h^2 - 2)\theta_k^n + \theta_{k-1}^n) \\ - \frac{1}{\tau_1^{1-\alpha} - \tau_0^{1-\alpha} h^2} \sum_{s=1}^{n-1} (\tau_{n-s+1}^{1-\alpha} - \tau_{n-s}^{1-\alpha}) \cdot (\theta_k^{s+1} - \theta_k^s) \end{aligned}$$

За попередніми обчисленнями та застосування методу декомпозиції і підставляючи функції з задачі Коші (14) обрахуємо наші значення  $\theta_{n+1}(x, t)$

$$\theta(x, t) = \sum_{n=0}^{\infty} \theta_n(x, t) = \cos \pi x \left[ 1 + 1 - \pi^2 \frac{t^\alpha}{\Gamma(\alpha+1)} + 1 - \pi^2 \frac{t^{2\alpha}}{\Gamma(2\alpha+1)} + 1 - \pi^2 \frac{t^{3\alpha}}{\Gamma(3\alpha+1)} + \dots \right]$$

Звідки отримуємо точний розв'язок

$$\theta(x, t) = \cos \pi x E_\alpha(1 - \pi^2 t^\alpha)$$

де  $E_p(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(1+pk)}$ ,  $p \geq 0$ ,  $z \in \mathbb{C}$  — функція Міттаг-Леффлера.

Будемо порівнювати графіки у фіксованих значеннях степеня дробової похідної  $\alpha = 0.25, 0.5, 0.75$  та часової змінної  $t = 0.5$ , а також зобразимо результати для зручності у тривимірному просторі.

- Розбиття для  $\alpha = 0.25$  : x на 21 точку, t на 14001 точку
- Розбиття для  $\alpha = 0.5$  : x на 41 точку, t на 2001 точку
- Розбиття для  $\alpha = 0.75$  : x на 51 точку, t на 401 точку

Отже, після виконання програми отримуємо такі результати:

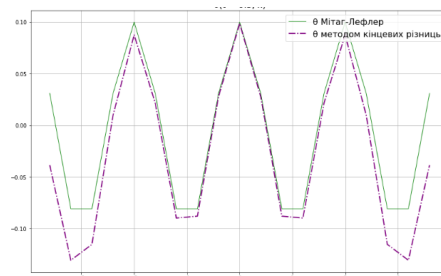


Рисунок 1. Порівняння отриманих значень функції  $\theta(0.5, x)$ ,  $\alpha = 0.25$



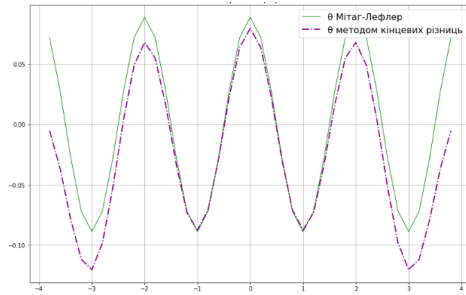


Рисунок 2. Порівняння отриманих значень функції  $\theta(0.5, x)$ ,  $\alpha = 0.5$

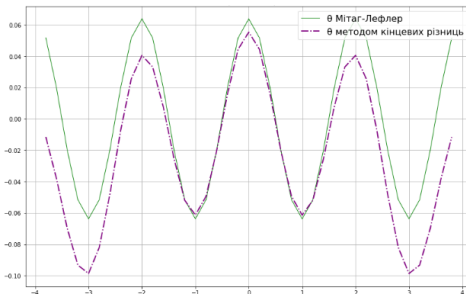


Рисунок 3. Порівняння отриманих значень функції  $\theta(0.5, x)$ ,  $\alpha = 0.75$

- Середнє значення помилки для  $\{\alpha = 0.25, t = 0.5\}$ : 0.0228
- Середнє значення помилки для  $\{\alpha = 0.5, t = 0.5\}$ : 0.0225
- Середнє значення помилки для  $\{\alpha = 0.75, t = 0.5\}$ : 0.0227

Як бачимо з рис. 1–3 та точних обчислень, методи дали доволі схожі результати, помилка досить незначна, найбільша для даного значення  $t = 0.5$  помилка вийшла для  $\alpha = 0.25$ , а найменша для  $\alpha = 0.5$ .

Тепер функцію  $\theta(t, x)$  зобразимо у трьох вимірах для фіксованих значень  $\alpha$  та порівняємо результати обох методів: справа (зелений) буде розв'язок за допомогою функцій Мітаг-Лефлера, а зліва (помаранчевий) – результат отриманий за допомогою застосування методу кінцевих різниць. Також наведемо значення похибки для всієї нашої часової області  $t$ .

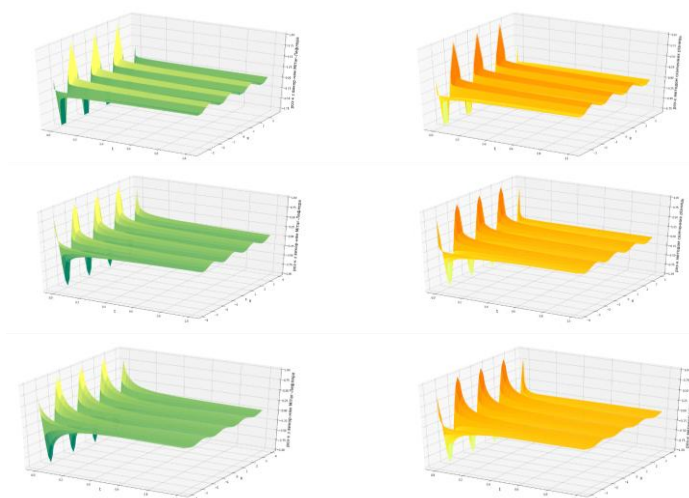


Рисунок 4. Порівняльний графік роботи обох досліджуваних методів для різних значень  $\alpha = 0.25, 0.5, 0.75$

- Середнє значення помилки для  $\alpha = 0.25$  по всім  $t$ : 0.0233
- Середнє значення помилки для  $\alpha = 0.5$  по всім  $t$ : 0.0247
- Середнє значення помилки для  $\alpha = 0.75$  по всім  $t$ : 0.0277

Як бачимо з графіків зображених на рис.4 та точних обрахунків, методи дали доволі схожі результати, помилка досить незначна, найбільша для усієї області помилка вийшла для  $\alpha = 0.75$ , а найменша для  $\alpha = 0.25$ , що звісно може бути обґрунтоване вищою кількістю вузлів просторової сітки для  $\alpha = 0.25$ .

#### 4. ВИСНОВКИ

В ході даного дослідження було оглянуто відомі процеси, що моделюються за допомогою застосування дифузійних диференціальних рівнянь, була проведена їх класифікація та детальний огляд, а також були наведені застосування дробового числення в цілому.

Результатами даної роботи було показано відмінності у поведінці змодельованого процесу в залежності від значення дробової похідної, було відстежено основні особливості результатів роботи програми для задачі Коші, що була проаналізована та побудована двома методами, і були прокоментовані відмінності між цими двома запропонованими методами (чисельним методом і методом з використанням операційного числення).

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Нахушев А. М. «Дробное исчисление и его применение». – М.: ФИЗМАТЛИТ, 2003.
2. Самко С.Г. «Интегралы и производные дробного порядка и некоторые их приложения.» С.Г. Самко, А.А. Килбас, О.И.Маричев // Минск. Наука и техника, 1987.
3. Учайкин В. В. «Метод дробных производных». Ульяновск: Изд. «Артишок», 2008. – 512 с.
4. Самарский А.А. Теория разностных схем. М.: Наука, 1977.
5. Самарский А.А., Гулин А.В. Устойчивость разностных схем. М.: Наука, 1973.
6. Шхануков МХ. О сходимости разностных схем для дифференциальных уравнений с дробной производной // Докл. РАН. 1996. Т. 348. № 6. С. 746-748.

# ОЦІНКА ADME-ТОХ ВЛАСТИВОСТЕЙ МОЛЕКУЛ

Возняк В.З.<sup>1</sup>, Островський З.Ю.<sup>2</sup>, Недашківська Н.І.<sup>3</sup>

Національний Технічний Університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>vozniak.v.z@gmail.com, <sup>2</sup>zakharostrovsky@gmail.com,  
<sup>3</sup>n.nedashkivska@gmail.com [0000-0002-8277-3095]

**Оцінка ADME-Тох властивостей молекул має надзвичайно важливе значення у сфері розробки, яка визначатиме ефективність препарату та його успішність проходження різних стадій випробувань. Метою цієї роботи є створення багатозадачної графової нейронної мережі, яка здатна швидко обробляти великі бази молекул, розміром від кількох мільярдів сполук, з метою відбору з них найперспективніших молекул для подальшого експериментального дослідження.**

**Ключові слова: Graph Neural Networks, Multi-Task Learning, Drug Discovery, Molecule Property Prediction.**

## 1. ВСТУП

Оцінка ADME-Тох (absorption – абсорбція, distribution – розподіл, metabolism – метаболізм, excretion – виділення, і toxicity – токсичність) властивостей молекул має надзвичайно важливе значення у сфері розробки ліків (Drug Discovery) [1, 2]. Під час досліджень вчені визначають яким значенням цих параметрів повинна відповідати молекула, для того щоб зможти дістатися до збудника хвороби, провзаємодіяти з ним і при цьому не бути токсичною для організму людини і легко виводитися з нього після лікувальної дії. Саме тому від точності передбачення ADME-Тох властивостей молекули залежатиме ефективність препарату та успішність проходження різних стадій випробувань.

Традиційний метод перевірки – in vivo, in vitro – експериментальні. Тобто, необхідно фізично синтезувати молекулу і провести реальні дослідження, щоб виміряти значення кожного параметру. Такий спосіб дуже часо-, трудо- та ресурсомісткий.

Для того, щоб забезпечити швидкий і дешевий процес розробки нових ліків, необхідно застосовувати сучасні in silico методи комп'ютерного моделювання, які дають попередню оцінку властивостей молекул без попереднього фізичного експерименту.

Тому метою цієї роботи є створення багатозадачної графової нейронної мережі, яка здатна швидко обробляти великі бази молекул, розміром від кількох мільярдів сполук, з метою відбору з них найперспективніших молекул для подальшого експериментального дослідження.

## 2. ВИБІР ГРАФОВОЇ АРХІТЕКТУРИ

Першим етапом побудови графової багатозадачної нейронної мережі є вибір саме графової архітектури. Це важливо, тому що в подальшому ми будемо обирати багатозадачний підхід і це потрібно робити із уже зафіксованою хорошою графовою архітектурою, щоб ми були впевнені, що, наприклад, погані результати багатозадачності не залежать саме від графової архітектури.

Алгоритм вибору графової архітектури наступний. Для проведення досліджень було відібрано 8 датасетів, які містять регресійні та/або класифікаційні значення цільової змінної. Далі було складено сітку гіперпараметрів для нейронних мереж. Спочатку обиратимемо одну

із графових архітектур і певний підбір гіперпараметрів для неї. Далі нейронну мережу, яка визначається цим набором гіперпараметрів і архітектурою, тренуватимемо на обраних 8 датасетах незалежно одне від одного й отримуватимемо два результати: усереднений на 4 параметрах скор для задач бінарної класифікації та усереднений на 4 параметрах скор для задач регресії. А тоді за допомогою Парето перетину двох усереднених скорів виберемо найкращу архітектуру й гіперпараметри.

Для задач бінарної класифікації нашим основним скором (основною метрикою) буде F1. Це універсальна метрика, яка мінімізує значення як хибно негативних, так і хибно позитивних. Для регресійних задач - R2. Це метрика, яка набуває значень від мінус нескінченності до 1, де від'ємні значення означають, що прогностична здатність моделі гірша за прогноз середнього для всіх даних, 0 – це те саме, що прогноз середнього, а додатні значення – прогноз кращий, ніж прогноз середнього. Тобто чим вище значення – тим краще, в ідеалі ми хочемо, щоб воно прямувало до 1.

Таблиця 1. Значення параметрів для підбору для графових архітектур

name	use edges features	activation	use batch norm	layers range	heads range	dropout range	units	unit's multiplier
GATv2	False	LeakyReLU	(True, False)	{1, 5}	{1, 5}	[0, 0.3]	-	-
GCN	False	LeakyReLU	(True, False)	{1, 5}	-	-	(32, 64, 128)	{1, 2}
GIN	False	LeakyReLU	(True, False)	{1, 5}	-	-	(32, 64, 128)	{1, 2}
GINE	True	LeakyReLU	(True, False)	{1, 5}	-	-	(32, 64, 128)	{1, 2}
MF	False	LeakyReLU	(True, False)	{1, 5}	-	-	(32, 64, 128)	{1, 2}

Графові архітектури будемо обирати серед GATv2 [3], GCN [4], GIN [5], GINE [6] та MF [7]. У табл. 1 наведено гіперпараметри, які ми будемо підбирати саме для графових архітектур. Круглі дужки означають набір категоріальних варіантів для даного параметру.

Фігурні дужки означають вибір із даного діапазону цілого значення для даного параметру із включеними крайовими значеннями. Квадратні дужки – аналогічно до фігурних, лише не ціле, а дійсне значення. Усі решта значень – додаткова інформація для архітектури.

У табл. 2 наведено всі решта гіперпараметри і значення, які ми будемо перебирати.

Таблиця 2. Значення решти параметрів для підбору

parameter	value
FC activation	LeakyReLU
FC use batch norm	(True, False)
FC layers range	{1, 3}
FC dropout range	[0.1, 0.5]
pooling type	(mean, add, max)
batch size	(32, 64, 128, 256)
learning rate	(1e-4, 1e-3)
weight decay	(0, 5*1e-5, 5*1e-4, 5*1e-3)

Перейдемо до аналізу результатів. Загалом ми зробили 54 різних експериментів в ортуна (тут і далі трайлів) із різними наборами гіперпараметрів. У табл. 3 зображено детальну інформацію про кожен із цих трайлів й обрано найкращі параметри для нової архітектури в стовпці Best.

Таблиця 3. Детальна інформація про трайли 10, 23 та 51

Parameter	Trial 23	Trial 10	Trial 51	Best
graph model	MF	GINE	GIN	MF
batch size	32	32	64	32
learning rate	1e-3	1e-3	1e-4	1e-3
weight decay	5*1e-3	5*1e-5	5*1e-4	5*1e-5
FC n_layers	2	3	3	2
FC use batch norm	True	False	False	False
FC dropout	-	0.36	0.44	0.15
graph n_layers	3	4	3	3
graph n_units	128	32	128	128
graph unit's multiplier	2	2	2	2
graph use batch norm	False	True	False	False
graph pooling type	max	max	max	max

Найкращою архітектурою виявився графовий підхід MF [7] із 3 графовими шарами із 128, 256 та 512 нейронами відповідно. Лінійна ж частина складалася із 2 шарів по 512 нейронів та дропауту між ними, що дорівнював 0.15.

## 3. ПОБУДОВА ГРАФОВОЇ БАГАТОЗАДАЧНОЇ НЕЙРОННОЇ МЕРЕЖІ

### 3.1 Терміни

Базові значення метрик – значення метрик моделей, натренованих за допомогою AutoML Pipeline в режимі “окремий датасет - окрема модель”.

AutoML Pipeline – внутрішньо розроблений пайплайн для автоматизації тренування моделей класичного машинного навчання, який включає етапи автоматичного генерування молекулярних дескрипторів, їх відбору та підбору гіперпараметрів за допомогою алгоритмів бассовської оптимізації.

Якість датасету – це введена для зручності характеристика датасету, яка може мати значення “висока якість”, “середня якість” та “низька якість”. Всі наявні молекулярні датасети були проранжовані по базових значеннях цільової метрики та відповідно погруповані на три класи якості.

### 3.2 Постановка досліджень

Отож, на початку досліджень було відомо наступне:

1. Більше 40 датасетів з різними розмірами та якістю.
2. Базові значення метрик моделей для кожного датасету.
3. 9 методів багатозадачного навчання нейронних мереж [8-15].
4. Проблеми, які виникають при багатозадачному навчанні, а саме “конфліктуючі градієнти” та різна область значень індивідуальних функцій втрат.

Задачі проведення досліджень:

1. Метод багатозадачного тренування, який буде найкращим компромісом між якістю отриманої моделі та швидкістю тренування.
2. Дослідити залежність метрик багатозадачної нейронної мережі та стабільність її тренування від обраного методу багатозадачного тренування, якості, кількості та типу датасетів, відібраних для сумісного тренування.
3. Визначити оптимальний набір датасетів для тренування багатозадачної нейронної мережі.

На початку дослідження була висунута наступна гіпотеза: враховуючи різну якість та розміри датасетів, найкращі результати багатозадачна нейронна мережа може досягнути тренуючись на їх підвибірці. Ця гіпотеза є підґрунтям для задачі 3.

### 3.3 Методика досліджень

Для отримання інформації необхідної для розв’язку поставлених задач пропонується провести серію експериментів.

Експеримент, в цьому контексті, визначається набором тренувальних датасетів (множини датасетів, на яких навчається багатозадачна нейронна мережа) та набором зовнішніх датасетів (множини датасетів, які не використовувалися для тренування багатозадачної нейронної мережі і слугують критерієм її узагальнюючої здатності).

Під час кожного експерименту нейромережа навчається на тому ж наборі тренувальних датасетів кожним із наперед визначених методів багатозадачного тренування.

Проведені експерименти мають вимірність 3, тобто є 3 осі параметрів по яким визначається експеримент, а саме:

1. Кількість датасетів у тренувальному наборі.
2. Типи датасетів у тренувальному наборі. Тобто, наявність лише класифікаційних, лише регресійних або датасетів обох типів.

3. Якість датасетів включених до тренувального набору. Наприклад, лише високої якості або наявність датасетів різної якості.

Таким чином, задаючи різні конфігурації експериментів та порівнюючи їх результати між собою, отримуємо можливість встановити шукані взаємозалежності, сформульовані у постановці задач.

Таблиця 4. Конфігурації експериментів

Тренувальний набір датасетів	Зовнішній набір датасетів	№ експерименту
AMES, HERG_cls	Bioavailability, Carcinogenicity_ISF, CYP_inh_2D6 (cls + rgr)	0
Acute, CYP_inh_3A4_rgr	Bioavailability, Carcinogenicity_ISF, CYP_inh_2D6 (cls + rgr)	1
AMES, HERG_cls, BBB	Bioavailability, Carcinogenicity_ISF, CYP_inh_2D6 (cls + rgr)	2
Acute, CYP_inh_3A4_rgr, Caco-2	Bioavailability, Carcinogenicity_ISF, CYP_inh_2D6 (cls + rgr)	3
AMES, HERG_cls, BBB, CYP_inh_3A4_cls	Bioavailability, Carcinogenicity_ISF, CYP_inh_2D6 (cls + rgr)	4
Acute, CYP_inh_3A4_rgr, Caco-2, HERG_rgr	Bioavailability, Carcinogenicity_ISF, CYP_inh_2D6 (cls + rgr)	5
AMES, Acute	Bioavailability, Carcinogenicity_ISF, CYP_inh_2D6 (cls + rgr)	6
AMES, HERG_cls, Acute, CYP_inh_3A4_rgr	Bioavailability, Carcinogenicity_ISF, CYP_inh_2D6 (cls + rgr)	7
AMES, HERG_cls, BBB, CYP_inh_3A4_cls, Acute, CYP_inh_3A4_rgr, Caco-2, HERG_rgr	Bioavailability, Carcinogenicity_ISF, CYP_inh_2D6 (cls + rgr)	8
AMES, HERG_cls, BBB, CYP_inh_3A4_cls, HIA, Bioavailability, CYP_sub_2D6, Carcinogenicity_ISF	Bioavailability, Carcinogenicity_ISF, CYP_inh_2D6 (cls + rgr)	13



HERG_rgr, HERG_cls, CYP_inh_3A4_rgr, CYP_inh_3A4_cls, CYP_inh_2D6_rgr, CYP_inh_2D6_cls	Bioavailability, Carcinogenicity_ISF, CYP_inh_2D6 (cls + rgr)	17
HIA, Clearance, CYP_sub_2D6	Bioavailability, Carcinogenicity_ISF, CYP_inh_2D6 (cls + rgr)	18
AMES, HERG_cls, BBB, CYP_inh_3A4_cls, Acute, CYP_inh_3A4_rgr, Caco-2, HERG_rgr, HIA, Clearance, Bioavailability_cls, Bioavailability_rgr, CYP_sub_2D6, Carcinogenicity_ISF_rgr, Carcinogenicity_ISF_cls, PGP_sub, CYP_inh_2D6_cls, CYP_inh_2D6_rgr	Bioavailability, Carcinogenicity_ISF, CYP_inh_2D6 (cls + rgr)	19

Таким чином, проаналізувавши результати експериментів з Таблиці 4, отримуємо наступні висновки:

Тренування багатозадачної нейронної мережі на наявних молекулярних датасетах є стабільним.

Найкращі метрики досягаються, якщо включати в тренувальний набір як регресійні, так і класифікаційні датасети високої якості і приблизно одного розміру. Тобто, гіпотеза висунута на початку дослідження, підтвердилася – для отримання найкращих результатів використовувати всі наявні датасети для багатозадачного тренування не потрібно.

Аналізуючи результати не було виявлено жодного методу багатозадачного навчання, який би систематично давав кращі результати. Тому для того, щоб обрати єдиний метод, який буде використовуватися для навчання фінальної моделі, було прийнято рішення зібрати статистику по всіх експериментах і визначити, які саме методи найчастіше давали для кожної із задач найвищі результати.

Отримані результати наведені у Таблиці 5.

Таблиця 5. Кількість разів, коли метод багатозадачного тренування дав найкращі з-поміж інших методів результати для датасету

Метод	Тестові вибірки		Валідаційні вибірки		Всього
	Тренувальний набір	Зовнішній набір	Тренувальний набір	Зовнішній набір	
DWA	16	19	17	19	71
EW	15	22	18	23	78
GradNorm	24	17	25	11	77
IMTL	8	30	4	42	84
RLW	18	15	11	14	58
UW	15	25	21	23	84

Отже, топ-3 методи, які найчастіше давали найкращі результати це IMTL [13], UW [9] та EW. З огляду на те, що IMTL має чітко виражене зміщення в сторону зовнішніх наборів та є найповільнішим з усіх методів, його кандидатуру вирішено було відхилити. Між UW та

EW немає принципової відмінності в статистиці, проте EW – найшвидший метод для тренування, і найпростіший. У свою чергу ідея підходу UW виглядає менш робастною до потенційно нестандартних комбінацій функцій втрат. Тому, було прийнято рішення тренувати багатозадачну нейронну мережу саме звичайним EW методом.

#### 4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Результатом нашого дослідження стала побудована графова багатозадачна нейронна мережа, яка дає змогу передбачати ADME-Тох властивості молекули, серед яких 26 задач бінарної класифікації та 15 задач регресії. У Таблиці 6 та Таблиці 7 наведені метрики кожного параметру ADME-Тох для класифікаційних та регресійних задач відповідно.

Таблиця 6. Метрики для класифікаційних задач фінальної моделі

ADME-Tox param	F1		Accuracy		Precision		Recall	
	CV	Test	CV	Test	CV	Test	CV	Test
Bioavailability	0.721	0.651	0.7	0.656	0.683	0.676	0.766	0.627
HIA	0.742	0.762	0.942	0.943	0.825	0.8	0.677	0.727
Pgp inhibitors	0.886	0.875	0.881	0.862	0.905	0.844	0.869	0.908
Pgp substrates	0.819	0.82	0.817	0.816	0.813	0.803	0.825	0.838
BBB	0.917	0.924	0.906	0.915	0.908	0.925	0.926	0.923
CYP1A2 inhibitor	0.865	0.839	0.848	0.827	0.85	0.866	0.881	0.814
CYP2C9 inhibitor	0.83	0.795	0.805	0.788	0.782	0.831	0.885	0.762
CYP2C19 inhibitor	0.858	0.859	0.848	0.838	0.859	0.804	0.857	0.923
CYP2D6 inhibitor	0.857	0.832	0.847	0.824	0.839	0.834	0.877	0.83
CYP3A4 inhibitor	0.834	0.835	0.809	0.796	0.826	0.772	0.841	0.908
CYP1A2 substrate	0.626	0.778	0.862	0.921	0.644	0.817	0.614	0.742
CYP2C9 substrate	0.635	0.544	0.885	0.871	0.634	0.607	0.641	0.493
CYP2C19 substrate	0.692	0.742	0.9	0.91	0.709	0.708	0.678	0.78
CYP2D6 substrate	0.715	0.729	0.883	0.886	0.731	0.718	0.703	0.74
CYP3A4 substrate	0.814	0.774	0.851	0.828	0.803	0.802	0.825	0.747
AMES	0.851	0.845	0.827	0.827	0.816	0.839	0.889	0.851
Carcinogenicity ISF	0.71	0.684	0.871	0.865	0.781	0.788	0.655	0.605
Carcinogenicity OSF	0.684	0.644	0.777	0.764	0.685	0.691	0.685	0.603
Androgen antagonist	0.906	0.908	0.925	0.924	0.905	0.886	0.908	0.931
Androgen agonist	0.922	0.942	0.957	0.967	0.944	0.926	0.902	0.958
Androgen binding	0.887	0.899	0.908	0.917	0.865	0.862	0.911	0.94
Estrogen antagonist	0.879	0.85	0.95	0.94	0.961	0.958	0.811	0.764
Estrogen agonist	0.522	0.551	0.846	0.861	0.658	0.731	0.433	0.442
Estrogen binding	0.881	0.85	0.952	0.94	0.97	0.958	0.808	0.764
hERG	0.886	0.886	0.865	0.867	0.875	0.885	0.897	0.887
DILI	0.758	0.742	0.686	0.689	0.671	0.696	0.874	0.795

Основною перевагою побудованої графової багатозадачної нейронної мережі є її швидкодія. Вона здатна передбачати ADME-Тох властивості для 500 тисяч молекул за 280 секунд на середньостатистичній машині із процесором GPU.

Таблиця 7. Метрики для регресійних задач фінальної моделі

ADME-Tox param	R2		MSE		MAE	
	CV	Test	CV	Test	CV	Test
Caco-2	0.657	0.727	0.2	0.166	0.345	0.302
PPB	0.71	0.732	301.07	280.3	11.38	10.84
VD	0.486	0.533	1.084	1	0.799	0.766
Half-life	0.295	0.209	1.316	1.583	0.876	0.974
Plasma Clearance	0.252	0.398	1.496	1.222	0.923	0.854
Renal Clearance	0.222	0.401	2.761	2.319	1.037	0.924
Acute	0.461	0.425	1.993	2.204	1.059	1.083
CYP1A2 inhibitor	0.537	0.585	1.753	1.467	0.925	0.86
CYP2C9 inhibitor	0.513	0.392	1.19	1.457	0.786	0.785
CYP2C19 inhibitor	0.499	0.536	1.247	1.175	0.793	0.763
CYP2D6 inhibitor	0.604	0.58	1.253	1.374	0.744	0.788
CYP3A4 inhibitor	0.623	0.626	1.356	1.374	0.798	0.791
Carcinogenicity ISF	0.532	0.648	12.425	11.314	2.122	1.79
Carcinogenicity OSF	0.495	0.4	15.365	14.029	2.866	2.67
hERG	0.675	0.702	1.418	1.224	0.778	0.801

## 5. ВИСНОВКИ

У ході роботи проаналізовано набори даних різного розміру для вирішення поставленої проблеми, знайшли можливий перетин усіх властивостей по молекулах і врахували це при поділі на тренувальні, валідаційні й тестові вибірки.

Також обрано найкращу графову архітектуру за допомогою тунінгу гіперпараметрів у бібліотеці Python optuna на невеликому наборі датасетів. Найкращою архітектурою виявився графовий підхід MF із 3 графовими шарами із 128, 256 та 512 нейронами відповідно. Лінійна ж частина складалася із 2 шарів по 512 нейронів та дропауту між ними, що дорівнював 0.15.

Наступним етапом стало обрання найкращого підходу до побудови мультизадачності. Ним виявився звичайний підхід із зрівноваженим зважуванням EW.

Як результат, побудовано графову багатозадачну нейронну мережу для передбачення ADME-Tox властивостей молекул, яка здатна обробляти 500 тисяч молекул за 280 секунд на середньостатистичній машині із процесором GPU.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Saeed Alqahtani. In silico ADME-Tox modeling: progress and prospects, 2017.
2. Vinícius Gonçalves Maltarollo, Jadson Castro Gertrudes, Patrícia Rufino Oliveira, Kathia Maria Honorio. Applying machine learning techniques for ADME-Tox prediction: a review, 2015.
3. Shaked Brody, Uri Alon, Eran Yahav. How Attentive are Graph Attention Networks? arXiv preprint arXiv:2105.14491, 2021.
4. Thomas N. Kipf, Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. arXiv preprint arXiv:1609.02907, 2016.
5. Keyulu Xu, Weihua Hu, Jure Leskovec, Stefanie Jegelka. How Powerful are Graph Neural Networks? arXiv preprint arXiv:1810.00826, 2018.
6. Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, Jure Leskovec. Strategies for Pre-training Graph Neural Networks. arXiv preprint arXiv:1905.12265, 2019.

7. David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, Ryan P. Adams. Convolutional Networks on Graphs for Learning Molecular Fingerprints. arXiv preprint arXiv:1509.09292, 2015.
8. Baijiong Lin, Feiyang Ye, Yu Zhang. A Closer Look at Loss Weighting in Multi-Task Learning. arXiv preprint arXiv:2111.10603, 2021.
9. Alex Kendall, Yarin Gal, Roberto Cipolla. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. arXiv preprint arXiv:1705.07115, 2018.
10. Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, Andrew Rabinovich. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. arXiv preprint arXiv:1711.02257, 2018.
11. Shikun Liu, Edward Johns, Andrew J. Davison. End-to-End Multi-Task Learning with Attention. arXiv preprint arXiv:1803.10704, 2019.
12. Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, Qiang Liu. Conflict-Averse Gradient Descent for Multi-task Learning. arXiv preprint arXiv:2110.14048, 2021.
13. Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, Wayne Zhang. Towards Impartial Multi-Task Learning, 2021.
14. Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, Chelsea Finn. Gradient Surgery for Multi-Task Learning. arXiv preprint arXiv:2001.06782, 2020.
15. Zirui Wang, Yulia Tsvetkov, Orhan Firat, Yuan Cao. Gradient Vaccine: Investigating and Improving Multi-task Optimization in Massively Multilingual Models, 2021.

# МЕТОДИ НЕСТАТИЧНОГО ПРОГНОЗУВАННЯ ДЛЯ ВІДНОВЛЕННЯ ЗОБРАЖЕНЬ

Клименко І.О.<sup>1</sup>, Подколзін Г.Б.<sup>2</sup>, Тимощук О.Л.<sup>3</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

<sup>1</sup> irka13klimenko@gmail.com

<sup>2</sup> glebpodkolzin@gmail.com [0000-0002-7120-2772]

<sup>3</sup> o.tymoshchuk@kpi.ua [0000-0003-1863-3095]

**У сучасному світі постає проблема розв'язання задач класифікації зображень та ідентифікації об'єктів на зображеннях. Метою дослідження є розробка програмного комплексу для відновлення та класифікації зображень. В роботі розглянуто новий напрям застосування нестатичного прогнозування, а саме відновлення зображення для подальшої класифікації. Результатом роботи є створення загальної системи класифікації, що одночасно відновлює зображення та класифікує його, що призводить до більш точних результатів класифікації, ніж при використанні класичних методів.**

**Ключові слова:** машинне навчання, відновлення, класифікація, модуль відновлення, класифікатор, модуль класифікації.

## 1. ВСТУП

У сучасному світі значного поширення набули методи машинного навчання для розв'язання задач класифікації зображень, або ж ідентифікації конкретних об'єктів на зображеннях [1]. Проте при розв'язанні таких задач з'являються певні проблеми, зокрема не завжди зображення яке отримує відео або фото-камера є чітким та зрозумілим. Ця нечіткість на зображеннях шкодить ідентифікації [2].

Методи, які будуть розглянути у цій роботі, дозволяють відновлювати зображення до стану, прийняттого для машини, яка буде їх обробляти. Це дозволить покращити результати ідентифікації, що у свою чергу надасть можливість приймати більш точне та зважене рішення базуючись на достовірних даних.

## 2. ПОСТАНОВКА ПРОБЛЕМИ ТА ШЛЯХИ ЇЇ РОЗВ'ЯЗАННЯ

Розглянемо стандартну задачу класифікації зображення. Ми маємо певну кількість класів, до яких можуть належати зображення. Необхідно побудувати алгоритм, що зможе приймати на вхід зображення і на виході видавати відповідь: до якого саме класу відноситься зображення.

Останнім часом для розв'язання цієї задачі здебільшого використовують машинне навчання, зокрема нейронні мережі, або ж так звані CNN (Convolutional Neural Network) [3], тобто згорткові нейронні мережі. Загорткова нейронна мережа – це алгоритм глибокого навчання, який приймає вхідне зображення, назначає важливість (навчені ваги та упередження) різним аспектам/елементам зображення та може відрізнити одне зображення від іншого. Архітектура таких мереж аналогічна структурі нейронів в мозку людини і схожа за організацією до зорової кори. Окремі нейрони реагують на подразники лише в обмеженій

області зорового поля, відомому як рецептивне поле. Множина таких полів перекривається, щоб охопити всю зорову зону. Згорткова мережа здатна успішно фіксувати просторові та тимчасові залежності в зображенні за допомогою застосування відповідних фільтрів. Така архітектура краще вчиться на наборі даних, завдяки зменшенню кількості задіяних параметрів та повторному використанню ваг. Приклад описаної вище загорткової нейронної мережі показано на рисунку (Рис.1).

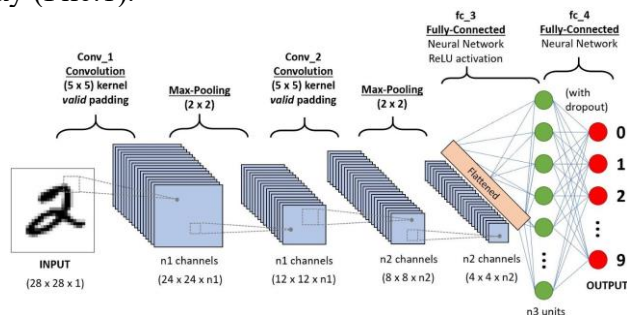


Рисунок 1. Приклад архітектури загорткової мережі для класифікації рукописного числа

Тобто для побудови алгоритму, що зможе класифікувати зображення використовуючи CNN нам необхідно спочатку зібрати набір зображень для тренування моделі. Ці зображення мають бути чітко розподілені на класи, їх має бути багато та для кращих результатів вони мають помітно відрізнятися одне від одного [4]. Після тренування моделі вона може бути використана для класифікації досі невідомих їй зображень. На вхід отримуючи зображення, а на виході повертаючи – до яких класів відноситься зображення та точність цього розподілу.

Цей метод працює для чітких, "лабораторних" зображень, проте життя набагато складніше і не завжди у нас є змога отримати чітке зображення. Наприклад, модель використовується для класифікації зображень отриманих від автономного автомобіля, такі зображення будуть створюватися на швидкості, а тому є ймовірність що не кожне з них буде чітким. А якщо зображення не чітке, то модель, яка бачила до цього лише чіткі зображення не буде здатна визначити, ні його належність до класів, ні будь-яку іншу інформацію. Навчати модель на нечітких зображеннях також не зарадило б ситуації, адже є багато видів нечіткості, та і сама природа нечіткості настільки різноманітна, що неможливо передбачити усі можливі варіанти.

Для подолання цієї проблеми можна використати різні підходи [5]:

- Класифікація на постійному потоці даних;
- Виявлення об'єктів;
- Відновлення зображень.

Перші два допоможуть пом'якшити проблему, проте не розв'яжуть її повністю. Наприклад, не завжди ми маємо постійний потік даних (старі фотографії), або ж виявлення не допоможе у випадках коли зображення спотворене цілком.

Тому у цій роботі ми сфокусуємось саме на відновленні зображень. Тобто ми фокусуємось на створенні системи, що зможе класифікувати навіть спотворені зображення, попередньо їх відновлюючи. Для розв'язання цієї задачі нам необхідно перш за все визначити яким чином ми будемо відновлювати зображення, а потім їх класифікувати. Оскільки класифікація зображень це доволі вивчена та розповсюджена тема, ми не будемо детально її пояснювати, а задачу відновлення, як ключову новизну у підході до таких задач ми розглянемо детальніше.

Процес відновлення зображень має на меті отримання незасміченого, природнього зображення із пошкодженого (будь яким чином). Тобто необхідно побудувати систему, яка зможе на вході приймати забруднене зображення, а на виході повертати відновлене.

Для реалізації процесу відновлення також можна застосувати нейронні мережі. Якщо розглядати вирішення задачі мінімізації функції зображення  $x$ :

$$\arg \min_x E(x; \bar{x}) + R(x), \quad (1)$$

де  $E(x; \bar{x})$  – це функція від даних;

$R(x)$  – це функція апіорності зображення.

Функцію  $E(x; \bar{x})$  зазвичай досить легко створити для широкого кола проблем, таких як підвищення роздільної здатності, зменшення шуму, видалення зайвих об'єктів тощо, а функцію апіорності –  $R(x)$  доволі важко. Сучасні тренди полягають у побудові функції апіорності через тренування згорткових мереж, навчаючи їх через велику кількість зашумлених зображень.

Проте можна помітити, що для сюр'єкції  $g : \theta \rightarrow x$  наступна процедура в теорії рівнозначна (2):

$$\min_{\theta} E(g(\theta); \bar{x}) + R(g(\theta)), \quad (2)$$

На практиці сюр'єкція  $g$  різко змінює спосіб пошуку простору зображень методом оптимізації. Крім того, вибравши "хороше"(можливо ін'єктивне) відображення  $g$ , ми могли б позбутися функції апіорності. Визначимо  $g(\theta)$ , як  $f_{\theta}(z)$ , де  $f$  – це глибока згорткова мережа з параметрами  $\theta$  та  $z$ , де  $z$  – це фіксований вхід мережі, що призводить до наступного формулювання:

$$\min_{\theta} E(f_{\theta}(z); \bar{x}), \quad (3)$$

Тут мережа  $f_{\theta}$  визначена випадково, а вхід мережі  $z$  наповнений шумом та фіксований.

Іншими словами, замість того, щоб шукати відповідь у просторі зображень, ми тепер шукаємо її у просторі параметрів нейронної мережі. Зазначимо, що ми не використовуємо попередньо навчену мережу чи базу даних зображень. Тільки пошкоджене зображення  $\bar{x}$ , що використовується в процесі реставрації.

Алгоритм відновлення зображення зображений на рисунку нижче (Рис. 2).

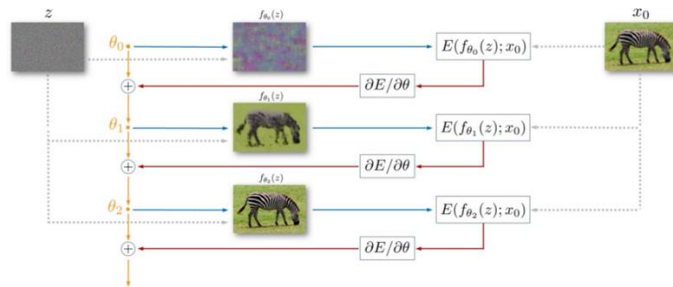


Рисунок 2. Алгоритм відновлення зображення

Перший наш крок – це ініціалізація  $z$ . Заповнити вхід  $z$  рівномірним шумом або будь-яким іншим випадковим зображенням.

Другий крок – розв'язати та оптимізувати функцію за допомогою методу градієнта:

$$\arg \min_{\theta} E(f_{\theta}(z); \bar{x})$$

$$\theta^{k+1} = \theta^k - \alpha \frac{\delta E(f_{\theta}(z); \bar{x})}{\delta \theta},$$

Наостанок, коли ми знайшли оптимальний  $\theta$ , ми можемо отримати оптимальне зображення, просто передаючи вперед фіксований вхід  $z$  до мережі з параметрами  $\theta$ .



$$x^* = f_{\theta^*}(z).$$

### 3. РЕЗУЛЬТАТИ РОБОТИ

Архітектура програмного комплексу має наступний вигляд, як зображено на Рис. 3:

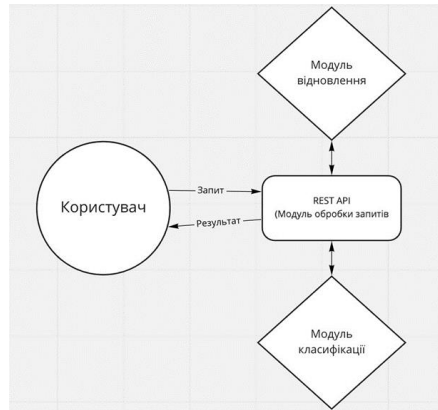


Рисунок 3. Концептуальна архітектура програмного комплексу

Модуль відновлення на вході приймає зображення, а на виході повертає відновлене зображення, що можна використовувати у подальшій роботі. Для побудови нейронної мережі була обрана бібліотека `pytorch`. В якості архітектури мережі для нашого завдання була обрана мережа типу ‘encoder-decoder’, з функцією активації ‘LeakyReLU’. Найскладніше на цьому етапі – це підбір параметрів, зокрема кількості епох тренування, адже занадто багато епох призведе до перетренування моделі, і вона фактично зійдеться до побудови вхідного зображення, що не є нашою ціллю. Для прикладу, який буде наведено пізніше, нормальною кількістю епох експериментальним шляхом було обрано 3000, що скоріш за все підходить і до інших зображень наведеної тематики.

Модуль класифікації на вхід приймає зображення, а на виході повертає клас, до якого він відноситься. У нашій роботі ми розглядаємо треновану класифікацію, тобто навчання проходить «з вчителем».

У поточні реалізації є кілька типів класифікаторів [5]:

- Plain Identifier (Простий класифікатор). Завжди повертає лише один клас з імовірністю 100%. Цей класифікатор має сенс у контексті багаторівневого класифікування.
- Single Identifier (Одиничний класифікатор). Розпізнає класи використовуючи лише одну модель.
- Multi Level Identifier (Багаторівневий класифікатор). Використовує описовий файл `modelmap.json` для побудови багаторівневого дерева класифікаторів.
- Lazy Identifier (Лінійний класифікатор). Переводить один із наведених вище класифікаторів у режим лінійної роботи, тобто класифікатор не буде насправді ініціалізовано до того, як він стане потрібним.

В основі усіх описаних класифікаторів лежить згортова нейронна мережа [6]. В якості архітектури мережі використана ResNet50. Вона добре справляється із своїм завданням, хоча і потребує ресурсів для тренування. Для реалізації ResNet50 також була використана бібліотека `pytorch`.

Для тренування мережі були експериментальним шляхом підбрані суперпараметри, зокрема швидкість навчання була закладена 0.003125, в якості планувальника використовується `tanh`, виконується 25 епох, а для швидшого навчання в якості базових ваг

використовуються ваги мережі, що натренована на наборі даних ImageNet. Загалом мережа показує непогані результати, досягаючи точності більше ніж 90% в топ 5 результатів. Також перед використанням класифікатор готує зображення для моделі, нормалізуючи його та обрізаючи до розмірів, що необхідні моделі.

Перший елемент, з яким будуть взаємодіяти користувачі це точка доступу, яка обробляє запити. За цей функціонал відповідає функція *identify.file*, вона отримує зображення із запиту і тимчасово зберігає його на диск для подальшої обробки.

В якості конкретного прикладу розглянемо повний процес класифікації зображення, від обробки запиту, через відновлення, класифікацію та до отримання відповіді. Будемо розглядати класифікацію рослини *Zamioculcas zamiifolia*, зображеної на Рис.4, що було пошкоджене певним шумом. Задача постає у тому щоб відновити зображення і отримати коректні результати класифікації.



Рисунок 4. Зашумлене зображення *Zamioculcas zamiifolia*

Алгоритм роботи буде наступний:

1. Відправлення зображення на сервер для класифікації.
2. Зображення отримує модуль обробки запитів і зберігає його на диск.
3. Зображення потрапляє до модуля відновлення, який покращує його якість.
4. Зображення потрапляє до модуля класифікації, який визначає його класи.
5. Модуль обробки запитів отримує результати від модуля класифікації та формує відповідь для клієнта.
6. Клієнт отримує результати класифікації.

Під час відновлення зображення проходить процес перетворення, який обумовлено тренуванням моделі. Нижче наведено приклад того, як зображення виглядає після певної кількості епох – Рис.5:



(a) Перша епоха

(b) 500 епох

(c) 3000 епох

Рисунок 5. Процес відновлення

Якщо виконати класифікацію без попереднього відновлення, то отриманий результат буде не вірний, тобто класифікатор вкаже не той клас на який ми сподіваємося:

```
{  
  "class_name": "Phlebodium",  
  "probability": 0.6576755362749099  
}
```

Проте після відновлення результати значно покращуються:

```
{  
  "class_name": "Zamioculcas_zamiifolia",  
  "probability": 0.3237182891368866  
}
```

Хоча точність ідентифікації не дуже висока (32%), але сам факт того, що класифікатор зміг розпізнати нашу рослину свідчить про те, що відновлення було вдалим і побудований комплекс справді справляється із своєю задачею, а саме класифікацією пошкоджених зображень.

#### **4. ВИСНОВКИ**

В роботі розглянуто новий напрям застосування нестатичного прогнозування, а саме відновлення зображення для подальшої класифікації. Враховуючи сучасні тенденції та активний розвиток і дослідження «комп'ютерного» зору цей процес може набути широкого комерційного та наукового застосування у різних сферах промисловості, сільському господарстві, автоматизації виробництва тощо.

Було запропоновано реалізацію програмного комплексу, що буде використовувати розглянуті алгоритми для виконання комплексної задачі відновлення та класифікації, і є цілком сумісним із сучасними потребами та системами.

#### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. A Gentle Introduction to Object Recognition With Deep Learning – 2021 [Електронний ресурс]. – Режим доступу: <https://machinelearningmastery.com/object-recognition-with-deep-learning/>.
2. Image Recognition with Deep Neural Networks and its Use Cases – 2019 [Електронний ресурс]. – Режим доступу: <https://www.altexsoft.com/blog/image-recognition-neural-networks-use-cases/>.
3. David Steinkraus, and John C. Platt Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. – 2003, 962 с.
4. Hinton, Geoffrey E.; Srivastava, Nitish; Krizhevsky, Alex; Sutskever, Ilya; Salakhutdinov, Ruslan R. Improving neural networks by preventing co-adaptation of feature detectors – 2012, 765 с.
5. V. Jain and S. Seung Natural image denoising with convolutional networks. – 2009, 776 с.
6. Mitchell T. McGraw-Hill Science/Engineering/Math. – 1997, 432 с.

# ПОБУДОВА ТОЧНИХ ДОВІРЧИХ ІНТЕРВАЛІВ ДЛЯ ПАРАМЕТРІВ ДЕЯКИХ РОЗПОДІЛІВ ТА ПОРІВНЯННЯ З НАБЛИЖЕНИМИ

Корабльов М.М.<sup>1</sup>, Каніовська І.Ю.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут ім. Ігоря Сікорського», Київ, Україна

<sup>1</sup>m.m.korablov@gmail.com

**Побудова довірчих інтервалів для параметрів розподілів є важливим аспектом при аналізі даних методами параметричної статистики. Метою дослідження є пошук точних довірчих інтервалів для невідомих параметрів звичайного й узагальненого експоненційних розподілів та порівняння їх з наближеними.**

**Ключові слова:** параметрична статистика, точний довірчий інтервал, наближений довірчий інтервал, експоненційний розподіл

## 1. ВСТУП

При дослідженні реальних даних отриманих в ході експериментів активно застосовуються прийоми та засоби параметричної статистики. Одною з основних задач параметричної статистики є інтервальне оцінювання невідомих параметрів закону розподілу генеральної сукупності.

При побудові довірчих інтервалів найчастіше виникають наступні проблеми [1]:

- Переважна більшість отриманих в ході дослідження точкових оцінок параметрів мають дуже складний вигляд, що унеможливує їх подальше застосування для інтервального оцінювання;
- Не розроблені стандартизовані методи побудови довірчих інтервалів для параметрів багатьох розподілів, такі як, наприклад, існують для параметрів нормального розподілу;
- Сучасна теорія наближених довірчих інтервалів погано застосовна на практиці та точність отриманих з її допомогою довірчих інтервалів значно залежить від обсягу вибірки.

У даному дослідженні на прикладі звичайного та узагальненого експоненційного розподілу пропонується точний метод побудови довірчого інтервалу невідомих параметрів та проводиться порівняння з асимптотичними методами.

## 2. ОСНОВНИЙ ЗМІСТ

Однією із спроб розробити узагальнений підхід до інтервального оцінювання параметрів можна вважати сучасну теорію наближених довірчих інтервалів. Даний підхід ґрунтується на центральній граничній теоремі (ЦГТ), і тому отриманий довірчий інтервал значно залежить від обсягу вибірки. Варто зауважити, що умова асимптотичної нормальності, яка вимагається від точкової оцінки невідомого параметра, досить часто не виконується, що робить застосування цього підходу неможливим.

Тому виникає необхідність у розробці підходу до інтервального оцінювання параметрів, який буде:

1. Достатньо універсальним для того, щоб застосовувати його для довільного гіпотетичного закону розподілу генеральної сукупності;
2. Не залежатиме від обсягу вибірки настільки сильно як відомі асимптотичні методи.

Пропонується розв'язати цю задачу шляхом розбудови теорії точних довірчих інтервалів, яка базується на безпосередньому використанні точних розподілів точкових оцінок відповідних параметрів. Для параметрів експоненційного розподілу з зсувом та без нього отримані закони розподілу наведені в пункті 3.

Розв'язання задачі знаходження точного розподілу точкових оцінок ґрунтувалося на застосуванні теорії не випадкових функцій від випадкових аргументів. А саме, маючи випадковий вектор з незалежними та однаково розподіленими координатами було знайдено розподіли випадкових величин, які є функціями цього випадкового вектора.

В якості альтернативи, пропонується розглянути побудову наближеного довірчого інтервалу, яка ґрунтується на наслідку з ЦГТ та подальшому переході до розгляду нормально розподіленої генеральної сукупності. У даному підході точкова оцінка параметра на основі якої будується довірчий інтервал має бути функцією від суми елементів випадкової вибірки, проте від неї не вимагається асимптотичної нормальності. Це дозволяє застосувати цей спосіб побудови довірчого інтервалу тоді, коли стандартна теорія наближених довірчих інтервалів незастосовна.

### 3. ПОБУДОВА ДОВІРЧИХ ІНТЕРВАЛІВ ДЛЯ ЕКСПОНЕНЦІЙНИХ РОЗПОДІЛІВ ТА ПОРІВНЯННЯ РЕЗУЛЬТАТІВ

Для побудови довірчих інтервалів при дослідженні генеральної сукупності  $\xi \sim \text{Exp } \alpha = 4.2889$  як точкова оцінка параметру була використана  $\alpha^* = \frac{1}{n} \sum_{i=1}^n \xi_i = \bar{\xi}$ , вона є незміщеною, консистентною, ефективною, асимптотично нормальною та розподілена за законом  $\text{Erlang } n, \frac{\alpha}{n}$ . Отримані довірчі інтервали з рівнем надійності  $\gamma = 0.95$  наведені у таблицях 1-3.

Таблиця 1. Класичний метод побудови наближених довірчих інтервалів для  $\alpha$

	Обсяг вибірки			
	50	100	250	1000
Значення точкової оцінки	4.0182	4.1839	4.1750	4.3616
Довірчий інтервал	(3.1461, 5.5590)	(3.4982, 5.2038)	(3.7145, 4.7657)	(4.1070, 4.6498)
Ширина інтервалу	2.4129	1.7055	1.0512	0.5427

Таблиця 2. Точний метод побудови наближених довірчих інтервалів для  $\alpha$

	Обсяг вибірки			
	50	100	250	1000
Значення точкової оцінки	4.0182	4.1839	4.1750	4.3616
Довірчий інтервал	(2.9823, 5.2060)	(3.4042, 5.0428)	(3.6734, 4.7081)	(4.0954, 4.6360)
Ширина інтервалу	2.2236	1.6386	1.0346	0.5406

Таблиця 3. Метод побудови наближених довірчих інтервалів на основі наслідку з ЦГТ для  $\alpha$ 

	Обсяг вибірки			
	50	100	250	1000
Значення точкової оцінки	4.0182	4.1839	4.1750	4.3616
Довірчий інтервал	(3.8567, 4.1797)	(4.1009, 4.2669)	(4.1420, 4.2078)	(4.3530, 4.3701)
Ширина інтервалу	0.3229	0.1660	0.0657	0.0171

Для побудови довірчих інтервалів при дослідженні генеральної сукупності  $\eta \sim \text{Exp } \alpha = 4.2889$ ,  $b = 2.8905$  в якості точкових оцінок параметрів були використані  $\alpha^* = \frac{n}{n-1} \eta - \min_{i \in \{1, n\}} \eta_i$ ,  $\alpha^{**} = \frac{n}{n-1} \eta$  та  $b^* = \min_{i \in \{1, n\}} \eta_i$ .  $\alpha^*$  є незміщеною, консистентною, асимптотично ефективною, не є асимптотично нормальною,  $\alpha^{**}$  розподілена за законом Erlang  $n, \frac{\alpha}{n-1}, \frac{nb}{n-1}$ ,  $b^*$  є асимптотично незміщеною, консистентною, не є ефективною та розподілена за законом  $\text{Exp } \frac{\alpha}{n}, b$ . Дослідження деяких властивостей цих оцінок наведено в [2]. Отримані довірчі інтервали з рівнем надійності  $\gamma = 0.95$  наведені у таблицях 4-6.

Таблиця 4. Точний метод побудови наближених довірчих інтервалів для  $b$ 

	Обсяг вибірки			
	50	100	250	1000
Значення точкової оцінки	2.8961	2.8937	2.8937	2.8935
Довірчий інтервал	(2.5907, 2.8961)	(2.7625, 2.8937)	(2.8387, 2.8937)	(2.8802, 2.8935)
Ширина інтервалу	0.3053	0.1311	0.0549	0.0132

Таблиця 5. Точний метод побудови наближених довірчих інтервалів для  $\alpha$ 

	Обсяг вибірки			
	50	100	250	1000
Значення точкової оцінки	5.0969	4.3779	4.5853	4.4290
Довірчий інтервал	(3.1432, 7.6902)	(3.0773, 5.9656)	(3.7119, 5.5758)	(3.9888, 4.8976)
Ширина інтервалу	4.5469	2.8882	1.8638	0.9088

Таблиця 6. Метод побудови наближених довірчих інтервалів на основі наслідку з ЦГТ для  $\alpha$ 

	Обсяг вибірки			
	50	100	250	1000
Значення точкової оцінки	5.0969	4.3779	4.5853	4.4290
Довірчий інтервал	(4.7733, 5.4206)	(4.2330, 4.5228)	(4.5262, 4.6443)	(4.4146, 4.4433)
Ширина інтервалу	0.6472	0.2897	0.1180	0.0287

#### 4. ВИСНОВКИ

Метод побудови точного довірчого інтервалу невідомого параметру експоненційного розподілу без зсуву показує кращий результат при будь-якому обсязі вибірки, ніж класичний метод побудови наближеного довірчого інтервалу. У випадку експоненційного розподілу із зсувом побудова точних довірчих інтервалів іноді показує гірший результат ніж наближений довірчий інтервал. В подальших дослідженнях планується розібратися з цим ефектом, який може бути пов'язаний з неоптимальністю програмної реалізації методу.

Крім того, з наведених таблиць видно, що запропонований альтернативний метод побудови наближених довірчих інтервалів на основі наслідку з ЦГТ показує надзвичайно гарні результати, ширина отриманого довірчого інтервалу в декілька разів менша ніж у інтервалів, отриманих іншими методами. Даний емпіричний результат становить самостійний інтерес та потребує подальшого дослідження.

#### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Турчин В.М. Теорія ймовірностей і математична статистика. Основні поняття, приклади, задачі: Підручник для студентів вищих навчальних закладів. – Дніпропетровськ: ІМА-прес, 2014. – 556 с.

2. Mezbahur. Rahman & Larry. M. Pearson (2001) Estimation in twoparameter exponential distributions, *Journal of Statistical Computation and Simulation*, 70:4,371-386, DOI: 10.1080/00949650108812128

# ПОБУДОВА S-КРИВИХ І ПАТТЕРНІВ S-КРИВИХ ДЛЯ ШВИДКОГО АНАЛІЗУ ДИФУЗІЇ ТЕНДЕНЦІЙ ТА ЕВОЛЮЦІЇ ІННОВАЦІЙ НЕЛІНІЙНИХ СИСТЕМ

Лопатін О.К.

Національний технічний університет України «Київський політехнічний інститут  
імені Ігоря Сікорського», Київ, Україна

lopatinalexeu142@gmail.com

**Ця робота присвячена економетричному моделюванню та прогнозуванню динаміки показників інновацій для моделей, причому на коротких вибірках їх спостереження. S-криві та патерни S-кривих є корисним інструментом для швидкого аналізу систем, особливо при розгляді дифузії тенденцій та еволюції інновацій. Вони можуть евристично визначати рішення та ймовірності, які інакше були б досить трудомісткими для з'ясування з використанням чогось на зразок повного системного чи функціонального аналізу. Пропонований матеріал пройшов апробацію в навчальних курсах для бакалаврів та магістрів на кафедрі «Математичні методи системного аналізу» ІІСА протягом останніх трьох років.**

**Ключові слова: non-linear systems, dynamics of innovation, diffusion of trends, evolution of innovation, forecasting**

## 1. ВСТУП

Нині в Україні відкриваються перспективи інтенсивного розвитку інноваційної діяльності: створення конкурентоспроможної економіки, здійснення технологічного прориву. Але цього не можна зробити без широкого використання досягнень сучасної науки, створення нових технологій, інноваційного продукту.

**Зауваження 1.** *Інновація - "винахід", "нововведення", "нововведення", "новаторство" і т.п. Інновацію можна розглядати як явище, як один із напрямів діяльності. Під інноваціями як явищем розуміються матеріалізовані зміни у товарі, технологіях, організаційній структурі та комунікаціях, зумовлені науково-технічним прогресом та спричинені прагненням підприємства до отримання конкурентних переваг. Процес інновацій є регульована сукупність дій, здійснюваних для послідовного зміни інновацій як явища. Інноваційна діяльність є одним із напрямів підприємницької діяльності, націлений на отримання конкурентних переваг за допомогою інновацій [1, 2].*

У розвинених країнах науково-технологічна революція призвела до перетворення технологічного базису виробництва, в якому зросла питома вага наукомісткого виробництва, що використовує високі технології. Цей процес у західній економічній та соціологічній літературі був кваліфікований як становлення економіки знання. Не тільки в тому сенсі, що вона заснована на знаннях, а й у тому, що в ній яскраво виявилася характерна для наукового знання тенденція до постійної зміни та оновлення, поліпшення показників не тільки використовуваної технології та продукції, а й соціально-економічних процесів.



Теоретичною основою подальших досліджень є методи системного аналізу та [1] статистичного дослідження, щодо змін соціально-економічних та інших масових явищ за часом [2-6].

## 2. НЕОБХІДНІ ВІДОМОСТІ ПРО S-КРИВИХ

На рис. 1 представлена логістична кумулятивна модель ЖЦП з виділенням точки перегину та укрупнених етапів розвитку: 1 – зростання, 2 – зрілість, 3 – насичення.

Крива прискорюється через петлю позитивного зворотного зв'язку, потім уповільнюється через обмеження.

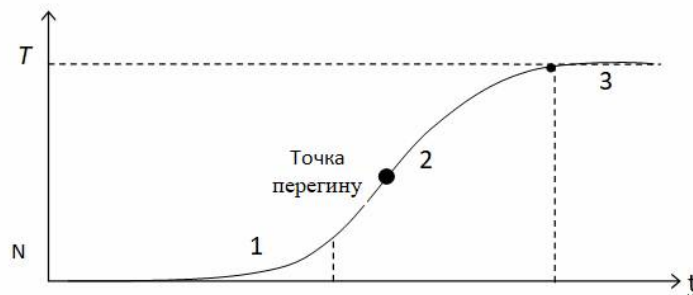


Рисунок 1. Логістична кумулятивна модель ЖЦП (ЖИТТЄВИЙ ЦИКЛ ПРОДУКТУ) з виділенням точки перегину та укрупнених етапів розвитку.

Логістична кумулятивна крива прискорюється через петлю позитивного зворотного зв'язку, потім уповільнюється через обмеження. У імпульсній моделі тренда ЖЦП зазвичай виділяють етапи, показані на рис. 2 використання, зростання, зрілість, насичення і спад. При цьому істотний інтерес додатків представляють координати піку імпульсу або точки перегину логісти.

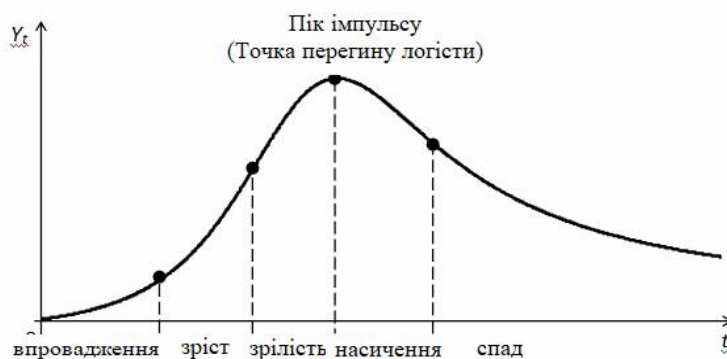


Рисунок 2. Імпульсна модель ЖЦП

Загальною основою конструювання багатьох кривих зростання, логістичної та імпульсної моделей можна вважати [2] вирішення диференціальних рівнянь щодо тренду.

$$\frac{1}{T} \frac{dT}{dt} = \alpha - \beta T \Rightarrow \frac{dT}{dt} = \alpha T - \beta T^2 \Rightarrow T(t) = \frac{\alpha}{1 + \exp(\beta - \gamma t)},$$

вид якої за зміни параметрів представлений на рис. 3. **Ошибка! Источник ссылки не найден.**

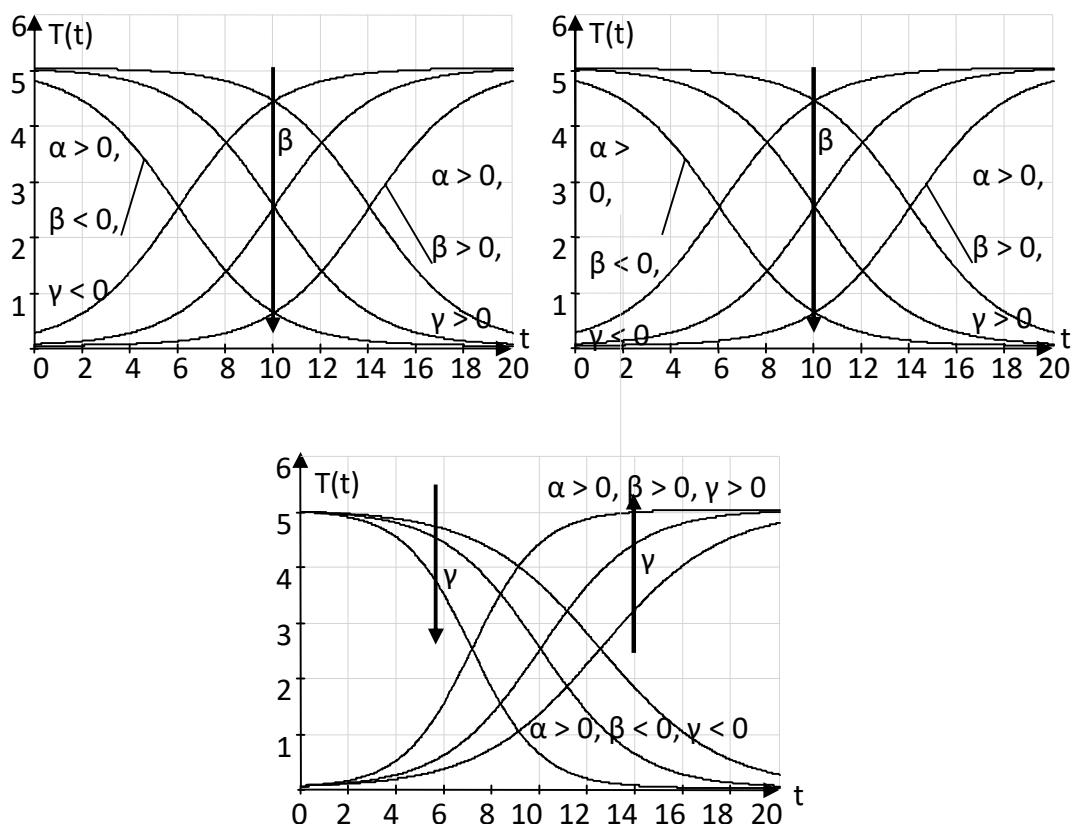


Рисунок 3. Вид логісти Верхулста при зміні знаків та величин параметрів

Точка перегину  $(t^*, T(t^*))$  логісти Верхулста, що визначається співвідношеннями

$$t^* = -\frac{1}{\gamma} \ln\left(\frac{1}{\beta}\right); T(t^*) = \frac{\alpha}{2},$$

відповідає половині рівня насичення  $\alpha$ . У цьому сенсі логіста

Верхулста є симетричною.

**Зауваження 2.** S-крива є фундаментальним патерном, який існує у багатьох системах, які мають петлі позитивного зворотного зв'язку та обмеження.

Відзначимо дві S-криві, які мають важливе значення для наших подальших міркувань [6].

### Дифузійні S-криві

Паттерн S-кривий досить поширений у поширенні ідей, практик та технологій. Наведений нижче приклад показує «дифузійні s-криві» - як технологія поширюється серед населення (у разі домогосподарств США).

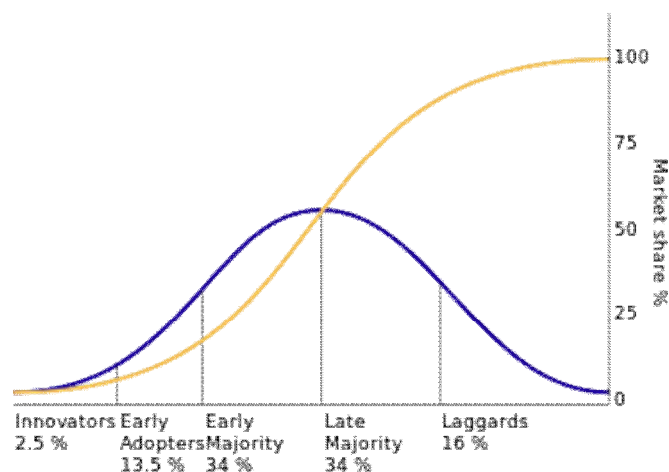


Рисунок 4. Дифузійні s-криві – як технологія поширюється серед населення

Петля позитивного зворотного зв'язку в цьому випадку є сарафанним радіо, а обмеження є фундаментальними бар'єрами для певних сегментів ринку або зростання, таких як простота, зручність використання, масштабованість, ціна і т. д. [8].

### Еволюційні S-криві

На додаток до дифузійних S-кривих у технології, ідеях та практиках, існують еволюційні S-криві. Вони є збільшенням цих ідей, які роблять їх придатними для використання у більшій кількості ситуацій та бажаними для більшої кількості людей. Коли ви долаєте обмеження в одній із цих властивостей за допомогою інновацій, це часто може збігатися з «розблокуванням» нової кривої дифузії, відкриваючи новий ринок, який раніше не використовував вашу технологію чи ідею.

У цьому випадку цикл позитивного зворотного зв'язку – це підвищене розуміння і досвід, який приходить від поширення нової інновації у вашій ідеї або технології, і обмеження є фундаментальними припущеннями в ідеї, практиці або технології, які повинні бути змінені за допомогою іншої інновації, щоб зробити ідею, практику чи технологію більш бажаними [8].

**Зауваження 3.** Логістична динаміка в макромасштабі відображає кумулятивну еволюцію розвитку економіки та обмеження зовнішнього середовища. Однією з причин криз називається незгодженість швидкостей інновацій в економічній та соціальних сферах.

## 3. ПОБУДОВА ТРЕНДУ У ВИГЛЯДІ S-КРИВОЇ ВЕРХУЛЬСТА ЗА СТАТИСТИЧНИМИ ДАНИМИ: GDP НА ДУШУ НАСЕЛЕННЯ НІМЕЧЧИНИ

### 3.1. Опис алгоритму по кроках

#### Крок 1. Знайомство з вихідними даними

World Development Indicators/ Data from World Bank.

[https://www.google.ru/publicdata/explore?ds=d5bncppjof8f9\\_](https://www.google.ru/publicdata/explore?ds=d5bncppjof8f9_)

Вихідні дані представлені в таблиці 1. Графік вихідних даних представлений на рис. 5.

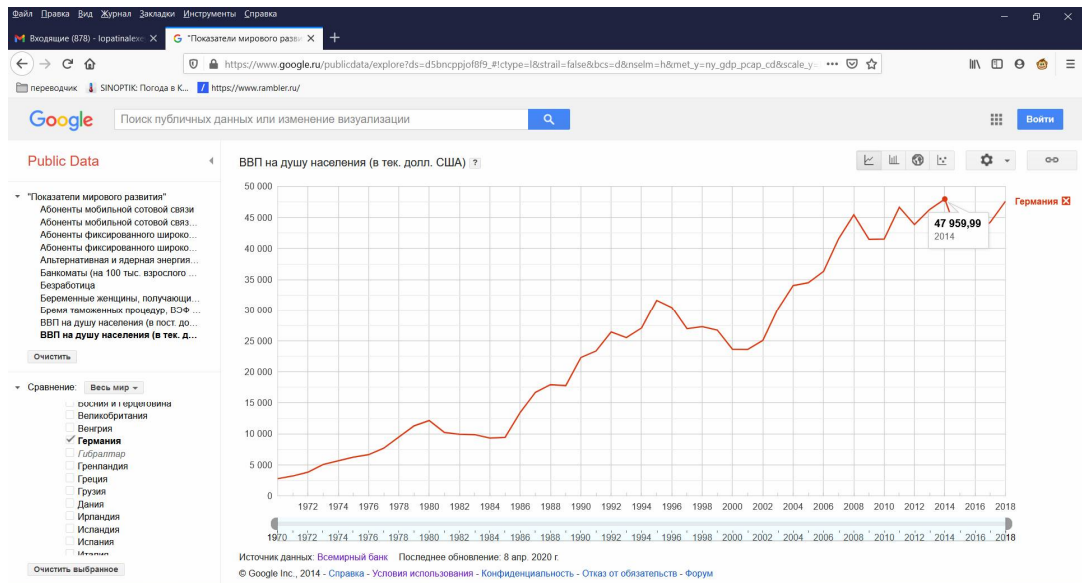


Рисунок 5. ВВП душу населения (в тек. дол. США) Німеччина 1972-2018 роки

Таблиця 1. Вихідні дані

No	y(t)	No	y(t)	No	y(t)	No	y(t)	No	y(t)
1	3,809	11	9,914	21	26,438	31	25,075	41	43,958
2	5,047	12	9,864	22	25,523	32	30,244	42	46,286
3	5,639	13	9,313	23	27,077	33	33,044	43	47,96
4	6,236	14	9,43	24	31,568	34	30,507	44	41,14
5	6,635	15	13,462	25	30,486	35	36,523	45	42,099
6	7,682	16	16,678	26	26,964	36	41,587	46	44,24
7	9,482	17	17,931	27	27,289	37	45,427	47	47,693
8	11,281	18	17,764	28	26,726	38	41,435		
9	12,138	19	22,304	29	23,636	39	41,532		
10	10,2	20	22,358	30	23,608	40	46,645		

Тут  $t$  – номер року,  $y(t)$  – ВВП на душу населення (в тек. дол. США).

## Крок 2. Графічна побудова S-тренду Верхульста

Як функцію апроксимації вибираємо S-криву Верхульста:

$$y'(t) = f(t) = u + \frac{A}{1 + B * \exp(-a(t - m))} \quad (1)$$

Тут  $A$ ,  $B$ ,  $a$ ,  $m$  – параметри, що змінюють форму S-кривий,  $u$ ,  $m$  – параметри, що змінюють положення S-кривої в першій чверті.

Алгоритм складається із серії послідовних наближень:

1) Змінюємо параметри кривої з метою її проходження через точкову діаграму вихідних даних з найменшими відхиленнями. Цей етап проводиться методом спроб і помилок та оцінюються візуально та кількісно за критерієм MAPE\*.

2) Проводиться серія подібних наближень до найменшого значення MAPE.

$$* \text{MAPE} = \frac{100\%}{N} \sum_{t=1}^N \frac{|y(t) - y'(t)|}{y(t)}$$

Для ілюстрації алгоритму наведемо результати трьох наближень.

Таблиця 2. Параметри тренду 1 наближення

A	B	a	m	u	MAPE%
126	45	0,15	-30	-6,4	38,2051

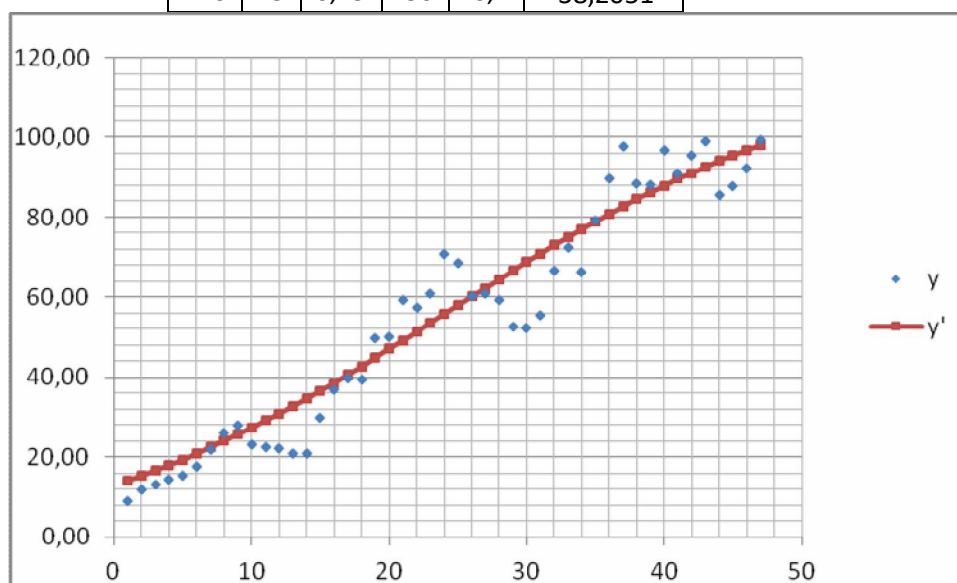


Рисунок 6. 1-е наближення. Таблиця 2 представляє параметри тренду

Таблиця 3. Параметри тренду 2 наближення

A	B	a	m	u	MAPE%
126	60	0,1	-30	-6,4	15,70

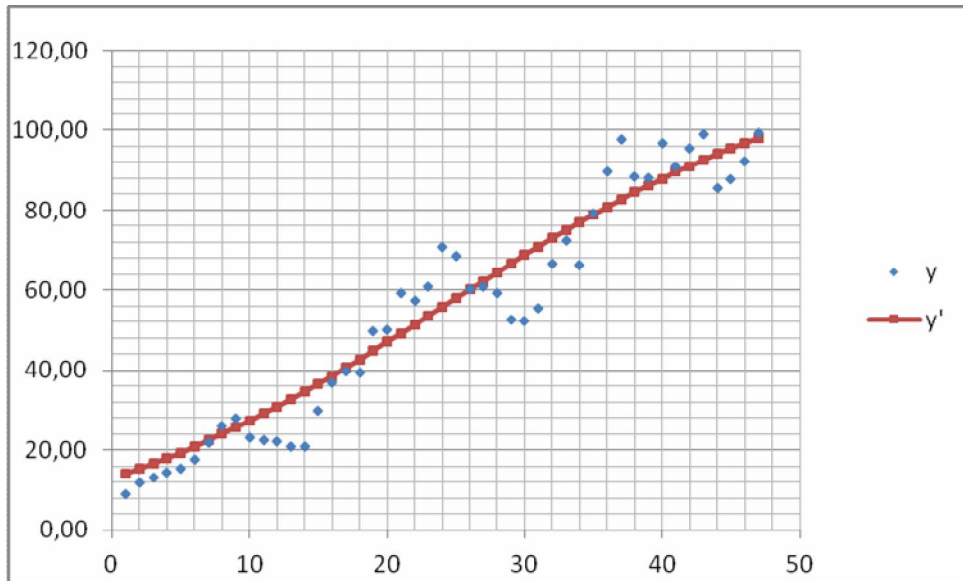


Рисунок 7. 2-е наближення. Таблиця 3 представляє параметри тренду

Таблиця 4. Параметри тренду 3 наближення

A	B	a	m	u	MAPE%
126	45	0,07	-30	-6,4	13,19

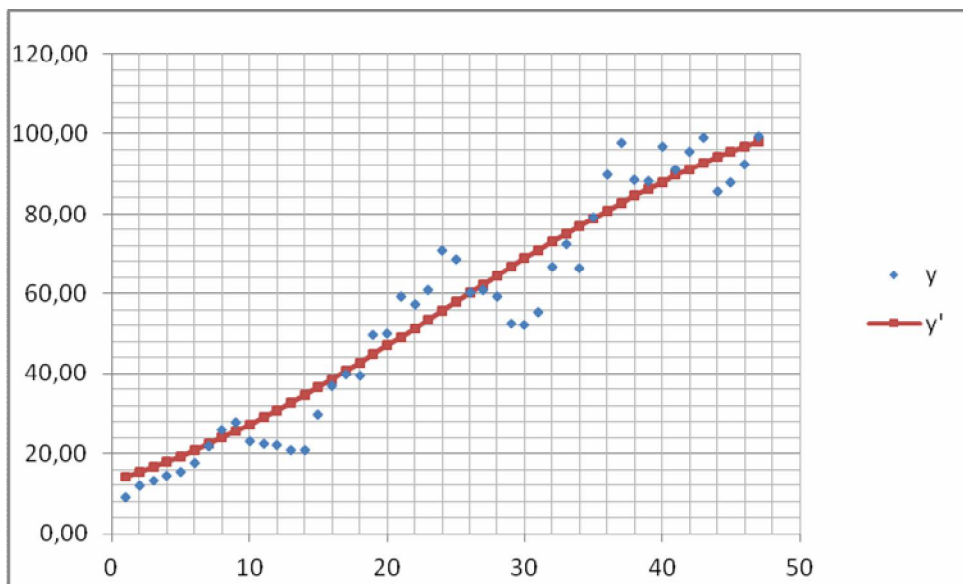


Рисунок 8. 3-е наближення. Таблиця 4 представляє параметри тренду

**Зауваження 4.** Подальше поліпшення наближень можна отримати шляхом нелінійної регресії (див., наприклад, статистичний пакет Minitab).

### Крок 3. Побудова прогнозу на 4 роки наперед

Побудова прогнозу за допомогою побудованої Логісти. Для цього у формулу (1) слід підставити значення  $N+1$ ,  $N+2$ , ... Результат розрахунків наведено в таблиці 5.

Таблиця 5. Побудований прогноз

t	рік	прогноз
48	2019	99,36
49	2020	100,52
50	2021	101,62
51	2022	102,68

### Крок 4. Знаходження точки перегину кривої тренда

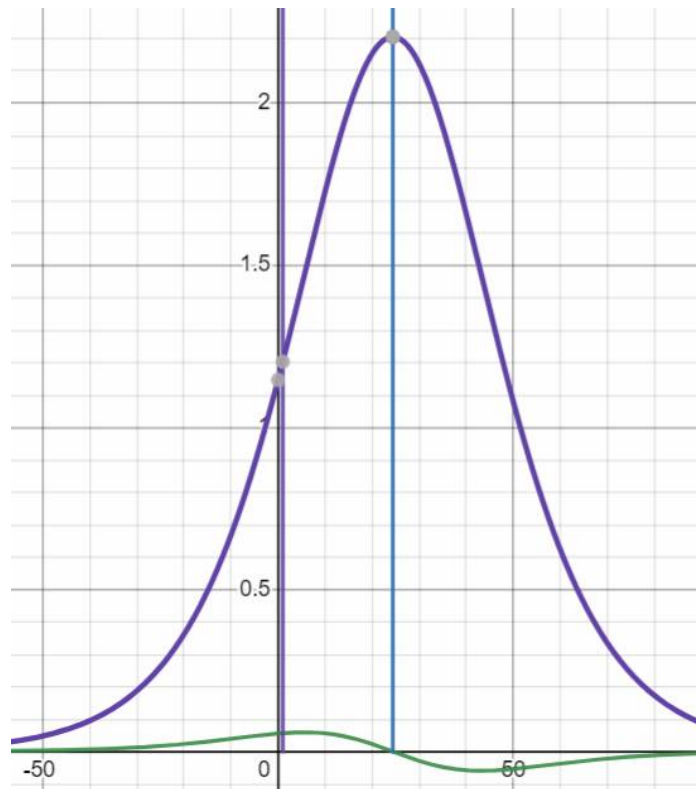


Рисунок 9. 1-а та 2-а похідні S-тренду Верхульста

- Як видно з рис. 9, точка перегину має координату (24.38, 2.2) або (1995, 2.2)
- Висновки:
- Запропонована логіста дає хорошу апроксимацію даних зростання ВВП на душу населення Німеччини и на період 1972-2018 років із метрикою MAPE=13%
- Точка перегину має координати: (1995, 2.2)
- З великою мірою надійності (87%) можна стверджувати, що зростання ВВП на душу населення почало зменшуватися з 1995 року.

- За досліджуваний період є два періоди (один із них досить явний) спаду економіки (1980-1984) та (1995-2001), зумовлених міжнародними причинами.
- На кривій, що розглядається, ці факти виявляються як відхід даних з апроксимуючою кривою.
- Можливий прогноз на кінцевому інтервалі до перетину логісти з бісектрисою з високою надійністю.
- Форма кривої, що розглядається, вказує на той факт, що справжній науково-технічний рівень даного регіону вичерпав свої можливості і повинен перейти на новий науково-технічний та економічний рівень.

## **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Згуровський М.З., Панкратова Н.Д., Системный анализ: Методология. Проблемы. Приложения, Київ, вид-во Наук. думка, 2005, 743 с.
2. Семёнычев В.К., Семёнычев Е.В. Параметрическая идентификация рядов динамики: структуры, модели, эволюция: монография. Самара: , вид-во «СамНЦ РАН», 2011. 364 с. ISBN 978-5-93424-558-1
3. Cıpra T. Time Series in Economics and Finance. Springer, 2020. 353 с.
4. Бідюк П. І., Романенко В. Д., Тимошук О. Л. Аналіз часових рядів : навч. посіб. Київ, Київ обл. : Політехніка, 2010. 317 с.
5. Montgomery D. C., Jennings C. L., Kulahci M. Introduction to Time Series Analysis and Forecasting. Wiley & Sons, Incorporated, John, 2011. 472 с.
6. Kotu V., Deshpande B. Data Science: Concepts and Practice. Morgan Kaufmann, 2018. 568 с.
7. Matt Goldenberg, S-Curves for Trend Forecasting, <https://www.lesswrong.com/posts/oaqKjHbgsoqEXBMZ2/s-curves-for-trend-forecasting>
8. Rogers, E.M. Diffusion of Innovations. (Free Press, New York, 2003), 512. ISBN 0743222091
9. Lopatin Alexey K., Technology progress implementation based on a modified version of R.M. Solow economic growth model: with production s-curve consisting of n-steps, Системні дослідження та інформаційні технології, 2021, № 2, 1-12, DOI: 10.20535/SRIT.2308-8893.2021.2.01



# АЛГОРИТМИ НАДАННЯ РЕКОМЕНДАЦІЙ НА ОСНОВІ НАБОРІВ ДАНИХ ТРАНЗАКЦІЙ

Недашківська Н.І.<sup>1</sup>, Деньгуб Д.<sup>2</sup>, Лавріненко В.<sup>3</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> nedashkovskaya.nadezhda@lil.kpi.ua [0000-0002-8277-3095],

<sup>2</sup> denhub.daria@lil.kpi.ua, <sup>3</sup> lavrinenko.veronika@lil.kpi.ua

**Розглядається задача знаходження асоціативних правил на основі бази даних транзакцій. Досліджено відомі алгоритми Apriori, Eclat і FP-growth отримання частих наборів об'єктів з метою їх подальшого використання для побудови асоціативних правил. Виконано порівняння результатів, отриманих алгоритмами Apriori, Eclat і FP-growth на двох вхідних наборах даних транзакцій залежно від різних значень вхідних параметрів цих алгоритмів; якість отриманих асоціативних правил оцінено за кількома показниками. Розроблено програмне забезпечення для побудови прогнозів та рекомендацій на основі даних транзакцій.**

**Ключові слова:** асоціативне правило, рекомендаційна система, алгоритм frequent-pattern growth

## 1. ВСТУП

Аналіз ринкових кошиків – це виявлення наборів товарів, які часто купляються разом або ніколи не купляються разом на основі бази даних транзакцій. Метою такого аналізу може бути формування персональних рекомендацій, оптимізація розміщення товарів на полицях, планування рекламних компаній, більш ефективного управління цінами і асортиментом. Суть аналізу ринкових кошиків полягає у побудові асоціативних правил, які в подальшому використовуються як для опису предметної області, так і для побудови прогнозу та рекомендації наступного товару споживачеві.

Задача знаходження асоціативних правил на основі бази даних транзакцій розбивається на дві: пошук частих наборів товарів та побудова правил за знайденими частими наборами. Використовуються кілька алгоритмів отримання частих наборів товарів. Найбільш популярні – Apriori [1], Eclat [2] і FP-growth [3].

Першим алгоритмом отримання всіх частих наборів об'єктів (товарів) в базі даних транзакцій був Apriori. Ключовою у ньому є властивість антимонотонності [1]: якщо деякий набір об'єктів  $F$  не є частим, то додавання нового об'єкту до набору  $F$  не робить результуючий набір більш частим. З іншого боку, якщо деякий набір об'єктів  $F$  частий, то всі його підмножини також часті. Таким чином, властивість антимонотонності значно зменшує простір пошуку асоціативних правил. Для усунення окремих недоліків, властивих Apriori, було запропоновано алгоритм Eclat [2]. Він заснований на тій самій властивості антимонотонності, що й Apriori, але створення наборів об'єктів-кандидатів здійснюється використовуючи рекурсивний пошук у глибину (depth-first search).

Алгоритм FP-growth (frequent-pattern growth) не потребує проміжного етапу створення кандидатів у часті набори. FP-growth використовує деревоподібну структуру даних – FP-дерево [3]. База даних з транзакціями представляється в компактному і узагальненому вигляді FP-дерева, і часті набори шукаються на основі елементів цієї структури без

звернення до початкової бази транзакцій. В останні роки пропонуються модифікації алгоритму FP-growth [4, 5] та нові алгоритми на основі FP-growth [6]. Розглядається паралельна реалізація алгоритму FP-growth у середовищі Hadoop MapReduce і для неї пропонується стратегія балансування навантаження [7]. Утворюють комплексні і, як наслідок, більш надійні та цікаві асоціативні правила на базі FP-growth та Apriori шляхом ансамблювання простіших правил за технологією бустингу [8].

Алгоритм FP-growth було використано також у сферах автоматизації охорони здоров'я [9] та розумного (smart) енергозбереження [10].

## 2. МАТЕРІАЛИ ТА МЕТОДИ

### 2.1. Алгоритм Apriori

Нехай  $L_k$  – множина  $k$ -елементних частих наборів:  $L_k = \{F_j \mid F_j = \{i_1, i_2, \dots, i_k\}, \text{Supp}(F_j) \geq \text{Supp}_{\min}\}$ ,  $C_k$  – множина кандидатів в  $k$ -елементні часті набори, отримана шляхом зв'язування множини  $L_{k-1}$  з собою, причому  $k$ -набори зв'язувані, якщо мають спільні перші  $k-1$  елементів.

Етапи алгоритму Apriori можна умовно поділити на наступні:

1. Генерація кандидатів в часті набори з одночасним скороченням множини кандидатів, використовуючи властивість антимонотонності.

2. Відбір частих кандидатів на основі параметру мінімальної підтримки  $\text{Supp}_{\min}$ .

Алгоритм Apriori перебирає всі часті підмножини деякої множини. Спочатку розглядаються одноелементні часті набори  $L_1$ . Далі до них додається по одному об'єкту, тобто формуються двоелементні множини. Нечасті набори відкидаються на основі параметру  $\text{Supp}_{\min}$ , так поступово нарощується потужність наборів, фільтруючи від нечастих наборів [1].

Алгоритм Apriori рекурсивний. На поточній його ітерації шукаємо часті набори потужності  $k$ , за умови, що часті набори потужності  $k-1$  нам відомі (вони отримані на попередній ітерації). Розглядаємо по одному елементу  $F$  з  $(k-1)$ -елементного частого набору  $L_{k-1}$ , та по одному об'єкту  $i$  з множини  $L_1 \setminus F$ , формуємо кандидата в часті набори – множину  $F \cup \{i\}$ . Якщо підтримка цього об'єднання не менша параметра  $\text{Supp}_{\min}$ , то записуємо це об'єднання в  $L_k$ . Якщо на поточній ітерації не знайдено жодного частого набору потужності  $k$ , то виходимо з циклу по  $k$ , оскільки за властивістю антимонотонності жодного частого набору в подальшому не отримаємо.

Переваги алгоритму Apriori – простота та швидке зменшення кількості згенерованих кандидатів при встановленні високого значення порогу мінімальної підтримки.

Основний недолік Apriori в тому, що він є дуже ресурсовитратним: потрібно багато разів, а саме на кожній ітерації алгоритму, сканувати базу даних транзакцій для розрахунку значення підтримки поточного аналізованого набору.

### 2.2. Алгоритм Eclat

Алгоритм Eclat сортує елементи у кожному наборі елементів відповідно до визначеного порядку та об'єднує два набори елементів, тільки якщо вони мають спільний елемент, крім останнього. Кожному набору елементів ставиться у відповідність вертикальна структура даних, так звана TID-множина, яка містить список транзакцій, у яких з'являється цей набір елементів.

При такому представленні підтримка набору розраховується безпосередньо на основі відповідної TID-множини, не потребує повторного сканування бази даних і дорівнює відношенню потужності TID-множини до загальної кількості транзакцій.

Позначимо, як і раніше,  $L_k$  – множина  $k$ -елементних частих наборів,  $L_k = \{F_j \mid F_j = \{i_1, i_2, \dots, i_k\}, \text{Supp}(F_j) \geq \text{Supp}_{\min}\}$ . Етапи алгоритму Eclat аналогічні відповідним етапам алгоритму Apriori, крім обчислення підтримки кандидатів.

На першому етапі дані переводяться з горизонтального до вертикального формату, формуються допоміжна множина  $S_1 = \{\{i, M_i\} \mid i \in I, M_i = D_i = \{T_r \mid i \in T_r, r = 1, \dots, m\}\}$  та множина одноелементних частих наборів  $L_1 = \{i \mid i \in I, \text{Supp}(i) \geq \text{Supp}_{\min}\}$ .

На другому етапі для всіх  $k = 2, \dots, n$  генерується  $S_k$  – множина кандидатів в  $k$ -елементні часті набори:

для всіх наборів  $F_a, F_b \in L_{k-1}$

$$S_k = \{\{F', M'_{F'}\} \mid F' = F_a \cup F_b, M'_{F'} = \{M_{F_a} \cap M_{F_b}\}, a < b, M_{F_a}, M_{F_b} \in S_{k-1}\}$$

Відбираються часті набори, формується  $L_k = \{F' \in S_k \mid \text{Supp}(F') \geq \text{Supp}_{\min}\}$ . Якщо  $L_k = \emptyset$ , здійснюється вихід із циклу по  $k$ .

Таким чином, вертикальна структура даних набору елементів-кандидатів  $S_k$  створюється шляхом перетину вертикальних структур  $M_{F_a}$  і  $M_{F_b}$  двох наборів елементів  $F_a$  і  $F_b$ , що об'єднуються, без перегляду бази даних. Це суттєво зменшує ресурсні затрати, оскільки в пам'яті може одночасно зберігатися менше наборів елементів, та є найважливішою перевагою алгоритму Eclat.

### 2.3. Алгоритм FP-росту

В алгоритмі FP-growth БД транзакцій представляється у вигляді компактного дерева – FP-дерева [3]. FP-дерево містить повну інформацію про всі часті набори БД транзакцій і забезпечує ефективне отримання частих наборів на відміну від витратної процедури, яка використовується в Apriori. Алгоритм FP-growth складається з двох загальних етапів:

1. Побудова FP-дерева на основі БД транзакцій.
2. Рекурсивний пошук частих наборів в FP-дереві.

Вершини FP-дерева – об'єкти  $i \in I$ . Причому різні вершини дерева можуть містити одні і ті самі об'єкти. Шлях від кореня  $g_0$  до вершини  $g$  позначає набір об'єктів  $F \subseteq I$ .

Нехай  $G(i) = \{g \in G : g = i\}$  – множина вершин для об'єкта  $i$ . Підтримка об'єкта  $i$  розраховується:

$$\text{Supp}(i) = \sum \text{Supp}(g), \text{ де сума береться за всіма } g \in G(i)$$

Тоді рівні FP-дерева відповідають об'єктам за спаданням величин  $\text{Supp}(i)$ . Таким чином, маємо порядок на множині об'єктів.

Розглянемо детальніше перший етап побудови FP-дерева на основі БД транзакцій. Дано:  $D = \{T_1, T_2, \dots, T_m\}$  – множина транзакцій  $\square$  – навчальна вибірка. Результуюче FP-дерево  $G$  представляється у вигляді:

$$G = \{g \mid g = (\text{Name}(g), \text{Supp}(g), \text{Child}(g))\},$$

де вузол  $g$  – структура, яка зберігає значення вузла  $\text{Name}(g)$ , значення  $\square$  підтримки  $\text{Supp}(g)$ , а також посилання на всі його дочірні елементи  $\text{Child}(g)$ .

Для побудови FP-дерева спочатку об'єкти  $i \in I$  впорядковуються за спаданням значення підтримки  $\text{Supp}(i)$ . Виконується фільтрація і вилучаються об'єкти зі значенням підтримки, меншим за заданий поріг мінімальної підтримки  $\text{Supp}_{\min}$ .

Для кожного елемента кожної відсортованої транзакції з вхідної БД  $D$  вузли FP-дерева будуються наступним чином:

- якщо для чергового елемента в поточному вузлі є нащадок, який містить цей елемент, то новий вузол не створюється, а підтримка знайденого нащадка збільшується на одиницю;
- в іншому випадку створюється новий вузол-нащадок з підтримкою рівною одиниці;
- поточним вузлом стає знайдений або побудований вузол.

Другий етап алгоритму FP-growth, під час якого безпосередньо шукаються часті набори, використовує поняття умовного FP-дерева.

Умовним  $G' := G \setminus i$  називається FP-дерево, побудоване на основі підмножини транзакцій  $D_i = \{T_k \mid i \in T_k\}$ . Ідея використання умовного дерева в тому, що БД  $D_i$  має набагато менший об'єм порівняно з початковою БД  $D$ . Тому умовне дерево  $G'$  буде, як правило, набагато меншим порівняно з початковим FP-деревом  $G$ . Як наслідок, алгоритм, побудований на FP-деревих, виявляється дуже економним щодо використання ресурсів.

Розглянемо FP-дерево  $G$  та зафіксуємо об'єкт  $i \in I$ . Щоб знайти умовне FP-дерево  $G' := G \setminus i$  для об'єкта  $i$  спочатку в дереві  $G$  треба вилучити шляхи, які не містять об'єкт  $i$ . Після цього в дереві  $G$  вилучити потомки вершин, які відповідають об'єкту  $i$ . Далі перерахувати підтримку всіх вузлів  $g \in G$ , що залишилися в  $G$ :  $Supp(g) := \sum Supp(x)$ , де сума береться по  $x \in Child(g)$ . На останньому етапі в дереві  $G$  вилучити вершини, які відповідають об'єкту  $i$ . Результуюче дерево буде шуканим умовним FP-деревом.

### 3. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ АЛГОРИТМІВ І ПОРІВНЯННЯ РЕЗУЛЬТАТІВ

Вхідними даними обрано два набори транзакцій.

Для кожного з цих наборів знайдено множини частих наборів та побудовано асоціативні правила для різних значень вхідних параметрів:  $Supp_{min}$  – мінімальної підтримки,  $max\_len$  – максимальної кількості елементів в наборі та  $Conf_{min}$  – мінімальної достовірності.

Кількість частих наборів, знайдених алгоритмами Apriori, Eclat і FP-growth, суттєво залежить від значення  $Supp_{min}$ : якщо  $Supp_{min}$  зменшуємо, кількість частих наборів буде більшою, і навпаки (Таблиці 1, 2).

Поріг мінімальної достовірності  $Conf_{min}$  має найбільший вплив на кількість отриманих асоціативних правил за умови, що множина частих наборів була отримана на попередньому етапі. Параметр  $max\_len$  практично не вплинув на результат для обох розглянутих вхідних наборів даних транзакцій (Таблиці 1, 2). Набори, що складаються з двох об'єктів (товарів), частіше зустрічалися порівняно з наборами розміром три і вище. Тому бути впевненим в зв'язку покупки двох товарів одночасно легше ніж покупки одночасно трьох і більшої кількості товарів.

Мінімальне значення lift збільшилося при збільшенні значення параметру  $Conf_{min}$  (Таблиці 1, 2). Показник lift – один з основних, які використовуються для аналізу якості побудованих асоціативних правил. Відомо, що правило корисніше за вгадування, якщо значення lift перевищує одиницю. Отже, всі правила, побудовані зі вказаними в Таблиці 1 значеннями параметрів, корисні для подальшого прогнозування та надання рекомендацій.

Таблиця 1. Результати застосування алгоритму FP-growth, перший набір транзакцій

№	Вхідні значення параметрів	Кількість частих наборів	Кількість правил	Мінімальне значення lift
1	$Supp_{min} = 0.02, max\_len = 2, Conf_{min} = 0.3$	120	32	1.205
2	$Supp_{min} = 0.02, max\_len = 2, Conf_{min} = 0.4$	120	12	1.572
3	$Supp_{min} = 0.02, max\_len = 5, Conf_{min} = 0.3$	122	37	1.205
4	$Supp_{min} = 0.02, max\_len = 5, Conf_{min} = 0.4$	122	15	1.572
5	$Supp_{min} = 0.03, max\_len = 2, Conf_{min} = 0.3$	64	14	1.205
6	$Supp_{min} = 0.03, max\_len = 2, Conf_{min} = 0.4$	63	5	1.572
7	$Supp_{min} = 0.03, max\_len = 5, Conf_{min} = 0.3$	63	14	1.205
8	$Supp_{min} = 0.03, max\_len = 5, Conf_{min} = 0.4$	63	5	1.572

9	$Supp_{min} = 0.05, max\_len = 2, Conf_{min} = 0.3$	31	3	1.205
10	$Supp_{min} = 0.05, max\_len = 2, Conf_{min} = 0.4$	31	1	1.572
11	$Supp_{min} = 0.05, max\_len = 5, Conf_{min} = 0.3$	31	3	1.205
12	$Supp_{min} = 0.05, max\_len = 5, Conf_{min} = 0.4$	31	1	1.572

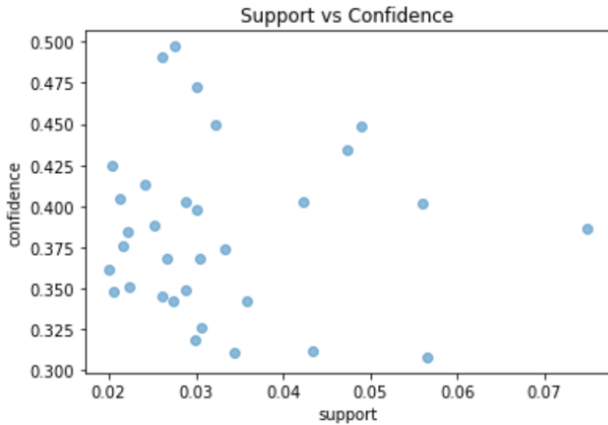
Таблиця 2. Результати застосування алгоритму FP-growth, другий набір транзакцій

№	Вхідні значення параметрів	Кількість частих наборів	Кількість правил	Діапазон зміни значень показника lift
1	$Supp_{min} = 0.05, max\_len = 2, Conf_{min} = 0.2$	11	2	[0.575, 1.102]
2	$Supp_{min} = 0.05, max\_len = 3, Conf_{min} = 0.2$	11	2	[0.575, 1.102]
3	$Supp_{min} = 0.05, max\_len = 4, Conf_{min} = 0.2$	11	2	[0.575, 1.102]
4	$Supp_{min} = 0.05, max\_len = 2, Conf_{min} = 0.5$	11	1	1.102
5	$Supp_{min} = 0.05, max\_len = 3, Conf_{min} = 0.5$	11	1	1.102
6	$Supp_{min} = 0.05, max\_len = 4, Conf_{min} = 0.5$	11	1	1.102
7	$Supp_{min} = 0.01, max\_len = 2, Conf_{min} = 0.2$	58	26	[0.575, 1.605]
8	$Supp_{min} = 0.01, max\_len = 3, Conf_{min} = 0.2$	61	31	[0.575, 1.938]
9	$Supp_{min} = 0.01, max\_len = 4, Conf_{min} = 0.2$	61	31	[0.575, 1.938]
10	$Supp_{min} = 0.01, max\_len = 2, Conf_{min} = 0.5$	58	11	[1.060, 1.472]
11	$Supp_{min} = 0.01, max\_len = 3, Conf_{min} = 0.5$	61	11	[1.060, 1.472]
12	$Supp_{min} = 0.01, max\_len = 4, Conf_{min} = 0.5$	61	11	[1.060, 1.472]
13	$Supp_{min} = 0.3, max\_len = 2, Conf_{min} = 0.2$	2	0	-
14	$Supp_{min} = 0.3, max\_len = 4, Conf_{min} = 0.2$	2	0	-
15	$Supp_{min} = 0.3, max\_len = 2, Conf_{min} = 0.5$	2	0	-
16	$Supp_{min} = 0.3, max\_len = 4, Conf_{min} = 0.5$	2	0	-

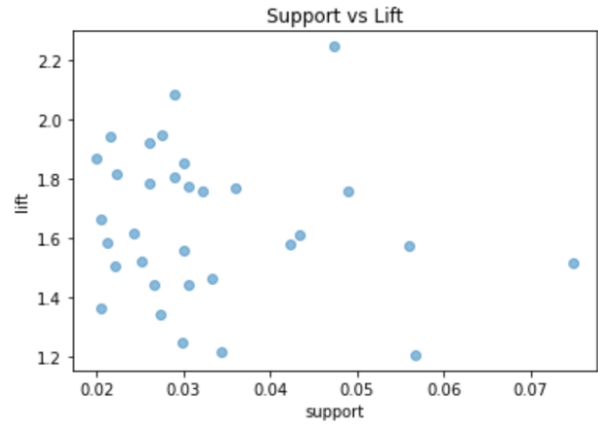
Значення достовірності (confidence) та підтримки (support), ліфта (lift) та підтримки (support) для множини асоціативних правил, отриманих на першому наборі даних, графічно відображено на Рисунках 1–3 для різних значень вхідних параметрів. З Рисунків 1 і 3 бачимо, що більшість знайдених правил отримано при значеннях  $Supp_{min}$  з інтервалу [0.010, 0.035]. Всі ці правила корисні, оскільки  $lift > 1$ .

У результаті застосування Apriori формується велика кількість кандидатів у часті набори при великому розмірі бази даних транзакцій або при низькому значенні порогу мінімальної підтримки  $Supp_{min}$ .

Алгоритм Eclat так як і Apriori, може витрачати значну кількість часу на створення наборів елементів-кандидатів, яких немає в базі даних транзакцій.

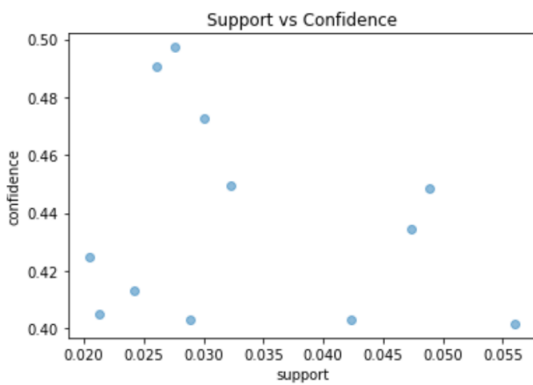


(а)

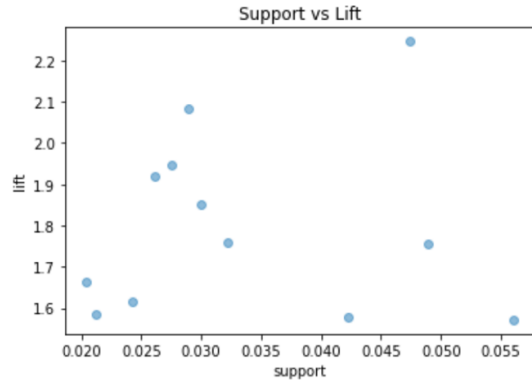


(б)

Рисунок 1. Значення показників якості (confidence, support) (а) та (lift, support) (б) для асоціативних правил, знайдених для  $Supp_{min} = 0.02$ ,  $max\_len = 2$  і  $Conf_{min} = 0.3$

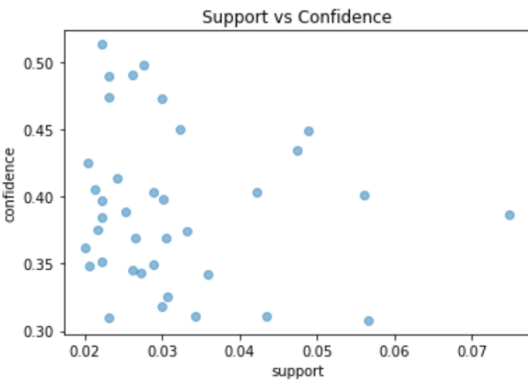


(а)

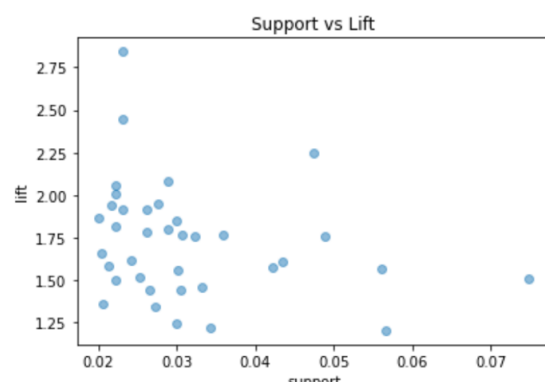


(б)

Рисунок 2. Значення показників якості (confidence, support) (а) та (lift, support) (б) для асоціативних правил, знайдених при  $Supp_{min} = 0.02$ ,  $max\_len = 2$  і  $Conf_{min} = 0.4$



(а)



(б)

Рисунок 3. Значення показників якості (confidence, support) (а) та (lift, support) (б) для асоціативних правил, знайдених при  $Supp_{min} = 0.02$ ,  $max\_len = 5$  і  $Conf_{min} = 0.3$

Виконаємо прогнозування і знайдемо які товари доцільно рекомендувати споживачу в якості наступної покупки (Рисунки 4 і 5). Нехай поточною покупкою є тости (Toast). Як видно з Рисунку 4, у побудованій множині асоціативних правил було знайдено лише одне

правило з умовою «тости (Toast)». Значення показника ліфта (lift) цього правила перевищує одиницю, тому правило корисне. На основі вказаного правила робиться висновок, що споживачу слід рекомендувати купити додатково каву (Coffee).

	antecedents	consequents	antecedent support	consequent support	support confidence	lift	leverage	conviction	
92	(Toast)	(Coffee)	0.033597	0.478394	0.023666	0.704403	1.472431	0.007593	1.764582

Рисунок 4. Асоціативне правило з умовою «тости (Toast)» та значення показників якості lift, leverage, conviction цього правила

	antecedents	consequents	antecedent support	consequent support	support confidence	lift	leverage	conviction	
31	(Pastry)	(Coffee)	0.086107	0.478394	0.047544	0.552147	1.154168	0.006351	1.164682
33	(Pastry)	(Bread)	0.086107	0.327205	0.029160	0.338650	1.034977	0.000985	1.017305

Рисунок 5. Асоціативні правила з умовою «випічка (Pastry)» та значення показників якості lift, leverage, conviction цих правил

Знайдемо прогнозний товар для споживача, який купив випічку (Pastry). Як видно з Рисунку 5, у побудованій множині асоціативних правил було знайдено два правила з умовою «випічка (Pastry)». Значення показника ліфт (lift) обох правил перевищує одиницю, тому правила є корисними. На основі цих правил робиться висновок, що споживачу слід рекомендувати купити додатково каву (Coffee) і хліб (Bread).

#### 4. ВИСНОВКИ

Проаналізовано алгоритми Apriori, Eclat і FP-growth на двох наборах даних транзакцій для задачі аналізу ринкових кошиків та рекомендації наступного товару покупцеві.

Використання алгоритмів Eclat і FP-growth потребує лише одного перегляду бази даних транзакцій. Це суттєва перевага вказаних алгоритмів у випадку надвеликих баз даних.

Експериментально підтверджено, що алгоритм FP-growth працює значно швидше порівняно з Apriori та Eclat.

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Agrawal R., Srikant R. Fast algorithms for mining association rules. *Proc. of the 20th international conference of very large databases VLDB*. 1994. Vol. 1215. P. 487–499.
2. Zaki M. J. Scalable algorithms for association mining. *IEEE Trans. on Knowledge and Data Engineering*. 2000. Vol. 12. P. 372–390.
3. Han J., Pei J., Yin Y., Mao R. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data mining and knowledge discovery*. 2004. Vol. 8. P. 53–87.
4. Lin K.-C., Liao I.-E., Chen Z.-S. An improved frequent pattern growth method for mining association rules. *Expert Systems with Applications*. 2011. Vol. 38, No. 5. P. 5154 – 5161, <https://doi.org/10.1016/j.eswa.2010.10.047>
5. Shabtay L., Fournier-Viger P., Yaari R., Dattner I. A guided FP-Growth algorithm for mining multitude-targeted item-sets and class association rules in imbalanced data *Information Sciences*. 2021. Vol. 553. P. 353–375. <https://doi.org/10.1016/j.ins.2020.10.020>
6. Hu S., Liang Q., Qian H., Weng J., Zhou W., Lin P. Frequent-pattern growth algorithm based association rule mining method of public transport travel stability. *International Journal of Sustainable Transportation*. 2021. Vol. 15, No. 11. P. 879 – 892. <https://doi.org/10.1080/15568318.2020.1827318>

7. Bagui S., Devulapalli K., Coffey J. A heuristic approach for load balancing the FP-growth algorithm on MapReduce. *Array*. 2020. Vol. 7. <https://doi.org/10.1016/j.array.2020.100035>
8. Augello A., Infantino I., Pilato G., Vella F. Sensing the Web for Induction of Association Rules and their Composition through Ensemble Techniques. *Procedia Computer Science*. 2020. Vol. 169. P. 851 – 859. <https://doi.org/10.1016/j.procs.2020.02.152>
9. Odu N. B., Prasad R., Onime C., Sharma B. K. How to implement a decision support for digital health: Insights from design science perspective for action research in tuberculosis detection. *International Journal of Information Management Data Insights*. 2022. Vol. 2, No. 2. <https://doi.org/10.1016/j.jjime.2022.100136>
10. Qi R., Guo X. Analysis of Intelligent Energy Saving Strategy of 4G/5G Network Based on FP-Tree. *Procedia Computer Science*. 2022. Vol. 198. P. 486-492.



# **МЕТОД РЕФІНАНСУВАННЯ В УПРАВЛІННІ КРЕДИТНИМ РИЗИКОМ ІПОТЕЧНОГО ПОРТФЕЛЯ КОМЕРЦІЙНОГО БАНКУ**

Півень О.К.<sup>1</sup>, Стулей В.А.<sup>2</sup>

Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”, Київ, Україна

<sup>1</sup> sandrpiven@gmail.com

<sup>2</sup> stuley.volodymyr@lil.kpi.ua

**Метою роботи є розробка методики управління кредитним ризиком іпотечного портфеля методом рефінансування, що ґрунтується на попередньому оцінюванні кредитного ризику портфеля на груповій основі, використовуючи статистичні методи аналізу в моделях виживаності: метод Каплана-Майєра та метод пропорційних ризиків Кокса. Результатом дослідження є пакет рішень, що реалізують метод рефінансування за умови наявності надлишкового попиту платоспроможних позичальників.**

**Ключові слова: кредитний ризик, іпотечний портфель, метод Каплана-Майєра, регресія Кокса, базова функція дожиття.**

## **1. ВСТУП**

Іпотечні кредити в Україні за даними 2020 року склали до 16% від загального портфеля кредитів домогосподарствам, в той час як у Європейському Союзі частка іпотек у загальному обсязі кредитів займає 78,8% [1]. Така відмінність між українською та європейською структурою ринку свідчить про великий потенціал іпотечного кредитування в нашій країні, особливо враховуючи євроатлантичний курс України та майбутні післявоєнні економічний розвиток і відбудову. Ці фактори обумовлюють актуальність обраної тематики досліджень. Банківська система України має бути готова до очікуваного стрімкого зростання частки іпотечних кредитів в банківських кредитно-інвестиційних портфелях і вже зараз вивчати та впроваджувати адаптивні методи управління іпотечним кредитним ризиком портфельними методами.

На сьогодні вітчизняні банки вже мають напрацьовану практику та налагоджені системи портфельного управління кредитними ризиками споживчого кредитування. Проте на відміну від споживчих кредитів, іпотечні кредити мають багато особливостей, які суттєво ускладнюють портфельне оцінювання ризику: застава, вартість якої змінюється в часі, значні терміни погашення боргу, що обумовлює високу чутливість кредитного ризику до впливу екзогенних макроекономічних чинників тощо.

Відтак постає проблематика застосування для великих пулів іпотечних кредитів портфельних методів управління кредитними ризиками.

## **2. УПРАВЛІННЯ КРЕДИТНИМ РИЗИКОМ ІПОТЕЧНОГО ПОРТФЕЛЯ**

Особливості іпотечного портфелю, які названі вище, обумовлюють специфічні методи портфельного оцінювання ризиків. За таких умов практично єдиним адекватним способом аналізу є базові підходи – метод Каплана-Майєра та метод пропорційних ризиків Кокса.

З іншого боку в порівнянні зі споживчим кредитом списання іпотеки за рахунок нарахованих резервів набагато складніше. В результаті „погані” іпотечні кредити набагато довше можуть залишатися в портфелі банку, що посилює пруденційні вимоги до нього з боку регулятора.

Для вирішення цієї проблеми пропонується замінювати позичальника для тих іпотек, що поки не є дефолтними, але на звітну дату мають підвищений кредитний ризик по відношенню до загального (портфельного) рівня. Заміна позичальника здійснюється шляхом викупу в нього кредитних зобов'язань іншим позичальником за певним посередництвом банку. Оскільки банк кредитує нового позичальника, цей спосіб управління ризиком називається методом рефінансування. Очікується, що послідовне застосування такого методу в результаті буде знижувати загальний ризик іпотечного портфелю. Очевидно, що метод рефінансування може досягти цілей лише за умови існування надлишкового попиту на іпотечні кредити з боку платоспроможних позичальників.

Таким чином, метод рефінансування передбачає виконання таких етапів:

1. Портфельна оцінка базового кредитного ризику іпотек.
2. Пошук статистично значущих коваріат, які визначають підвищений кредитний ризик.
3. Виділення на кожну звітну дату з іпотечного портфеля субпортфель кредитів з підвищеним кредитним ризиком.
4. Проведення рефінансування субпортфеля кредитів з підвищеним кредитним ризиком протягом певного об'єму планування.

Для виконання 1 етапу використовується метод Каплана-Майєра, а для 2 – метод пропорційних ризиків Кокса. Ці методи широко використовуються в медицині для аналізу виживаності пацієнтів, тому досить часто називаються методами аналізу виживаності.

В даному аспекті ці методи використовуються для статистичного моделювання процесу настання дефолту позичальника на основі аналізу таких подій в когорті спостережень за „виживаністю” іпотечного кредиту. Найбільш важливою особливістю, що відрізняє даний аналіз від інших статистичних методів, є можливість дослідження двох аспектів задачі: аналіз настання події та часовий проміжок до її настання.

Основними компонентами аналізу виживаності, як і будь-якої моделі тривалості, є функція дожиття та функція ризику. Функція дожиття відображає ймовірність того, що банкрутство позичальника не відбудеться як мінімум  $t$  одиниць часу. Функція ризику – ризик дефолту у наступний період часу при умові, що боржник проіснував без дефолту  $t$  одиниці часу [2].

Метод Каплана-Мейєра використовується для оцінки частки об'єктів спостереження (боржників), у яких подія (дефолт) не відбулася, для будь-якого моменту часу протягом усього періоду спостереження. Оцінка функції дожиття в методі Каплана-Майєра є добутком дожиття в момент часу на дожиття в наступний момент часу, коли подія відбулась.

Метод пропорційних ризиків Кокса спирається на побудову регресійної моделі для відносного ризику. В якості вихідного параметра модель повертає значення відношення ризиків і його довірчий інтервал [3].

Застосування цих методів останнім часом стимулювалось впровадженням Міжнародним стандартом фінансової звітності (МСФЗ 9) [4]. Ці методи дозволяють статистично адекватно враховувати часову структуру ймовірності дефолту, застосовуючи так звані цензуровані спостереження (фінансові інструменти, які не погашені, тобто знаходяться під ризиком в момент розрахунку).

Модель пропорційних ризиків Кокса, крім цього, дозволяє дослідити вплив кількісних чинників на ризик дефолту. Застосування методів стратифікації когорт дозволяє виділити статистично однорідні групи за вимогами МСФЗ 9. Розбиття портфеля на окремі страти

(сегменти), що статистично розрізняються показниками ймовірності дефолту, дозволяє розрахувати групові значення такої ймовірності [5].

Таким чином на 1 етапі запропонованого метода рефінансування для кожного сегменту розраховується кумулятивне значення ймовірності дефолту  $kPD_{TTC}$  на основі базових функцій «групового дожиття» Каплана-Майєра фінансових інструментів певного сегменту за формулою:  $kPD_{TTC} = 1 - S_{Tm}$ , де  $Tm$  – оцінка кількості місяців спостережень за дефолтами позичальників (медіани часу спостереження).

На 2 етапі для стратифікованого кредитного портфеля методом пропорційних ризиків Кокса визначаються статистично значущі коваріати ризику. Надалі з метою виконання 3 етапу запропонованого метода рефінансування виконується первинний розподіл за кошиками знецінення. Зазвичай для цього використовується критерій низького ризику, запропонований МСФЗ 9. Надалі ж формуються додаткові критерії, за яким фінансовий інструмент може бути перенесений із одного кошика до іншого:

- критерій за кумулятивною ймовірністю дефолту (МСФЗ 9) [4];
- критерій «вікна» можливостей.

В результаті спрацювання подібних критеріїв, змінення кредитного ризику слід скорегувати в бік збільшення  $PD_{TTC}$ , виходячи з аналітичного виразу для функції ризику в моделі пропорційних ризиків Кокса:  $PD_{TTC} = 1 - 1 - PD^\gamma$ , де  $PD$  – базовий ризик сегменту,  $\gamma$  – коефіцієнт, що характеризує ризик, притаманний конкретному позичальнику та залежить від коваріат, які були визначені методом пропорційних ризиків Кокса на етапі 2.

На етапі 4 при підвищенні кредитного ризику фінансового інструменту  $i$ , відповідно, потраплянні його на другу стадію знецінення, застосовується метод рефінансування позичальника. Це дозволяє банку позбавлятися боржників, що мають підвищений ризик дефолту, і знаходити нових, більш платоспроможних клієнтів. Тим самим банк управляє кредитним ризиком свого іпотечного портфеля, намагаючись послідовно зменшити його на кожну звітну дату оцінювання.

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для дослідження, одним з вітчизняних банків, був наданий іпотечний портфель, що охоплює період кредитної діяльності банку 1996 – 2021 років.

Було визначено фактори впливу на «виживаність» фінансових інструментів, за якими досліджуваний портфель був стратифікований на 24 відповідні страти (сегменти). На вказаній стратифікованій когорті проведена оцінка методом пропорційних ризиків Кокса та визначені відповідні коваріати кредитного ризику, що визначають „індивідуалізований” показник  $\gamma$ .

Базові функції дожиття  $S(t)$  були розраховані за моделлю пропорційних ризиків Кокса окремо для кожного сегменту разом з нижньої границею 95% довірчого інтервалу, яка відповідає верхній границі ймовірності дефолту відповідно до визначення функцій  $S(t)$  за Капланом-Майєром [6]. Нижче для прикладу (Рис.1) представлені розрахункові  $S(t)$  криві для іпотек сегментів з термінами погашення меншими ніж 10 років, що мають різні фактори впливу на базовий кредитний ризик: інвестиція (Inv) або вторинний ринок (Buy), періодичний спосіб оплати (P) або ні (E), найманий працівник (W) або ні (unW).

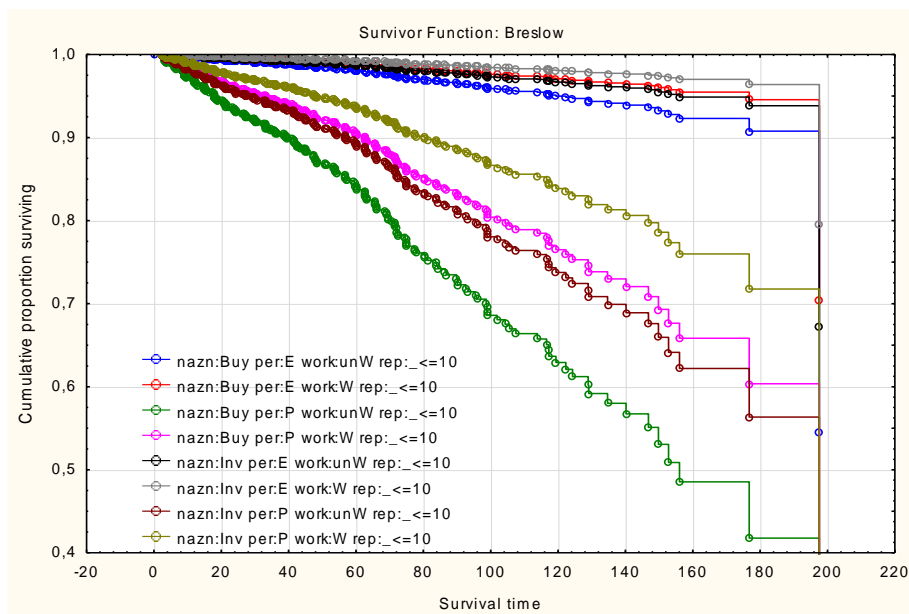


Рисунок 1. Базові функції дожиття для сегментів зі строком погашення до 10 років

Виходячи з результатів розрахунку медіани часу спостереження, за часову основу розрахунку  $PD_{TTC}$  для сегментів приймалась величина, що дорівнює  $T_m = 23$  місяці. Надалі визначалася величина  $kPD_{TTC}$ , яка після очевидного перерахунку на річну основу, є базовим значенням річної ймовірності дефолту  $PD_{TTC}$  відповідного сегменту.

На кожну звітну дату з іпотечного портфелю виділявся субпортфель кредитів з підвищеним кредитним ризиком, щодо якого здійснювалася процедура рефінансування.

Результати розрахунків показані нижче (Рис. 2).



Рисунок 2. Порівняльна динаміка рівнів резервування іпотечного портфеля

На рисунку 2 представлена динаміка рівня резервування іпотечного портфелю, щодо якого з січня 2021 року був застосований метод рефінансування для управління портфельним ризиком, що станом на 31.12.2021 склало 10,8%. Для порівняння пунктиром наведений прогноз рівня резервування у припущенні пасивної еволюції кредитного портфеля, що станом

на 31.12.2021 склало 19,5% (процедура рефінансування не застосовується, проте розрахунок ризику здійснюється аналогічно, як у методі рефінансування).

#### **4. ВИСНОВКИ**

В період після закінчення війни з РФ очікується стрімке зростання частки іпотечних кредитів в балансах вітчизняних комерційних банків, тому виникає необхідність пошуку ефективних та якісних інструментів управління ризиками іпотечних портфелів в умовах надлишкового попиту на кредити платоспроможних позичальників.

Одним із таких інструментів може стати метод рефінансування, що спирається в своїй основі на оцінку кредитного ризику портфельним методом з використанням статистичного аналізу Каплана-Майєра та застосування регресійної моделі Кокса, що дозволяє враховувати особливості іпотечних портфелів, порівняно з портфелями інших фінансових інструментів.

У ході дослідження продемонстровано ефективність застосування запропонованого методу на прикладі іпотечного портфеля одного з комерційних банків України. Фактичний рівень резервування склав 10,8% проти прогнозного рівня 19,5% станом на 31.12.2021, а середній рівень резервування за 2021 рік знизився на 25,5%.

#### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Дослідження ринку кредитування в Україні: презентація. URL: [https://nabu.ua/images/uploaded/sys\\_media\\_doc/doc\\_51de4af05909c502efee8558d70b54b0.pdf](https://nabu.ua/images/uploaded/sys_media_doc/doc_51de4af05909c502efee8558d70b54b0.pdf)
2. Abad R.C., Fernández J.M.V., Rivera A.D. Modelling consumer credit risk via survival analysis. 2009. Т. 33. № 1. Р. 3–30.
3. Aharony J., Swary I. A note on corporate bankruptcy and the market model risk measures. 1988. Т. 15. № 2. Р. 275–281.
4. МСФЗ 9 Фінансові інструменти. URL: [https://mof.gov.ua/storage/files/IFRS\\_9\\_Ukrainian-compressed.pdf](https://mof.gov.ua/storage/files/IFRS_9_Ukrainian-compressed.pdf)
5. Perperoglou A., Cessie S. le, Houwelingen H.C. van. A fast routine for fitting Cox models with time varying effects of the covariates. 2006. Т. 81. № 2. Р. 154–161.
6. Thomas L., Reyes E.M. Tutorial: survival estimation for Cox regression models with time-varying coefficients using SAS and R. 2014. Т. 61.

# СИСТЕМНИЙ АНАЛІЗ ФІНАНСОВИХ РИЗИКІВ

Саркісян А.О.<sup>1</sup>, Бідюк П.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

<sup>1</sup>a.s.ohanesovna@gmail.com

**Фінансові ризики (ФР) є невід’ємною складовою ведення фінансової діяльності та будь-якої іншої діяльності пов’язаною з грошовими операціями. ФР – це безпосередній наслідок невизначеностей, тому для ефективного керування та визначення потенційних операційних ризиків доцільно застосовувати підходи системного аналізу, одним із яких є побудова інтелектуальної системи підтримки прийняття рішень (ІСПР), яка дає можливість динамічно змінювати підхід до аналізу та полегшує прогнозування ризиків. Метою роботи є аналіз операційних ризиків за допомогою системного аналізу, представленого у створенні СПР, що застосовуватиме байєсівські підходи для прогнозування та оцінювання ризиків. Результатом дослідження є створена СПР, що оцінює операційні ризику валютного і грошового ринку із застосуванням методології байєсівських мереж, методу Монте-Карло, VaR та CVaR. У роботі використано теоретичні та емпіричні методи дослідження.**

**Ключові слова: фінансові ризики, операційні ризики, системний аналіз, байєсівські мережі, аналіз ризиків.**

## 1. ВСТУП

Операційний ризик – це будь-який ризик пов’язаний з потенційною втратою капіталу чи інвестицій, який є актуальним для кожної фізичної чи юридичної особи, у розпорядженні якої є певним грошовий капітал. Операційний ризик, як і всі інші види ризику, пов’язаний у першу чергу з невизначеністю, яка його обумовлює. Невизначеність, в свою чергу, формує випадковість ризику як досліджуваної величини. Частіше за все поняття операційного ризику асоціюється з банківською справою, а саме з валютним та грошовими ринками, проте отримати достовірні дані обробки операційних ризиків із банківської справи є досить важко через високий рівень конфіденційності та неможливість чітко відрізнити операційний ризик від ринкового чи кредитного. Також з обробкою даних по операційних ризиках у банківській справі пов’язані складність кількісного оцінювання ризику і нетривалість часу послідовного збору даних.

З огляду на це, найважливішою вимогою застосування системного підходу є міждисциплінарна компетентність, адже для аналізу ризику важливо розуміти його природу походження, залежність та інтегрованість від зовнішніх факторів. Оскільки навчити фахівця бачити і прогнозувати залежності між випадковими подіями чи об’ємами потенційних втрат від ризиків є досить складно, у сучасному світі використовуються програмні підходи до аналізу та проектування залежностей і ризиків, але саме присутність фахівця надає програмним підходам одну з основних вимог системного аналізу – наявності експертного оцінювання. Саме системний підхід дає можливість визначати міру і залежність операційних ризиків від інших видів, тим самим підвищуючи точність оцінювання.

## 2. МЕТОДИ ОЦІНЮВАННЯ ОПЕРАЦІЙНИХ РИЗИКІВ

Одним із найважливіших аспектів прогнозування ризиків є боротьба із невизначеністю. Невизначеності – це фактори негативного впливу, які впливають на всю послідовність обробки даних і призводять до зниження якості проміжних і кінцевих результатів. Усі типи математичного моделювання, заснованого на статистичних даних, як правило, повинні справлятися з різними видами невизначеностей, пов'язаних з даними, фактичною структурою досліджуваних процесів та їх моделями, невизначеністю оцінок параметрів і невизначеністю, що стосується якості прогнозування за моделями.

Математичні методи та прийоми, які використовуються для розробки та впровадження СППР, можна розділити на дві групи: а) – методи загального призначення, які забезпечують реалізацію функцій системи; б) – методи та прийоми спеціального призначення, які необхідні для розв'язання конкретних завдань щодо попередньої та базової обробки даних, оцінки моделей, генерації альтернатив, вибору найкращої альтернативи для практичної реалізації та прогнозування наслідків реалізації.

У багатьох випадках моделювання на основі статистичних даних досліднику доводиться мати справу з такими основними видами невизначеностей: структурною, статистичною та параметричною. Структурні невизначеності виникають у випадках, коли структура досліджуваного процесу і, відповідно, його модель невідомі або визначені недостатньо чітко, або відомі частково. Наприклад, коли застосовується функціональний підхід до побудови моделі, зазвичай ми не знаємо структури об'єкта/процесу, і вона оцінюється за допомогою відповідних методів оцінювання структури моделі, таких як кореляційний аналіз, оцінка взаємних ймовірностей, оцінка лагів, тестування на нелінійність і нестационарність та інших спеціальних ефектів, визначення типу зовнішніх збурень тощо. Послідовність дій, необхідних для ідентифікації, обробки та врахування невизначеностей, подана на рис. 1. Ця послідовність обчислювальних процедур відрізняється від багатьох відомих схем обробки даних явним застосуванням методів ідентифікації невизначеності та процедур зменшення невизначеності. Зазвичай усі завдання, згадані на малюнку, успішно вирішуються за допомогою належним чином розробленої та реалізованої СППР, що має спеціальні функції для обробки згаданих невизначеностей.

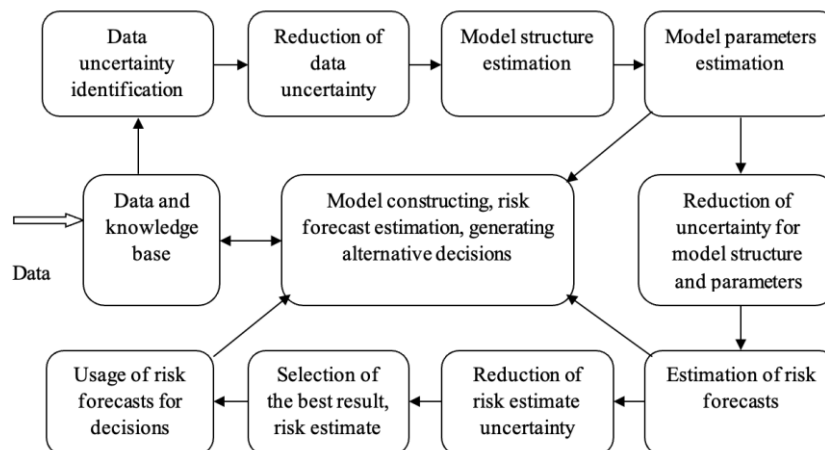


Рисунок 1. Послідовність дій, необхідних для ідентифікації невизначеностей, обробки та їх врахування

Побудова байєсівської мережі є ітераційним процесом, а також компромісом між дуже детальною та насиченою моделлю для отримання «точних» результатів з одного боку, та

вартістю та складністю отримання детальних ймовірностей і підтримки мережі після цього, з іншого боку. Галан Кабальєро (2005) надає просту блок-схему (рис. 2) для реалізації БМ.

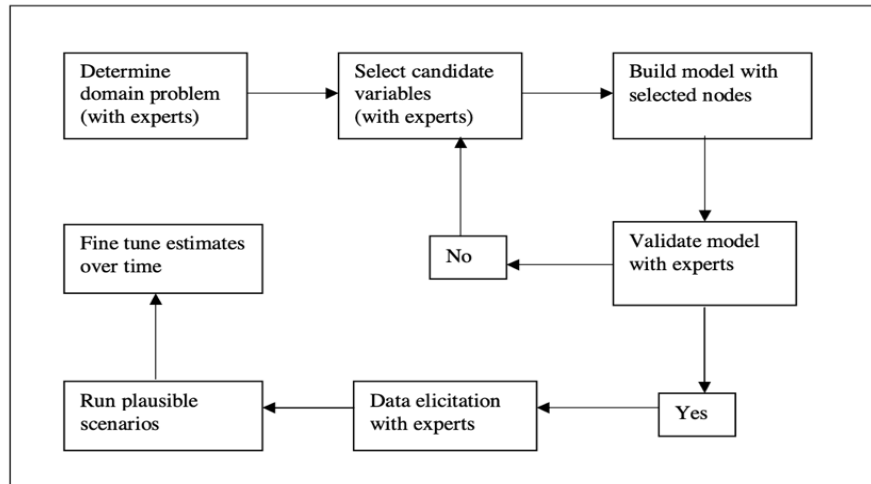


Рисунок 2. Блок-схема баєсівської мережі

Процес прийняття рішень включає досить складні процедури, які можуть бути частково або повністю ітераційними, тобто виконуватися неодноразово, коли знайдена альтернатива не є задовільною для особи, яка приймає рішення (ОПР). СППР повертається автоматично (або за ініціативою ОПР) до попередніх етапів для обробки наявних даних і аналізу знань. Одним із фінансових ризиків, які аналізує ОПР, є кредитний ризик. Для аналізу якості моделі класифікації кредитних позичальників використовуються наступні критерії якості: загальна точність, помилки I та II типу, ROC-крива та індекс Джині [2]. Загальна точність обчислюється так:

$$CA = \frac{\text{Correct Forecast}}{N},$$

де  $CA = \text{Correct Forecast}$  кількість правильно класифікованих випадків;  $N$  це кількість розглянутих справ (клієнтів). Цей критерій певною мірою є суб'єктивним, оскільки залежить від ряду стандартних значень, а також від порогового значення відсікання. Крива ROC (операційна характеристика приймача) показує співвідношення між кількістю правильно класифікованих позитивних випадків (позитивних) і кількістю неправильно класифікованих негативних випадків (негативних). Отже, провівши експериментальні обчислення у спроектованій СППР, можна не лише визначити розмір можливих втрат, а й зробити висновок про те, який з методів обчислення є найкращим для використання на практиці.

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Головною метою БМ, розробленої в цій роботі, є допомога в управлінні операційним ризиком, а також обчислення економічного капіталу, але виключно для цілей внутрішнього управління. Байєсівські мережі можна кількісно визначити за допомогою суб'єктивних чи об'єктивних даних або обох. Зазвичай використовуються об'єктивні дані, якщо вони доступні та можуть бути легко закодовані у мережі. У нашому домені моделювання справді є деякі об'єктивні дані. Проте мережа була розроблена на основі суб'єктивних даних (думка експертів), а наявні об'єктивні дані (історичні дані про втрати) використовувалися лише як засіб «перевірки» розробленої моделі. Статистичний аналіз роботи виконано за допомогою пакета статистичних програм «R».



Для дослідження було обрано опубліковану банківську програму передової моделі операційних ризиків, що діє як свого роду перевірка модулів впроваджувальної структури, але на відміну від впроваджувальної структури, вона не має методу дискримінаційного інтервалу для прийняття рішень, а також не говорить про те, як знайти і контролювати правильні дані.

Коли до набору даних застосовували підхід постійної смуги, проблема порожнього сусіда була дуже помітною для всіх окремих валют, за винятком турецьких лір (TKL) і бразильського реала (BRL). Виявлено, що використання окремих смуг пропускання для кожної точки підгонки наших даних є дорогим з точки зору обчислень (наприклад, валюта євро має понад 17 000 спостережень). Використання локально адаптивних методів підгонки разом із методом ширини смуги найближчого сусіда [3] дало найбільш задовільні результати для набору даних і тому використовувалося для всіх валют, крім TKL і BRL. Для цих двох використовувався підхід із постійною пропускнуою здатністю з додатковою технікою відсікання негативних значень оцінки та зміни масштабу щільності для забезпечення позитивної підтримки оцінок. На рисунку 3 показано деякі приклади оцінок щільності ядра.

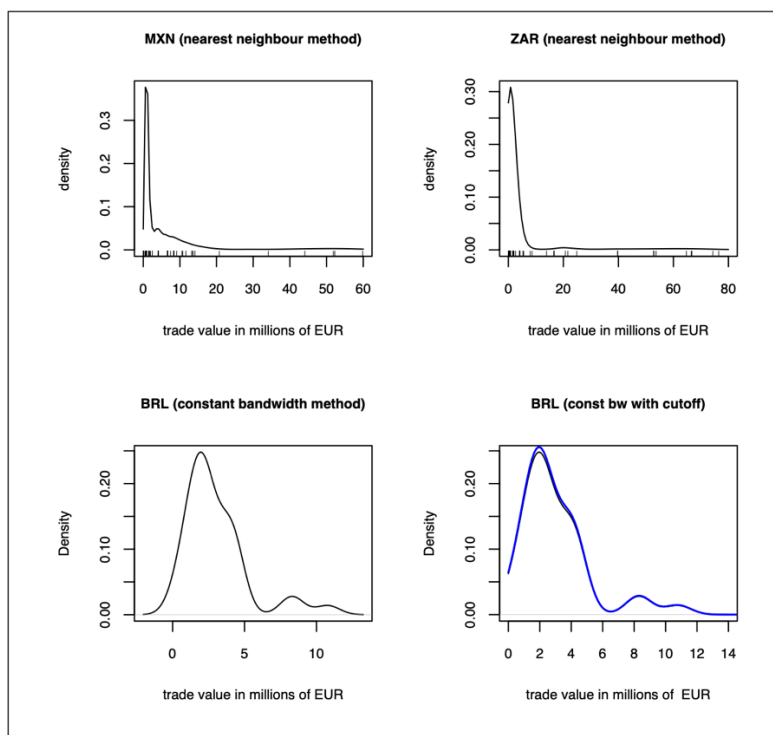


Рисунок 3. Оцінка щільності ядра

Випробування на відповідність проводилися наступним чином: для різноманітних параметрів згладжування оцінювалася щільність. Потім було проведено графічне порівняння між емпіричною функцією розподілу та інтегрованими оцінками щільності. Акцент порівняння робиться на хвостах, які становлять великий інтерес для операційного ризику. Квантиль-квантиль (qqplots) залишків, отриманих від відповідності, також досліджували, щоб оцінити ступінь відсутності відповідності.

Деякі приклади тестів на відповідність показані на рисунку 4.

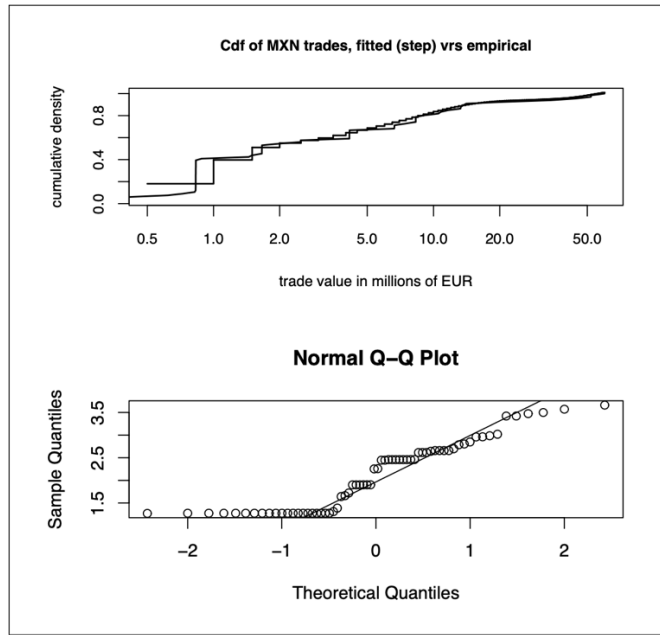


Рисунок 4. Оцінка відповідності

Розроблені байесівські мережі дають два набори результатів (від кінцевих вузлів): 1) частота відмов і 2) серйозність втрат. Ці два розподіли складаються за допомогою моделювання Монте-Карло, щоб отримати розподіл потенційних збитків, який показує потенційні збитки, які можуть виникнути в процесі розрахунку з їхньою ймовірністю (див. рис. 5).

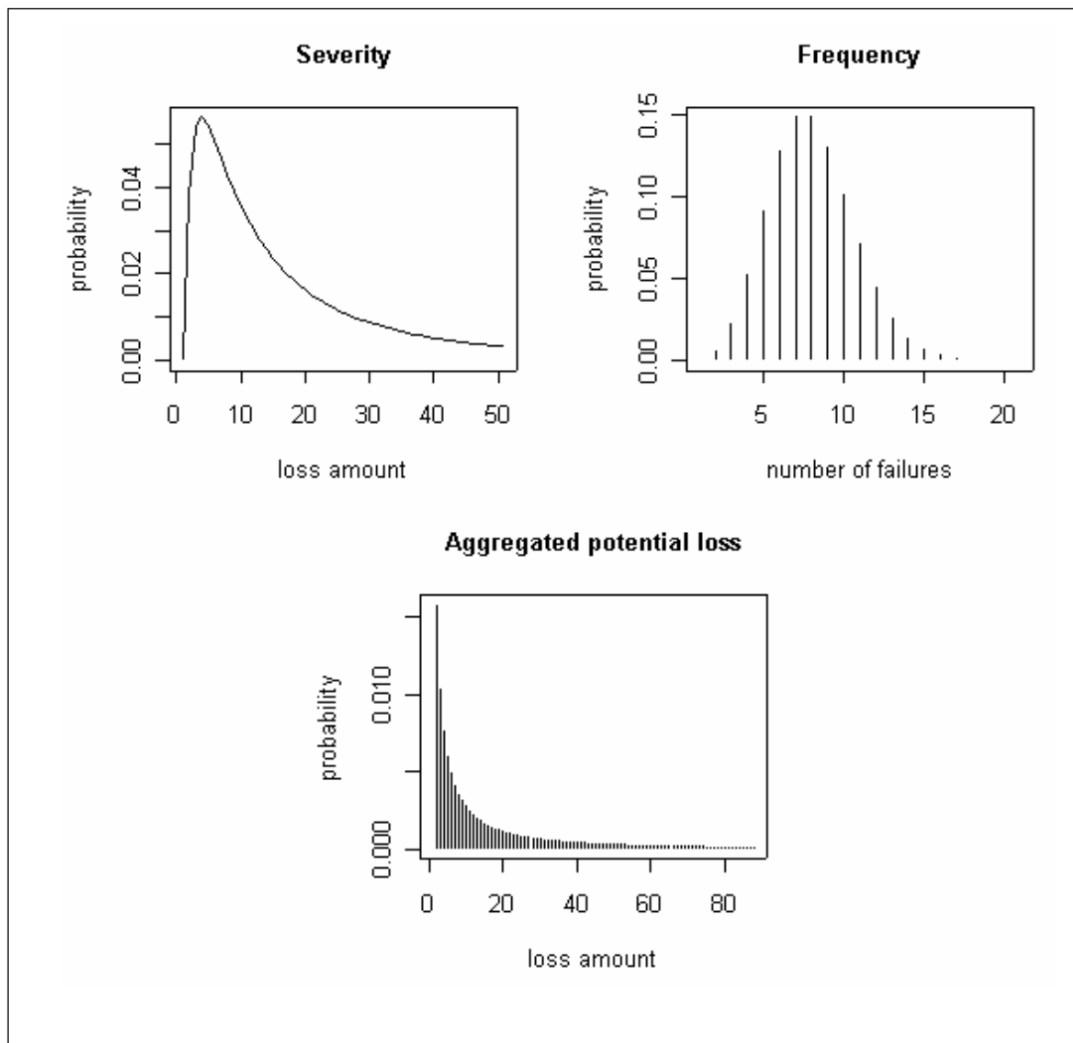


Рисунок 5. Агрегування розподілів частоти та тяжкості

Розподіл ймовірностей для термінального вузла (гіпотетичного вузла) показує, що протягом одного тижня ймовірність відсутності торгової невдачі становить приблизно 11%, однієї невдачі – приблизно 23%, двох невдач – приблизно 24% і три відмови - близько 18%. Пізніше ці результати порівнюються з історичними даними.

На рис. 6 показано мережу серйозності з деякими апостеріорними ймовірностями. Розподіл вузла інтересу, серйозність збитку вказує на те, що найімовірніша сума збитку враховуючи невдачу розрахунку за тиждень становить менше 1000 євро. Відповідна ймовірність появи становить 90%. Інші особливості включають збитки від 5 000 євро до 10 000 євро, від 100 000 євро до 250 000 євро та від 750 000 євро до 1 мільйона євро з імовірністю 1,46%, 0,129% і 0,005% відповідно.

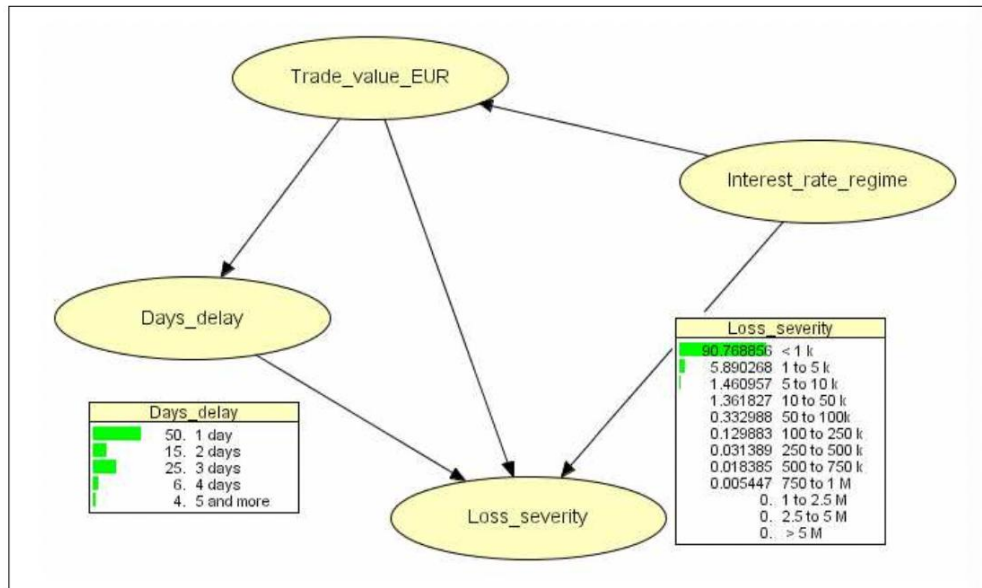


Рисунок 6. Мережа серйозності, яка показує деякі апостеріорні ймовірності

Короткий виклад результатів, отриманих від агрегування за допомогою моделювання Монте-Карло, наведено в таблиці 1. Гістограма потенційних втрат показана на рис. 7. Результати показують, що середня втрата, яка може статися протягом торгового тижня, становить приблизно 11 290 євро з відхиленням у 136 євро. Максимальний потенційний збиток, який можна зазнати, становить 3 142 000 євро. Розподіл потенційних втрат, як і розподіл тяжкості втрат, також має жирний хвіст.

Таблиця 1. Зведена статистика потенційних втрат

	Min	1st quartal	Median	Mean	3rd quartal	Max	Standard deviation
Model results	0.050	760	1,400	11,300	3,000	3,142,000	130,438

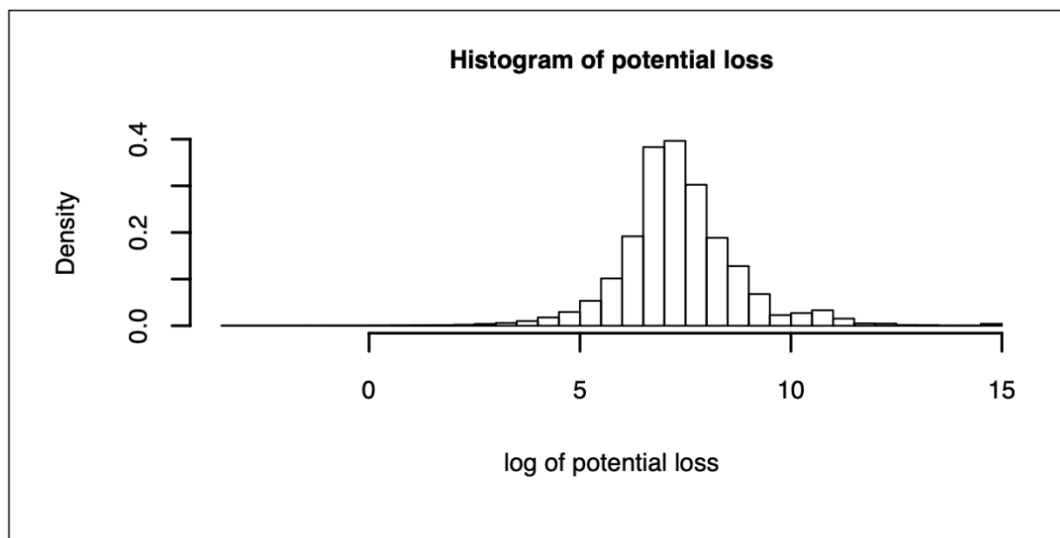


Рисунок 7. Гістограма потенційних втрат

Європейські міжбанківські рекомендації щодо компенсації передбачають, що збитки в розрахунках, менші або рівні 100 євро, не пред'являються. Це дає точку скорочення 100 євро.

Таким чином, набір даних може бути скорочений на 100, 500 або 1000 євро. Для спрощення порівняння результати моделі також скорочуються на 100, 500 і 1000 євро для даних про серйозність збитків перед їх порівнянням з історичними даними про збитки.

Як показано на малюнку 8, ймовірність відчуту одну або нульову кількість невдач протягом торгового тижня є вищою для історичних даних (приблизно 30 і 40 відсотків відповідно), ніж для результатів моделі (приблизно 10% і 25% відповідно). З іншого боку, шанси зазнати двох або більше збоїв вищі для результатів моделі, ніж для історичних даних. Крім того, у той час як імовірна кількість помилок для історичних даних закінчується п'ятьма, для результатів моделі закінчується сімнадцятьма, але з невеликою ймовірністю появи (0,028%).

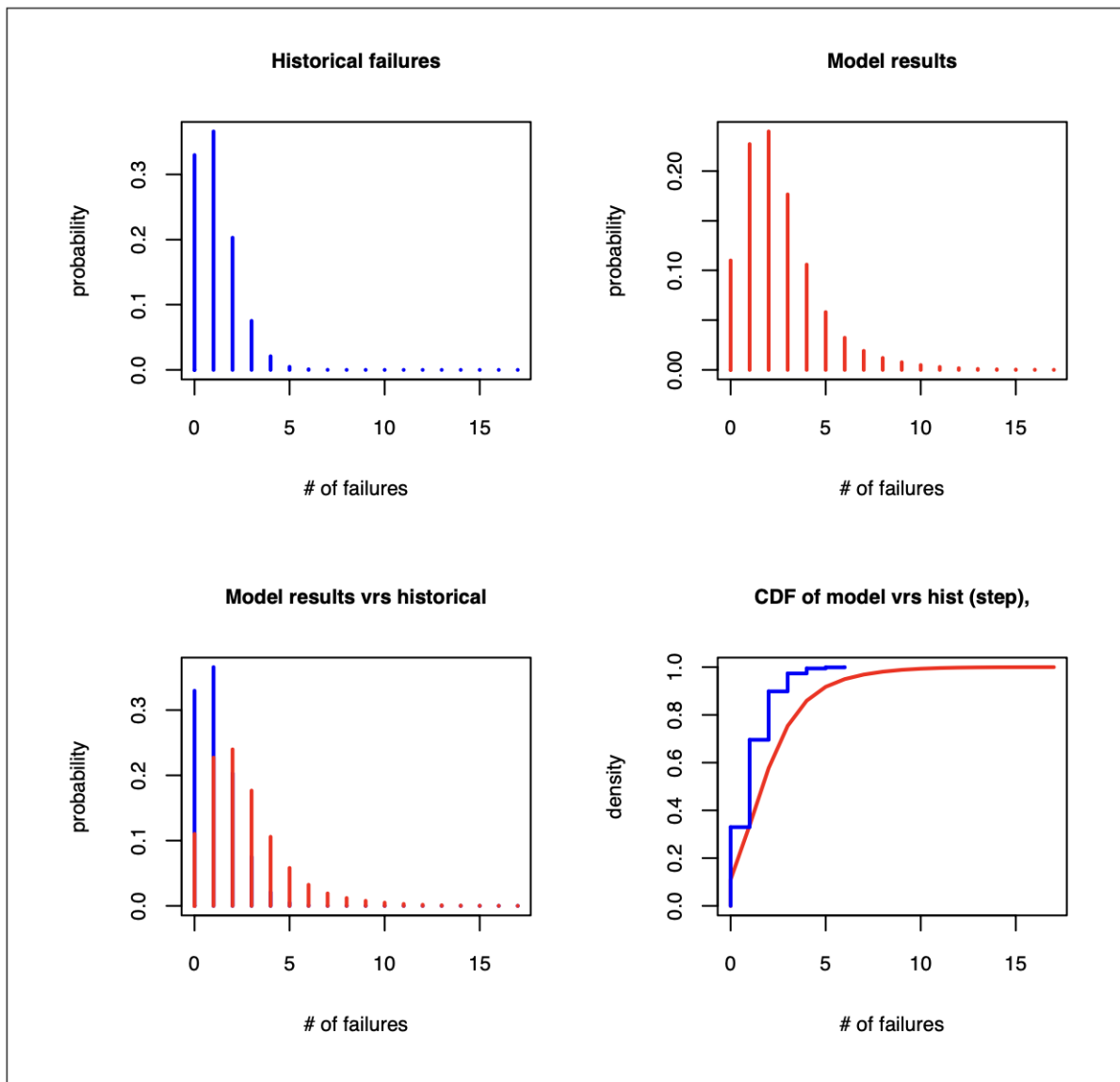


Рисунок 8. Порівняння частоти відмов

## 4. ВИСНОВКИ

На сьогоднішній день поточний акцент в управлінні операційним ризиком, як очікується, переміститься з макрорівня (верхнього) на мікрорівень (бізнес-підрозділ), де фактично відчуються втрати від операційного ризику. Це очікуване зміщення акцентів вимагатиме моделей, які можна використовувати для управління деталями бажаного процесу. З існуючих моделей управління операційним ризиком байєсівські мережі виділяються як модель вибору.

Корисність байєсівських мереж для управління операційним ризиком на рівні бізнес-одиниці було продемонстровано на прикладі реального бізнес-світу. Завдяки застосуванню BN до міжбанківського процесу розрахунків за іноземною валютою в цьому випадку дослідження вдалося змоделювати причинно-наслідкові зв'язки між атрибутами операційного ризику, такими як причинні фактори, ключові показники ризику, частота невдач розрахунків і втрати дана невдача розрахунку. Розроблена байєсівська мережа показала кілька рівнів залежності між факторами ризику. Це дозволило використати модель для детального аналізу ризику та причинних факторів у процесі врегулювання.

Мета цього дослідження полягала в тому, щоб обґрунтувати це твердження. Існуючу практику управління операційним ризиком за допомогою KRI, факторів ризику та підходів до контролю можна розширити шляхом об'єднання основних елементів цих підходів в одну структуру мережі Байєса. Така структура байєсівської мережі має дві переваги: 1) здатність моделювати причинно-наслідкові зв'язки між факторами ризику, KRI та бажаним атрибутом операційного ризику та 2) здатність поєднувати та використовувати переваги основних елементів передовий підхід до вимірювання (дані про внутрішні та зовнішні втрати, аналіз сценаріїв або експертне судження, бізнес-середовище та середовище контролю ризиків).

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кузнецова Н.В., Бідюк П.І. Теорія і практика аналізу фінансових ризиків: системний підхід, монографія. Київ 2020, 400 с.
2. С. Олександр. Операційний ризик, аналіз регулювання та управління. Pearson Education, Лондон, 2013.
3. Дж.Х. Буз і М. Шоу. Отримання знань для систем, заснованих на знаннях. На одинадцятій Міжнародній конференції зі штучного інтелекту, Детройт, Мічіган, 2019 р.
4. Г.Ф. Купер. Обчислювальна складність імовірного висновку з використанням байєсівських мереж переконань. Штучний інтелект, 42:393–405, 2010.
5. У. К'ерульф. Зменшення обчислювальної складності в байєсівських мережах шляхом усунення слабких залежностей. У матеріалах десятої конференції з невизначеності в штучному інтелекті, 1994 р.
6. С. Ліхтенштейн, П. Словік, Б. Фішхофф, М. Лейман і Б. Комбс. Оцініть частоту летальних подій. Експериментальна психологія: навчання та пам'ять людини, 4(6):551–578, 2018.
7. Дж. Перл і С. Рассел. Байєсівські мережі. У М. Arbib, редактор, Handbook of Brian Theory and Neural Networks. MIT Press, 2013.

# МОДЕЛЮВАННЯ НАЙКРАЩОЇ ЦІНИ МОБІЛЬНОГО ДОДАТКУ ЗА ПОВЕДІНКОЮ КОРИСТУВАЧА

Семиволос О.Р.<sup>1</sup>, Каніовська І.Ю.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

<sup>1</sup>helen.7y@gmail.com

**Існує багато способів використовувати методи машинного навчання у сфері ІТ продуктів, включаючи мобільні застосунки. Персоналізація ціни підписок для користувачів є швидкозростаючою нішею задач машинного навчання. Це дає змогу максимально швидко побачити результат роботи мережі й отримати додатковий прибуток. Побудована система дає можливість, аналізуючи дії користувача, запропонувати йому кращу пропозицію. Метою є аналіз алгоритмів машинного навчання для побудови такої моделі. Результатом дослідження є модель, що підбирає для кожного користувача підписку, яка буде кращою для нього та для прибутку компанії.**

**Ключові слова:** машинне навчання, нейронні мережі, персоналізація.

## 1. ВСТУП

Нейронні мережі є невід'ємною частиною сектору інформаційних технологій. Вони імітують поведінку людського мозку, що в поєднанні з великими машинними потужностями, дає їм змогу давати результати краще та швидше, ніж людина. Тому нейронні мережі використовують для автоматизації багатьох сфер нашого життя [2].

Персоналізація мобільних застосунків – це досить активна тема, яка набирає популярності у можливих напрямках розвитку застосунку. Завжди краще, якщо для кожної людини обирають більш персоналізований контент або пропозицію, аніж якусь загальну для всіх. Персоналізація цінової пропозиції є найбільш вигідним напрямком, тому що дає найбільший прибуток компаніям [3]. Складність роботи полягає в тому, що протестувати виконану модель можна тільки на реальних користувачах за допомогою тестів, що при поганій моделі може призвести до ще більших витрат можливого прибутку.

## 2. ОСНОВНИЙ ЗМІСТ

Одним з етапів роботи у побудові системи машинного навчання є аналіз початкових даних. Аналіз показав, що більшість користувачів беруть підписку на етапі реєстрації, коли вони ще не бачили та не користувалися основними функціями додатку. Це можна аргументувати тим, що людина, побачивши рекламу та опис, сформувала очікування та вже на початку бере безкоштовну пробну версію. Складність полягає в тому, що нам мало що відомо про вподобання користувача, а також, яку саме рекламу він побачив. Тому було вирішено змінити початкові дії та запитати у користувача інформацію про його бажання.

Складені запитання дають необхідну інформацію для нейронної мережі та водночас не займають багато часу на їх проходження. За своєю суттю вони характеризують те, наскільки гарно користувач розбирається у тематиці застосунку та наскільки є актуальним додаток для

нього. Важливо не погіршити початкові метрики проходження етапу реєстрації у застосунку, а саме, щоб питання не були відлякуючими та їх кількість не була занадто великою. В процесі роботи ці питання були протестовані на реальних користувачах, а також зібрані дані по їх відповідях.

Проаналізовані відповіді були додані до загального списку параметрів, що відомі про клієнта на самому початку користування застосунком до моменту, як він побачить підписку. Було протестовано різні види підписок для того, щоб зібрати дані для моделі.

Наступний етап полягав у побудові моделей з використанням машинного навчання. Для вирішення задачі були обрані: лінійна регресія, випадковий ліс, дерево рішень, а також нейронна мережа.

При побудові моделей враховувався факт різних цін на кожну підписку. Найбільш ймовірно, що користувач візьме найдешевшу підписку, але це не буде вигідно розробникам. Тому кожній з підписок присвоювалось своє значення прибутковості, що дало можливість побудувати модель, яка вирішує питання збільшення прибутковості компанії [6].

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Труднощі полягали в тому, що протестувати модель можна тільки реалізувавши її роботу в застосунку. Адже неможливо знати фактичні ймовірності того, що користувач візьме ту чи іншу підписку.

Одним з показників моделі є час на видачу відповіді. Вона повинна дати свій результат у короткій проміжок часу, адже, якщо підписка на екрані не встигне відобразитися, то користувач закrije екран та не візьме підписку. Тому це обмеження також було враховано при побудові кожної з моделей.

Результати роботи кожної з моделей наведені у таблиці нижче (Табл. 1).

Таблиця 1. Значення покращення метрики прибутковості

	Logistic Regression	Random Forest	Decision Tree	Neural Network
Improvement	+3.7%	+7.4%	+4.9%	+7.0%

Дерево рішень дало досить непогані результати. Але значення, що дали моделі, побудовані за допомогою нейронних мереж та випадкового лісу значно краще за інші. Між ними є невелика різниця в покращеннях метрик. По часу виконання роботи нейронна мережа давала свою відповідь значно повільніше за інші моделі.

Саме тому найкращою моделлю можемо вважати випадковий ліс. Модель, що побудована цим методом, дала найбільше покращення та дає відповідь значно швидше за інші моделі.

### 4. ВИСНОВКИ

На сьогоднішній час нейронні мережі використовують для різноманітних задач, в тому числі й тих, що допомагають компаніям отримувати гарні прибутки. Одним із напрямків використання методів машинного навчання у мобільних застосунках – це персоналізація цін для кожного з користувачів. Це є найбільш ефективний підхід до збільшення прибутковості компанії, бо завдяки ньому можна швидко побачити результат, що збільшує прибутковість.

У ході дослідження було реалізовано таку систему, яка дозволяє виконувати оцінку ймовірностей того, що користувач візьме кожну з трьох видів підписки та вибере з них ту,



показ якої призведе до збільшення метрики повернення користувача у застосунок. В ході роботи було побудовано декілька моделей за допомогою лінійної регресії, дерева рішень, випадкового лісу та нейронних мереж. Найкращі результати показала модель, що була побудована за допомогою одного з методів машинного навчання – випадкового лісу.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кузнецова Н.В., Бідюк П.І. Теорія і практика аналізу фінансових ризиків: системний підхід, монографія. Київ 2020, 400 с.
2. How Target figured out a teen girl was pregnant before her father did <https://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did/?sh=1b4e41886668>
3. Mobile App Personalization: The future of User Experience [Електронний ресурс]. – Режим доступу : <https://www.storyly.io/post/mobile-app-personalization-the-future-of-user-experience>
4. Ali, J., Rehanullah Khan, Ahmad, N. and Imran Maqsood (2012). Random Forests and Decision Trees. ResearchGate. [Електронний ресурс]. – Режим доступу : [https://www.researchgate.net/publication/259235118\\_Random\\_Forests\\_and\\_Decision\\_Trees](https://www.researchgate.net/publication/259235118_Random_Forests_and_Decision_Trees)
5. Altman, N. and Krzywinski, M. (2015). Simple linear regression. Nature Methods, 2(11), 999–1000 ст. [Електронний ресурс]. – Режим доступу : <https://www.nature.com/articles/nmeth.3627>
6. Berrar, D. (2018). Cross-Validation. ResearchGate. [Електронний ресурс]. – Режим доступу : [https://www.researchgate.net/publication/324701535\\_Cross-Validation](https://www.researchgate.net/publication/324701535_Cross-Validation)

# ГЕНЕРАЦІЯ ЗОБРАЖЕНЬ ЕМОЦІЙ ЛЮДИНИ З ВИКОРИСТАННЯМ ГЕНЕРАТИВНИХ ЗМАГАЛЬНИХ МЕРЕЖ

Столяр М.В.<sup>1</sup>, Каніовська І.Ю.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна  
<sup>1</sup>misha.stolyar99@gmail.com, <sup>2</sup>ikaniovska@gmail.com

**Генерація зображень людей - актуальна проблема сучасності. Метою даної роботи є розробка моделі синтезу емоції людини на зображенні. До головних переваг моделі над вже існуючими можна віднести: гнучкість, швидкість роботи та розмір зайнятого простору на диску. Під гнучкістю розуміється, що для якісної заміни емоції у людини, яка не була в тренувальній вибірці, перенавчати модель не має потреби. Таким чином, підхід можна використовувати у промисловості.**

**Ключові слова: генерація зображень, VAE, GAN**

## 1. ВСТУП

Станом на сьогоднішній день генерація зображень людини, тобто створення DeepFake контенту, є актуальною проблемою. Існує багато платформ, таких як Kaggle [1], що організовують міжнародні змагання у даних напрямках. Необхідно відмітити, що розробляються, як методи синтезу штучного контенту, так і методи його розпізнавання.

Розглядається задача генерації деякої емоції людини на зображенні, конкретно, задача заміни емоції людини на вхідному зображенні на одну з: радість, страх, гнів, сум, здивування, розчарування та нейтральне обличчя. Головними обмеженнями є:

- Модель має видавати якісний контент. Синтезоване зображення має бути максимально правдоподібним;
- Швидкість роботи та вага моделі. Необхідно, щоб модель була швидкою (до 10 секунд робота моделі) та була малого розміру (до 100 Mb) - у такому разі її можна буде використовувати на мобільних пристроях.

## 2. ОПИС МОДЕЛІ

У подальшому називаємо модель EmotionSwap.

EmotionSwap базується на одному з підходів переносу руху з одного фрейма на інший - First Order Model [2], з використанням умовної генерації - AdaIN [3]. Отже, процедура роботи моделі наступна:

1. Виділяємо ключові точки на обличчі людини (ніс, очі, губи...) та знаходимо де саме мають бути ці точки для отримання бажаної емоції;
2. Відновлюємо функцію переходу між бажаним розташуванням точок та поточним;
3. Перебудовуємо зображення;
4. Відновлюємо зображення.

Детекція ключових точок та відновлення зображення реалізовані за допомогою глибокого навчання, пошук функції переходу між бажаним розташуванням точок та поточним відбувається на основі афінних перетворень. На рис. 1 зображено архітектуру EmotionSwap.

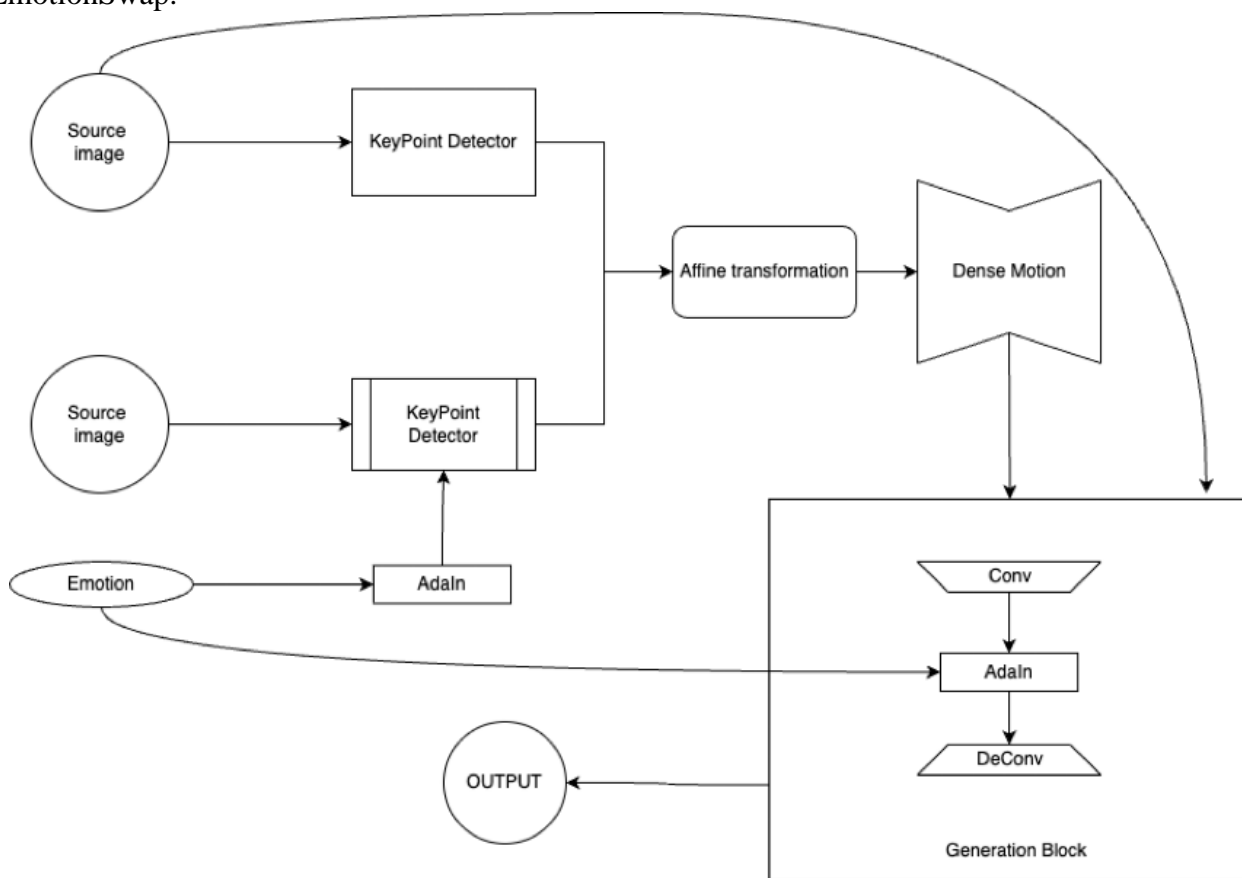


Рисунок 1. Архітектура EmotionSwap

### 3. ТРЕНУВАННЯ МОДЕЛІ ТА РЕЗУЛЬТАТИ ГЕНЕРАЦІЇ

Під час навчання моделі, було використано дві функції витрат, а саме: MSE - для витрат пошуку необхідних координат точок, perception loss [4] - у якості витрат при реконструкції. Було проведено два експерименти:

- з однаковими вагами функцій витрат. Графік зображено на рис. 2.
- Функції витрат при реконструкції мали в 10 разів більший вплив (вагу) на цільові витрати. Графік зображено на рис. 3.

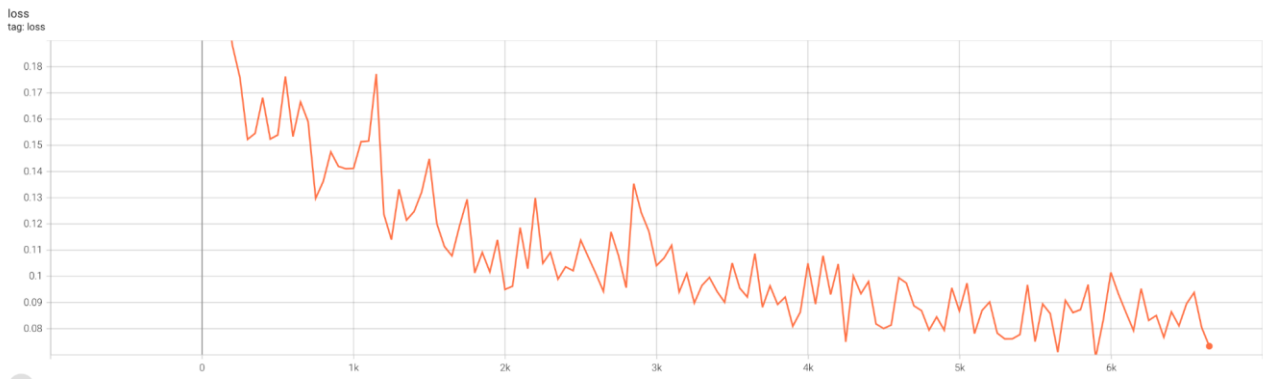


Рисунок 2. Функція витрат першого експерименту

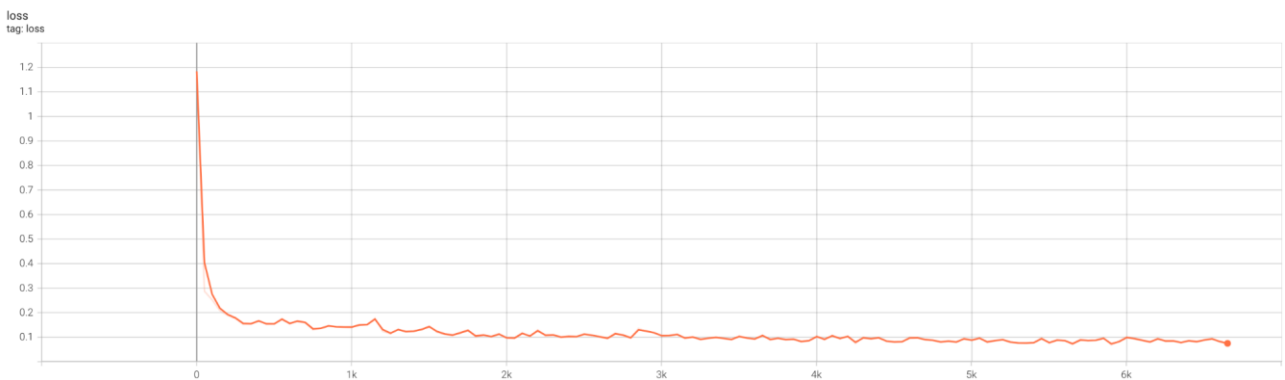


Рисунок 3. Функція витрат другого експерименту

Результати генерації нової емоції зображено на рис. 4 та рис. 5 відповідно для першого типу витрат та для другого.



Рисунок 4. Результат генерації першого експерименту



Рисунок 5. Результат генерації другого експерименту

Дані для навчання було згенеровано за допомогою моделі Conditional StyleGan [5].

#### 4. ВИСНОВКИ

Обидва експерименти показали приблизно однакові результати генерації. На жаль, на цей час не створена якісна метрика для їх порівняння (особливо для синтезованої тренувальної вибірки). Важливим показником результатів тренування є те, що в обох експериментах функція витрат збігалася до нуля, проте в другому експерименті це відбулось значно швидше, що видно на рис. 2 та рис. 3. Отже, отримано модель, що дозволяє швидше, без втрати якості, змінювати емоції людини на зображенні.

У подальших дослідженнях можна додати інші типи функцій витрат, наприклад, класифікувати чи дійсно на виході моделі було отримано бажану емоцію (Cross-Entropy Loss).

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Платформа Kaggle [Електронний ресурс] - <https://www.kaggle.com/>
2. Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. In ICLR, 2017.
3. Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. In NIPS, 2015.
4. Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In ECCV, 2016.
5. Hao Tang, Dan Xu, Wei Wang, Yan Yan, and Nicu Sebe. Dual generator generative adversarial networks for multi-domain image-to-image translation. In ACCV, 2018.

# **СИСТЕМНИЙ ПІДХІД ДО ПРОЦЕСУ КЕРУВАННЯ ДОСТУПАМИ В ОБ'ЄКТАХ КРИТИЧНОЇ ІНФРАСТРУКТУРИ В УМОВАХ ВОЄННОГО СТАНУ**

Тимошук О.Л.<sup>1</sup>, Пастушок О.О.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>o.tymoshchuk@kpi.ua [0000-0003-1863-3095], <sup>2</sup>olenapastusok@gmail.com

**Процес керування доступами є основним елементом захисту інформаційного простору об'єктів критичної інфраструктури від кібератак зі сторони хакерів та хактивістів, тому для ефективного управління даним процесом доцільно використовувати системний підхід, одним із результатів застосування якого може стати створення надійної системи безпечного доступу. Метою роботи є розробка системного підходу до процесу керування доступами в умовах воєнного стану. Результатом дослідження є розроблена методика, яка забезпечує конфіденційність, цілісність та доступність інформації.**

**Ключові слова:** кібератаки, системний підхід, енергетичний сектор, ІТ стратегія, керування доступами.

## **1. ВСТУП**

З кожним роком об'єкти критичної інфраструктури потерпають все більшої і більшої кількості кібератак, особливо під час воєнного стану. На сьогоднішній день в небезпеці знаходиться не тільки банківський сектор, а енергетичний, від якого залежить стабільність роботи всієї країни. Енергетика України перебуває у стані війни з 2014 року, тому з 24 лютого 2022 року, з повномасштабним вторгненням на територію України, певні рішення вже були відпрацьовані на територіях України, де раніше вели активні бойові дії та тимчасово окуповані території. Водночас, українська енергетика зустрілася з переліком нових, ще більш загрозливих викликів, як ядерний тероризм із захопленням АЕС, численні пошкодження критичної інфраструктури – електричних та газових мереж, критичне зниження попиту на енергетичні продукти у зв'язку з виїздом населення та припиненням бізнесу, ще більш критичне зниження рівня оплат в енергетичній системі, і рішення, незважаючи на бойові дії по всій території країни, продовжуватиме синхронізацію енергетичної системи України з енергосистемою Континентальної Європи, паливну кризу тощо. Окрім, всього вищесказаного енергетичний сектор попав під вплив хактивістів та кіберзлочинців, які намагаються розхитати енергетичну структуру з середини [1].

Для того щоб забезпечити цілісність енергетичного сектору, то потрібно забезпечити єдність не тільки фізичну та роботоздатність ТЕЦ, ГЕС, та АЕС, але і не дати хакерам зруйнувати ядро, скомпрометувати критичну інформацію, модифікувати її, або використати проти нас самих. Забезпечити цілісність, доступність та конфіденційність інформації можна лиш побудувавши надійну структуру доступу до даних.

## **2. МЕТОД ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ДАНИХ**

Контроль доступу є фундаментальним компонентом безпеки даних, який визначає, кому дозволено отримувати доступ і використовувати інформацію та ресурси компанії.

Завдяки автентифікації та авторизації політики контролю доступу гарантують, що користувачі є тими, за кого себе видають, і що вони мають відповідний доступ до даних компанії. Контроль доступу також можна застосувати для обмеження фізичного доступу до кампусів, будівель, кімнат і центрів обробки даних [2].

Контроль доступу ідентифікує користувачів шляхом перевірки різних облікових даних для входу, які можуть включати імена користувачів і паролі, PIN-коди, біометричні скани та маркери безпеки. Багато систем контролю доступу також включають багатофакторну автентифікацію (MFA), метод, який вимагає кількох методів автентифікації для перевірки особи користувача.

Після автентифікації користувача система керування доступом авторизує відповідний рівень доступу та дозволені дії, пов'язані з обліковими даними та IP-адресою цього користувача.

Управління доступом в організації — це перевага того, наскільки добре організація функціонує, коли йдеться про авторизацію активів для користувачів. Проблемний аспект неавторизованого доступу усувається завдяки ефективному управлінню доступом у будь-якій організації. Для регулювання та створення ефективного процесу управління доступами в ІТ системах існує декілька міжнародних стандартів. Найбільш використовуваними міжнародними стандартами є CISCO Benchmarks та ITIL. Але окрім міжнародних стандартів критична інфраструктура також повинна відповідати державним стандартам КСЗІ (Комплексні системи захисту інформації).

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для того, щоб уникнути вразливостей процесу керування доступом, а саме процесу автентифікації та подолати вектори загроз потрібно слідувати наступному підходу:

1. Надати високий пріоритет автентифікації без пароля. Паролі та процеси автентифікації на основі паролів страждають від загроз безпеці, таких як фішинг і злом паролів. І підхід автентифікації без пароля є способом усунення цих загроз.
2. Впровадити федеративний вхід і єдиний вхід (SSO). Система єдиного входу (SSO) дозволяє користувачам входити в кілька програм за допомогою одного набору облікових даних. Використовувати протоколи SAML і OAuth2.
3. Посилена функція автентифікації. Посилення функціональності автентифікації є важливою частиною правильного впровадження потоків автентифікації.
4. Захист від автоматизованих атак. Впровадити обмежувачі швидкості. Впровадження обмежувачів швидкості допомагає зупинити атаки грубої сили. Простий обмежувач швидкості, який зараховує невдалі спроби автентифікації та блокує користувачів на певний період, може значно підвищити безпеку та зупинити атаки грубої сили.
5. Посилити автентифікацію на основі пароля. Хоча загальна порада полягає в тому, щоб надавати перевагу безпарольному рішення, у багатьох випадках це нелегка міграція.
6. Не винаходьте заново методи автентифікації. Легко написати функцію автентифікації, але важче правильно реалізувати. Бібліотеки, сертифіковані стандартними специфікаціями (наприклад, сертифіковані OpenID Connect) і перевірені на безпеку бібліотеки є безпечнішим вибором.

Найкращі рекомендації для захисту процесу автентифікації — це пріоритет автентифікації без пароля та впровадження федеративного входу або SSO із двофакторною автентифікацією. На основі цього було розроблено графічний тестовий приклад аутентифікації в системі використовуючи SSO та двофакторну аутентифікацію:



1. Заходимо на сторінку аутентифікації (Рис. 1) та вводимо SSO ID.

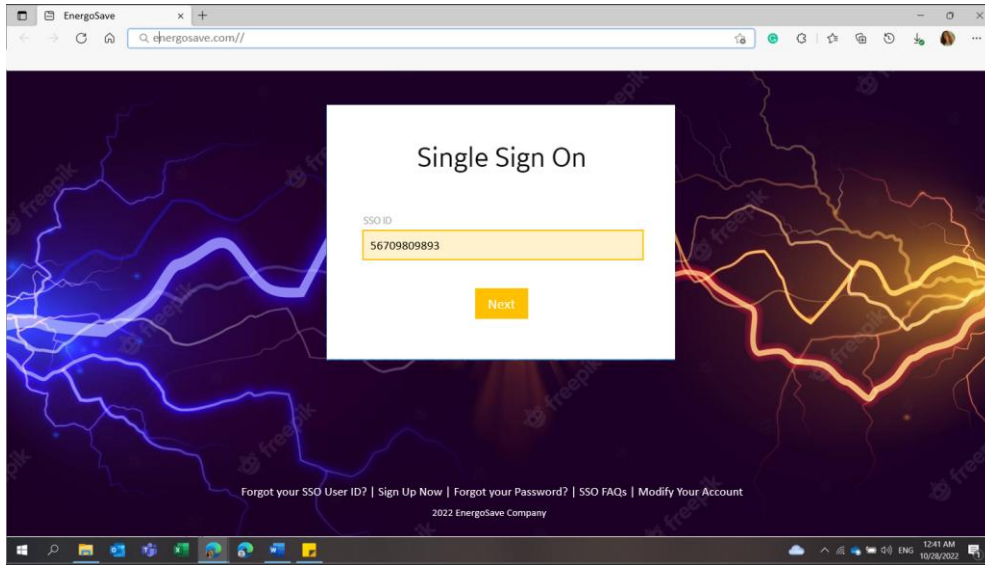


Рисунок 1. Сторінка входу (SSO)

2. Вводимо пароль (Рис. 2).

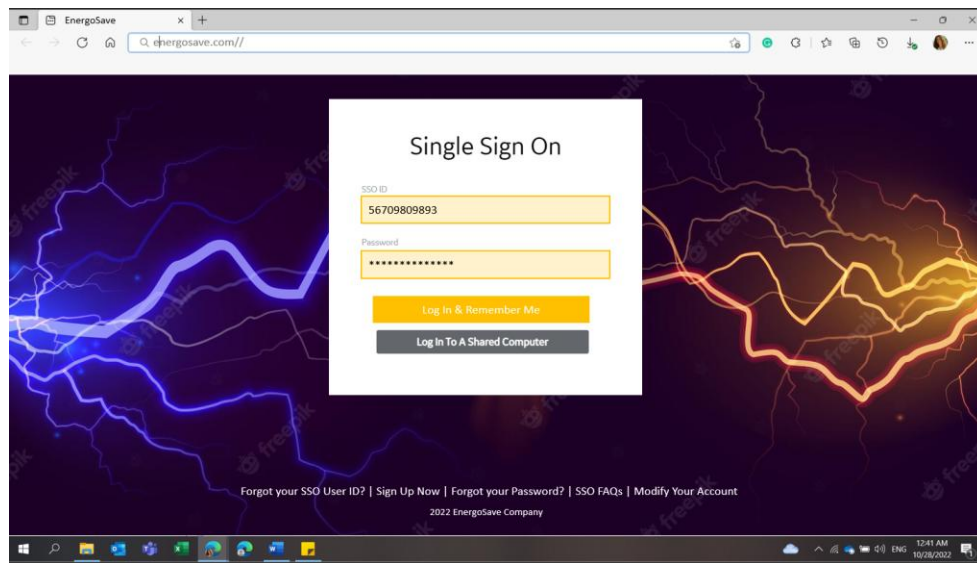


Рисунок 2. Процес аутентифікації

3. Використання двофакторної авторизації (Рис. 3, Рис. 4).

Дані міркування про безпеку необхідно враховувати на ранніх стадіях процесу проектування, щоб реалізувати елементи керування, які запобігають логічним помилкам. Регулярні аудити безпеки також важливі для перевірки на наявність логічних недоліків, і слід ретельно обміркувати вибір бібліотек і залежностей для процесу аутентифікації.



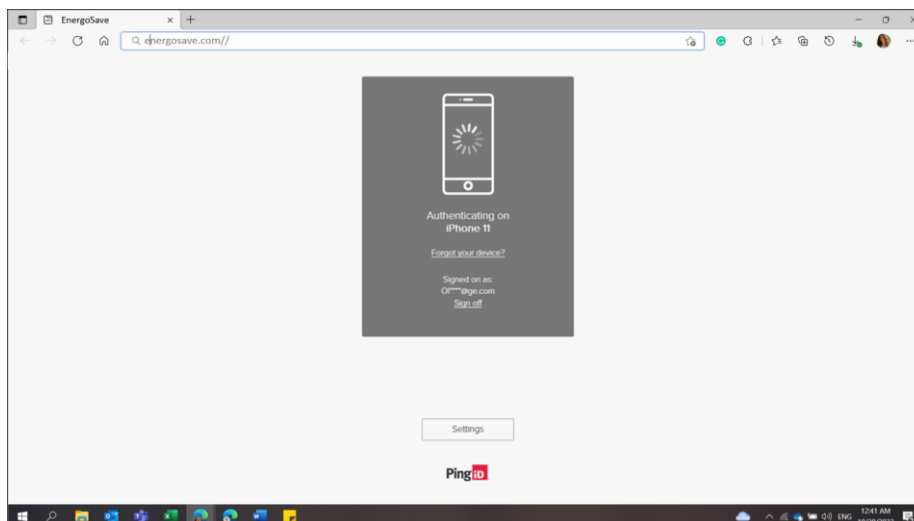


Рисунок 3. Використання двофакторної аутентифікації

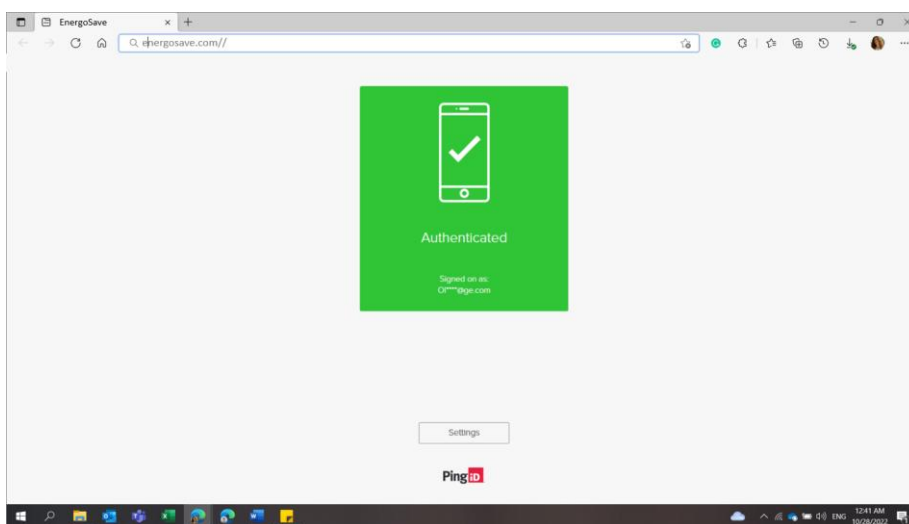


Рисунок 4. Успішний вхід

## 4. ВИСНОВКИ

На сьогоднішній день кібератаки тільки посилюються, тому ефективним методом захиститися від них є посилання функції керування доступом та використовувати системний підхід, який забезпечить цілісність, конфіденційність та доступність інформації.

В результаті розроблені методики та моделі було об'єднано в підхід впровадження технічних рішень інформаційної безпеки на підприємстві, що є однією з основних складових стратегії інформаційної безпеки. Розроблене рішення є унікальне тим, що надає практичні для застосування методики з оцінки актуального стану ІТ та ІБ архітектури та процесів підприємства, демонструє приклади класифікації ризиків, збитків та активів.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Енергетика України під час воєнного стану. Зміни у вимогах. URL: [https://jurliga.ligazakon.net/ru/aktualno/12602\\_energetika-vo-vremya-voyny-v-ukraine-kakie-izmeneniya-v-regulirovanii](https://jurliga.ligazakon.net/ru/aktualno/12602_energetika-vo-vremya-voyny-v-ukraine-kakie-izmeneniya-v-regulirovanii) (дата звернення 15.11.2022).
2. What is access control? URL: <https://www.citrix.com/solutions/secure-access/what-is-access-control.html> (дата звернення 15.11.2022).

# АНАЛІЗ ГЕТЕРОСКЕДАСТИЧНИХ ПРОЦЕСІВ ТА ФІНАНСОВОГО РИЗИКУ

<sup>1</sup>Федоренко А.П., <sup>2</sup>Бідюк П.І.,

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>feanneta@gmail.com

**Пропоноване дослідження спрямоване на моделювання фінансових процесів, що широко використовуються при роботі з фінансовими ринками та прогнозування цін на цьому. Неодноразово підтверджувалося, що фінансовий ринок є одним із найбільших і найдинамічніших явищ на Землі. Як наслідок, для його аналізу потрібні передові методи і інструменти. З цієї причини для навчання обрано різноманітні гетероскедастичні моделі. Такі моделі для оцінювання ризику як VAR (Value-at-Risk) та авторегресійна умовна гетероскедастична модель (ARCH та Generalized ARCH model (GARCH)) для опису динаміки дисперсії. Приклад аналізу реальних даних фондового ринку ілюструє можливість і доцільність застосування моделі гетероскедастичності для прогнозування динаміки дисперсії.**

**Ключові слова: фінансові ризики, системний підхід, VAR, CVAR, аналіз ризиків, ринковий ризик.**

## 1. ВСТУП

Важливе місце в економічній науці та практиці займають категорії ризику. Ризик є елементом здійснення будь-якої фінансової діяльності. Ризики є результатом невизначеності та є випадковими величинами, оскільки вони визначаються наявністю багатьох факторів, вплив яких на результат фінансової діяльності неможливо точно визначити заздалегідь.

Сучасні фінансові процеси характеризуються високою динамікою, нестаціонарністю, нелінійністю, великою і змінною в часі волатильністю, наявністю детермінованих і випадкових компонентів у даних часових рядів [1, 2]. Фінансові процеси перебувають під впливом множини випадкових збурень різної природи (шумові компоненти), які додають значну невизначеність в аналізі даних. Зміни у фінансових процесах характеризуються оцінками волатильності, але не можуть описуватися однорідними моделями за рахунок різких змін у факторах, що впливають на функціонування цих фінансових процесів. Тому для того, щоб охарактеризувати сучасні фінансові процеси на ринках, такі моделі зазвичай використовуються для врахування змін обох фінансових параметрів і самого процесу, порівняно з його попередніми станами, а також як процеси та фактори, що на нього впливають.

У цьому дослідженні розглядаються деякі типи моделей гетероскедастичних процесів для оцінювання волатильності фінансових процесів і можливих збитків. Багато сучасних наукових досліджень присвячено моделюванню та прогнозуванню волатильності. Особливості фінансових процесів потребують подальшого вдосконалення математичних моделей і методів оцінювання їх параметрів. Таким чином, постає проблема отримання якісних прогнозів цього важливого параметра для прийняття статистичного рішення, яке потребує подальших серйозних досліджень.

## 2. ПРОГНОЗУВАННЯ ВОЛАТИЛЬНОСТІ ЧАСОВИХ РЯДІВ

Фінансові установи зазвичай поділяють фінансові ризики на чотири основні групи. Ці групи включають операційний ризик, ризик ліквідності, кредитний ризик і ринковий ризик. Як правило, ці ризики розглядаються окремо під час формування вартості портфеля капіталу. Для визначення рівня можливих втрат та забезпечення інвесторів від потенційних збитків при настанні несприятливих ситуацій на ринку надзвичайно важливим є запровадження ефективного процесу управління ризиками.

Оскільки фінансовий ризик за своєю природою передбачає невизначеність щодо здійснення певного виду фінансової операції у майбутньому, то не завжди вдається отримати прийнятні оцінки [2]. Тому одним із високо актуальніших напрямів розвитку економічної науки сьогодні є пошук ефективних, науково обґрунтованих методів і способів аналізу та оцінювання фінансових ризиків.

Одним із таких підходів є побудова математичних моделей, які описують дискретну динаміку фінансових часових рядів на основі гетероскедастичних процесів. Існують різні підходи до оцінювання фінансового ризику, метою яких є розробка практичної, ефективної та достатньо точної моделі для певних класів ризику з метою складання достатньо точних прогнозів та визначення стратегій мінімізації ризику. До таких додати можна методологію VaR, а також деякі математичні моделі для прогнозування дисперсії втрат, такі як ARCH та GARCH.

Метод Value at Risk (VaR) сьогодні достатньо популярний у використанні. Можна виділити ряд переваг, які зумовлюють таку популярність, а саме: можливість вимірювання ризиків на різних ринках універсальним чином; ризики можуть бути оцінені в межах можливих втрат, пов'язаних з імовірністю їх виникнення; агрегувати ризик окремих позицій в єдине значення для всього портфеля (загальний ризик), беручи до уваги інформацію про кількість позицій і волатильність ринку.

Загалом, VaR відображає мінімальний збиток інвестиції, який буде понесений на певному рівні ймовірності (довірчому інтервалі). Однак, чим менш стабільна інвестиція, тим більша ймовірність того, що VaR не дасть повної картини ризиків, оскільки ця міра байдужа до всього, що виходить за межі її власного порогу. Саме тому для оцінки екстремальних подій поруч із показником VaR також розраховують показник очікуваних втрат Expected Shortfall, найбільш відомий як CVaR або Conditional Value at Risk (Рис.1).

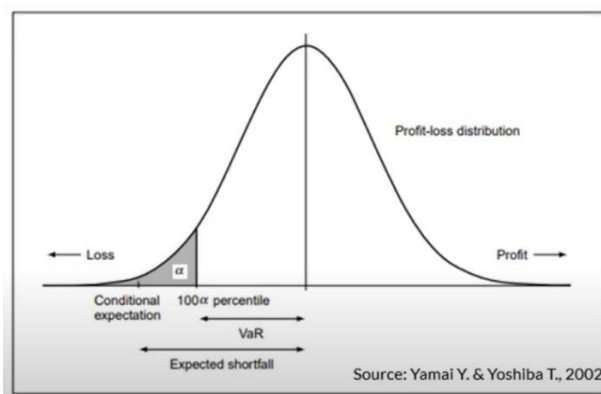


Рисунок 1. Візуалізація VaR та CVaR на графіку розподілу портфелю

До методів оцінювання VaR відносяться: історичний метод, параметричний метод та метод моделювання Монте Карло, який відрізняється від попередніх більшою складністю обчислень, однак показує більш точні результати. Під час обчислення VaR волатильність

доходності інструменту є основною проблемою. Через це дельта-нормальні методи зазвичай включають дані на основі історичної дисперсії вибірки.

Модель ARCH формує волатильність як суму постійної базової волатильності та лінійної функції абсолютного значення кількох останніх змін ціни. Розширенням моделі ARCH є модель волатильності GARCH. Популярність методів ARCH/GARCH, яку ми спостерігали протягом останнього десятиліття, свідчить про те, що ці методи мають ряд недоліків у прогнозуванні волатильності. Існує точка зору, згідно з якою прибутковість не можна розглядати як незалежну, рівномірно розподілену випадкову змінну. Тому з'явилася велика кількість моделей прогнозування, таких як GARCH, FGARCH, GJR-GARCH і т. ін.

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для дослідження сформовано інвестиційний портфель з цін акцій компаній Ford Motor Company (F), Tesla (TESLA) та Intel Corporation (INTC). У якості часового горизонту взято період у 800 днів. Початкова інвестиція була обрана у розмірі 10 000 \$, а в якості періоду для прогнозування обрано один торговельний рік – 252 дні.

На основі цих даних безпосередньо оцінюється VAR. Для історичного моделювання ряд відносних змін сортується та очищається за підмножиною найгірших значень, що виходить за межі прийнятного рівня довіри. Найгірше з решти значень відповідає максимально можливому значенню втрати в межах прийнятного рівня довіри, яким є VAR - розрахований у відносній варіації відповідно до порядку розрахунку. На Рис. 2 показано доходність акцій компанії Ford Motor Company. У таблиці 1 показано похибку прогнозування втрат за рівнем довірчої ймовірності.

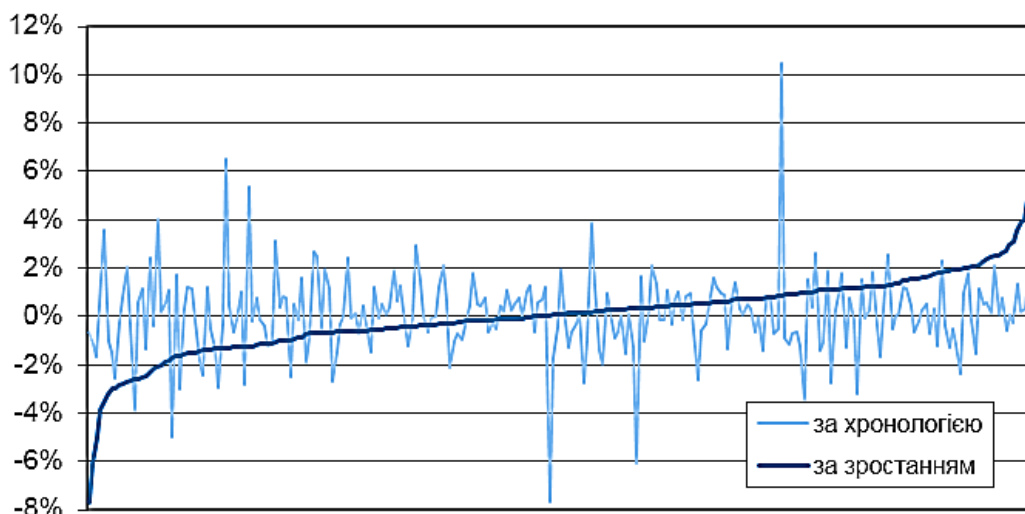


Рисунок 2. Доходність акцій Ford Motor Company

Таблиця 1. Прогноз похибки для втрат, історичний метод (Ford Motor Company)

Довіра (%)	Точки, що вилючаються з розгляду	Втрати: похибка у прогнозі
99	3	6,3%
97	8	3,1%
95	12	2,8%

Для оцінки VAR параметричним методом використано дисперсійно-коваріаційну модель. Робиться припущення щодо розподілу доходів і використовуються історичні доходи портфелю та стандартні відхилення (оцінки) для визначення параметрів моделі. Загальна схема оцінки VAR методом змінної коваріації проводилась таким чином. На першому етапі визначалися параметри нормального розподілу, які найбільше наближені до фактичного розподілу показника, що розглядається, - отриманого вище ряду відносних змін. Далі необхідно було визначити значення оберненого нормального розподілу - для довгої позиції, а отже, негативної зміни, на основі параметрів, отриманих раніше. У таблиці 2 наведено параметри нормального розподілу, а в таблиці 3 похибка прогнозування втрат за рівнем довіри для компанії Ford Motor Company.

Таблиця 2. Нормальний розподіл

	F
Середнє значення	0,12%
Станд. відхилення	1,83%

Таблиця 3. Витрати: похибка у прогнозі, варіаційно-коваріаційна модель (Ford Motor Company)

Довіра (%)	Втрати: похибка у прогнозі
99	4,0%
97	3,2%
95	2,9%

Найскладнішим у реалізації виявився метод Монте-Карло, який хоч і відрізняється високою обчислювальною складністю, проте вважається найбільш точним та гнучким серед усіх методів обчислення VaR. Основна відмінність між VAR методом Монте-Карло та моделюванням історичної оцінки і коваріації дисперсії полягає в тому, що моделюється не лише кількість втрат, а і вартість самого інструменту. У рамках цього підходу визначення збитків пов'язане не з поточною вартістю інструменту, а з його найбільш вірогідною майбутньою вартістю, що з формальної точки зору правильніше. Оцінка VAR здійснювалась за допомогою імітаційного моделювання таким чином.

Першим кроком було визначення форми розподілу та параметрів (функціональних залежностей) ряду, що розглядається. Як зазначалося раніше, відповідно до нормального розподілу випадкових величин на фінансових ринках варіація показників на основі нормального розподілу приблизно домінує в моделюванні фінансового ризику.

Похибка прогнозування втрат методом імітаційного моделювання наведена в таблиці 4.

Таблиця 4. Витрати при імітаційному моделюванні (Ford Motor Company)

Довіра (%)	Втрати: похибка у прогнозі
99	4,4%
97	3,0%
95	2,7%

Не дивлячись на існування певної випадкової складової в результатах моделювання, загалом, при правильному розрахунку вони відповідають оцінкам, отриманим за допомогою інших методів, але можуть трохи відхилитись через характеристики конкретної моделі.

Щоб забезпечити зручність використання (G)ARCH-подібних моделей, необхідно виконати залишкові тести на гетероскедастичність та ARCH-ефект. Спочатку ми виконали графічну перевірку залишків. Враховуючи авторегресію четвертого порядку, графіки нижче показують графіки залишків моделі.

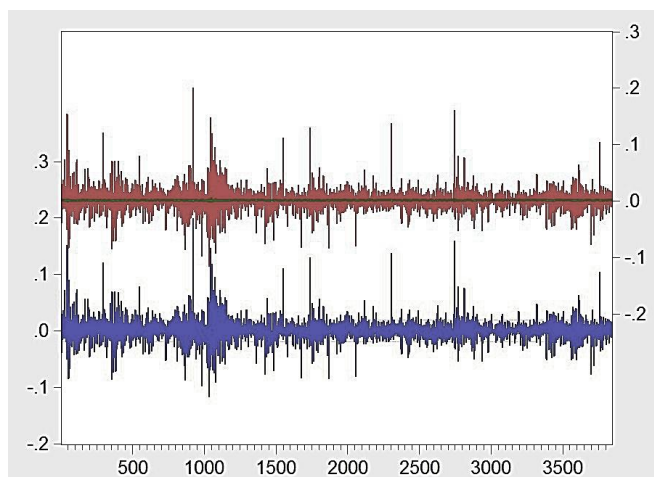


Рисунок 3. Графічний аналіз залишків моделі (Ford Motor Company)

Графічний аналіз дозволив нам підтвердити існування кластерів волатильності: ми могли візуально розрізнити періоди низької та високої волатильності на кожному графіку. Далі ми перевірили ефект ARCH. Нульова гіпотеза  $H_0$  тесту полягає у відсутності ефекту ARCH, а заміна ( $H$ ) - у його наявності. Результати випробувань подані в таблиці 5.

Таблиця 5. ARCH-ефект: результативність теста (Ford Motor Company)

F-statistic	33.22418	Prob. F (1.3835)	0.0000
Obs*R-squared	32.95599	Prob. Chi-Square (1)	0.0000

Prob. Chi-Square відповідає за перевірку нульової гіпотези про відсутність ефекту ARCH в залишках, а значення  $p$  близьке до 0, що дозволяє нам відхилити основну гіпотезу про відсутність ефекту ARCH. Тому ми визначили наявність ефекту ARCH, що означає, що модель ARCH для групи (G) рекомендована для аналізу ряду даних. На Рис. 4 показано гістограму залишків для моделі зазначені вище.

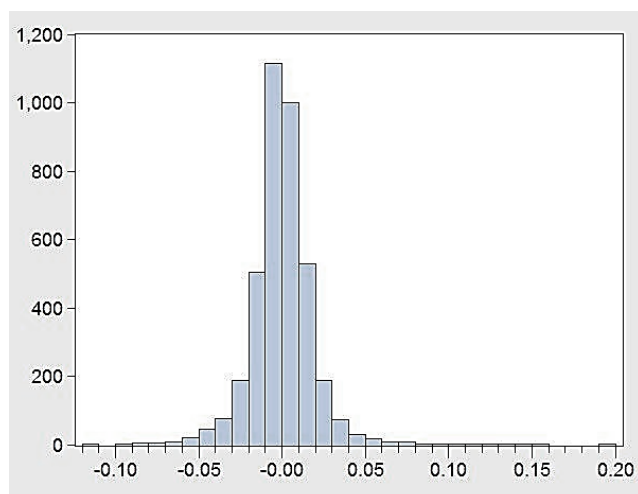


Рисунок 4. Залишки авторегресійної моделі 4 порядку у вигляді гістограми (Ford Motor Company)

Таблиця 6. Статистика для ряду залишків (Ford Motor Company)

Observations	3839
Mean	0,00143
Median	-0,000429
Maximum	0,199396
Minimum	-0,116702
Std. Dev.	0,018955
Skewness	0,903999
Kurtosis	14,44545
Jarque-Bera	21477,15
Probability	0,000000

Оскільки критерій Жака-Бера відкидає нульову гіпотезу, а значення ексцесу більше 3, ми можемо зробити висновок, що залишки розподілені ненормально, що свідчить про доцільність використання ARCH-подібної (G) моделі. У таблиці 7 наведено моделі, які було побудовано для компанії. У таблиці 8 наведено результати оцінювання моделей.

Таблиця 7. Побудовані моделі (Ford Motor Company)

	Ford Motor Company
AARCH(1)	$\sigma^2(t) = 0,00027 + 0,30466 * \varepsilon^2(t-1)$
GAARCH(1,1)	$\sigma^2(t) = 0,00117 + 0,073061 * \varepsilon^2(t-1) + 0,8959 * \sigma^2(t-1)$

Таблиця 8. Оцінювання моделей (Ford Motor Company)

	C	SC
AARCH(1)	-5,153	-5,14
GAARCH(1,1)	-5,2679	-5,2614

Використовуючи інформаційний критерій Akaike для вибору найкращої моделі, порівнюючи моделі для серії " Ford Motor Company ", ми бачимо, що C = -5,26, GAARCH(1,1) є найбільш точним для опису серії,

У таблиці 9 подано оцінки якості прогнозу.

Таблиця 9. Оцінка якості прогнозу.

№	Тип моделі	Прогнозні характеристики		
		MAE	MAPE	Theil
Ford Motor Company	AARCH(1)	0,012600	136,4292	0,9366
	GAARCH(1,1)	0,012592	130,1193	0,94489
TESLA	AARCH(1)	0,019850	94,73257	0,958752
	GAARCH(1,1)	0,019911	98,33821	0,942835
Intel Corporation	AARCH(1)	0,023527	119,9012	0,949652
	GAARCH(1,1)	0,023501	113,6411	0,960401

Підсумовуючи сказане, найкращі результати для оцінювання можливих втрат показує метод моделювання Монте-Карло, який передбачає врахування всіх можливих змін валютного курсу на ринку. Помилки в прогнозуванні можливих втрат виникають лише при несподіваних різких змінах курсу валют, але моделі, засновані на такому підході, можуть швидко адаптуватися до змін на ринку.

Побудовано та порівняно адекватність моделей AARCH та GAARCH на реальних даних, а також проаналізовано якість оцінювання прогнозів умовної дисперсії. Крім того, в практичній частині вибір найкращої моделі здійснюється на реальних прикладах.

На основі виконаної роботи можна зробити деякі висновки та спостереження: обчислено ексцес моделей AARCH(1) і GAARCH(1,1), який у всіх трьох випадках більший, ніж ексцес моделі для нормального розподілу. Це означає розподіл стрибків навколо математичного сподівання. Загалом моделі GAARCH показують кращі результати, ніж моделі AARCH, про що свідчать критерії оцінки моделей. Оцінки якості прогнозу також показують, що модель GAARCH перевершує модель AARCH.

#### 4. ВИСНОВКИ

На сьогоднішній день фінансові системи стають все більш складними та невизначеними, тому виникає необхідність знаходження більш ефективних інструментів аналізу та прогнозування фінансових ризиків для якісного управління ризиками.



Для вимірювання ризику у моїй роботі використовується волатильність акцій найбільших компаній світу. Волатильність також використовується як характеристика для прогнозування ринкових рухів, що є прямим підходом до аналізу проблеми наявності невизначеності.

Найкращі результати для оцінки можливих втрат показує метод моделювання Монте-Карло, який передбачає врахування всіх можливих змін валютного курсу на ринку. Помилки в прогнозуванні можливих втрат виникають лише при несподіваних різких змінах курсу валют, але моделі, засновані на такому підході, можуть швидко адаптуватися до змін на ринку.

Моделювання та вимірювання змін курсу акцій є важливою частиною теорії та практики управління ризиками фінансового ринку. У цій статті розглядається концепція волатильності та аналізуються різні способи її моделювання, показуючи їх відмінності та зв'язки.

Під час виконання роботи була детально вивчена проблема прогнозування волатильності, зокрема за допомогою моделі GARCH. З цією метою розглядаються різні сучасні модельні структури для оцінки динаміки дисперсії.

Це також показує важливість вирішення проблеми невизначеності та ризику у фінансових установах та організаціях. Як запобіжний ризик розглядається волатильність фінансових інструментів, яка також використовується для прогнозування, що, у свою чергу, є способом вирішення проблеми невизначеності. Також вивчаються різні методи розрахунку волатильності.

Варто зазначити, що передбачення умовної дисперсії (тобто волатильності) можна використовувати як практичне застосування для вимірювання ризику фінансових активів, наприклад, під час прийняття рішень стосовно інвестування фінансових активів.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кузнецова Н.В., Бідюк П.І. Теорія і практика аналізу фінансових ризиків: системний підхід, монографія. Київ 2020, 400 с.
2. Бідюк П.І., Романенко В.Д., Тимощук О.Л. Аналіз часових рядів: монографія. Київ: Політехніка, 2012. 520 с.
3. Башкіров О.В. Порівняльний аналіз VAR-методів оцінки ризику активів банку О.В. Башкіров, Проблеми і перспективи розвитку банківської системи України: зб. наук. праць ДВНЗ «УАБС НБУ». – Вип. 14. –С. 302–309.
4. Moraux F. How valuable is your VaR? Large sample confidence intervals for normal VaR. F. Moraux. Journal of risk management in financial institutions. – 2011. – № 4.2. – P. 189–200. – [Електронний ресурс]. – Режим доступу : <https://perso.univ-rennes1.fr/franck.moraux/research/JRMFI.pdf>
5. Engle R. F. Autoregressive conditional heteroskedasticity with estimates of the variance of U. K. inflation. *Econometrica*. 1982. Vol. 50. P. 987–1008.
6. Theodore Barnhill and William Maxwell. Modeling correlated market and credit risk in fixed income portfolios. *Journal of Banking and Finance*, 26:347–374, 2002
7. Lin C. and Shen S. (2006), "Can the student-  $t$  distribution provide accurate value at risk?", *Journal of Risk Finance*, Vol. 7 No. 3, pp. 292-300.

# СПЕЦІАЛЬНІ ФУНКЦІЇ МАТРИЧНОГО АРГУМЕНТУ

Шутяк Д.О.<sup>1</sup>, Подколзін Г.Б.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>dima.shutyak@gmail.com, <sup>2</sup>podkolzin.gleb@lil.kpi.ua [0000-0002-7120-2772]

**Вирішення задач та робота з функціями аргументів вищих розмірностей завжди були складними процесами, до яких найчастіше використовувався прямий підхід чисельного наближення, який в собі несе ряд недоліків та складностей імплементації та застосування. На противагу цьому в даній роботі за мету ставилось розглянути більш аналітичний підхід до вирішення проблеми, а саме з використанням властивостей матричної експоненти та її зв'язків з жордановою канонічною формою матриці і фокус на функціях двох аргументів. В результаті дослідження було виведено їх означення, зв'язки зі скалярними аналогами, властивості та тотожності, та певні загальні засади щодо роботи з конструкціями такого вигляду.**

**Ключові слова: спеціальні функції, матричні функції, жорданова матриця, матрична експонента.**

## 1. ВСТУП

Дуже часто в сучасних практичних задачах з'являється потреба в роботі з великими масивами даних і зі змінними більш складного характеру. Це можуть бути як і складні системи керування об'єктами, комп'ютерне моделювання комплексних процесів (напр. терморідинне моделювання) так і навчання просторових моделей зі складними вхідними параметрами.

З стрімким ростом обчислювальних можливостей комп'ютерів, задачі, які перед ними ставляться продовжать ставати ще складнішими, тому дуже важливо виявляти, розуміти та позбавлятися будь-яких проблем та лімітацій, які вони мають. Великою проблемою є те, що більшість комп'ютерних алгоритмів, що використовуються в цих задачах, базуються на чисельних наближеннях складних функцій та конструкцій на кожному кроці, що може призвести до накопичення обчислювальної похибки і негативно вплинути на результати. Саме тому розробка аналітичних методів та загальних засад роботи з цими структурами це дуже важлива зараз задача, особливо для функцій та рівнянь, які використовуються в високоточних обчисленнях комплексних динамічних систем, де мінімальні відхилення можуть мати дуже суттєві наслідки

## 2. МЕТОДИ ОБЧИСЛЕННЯ СПЕЦІАЛЬНИХ ФУНКЦІЙ

На сьогоднішній день існує багато підходів до роботи з складними функціями матричного аргументу, але дуже часто всі ці підходи були створені для вирішення певного типу задач і розроблені абсолютно незалежно одне від одного, без використання нароблених матеріалів інших сторін. При вивченні нескінченновимірної теорії представлень була розроблена теорія операторнозначних функцій Бесселя матричного аргументу над дійсними алгебрами ділення.

Ці операторнозначні функції, в свою чергу були виведені і базувались на теорії Пітера-Вейля, застосованої до дій компактної групи на евклідових просторах [5].

З іншої сторони, абсолютно інший підхід і клас функцій матричного аргументу існує в теорії квантової механіки, де за допомогою «принципу скорочення», який використовується в квантовій фізиці, було виведено метод, який спирався на розкладання в ряди функцій комплексно значного матричного аргументу [4].

Ще один із популярних підходів до знаходження більш загальних функцій матричного вигляду і побудови певних збіжностей у вигляді це індуктивний підхід [5]. Ця методологія базується на знаходженні поведінкових шаблонів результуючої функції, починаючи спочатку з різних підмножин множини визначення досліджуваних функцій. Цей підхід, порівняно з попередніми, є набагато більш універсальним і мав в собі багато вагомих результатів, більш конкретно в узагальненні гіпергеометричних функцій, але все ще мав в собі ряд недоліків, наприклад основний фокус на репрезентації функцій за допомогою суми рядів, що не є недоліком саме по собі, але в контексті подальшого практичного застосування знайдених функцій мало не найкращі успіхи.

В даній роботі для дослідження був обраний підхід з використанням жорданової канонічної форми матриць і як об'єкт дослідження були обрані функції які містять в собі операції над показниковими функціями, оскільки це дозволяє користуватись властивостями матричної експоненти та її зв'язку з жордановою формою [1]. Також саме серед цих функцій є багато дуже відомих та значущих функцій, які або використовуються самі по собі, або як частини складнішої системи у багатьох сферах наукової діяльності [3]. На відміну від попередньої роботи на цю тематику, в даній роботі було розглянуто функції двох матричних аргументів, оскільки в процесі дослідження було виявлено, що в багатьох пакетах відкритого доступу різних серед програмування, часто ці функції розглядались як функції одного матричного та одного скалярного аргументу, що дуже суттєво звужує можливості застосування цих функцій.

Загальна процедура аналізу обраних функцій починалась зі знаходження явного вигляду означення обраної функції для жорданових матриць  $J_{r_i} \lambda_i$  та наступне узагальнення до довільних відповідних матриць  $A_i$ . На даному етапі дослідження було також прийняте рішення розглядати аргументи матриці, які між собою комутують, оскільки без цього складність поставленої задачі ставала практично не підйомною. Це спричинене тим, що обрані функції представлені у вигляді добутку двох показникових компонент і це призводить на етапі узагальнення до утворення структури наступного вигляду:

$$U_1 e^{f_1 t} J_1 U_1^{-1} U_2 e^{f_2 t} J_2 U_2^{-1},$$

тобто добуток двох різних не вироджених матриць  $U_1$  та  $U_2$  різних базисів. Це рішення перейти до дослідження комутуючих матриць також дало можливість узагальнити певні властивості функцій скалярного аргументу, які б без цього також не вдалось вивести в іншому випадку.

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для дослідження було обрано декілька відомих спеціальних функцій двох змінних і в остаточному варіанті задокументовано три із них, тому що вони добре репрезентують різні степені успішності використання підходу, а також добре ілюструють як потенціал так і складності поставленої задачі.

Першою було розглянуто Бета-функцію. Її було обрано через зв'язок та певні збіжності з Гама-функцією, яка розглядалась у попередній роботі на цю тематику. Також вона має

широкий ряд застосувань в різних дослідницьких сферах, а також має багато властивостей і пов'язаних тотожностей, які також були винесені до узагальнення. Як результат було отримано явний вигляд означення для двох довільних жорданових матриць  $J_1 \lambda_1, J_2 \lambda_2$  однакової розмірності і результат виглядав наступним чином:

$$B_{J_1 \lambda_1, J_2 \lambda_2} = \begin{matrix} B_{\lambda_1 + 1, \lambda_2 + 1} & \dots & \frac{1}{j!} \frac{\partial^{m-l} B(x, y)}{\partial x^j \partial y^{m-l-j}} & \begin{matrix} x = \lambda_1 + 1 \\ y = \lambda_2 + 1 \end{matrix} \\ \vdots & \ddots & \vdots & \\ 0 & \dots & B_{\lambda_1 + 1, \lambda_2 + 1} & \end{matrix}$$

Як ми бачимо, результуюча матриця – це верхньотрикутна матриця, на основній діагоналі якої знаходиться скалярна Бета-функція з відповідними значеннями, а всі наступні елементи являють собою змішані похідні порядку, який відповідає різниці індексів. Це означає, що кожен елемент матриці залежить не від своєї абсолютної позиції в ній, а лише від різниці індексів, а тобто всі елементи на відповідних діагоналях є між собою рівними. Цей факт суттєво зменшує кількість обчислень, які потрібно зробити для того, щоб знайти значення функції для заданих матриць і пришвидшує роботу з нею.

Тепер, враховуючи попередньо вказану комутативність аргументів функції, узагальнення до довільних матриць зводиться до перетворення, аналогічного до самої жорданової форми:

$$B_{X, Y} = U B_{J_1, J_2} U^{-1},$$

де  $J_1, J_2$  – відповідні жорданові матриці до  $X, Y$ , а  $U$  невинроджена матриця, яка, в зв'язку з існуванням спільного базису комутуючих матриць, є однаковою для  $X$  та  $Y$ .

Наступними було отримано декілька основних властивостей Бета-функції, які вдалось узагальнити в повному або частковому обсязі:

$$1. B_{I, J \lambda_1} = \begin{matrix} \frac{1}{\lambda_1 + 1} & \dots & \frac{-1^{k-1}}{(k-1)! \lambda_1 + 1^{k-1}} \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{\lambda_1 + 1} \end{matrix};$$

$$B_{I, X} = U B_{I, J} U^{-1}.$$

$$2. B_{J_1 \lambda_1 + I, J_2 \lambda_2} = \begin{matrix} B_{\lambda_1 + 1, \lambda_2 + 1} \cdot \frac{\lambda_1 + 1}{\lambda_1 + \lambda_2 + 2} & \dots & \frac{1}{j!} \frac{\partial^{m-l} B(x, y)}{\partial x^j \partial y^{m-l-j}} & \begin{matrix} x = \lambda_1 + 2 \\ y = \lambda_2 + 1 \end{matrix} \\ \vdots & \ddots & \vdots & \\ 0 & \dots & B_{\lambda_1 + 1, \lambda_2 + 1} \cdot \frac{\lambda_1 + 1}{\lambda_1 + \lambda_2 + 2} & \end{matrix};$$

$$B_{X + I, Y} = U B_{J_1 + I, J_2} U^{-1}.$$

3. Тотожність Паскаля або різницеве рівняння:

$$B_{X + I, Y} + B_{X, Y + I} = B_{X, Y}.$$

Слід зазначити, що хоча властивість 1 та тотожність Паскаля були явні і лаконічно виведені з використанням лише базових елементів, властивість 2, а саме ітеративність функції, яка б дозволила зводити невідомі результати ітеративного процесу до попередньо відомих значень, не була. Через додану змішану похідну в означенні, явного вигляду, який би залежав лише від  $\lambda_1 + 1$  та  $\lambda_2 + 1$  знайти не вдалось, але цей результат все ще є вагомим і не повинен задати складностей в обчисленні значень функції.

Наступною функцією є Тета-функція Якобі, а саме основна функція вигляду

$$v_{J_{r_1}, J_{r_2}} \lambda_2 = \sum_{n=-\infty}^{\infty} Q^{n^2} H^n,$$

де  $Q = e^{\pi i J_{r_2} \lambda_2}$ ,  $H = e^{2\pi i J_{r_1} \lambda_1}$ .

Явний матричний вигляд її був отриманий наступний:

$$v_{J_{r_1} \lambda_1, J_{r_2} \lambda_2} = \begin{matrix} v_{\lambda_1, \lambda_2} & \dots & \frac{1}{j!} \frac{\delta^{m-l} v_{z, \tau}}{\delta z^{m-l-j} \delta \tau^j} & z = \lambda_1 \\ \vdots & \ddots & \vdots & \tau = \lambda_2 \\ 0 & \dots & v_{\lambda_1, \lambda_2} & \end{matrix},$$

для  $Z, T$  комутуючих  $= U v_{J_r, J_r} U^{-1}$ .

Аналогічно Бета-функції, матриця має скалярну Тета-функцію на основній діагоналі та суму змішаних похідних на наступних, що привело до міркувань, щодо загального вигляду функцій двох змінних, які мають схожу структуру.

Як властивості Тета-функції Якобі були виведені дві основні, а саме 1-періодичність та  $\tau$ -квазіперіодичність:

1. 1-періодичність по першій координаті:

$$v_{Z+I, T} = v_{Z, T};$$

2.  $T$ -квазіперіодичність по першій координаті:

$$= \begin{matrix} v_{\lambda_1, \lambda_2} \cdot e^{-\pi i \lambda_2 + 2\lambda_1} & \dots & \frac{1}{j!} \frac{\delta^{m-l} v_{z, \tau}}{\delta z^{m-l-j} \delta \tau^j} & z = \lambda_1 + \lambda_2 \\ \vdots & \ddots & \vdots & \tau = \lambda_2 \\ 0 & \dots & v_{\lambda_1, \lambda_2} \cdot e^{-\pi i \lambda_2 + 2\lambda_1} & \end{matrix};$$

Як ми бачимо у властивості 2, знову ж таки основним бар'єром є наявність змішаних похідних досліджуваної функції, через які не вдається чітко повернутись до початкових координат  $\lambda_1$  та  $\lambda_2$ .

Останньою дослідженою в роботі функцією є Дзета-функція Гурвіца, яка спочатку була винесена на дослідження як і попередні дві, а більш конкретно через її зв'язок з попередньою Тета-функцією Якобі та тотожністю, через яку вони пов'язані. Проте, через структуру, відмінну від двох попередніх функцій, Дзета-функція Гурвіца є гарним прикладом складності поставленої задачі і експоненційного зростання комплексності необхідних обчислень з додаванням додаткових матричних операцій. Ця функція в скалярному випадку має наступний вигляд:

$$\zeta(s, a) = \frac{1}{\Gamma(s)} \cdot \int_0^{\infty} \frac{t^{s-1} e^{-at}}{1 - e^{-t}} dt,$$

що для матричних аргументів набуває вигляду:

$$\zeta(S, A) = \Gamma(S)^{-1} \cdot \int_0^{\infty} \frac{1}{1 - e^{-t}} e^{\ln t S} e^{-tA} dt.$$

Підінтегральний вираз є дуже схожим на попередні дві функції, і дійсно, вигляд результуючої матриці є ідентичним до них, але тепер до неї додається ще один множник  $\Gamma(S)^{-1}$  – матриця, обернена до Гамма-функції матричного аргументу. Цей приклад ілюструє багато потенційних складностей та обмежень, які з'являються для поставленої задачі, але в цей же час слід зазначити, що ця ситуація зумовлена безпосередньо складною структурою функції, аніж недовіками обраного методу. Для цього було все ще в загальному вигляді перевірено розклад в жорданову канонічну форму і показано, що з додатковою складністю механічних обчислень цей зв'язок не пропадає і, з використанням більш надійних обчислювальних методів, можна досягти бажаного результату.

$$\zeta(S, A) = U \zeta(J, \lambda_1, J, \lambda_2) U^{-1}.$$

#### 4. ВИСНОВКИ

Потреба в вирішенні складніших задач та моделюванні складніших систем вищих розмірностей продовжить з'являтися у всіх сферах прикладних наукових досліджень, а тому і методи роботи з цими конструкціями будуть ставати лише більш актуальними. Обраний у роботі метод це всього лише один із підходів до цієї проблеми і він несе в собі ряд унікальних переваг та недоліків, які добре ілюструються в отриманих результатах.

Акцентування уваги саме на аналітичному загальному вигляді функції та її властивостях дозволяє використовувати їх у більш широкому спектрі різних сфер, а тобто надає більш універсальне рішення проблеми, ніж методи, які базуються на конкретному використанні певної функції у певному типі задач.

Як загальний висновок до розглянутих функцій, можна сказати, що функції двох змінних, представлених у вигляді добутку показникових функцій матричного аргументу та певних базових операцій з ними (інтегрування, сума ряду та ін..) мають багато збіжностей між собою у вигляді результуючої матриці та поведінці з боку властивостей.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kishka, Z., Saleem M., Elrawy A. The matrix of matrices exponential and application. *International Journal of Mathematical Analysis*, 2019. Vol. 13. P. 81–97.
2. Jódar L., Cortés J. C. Some properties of Gamma and Beta matrix functions. *Applied Mathematics Letters*, 1998. Vol. 11, No. 1. P. 89–93.
3. Andrews, G., Askey, R., & Roy, R. (1999). *Special Functions (Encyclopedia of Mathematics and its Applications)*. Cambridge: Cambridge University Press. doi:10.1017/CBO9781107325937
4. Whittaker, E., & Watson, G. (1996). *A Course of Modern Analysis, Part 2, Transcendental Functions*. (4th ed., Cambridge Mathematical Library). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511608759
5. Alan T. James (1975). *Special Functions of Matrix and Single Argument in Statistics, Theory and Application of Special Functions*, Academic Press, cc. 497-520, ISBN 9780120648504, <https://doi.org/10.1016/B978-0-12-064850-4.50016-1>.

# МОДИФІКОВАНИЙ МЕТОД «АКТОР-КРИТИК» ДЛЯ ФОНДОВОГО РИНКУ ЯК ЧАСТИНА СИТУАЦІЙНО-АНАЛІТИЧНОЇ ПЛАТФОРМИ

Батейко Е.А.<sup>1</sup>, Касьянов П.О.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>bateikoeduard@gmail.com [0000-0002-6729-5611],

<sup>2</sup>kasyanov@i.ua [0000-0002-6662-0160]

**Пропонується модифікований метод «актор-критик» (SAC) [1] та метод глибокого детермінованого градієнту стратегії (DDPG) [2] у період з 01.01.2022 по 28.09.2022. Визначений інтервал тренування дозволяє навчити модель з обмеженими обчислювальними ресурсами. Ефективність роботи алгоритмів визначено в порівнянні з ETF [3] за допомогою метрик річного прибутку, кумулятивного прибутку, коефіцієнта Шарпа. Отримані результати на заданому часовому інтервалі кращі за ETF. Дана робота було здійснена як частина імплементації ситуаційно-аналітичної платформи фінансового ринку.**

**Ключові слова: Multiple stock trading, Nasdaq 100, QQQ, DDPG, PPO, SAC, Reinforcement Learning, A2C, TD3.**

## 1. ВСТУП

Розглядається система прийняття рішень по збалансованості портфеля активами за допомогою методів навчання з підкріпленням. Для реалізації було використано фреймворк FinRL[4] який надає можливість встановлювати своє середовище, агентів, тощо для вирішення задач з торгівлі за допомогою навчання з підкріпленням. Pyfolio[5] для порівняння метрик портфеля та бейзлайна. Weight&Biases[6] фреймворк для збереження результатів експериментів та зручної агрегації метрик. Фінансові дані отримані з Yahoo Finance [7].

Завдання натренувати агента, який буде торгувати акціями з індексу Nasdaq 100 та покаже результати прибутку більші ніж QQQ ETF фонд[8], в подальшому будемо називати його бейзлайн або бенчмарк. Оскільки QQQ ETF наслідує Nasdaq 100, тому фонд розглядається як альтернатива нашому агенту. Було розглянуто задачу формування портфеля. Початковою сумою в 100.000 доларів, агенту надано перелік компаній які належать індексу Nasdaq 100. Початкова кількість акцій 0. Математичне формулювання продемонстровано в розділі 2.

Було проведено 25 експериментів з 5-ма алгоритмами навчання з підкріпленням: жадібний «актор-критик» (a2c), DDPG, «уповільнення близнюків» DDPG (td3), оптимізація проксимальної політики (PPO) та SAC[1]. З різними часовими інтервалами для тренування та тестування моделей. Тестування роботи алгоритму будемо називати також торгівлею. Часові проміжки на тренуванні 2010-01-01 по 2021-07-01 (експерименти під номером 0 та 1); 2010-01-01 по 2021-01-01 (експеримент 2); 2022-01-01 по 2022-06-01 (експерименти 3,4). Періоди на торгівлі 2021-01-01 по 2022-09-28 (експерименти 0, 1); 2021-01-01 по 2021-12-30 (експеримент 2); 2022-06-01 по 2022-09-28 (3, 4) відповідно. Часові проміжки були обрано таким чином, щоб покрити різні умови, такі як: агент тренований на зростаючому тренді, а



торгівля на змінному із зростаючого до спадного; агент тренований на зростаючому ринку та торгує на зростаючому; агент тренований на невизначеному ринку та торгує на змінному ринку зі спадного до частково зростаючого. Результати проведених експериментів та код збережені на GitHub репозиторій[9]. Кількість ітерацій тренування для експериментів 0-3 це 50.000, для 4 це 100.000 кроків.

Навчання з підкріпленням в задачі торгівлі розглядаються в статті [10], де автори пропонують нову архітектуру під назвою тимчасовий адаптер маршрутизації (TRA), щоб розширити можливості існуючих моделей прогнозування акцій із можливістю моделювання декількох моделей торгівлі акціями. По суті, TRA — це відносно легкий модуль, який складається з набору незалежних предикторів для вивчення кількох шаблонів, а також блока для відправлення зразків до різних предикторів. В роботі [11] розглядалась торгівля як модель частково спостережуваного марковського процесу прийняття рішень (POMDP), враховуючи обмеження, які накладає фондовий ринок, такі як ліквідність і транзакційні витрати. Останнім часом було проведено багато досліджень для розробки рішень глибокого навчання для прогнозування акцій. Серед них [12] пропонує варіант LSTM для виявлення багатьох частотних шаблонів у даних фондового ринку, [13] розробляє двоетапний LSTM на основі шар уваги яка дозволяє фокусуватися на найважливіших речах для прогнозування акцій, [14] додатково покращують шар уваги у LSTM шляхом використання змагального навчання, в [15] побудовано трансформер, який може принести приріст продуктивності, хоч він і потребує великих об'ємів даних та обчислювальних ресурсів. В [16] розробляють стохастичну рекурентну модель (SRM) для розгляду адаптивності фондових ринків і [17] розробляють структуру для використовувати графік знань для прогнозування акцій на основі подій. В даній роботі пропонується модифікований метод «актор-критик» (SAC) та метод глибокого детермінованого градієнту стратегії (DDPG) у період з 01.01.2022 по 28.09.2022. Визначений інтервал тренування дозволяє навчити модель з обмеженими обчислювальними ресурсами. Ефективність роботи алгоритмів визначено в порівнянні з ETF за допомогою метрик річного прибутку, кумулятивного прибутку, коефіцієнта Шарпа. Отримані результати на заданому часовому інтервалі кращі за ETF. Дана робота було здійснена як частина імплементації ситуаційно-аналітичної платформи для фінансового ринку

## 2. ФОРМАЛІЗАЦІЯ ЗАВДАННЯ

### 2.1. Простір станів, простір дій і функція винагороди

Простір станів  $S$ . Простір станів описує спостереження, які агент отримує від середовища. Подібно до того, як людині-трейдеру потрібно проаналізувати різну інформацію перед здійсненням угоди, наш торговий агент спостерігає за багатьма різними функціями, щоб краще навчатися в інтерактивному середовищі[18]:

- баланс  $b_t \in R_+$ : сума грошей, що залишилася на рахунку на поточному кроці часу  $t$ ;
- Власні акції  $h_t \in Z_+^n$  : поточні акції для кожної акції,  $n$  представляє кількість акцій[18];
- ціна закриття  $p_t \in R_+^n$  : одна з найбільш часто використовуваних характеристик;
- відкриття/високі/низькі ціни  $o_t, h_t, l_t \in R_+^n$  : використовується для відстеження змін курсу акцій;
- обсяг торгів  $\mathbb{Q}_t \in R_+^n$ : загальна кількість акцій, якими торгують протягом торгової сесії;
- технічні індикатори: розбіжність конвергенції ковзного середнього (MACD)  $M_t \in R^n$  та індекс відносної сили (RSI)  $R_t \in R_+^n$  тощо. Багаторівнева деталізація: дозволяє частоту даних вищевказаних функцій на щоденній, погодинній або хвилинній основі[18].

Простір дій  $A$ . Простір дій описує дозволені дії, які агент взаємодіє з середовищем. Зазвичай  $m \in A$  включає три дії:  $a \in \{-1, 0, 1\}$  де -1; 0; 1 означає продаж, утримання та купівлю однієї акції. Крім того, дія може виконуватися на кількох акціях. Автори використовують простір дії  $\{-k, \dots, -1, 0, -1, \dots, k\}$ , де  $k$  позначає кількість куплених акцій. Наприклад, «Купити 10 акцій AAPL» або «Продати 10 акцій AAPL» — це 10 або -10 відповідно.[19]

Функція винагороди  $r(s, a, s')$  є механізмом стимулювання агента навчитися здійснювати кращі дії. Існує багато форм функцій винагороди. FinRL пропонує наступні[18]: зміна вартості портфеля, коли дія  $a$  виконується в стані  $s$  і досягає нового стану  $s'$ , тобто  $r(s, a, s') = v' - v$ , де  $v', v$  представляють вартість портфеля в стані  $s'$  та  $s$  відповідно; логарифмічний прибуток портфеля отримується, коли дія  $a$  виконується в стані  $s$  і переходить у новий стан  $s'$ , тобто  $r(s, a, s') = \log(v'/v)$ ; коефіцієнт Шарпа для періодів  $t = \{1, \dots, T\}$ , тобто,  $S_T = \text{mean}(R_t) / \text{std}(R_t)$  де  $R_t = v_t - v_{t-1}$ . Ми використовуємо визначені за замовчуванням параметри для середовища та агентів які задані в FinRL.

### 3. ЗАГАЛЬНІ РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для кращого аналізу та розуміння відмінності в метриках між бейзлайном та бектестуванням в таб. 1 по таб. 4 наявні стовпці з назвами baseline, backtest, diff\_abs, normalized. Де baseline це значення метрики для бейзлайна, backtest це значення метрики для бектестування, diff\_abs обчислюється за формулою  $\text{diff\_abs} = \text{backtest} - \text{baseline}$ ,  $\text{normalized} = \text{diff\_abs} / |\text{baseline}|$ . Агрегація diff\_abs демонструє різницю між бектестуванням та бейзлайном, а normalized демонструє відсоткову різницю в обрахунку на бейзлайн.

Експерименти мають назву відповідно структури <НАЗВА\_АЛГОРИТМА><СИМВОЛ\_БЕЙЗЛАЙНА><НОМЕР\_ЕКСПЕРИМЕНТА>.

З усіх експериментів отримали шість які показали себе краще за бейзлайн. Це експерименти під назвами sac^NDX\_3, ddpq^NDX\_4, ppo^NDX\_4, sac^NDX\_2, a2c^NDX\_0, ddpq^NDX\_3 з різницею в річному прибутку між бейзлайном в 56.3%, 20.9%, 8.5%, 5.4%, 3.5% та 1.2%, відповідно табл. 1. Різницею в кумулятивному прибутку в 18.5%, 7.8%, 5.5%, 4%, 3.3%, 0.5% відповідно експерименти sac^NDX\_3, ddpq^NDX\_4, sac^NDX\_2, a2c^NDX\_0, ppo^NDX\_4 та ddpq^NDX\_3, продемонстровано в табл. 2.

Таблиця 1 – annual\_return для бектестування та бейзлайн

Name	baseline	backtest	diff_abs	normalized	Name	baseline	backtest	diff_abs	normalized
sac^NDX_3	-0.293	0.27	0.563	1.923	sac^NDX_2	0.303	0.357	0.054	0.179
ddpq^NDX_4	-0.293	-0.084	0.209	0.713	a2c^NDX_0	-0.185	-0.15	0.035	0.187
ppo^NDX_4	-0.293	-0.208	0.085	0.289	ddpq^NDX_3	-0.293	-0.28	0.012	0.043

Таблиця 2 – cumulative\_return для бектестування та бейзлайн

Name	baseline	backtest	diff_abs	normalized	Name	baseline	backtest	diff_abs	normalized
sac^NDX_3	-0.105	0.08	0.185	1.759	a2c^NDX_0	-0.223	-0.183	0.04	0.181
ddpq^NDX_4	-0.105	-0.028	0.078	0.736	ppo^NDX_4	-0.105	-0.072	0.033	0.314
sac^NDX_2	0.299	0.354	0.055	0.184	ddpq^NDX_3	-0.105	-0.1	0.005	0.048

Таблиця 3 – sharpe\_ratio для бектестування та бейзлайн

Name	baseline	backtest	diff_abs	normalized	Name	Baseline	backtest	diff_abs	normalized
sac^NDX_3	-1	0.818	1.818	1.819	ddpg^NDX_3	-1	-0.781	0.218	0.218
ddpg^NDX_4	-1	-0.01	0.989	0.99	ppo^NDX_4	-1	-0.839	0.161	0.161
sac^NDX_2	1.521	2.156	0.634	0.417	a2c^NDX_4	-1	-0.953	0.047	0.047

Таблиця 4 – max\_drawdown для бектестування та бейзлайн

Name	baseline	backtest	diff_abs	normalized	Name	Baseline	backtest	diff_abs	normalized
a2c^NDX_0	-0.329	-0.252	0.076	0.233	td3^NDX_1	-0.329	-0.294	0.034	0.104
ddpg^NDX_3	-0.172	-0.169	0.003	0.019	ddpg^NDX_4	-0.172	-0.182	-0.01	-0.058
sac^NDX_2	-0.109	-0.065	0.044	0.402	ppo^NDX_4	-0.172	-0.153	0.019	0.112

## 4. АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ АЛГОРИТМІВ

### 4.1. «Актор-критик» метод

Даний експеримент проводився в часових рамках торгування 2022-06-01 – 2022-09-28 та період тренування 2022-01-01 – 2022-06-01. sac^NDX\_3 продемонстрував найкращі результати при даних умовах на відміну від інших алгоритмів, в той самий час ddpг^NDX\_3 теж дав результати кращі ніж бенчмарк, однак гірші за sac^NDX\_3. Із значеннями основних метрик річний прибуток для QQQ це -29.3%, для поточного алгоритму це 27%, різниця у 56.3% річних таб. 1. З іншого боку кумулятивний прибуток показав для бенчмарка -10.5%, для нашого портфолію 8%, що в свою чергу являє більше на 18,5% від бейзлайна, таб. 2. Коефіцієнт Шарпа продемонстрував цифри бенчмарка -1 та алгоритму 0.82 таб. 3. , як бачимо на цьому періоді індекс продемонстрував більший ризик ніж поточний підхід. Порівнюючи максимальну просадку даного портфеля порівняно з бенчмарком маємо -17.2% та -12.1 %, таб. 4 для бенчмарку та портфеля відповідно, відповідно алгоритм виявився більш стійкий до просадок. Портфель мав 4 періоди просадки: 02.06.2022 – 22.08.2022, 16.08.2022 – 08.09.2022, 09.09.2022– невідомо, 10.08.2022 – 15.08.2022 з чистими падіннями на 12,07%, 10,72%, 5,50% та 2,36%. З тривалістю в 50, 18, невідомо (спад не закінчився) та 4 дні. Основний прибуток в останньому місяці 14%, що й дало такий великий розрив з бенчмарком оскільки індексний фонд падав на цьому проміжку.

### 4.2 Алгоритм глибокого детермінованого градієнта стратегії

Даний експеримент проводився в тих же часових інтервалах що й SAC. ddpг^NDX\_4 показав найкращі результати при даних умовах на відміну від інших алгоритмів, в той самий час ppo^NDX\_4 теж дав результати кращі ніж бенчмарк, однак гірші за ddpг^NDX\_4. Що в свою чергу продемонструвала, що для алгоритму SAC необхідно менше кількість тренувань на відміну з алгоритмами DDPG та PPO. Із значеннями основних метрик річний прибуток для QQQ це -29.3%, для поточного алгоритму -8,4%, різниця у 20.9% річних таб. 1. Хоча портфель має негативний результат однак він кращий за бенчмарк, що дає змогу говорити про доцільність його використання. З іншого боку кумулятивний прибуток показав для бейзлайна -10.5%, для порфолію -2.8%, що в свою чергу являє більше на 7,8%, таб. 2. Коефіцієнт Шарпа продемонстрував цифри бенчмарка -1 та алгоритму -0.01 таб. 3, як бачимо на цьому періоді індекс більш ризикований. Однак коефіцієнт демонструє що портфель мав

прибуток близький до нуля також демонструвати від'ємні результати. Порівнюючи максимальну просадку даного портфеля порівняно маємо -17.2% та -18.2 %, таб. 4 для QQQ та портфеля відповідно. Як видно хоча портфель має майже такий самий рівень просадок, однак результуючі значення набагато кращі. Портфель мав багато періодів просадки, що демонструє кумулятивний прибуток. П'ять найбільших по величині просадки: 12.08.2022 – невідомо (спад не закінчився), 02.06.2022 – 21.07.2022, 21.07.2022 – 28.07.2022, 04.08.2022 – 10.08.2022, 10.08.2022 – 12.08.2022 та з чистими падіннями на 18,25%, 15,95%, 4,48%, 3,48% та 1,77%. З тривалістю в невідомо (спад не закінчився), 36, 6, 5 та 3 днів. Основний прибуток полягає в липні це 16% прибутку, що й дало такий великий розрив з бенчмарком. Хоча індексний фонд теж ріс на цьому проміжку, проте портфель показав кращий результат.

#### 4. ВИСНОВКИ

Результати дослідження демонструють що поточний підхід є ефективним на заданому інтервалі тренування 2022-01-01 – 2022-06-01 та інтервалі тестування 2022-06-01 – 2022-09-28. Для інвестиційного фонду QQQ, на проміжку тестування річний прибуток має значення в -23.3%. В свою чергу для алгоритму SAC це 27%, для DDPG – -8.4%. Різниця між бенчмарком та SAC це 56.3% річних, між бейзлайном та DDPG це 20.9%. Хоча DDPG показав від'ємні результати для річного прибутку, він продемонстрував кращі результати ніж QQQ, на 20.9%. Кумулятивний прибуток для QQQ має значення в -10.9%, для SAC це значення 8%, для DDPG – -2.8%. Різниця в прибутках між QQQ та SAC це 18.5%, між QQQ та DDPG це 7.8%. Зауважимо, що кумулятивний прибуток для DDPG має від'ємні значення, однак кращі на 7.8% ніж індексний фонд. Коефіцієнт Шарпа продемонстрував цифри для бенчмарка -1 та алгоритму SAC 0.82, для DDPG – -0.01. В майбутніх роботах слід дослідити дієздатність методу на ширшому часовому проміжку, перевірити стійкість алгоритмів відносно часу, визначити періоди перетренування моделей.

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. International Conference on Machine Learning, 2018
2. Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actorcritic for mixed cooperative-competitive environments. In Advances in Neural Information Processing Systems, pages 6379–6390, 2017.
3. What Is an Exchange-Traded Fund (ETF)? URL: <https://www.investopedia.com/terms/e/etf.asp> (дата звернення: 05.11.2022)
4. FinRL: Financial Reinforcement Learning URL: <https://github.com/AI4Finance-Foundation/FinRL> (дата звернення: 06.11.2022)
5. Pyfolio URL: <https://quantopian.github.io/pyfolio/> (дата звернення: 06.11.2022)
6. Weight and Biases URL: <https://wandb.ai/> (дата звернення: 06.11.2022)
7. Yahoo Finance URL: <https://finance.yahoo.com/> (дата звернення: 07.11.2022)
8. Invesco QQQ ETF URL: <https://www.invesco.com/qqq-etf/en/home.html> (дата звернення: 10.11.2022)
9. Actor-Critic-and-Deep-Deterministic-Policy-Gradient-in-the-2022 URL: <https://github.com/bateikoEd/Actor-Critic-and-Deep-Deterministic-Policy-Gradient-in-the-2022/tree/master> (дата звернення: 12.11.2022)
10. Learning Multiple Stock Trading Patterns with Temporal Routing Adaptor and Optimal Transport arXiv:2106.12950 (2021).

11. Deep Reinforcement Learning Approach for Trading Automation in The Stock Market arXiv:2208.07165 (2022)
12. Dawei Cheng, Fangzhou Yang, Xiaoyang Wang, Ying Zhang, and Liqing Zhang. 2020. Knowledge Graph-based Event Embedding Framework for Financial Quantitative Investments. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2221–2230.
13. Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. 2017. A dual-stage attention-based recurrent neural network for time series prediction. arXiv:1704.02971 (2017)
14. Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, and Tat-Seng Chua. 2019. Enhancing Stock Movement Prediction with Adversarial Training. IJCAI (2019)
15. Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo, and Jian Guo. 2020. Hierarchical Multi-Scale Gaussian Transformer for Stock Movement Prediction. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20. International Joint Conferences on Artificial Intelligence Organization, 4640–4646.
16. Yumo Xu and Shay B Cohen. 2018. Stock movement prediction from tweets and historical prices. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 1970–1979.
17. Dawei Cheng, Fangzhou Yang, Xiaoyang Wang, Ying Zhang, and Liqing Zhang. 2020. Knowledge Graph-based Event Embedding Framework for Financial Quantitative Investments. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2221–2230.
18. FinRL-Meta: A Universe of Near-Real Market Environments for Data-Driven Deep Reinforcement. Learning in Quantitative Finance arXiv:2112.06753 (2022)

# МОДЕЛІ ТА МЕТОДИ ОРГАНІЗАЦІЇ РОЗПОДІЛЕНИХ БАЗ ДАНИХ З ДИНАМІЧНОЮ СТРУКТУРОЮ

Васильченко І.В.<sup>1</sup> Мухін В.Є.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>vasylchenko.illia@gmail.com, <sup>2</sup>v.mukhin@kpi.ua [0000-0002-1206-9131]

**Розподілені бази даних є складною системою, яка має багато параметрів. Для ефективного керування процесами під час обробки інформації - необхідно оцінювати та прогнозувати витрати ресурсів на дані процеси, а для цього необхідно враховувати більшість цих параметрів. У запропонованій моделі враховується значна частина важливих параметрів. Створено модифікацію моделі для оцінки параметрів обробки інформації у розподілених базах даних з динамічною структурою, яка допоможе врахувати завантаженість системи користувачами. Дана модель допоможе якісніше оцінювати витрати ресурсів на обробку інформації, що надає можливість для прогнозування витрат як часових, так і обчислювальних.**

**Ключові слова: розподілена база даних, динамічна структура, модель оцінювання параметрів, обробка запитів.**

## 1. ВСТУП

У наш час бази даних відіграють ключову роль під час обробки великої кількості інформації. За їх допомогою інформацію можна зберігати, вилучати та обробляти, а також надавати різні рівні доступу певним користувачам для маніпуляції даними. Бізнеси, що мають хоча б якусь інформацію, наприклад, про продажі, клієнтів, товари тощо, зберігають її у різноманітних видах і формах: від паперових зошитів та блокнотів, до табличних процесорів та баз даних. Ті компанії, що бажають швидко та зручно отримувати інформацію про стан свого бізнесу, все частіше переходять до використання баз даних, чим пришвидшують обробку інформації та зменшують вірогідність помилки внаслідок втрати інформації, або людської помилки під час ручної маніпуляції з даними.

Через світовий прогрес інформація почала стрімко накопичуватися і тому почали використовувати розподілені бази даних, для швидкого та постійного доступу до даних.

Для прогнозування часу та ресурсів витраченого на роботу з даними, виникає потреба у прогнозуванні вищезазначених витрат на обробку інформації. Для цього створюються моделі, що допомагають оцінити витрати ресурсів, у тому числі й часу.

Окрім цього, навантаження системи у розподілених базах даних з динамічною структурою залежить від багатьох параметрів, наприклад, швидкість з'єднання, кількість вузлів, кількість активних користувачів, що працюють у системі тощо.

У даній статі запропоновано модель для оцінювання параметрів обробки інформації.

## 2. МОДЕЛЬ ОЦІНЮВАННЯ ПАРАМЕТРІВ ОБРОБКИ ІНФОРМАЦІЇ В РОЗПОДІЛЕНИХ СИСТЕМАХ З ДИНАМІЧНОЮ СТРУКТУРОЮ, В ЯКИХ РЕАЛІЗОВАНО ГЕТЕРОГЕННІ РОЗПОДІЛЕНІ БАЗИ ДАНИХ

Для формалізації обробки запитів в розподіленій системі з використанням баз даних потрібно розробити математичну модель, яка опише всі процеси, що проходять при обробці пакетів з запитом, наложити обмеження на систему та дозволить використати її в подальшій розробці моделей та методів.

Система обробки запитів (Query Processing System) в розподіленій системі з використанням гетерогенних розподілених баз даних задається функцією від наступної множини характеристик [1, 2]:

$$QPS(t, n) = F(SN, SC, SP, SPDB, SPCN),$$

де SN – множина вузлів (Set of Nodes) розподіленої системи обробки даних;

SC – множина зв'язків (Set of Connections) між вузлами розподіленої системи обробки даних;

SP – множина параметрів (Set of Parameters) розподіленої системи обробки даних;

SPDB – множина параметрів (Set Parameters Database) бази даних;

SPCN – множина параметрів вузлів керування (Set Parameters of Control Nodes) розподіленою системою обробки даних;

t – одиниця часу;

n – кількість вузлів системи в даний момент часу.

Множину вузлів розподіленої системи обробки даних позначимо як функцію від кількох параметрів та опишемо їх [1, 2]:

$$SN(t, n) = fN(scn, sdbn, sein, srn)$$

де scn – множина вузлів (Set of Control Nodes) в яких встановлено програмне забезпечення для керування розподіленою системою обробки даних;

sdbn – множина вузлів (Set of Database Nodes) в яких встановлено програмне забезпечення для гетерогенних розподілених баз даних;

sein – множина проміжних вузлів (Set of Intermediate Nodes), які виконують функцію маршрутизації пакетів;

srn – множина вузлів (Set of Requests Nodes), з яких надходять запити до розподіленої системи обробки даних.

Відома модель, в якій множина параметрів розподіленої системи обробки даних описана наступною функцією [1, 2]:

$$SP t, n = fp P, V, R, DB, Tdb, Tc$$

Нами запропонована модифікація цієї моделі:

$$SP t, n = fp P, V, R, DB, Tdb, Tc, U$$

де P – продуктивність вузла обробки даних в розподіленій системі обробки даних;

V – швидкість передачі даних по каналу зв'язку в розподіленій системі обробки даних;

R – надійність вузла розподіленої системи обробки даних;

DB – індикатор присутності в вузлі гетерогенної розподіленої бази даних;

Tdb – час обробки пакету – запиту в вузлі розподіленої системі обробки даних в якому знаходиться система управління базою даних;

Tc – час оновлення програмного забезпечення в вузлі керування розподіленої системи обробки даних.

U – навантаження системи користувачами, тобто яка частина користувачів в один момент працює в розподіленій системі баз даних.

Введемо обмеження на параметри розподіленої системи обробки даних, а саме на продуктивність вузла [1]:

$$P_{min} \leq P$$

де  $P_{min}$  – залежить від типу обладнання на якому знаходиться вузол та його характеристик, але не повинна бути меншою за 0.5 терафлопси.

Опишемо початкові умови обмеження швидкості передачі даних по каналу зв'язку в розподіленій системі обробки даних [1].

$$V_{min} \leq V,$$

де  $V_{min}$  – визначатиметься від типу вузлів, які під'єднанні до розподіленої системи, але не менше ніж 10Мбіт/с. В зв'язку з тим, швидкість передачі впливає на час доставки пакетів між вузлами, то вона є ключовим параметром, як провідних розподілених системах обробки даних так і в безпроводних системах [1].

Введемо обмеження на параметри розподіленої системи обробки даних, а саме на надійність вузла [1]:

$$R_{min} \leq R,$$

де  $R_{min}$  – визначається як безвідмовність вузла системи, яка визначається, як вірогідність безвідмовної роботи системи і залежить від часу  $t$ . Відповідно безвідмовність всієї системи обчислюється [1]:

$$R(n) = \prod_{i=1}^n e^{-\lambda_i t}$$

де  $\lambda_i$  – інтенсивність виходу з ладу  $i$ -го елемента системи. Згідно з середніми статистичними даними, інтенсивність виходу з ладу елементів системи знаходяться в межах інтервалу від  $10^{-3}$  до  $5 * 10^{-3}$ . Тоді  $R_{min}$  залежить від кількості вузлів, але не менше ніж 0,5 [1].

Прийmemo, що признак присутності в вузлі гетерогенної розподіленої бази даних буде приймати значення [1]:

$$DB = \frac{0 - NoDB}{1 - YesDB}$$

де 0 – приймається у випадку, коли в вузлів відсутня система управління базою даних;

1 – приймається у випадку, коли в вузлі встановлене програмне забезпечення баз даних [1].

Введемо обмеження на час обробки пакету — запиту в вузлі розподіленої системи обробки даних в якому розміщена система управління базою даних, а саме [1]:

$$T_{dbmin} \leq T_{db} \leq T_{dbmax}$$

де  $T_{dbmax}$  – визначається як максимальний час затримки обробки пакетів в вузлі бази даних.

А  $T_{dbmin}$  – повинно бути як найменшим і прямувати до нуля, але значення нуля ніколи не приймати [1, 3].

Опишемо початкові умови оновлення програмного забезпечення в вузлі керування розподіленої системи обробки даних, а саме час затримки обробки пакетів [1, 3]:

$$0 \leq T_c \leq T_{cmax}$$

де  $T_{cmax}$  – визначається як подвоєна сума максимальних часів затримки пакетів оновлення в вузлах на шляху до вузла керування та максимальний час оновлення програмного забезпечення (планувальник та брокер) в вузлі керування [1, 3].

Введемо обмеження на навантаження системи користувачами. Оскільки це відносне значення, що відображає, яка частина користувачів в один момент працює в розподіленій системі баз даних, то обмеження є наступними:

$$0 \leq U \leq 1,$$

де  $U = au/srn$ ,

де  $au$  – кількість активних користувачів (Active Users) - кількість активних користувачів, які працюють у системі;



srn – множина вузлів (Set of Requests Nodes), з яких надходять запити до розподіленої системи обробки даних.

### 3. РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ЗНАХОДЖЕННЯ ЧАСУ ОБРОБКИ ЗАПИТІВ В РОЗПОДІЛЕНІЙ СИСТЕМІ ОБРОБКИ ДАНИХ

Застосуємо регресійну модель для знаходження часу обробки запитів в розподіленій системі обробки даних в залежності від параметрів розподіленої системи. Для цього візьмемо дані з джерела [1], а також додамо параметр  $U$  та 5 розподілену систему. Дані наведені у таблиці 1.

Таблиця 1. Час обробки запитів в розподіленій системі обробки даних в залежності від параметрів

<b>P</b>	<b>V</b>	<b>T<sub>db</sub></b>	<b>T<sub>c</sub></b>	<b>U</b>	<b>T</b>
0,11	5	5	3	0,2	8
0,9	4	4	5	0,05	12
0,12	4	8	6	0,2	21
0,8	2	4	6	0,2	14
0,7	3	6	5	0,25	20

Тобто нам потрібно вирішити наступну систему рівнянь, яка нам дасть можливість знайти коефіцієнти:

$$\begin{aligned} 0,11c_1 + 5c_2 + 5c_3 + 3c_4 + 0,2c_5 &= 8, \\ 0,9c_1 + 4c_2 + 4c_3 + 5c_4 + 0,05c_5 &= 12, \\ 0,12c_1 + 4c_2 + 8c_3 + 6c_4 + 0,2c_5 &= 21, \\ 0,8c_1 + 2c_2 + 4c_3 + 6c_4 + 0,2c_5 &= 14, \\ 0,7c_1 + 3c_2 + 6c_3 + 5c_4 + 0,25c_5 &= 20 \end{aligned}$$

Вирішивши дану систему рівнянь ми знаходимо єдине правильне рішення в вигляді:

$$\begin{aligned} c_1 &= 8,5052, \\ c_2 &= -2,2216, \\ c_3 &= 4,3067, \\ c_4 &= -0,7423, \\ c_5 &= -5,6701. \end{aligned}$$

Звідси вираз для аналізу і прогнозування часу обробки пакетів в розподіленій системі обробки даних в залежності від параметрів системи має вигляд:

$$T = 8,5052P - 2,2216V + 4,3067T_{db} - 0,7423T_c - 5,6701U$$

Введення додаткового параметру  $U$  дозволяє врахувати навантаження користувачами розподіленої системи, чим допомагає більш точно оцінити витрати ресурсів.

### 4. ВИСНОВКИ

Розподілені бази даних з динамічною структурою – це ефективний, але досить складний інструмент для обробки інформації. Для прогнозування витрат ресурсів важливо враховувати параметри системи. Тому для оцінки були застосовані такі параметри як продуктивність, швидкість, час обробки пакету, час оновлення програмного забезпечення, а також завантаженість системи користувачами [4, 5].

У результаті створено модифікацію моделі для оцінки параметрів обробки інформації у розподілених базах даних з динамічною структурою, яка допоможе врахувати завантаженість системи користувачами. Дана модель допоможе якісніше оцінювати витрати ресурсів на обробку інформації, що надає можливість для прогнозування витрат як часових, так і обчислювальних.

## **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. КОРНАГА Я. І. МОДЕЛІ ТА МЕТОДИ ОРГАНІЗАЦІЇ ТА УПРАВЛІННЯ ГЕТЕРОГЕННИМИ РОЗПОДІЛЕНИМИ БАЗАМИ ДАНИХ З ДИНАМІЧНОЮ СТРУКТУРОЮ НА ОСНОВІ МЕРЕЖЕЦЕНТРИЧНОГО ПІДХОДУ : дис. докт. техн. наук : 05.13.06 / КОРНАГА ЯРОСЛАВ ІГОРОВИЧ – Київ, 2020. – 328 с.

2. V. Mukhin, Y. Kornaga, V. Bondarenko, V. Zavgornii, O. Herasymenko and O. Sholokhov, "Mathematical Model for Heterogeneous Databases Parameters Estimation in Distributed Systems with Dynamic Structure," 2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (ATIT), 2020, pp. 158-161, doi: 10.1109/ATIT50783.2020.9349331.

3. Е. А. Mikrin, S. K. Somov, "The optimal online information backup in the data processing systems based on computer networks redundancy", Probl. Upr., 2016, no. 5, 47–56.

4. Ярцев В.П. Розподілені бази даних: навчальний посібник. - К. ДУТ 2018. - 97с.

5. Дейт, К. Дж. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом "Вильямс", 2005. — 1328 с.: ил. — Парал. тит. Англ.

# ОЦІНЮВАННЯ РИНКОВИХ РИЗИКІВ ТА ПОРІВНЯННЯ РІЗНИХ МЕТОДІВ

Воловоденко Т.О.<sup>1</sup>, Тимошук О.Л.<sup>2</sup>, Бідюк П.І.

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>ta.volovodenko@gmail.com,

<sup>2</sup>o.tymoshchuk@kpi.ua [0000-0003-1863-3095]

**Ринкові ризики — основний компонент фінансових ризиків, від прогнозування якого має суттєву вагу у управлінні фінансовою діяльністю. Більшість установ використовують традиційні та застарілі методи оцінювання ризиків. Метою роботи є оцінювання ринкових ризиків різними методами та порівняння їх з традиційними. Результатом дослідження є СППР, що виконує оцінку ринкового ризику фінансового портфелю із застосуванням різних підходів та методів. У роботі було використано теоретичні та емпіричні методи дослідження.**

**Ключові слова: фінансові ризики, VAR, CVAR, прогнозування волатильності, ринковий ризик.**

## 1. ВСТУП

Ринкові ризики — розповсюджена тема серед аналітиків. Будь-яка фінансова установа, що оперує з фінансовими ринками, має справу в тому числі з ринковими ризиками. Ризик і прибуток дуже пов'язані між собою. Зазвичай, більш високий ризик приносить потенціал для більшої високої віддачі, і навпаки. Але це не аксіома. Ви не можете очікувати однакового результату від активів з низьким ризиком (наприклад, грошові кошти) та активів з більш високим ризиком (наприклад, акції).

Методи оцінювання ринкових ризиків стрімко розвиваються. Більш сучасне та більш точне визначення ризику дозволяє запобігти несподіваних та зайвих витрат.

## 2. МЕТОДИ ОЦІНЮВАННЯ РИНКОВИХ РИЗИКІВ

Одним з найбільш популярних підходів до оцінювання величини можливих втрат є методологія Value at Risk (VaR) — інструмент вимірювання ризику, заснований на розподілі збитків.

«Вартість під ризиком» (VaR) є важливою мірою ризику в фінансах. Вона спрямована на те, щоб знайти реально найгірший результат в тому сенсі, що щось гірше відбувається із заданою (малою) ймовірністю [1]. VaR — найгірша втрата за горизонт, яка не буде перевищена при даному рівні достовірності [2]. За домовленістю є позитивним числом.

Вартість під ризиком (VaR) — метод, який використовується для оцінки фінансового ризику шляхом оцінки потенційно можливої несприятливої зміни в ринковій вартості портфелю із заданим рівнем довірчої ймовірності за певний період часу. Деякі організації використовують рівень довірчої ймовірності рівний 99%, а горизонт втрат в 10 робочих днів.

Value at Risk має такі переваги:

- об'єднання ризиків окремих позицій в єдину величину для всього портфелю;
- оцінювання ризику можливих втрат, співвіднесених з імовірностями їх появи;
- вимірювання ризику на різних ринках універсальним чином.

До переваг VaR можна віднести наочність і простоту обчислень, можливість порівняльного аналізу втрат і ризиків, які їм відповідають, а також консолідацію інформації. Однак у методології VaR існують і недоліки, до яких відносяться слабкі і сильні припущення про властивості фінансових ринків, поведінку економічних агентів на цих ринках, про чутливості портфеля і інші. Також при оцінюванні VaR майже не враховується ліквідність, а це важлива характеристика всіх ринків. Це може привести до того, що в певні моменти зміна структури портфеля для зменшення ризику може виявитися марною.

Коли на ринках відбуваються швидкі або різкі зміни методологія Value at Risk перестає адекватно відображати величину ризику, оскільки вона може бути застосована на стабільних ринках. При істотній зміні ринкових умов, наприклад, різко зміниться ліквідність ринку або стрибкоподібно зміниться ціна, то VaR зможе врахувати ці зміни тільки через певний проміжок часу, накопичивши необхідну статистику даних і подій. Крім того, методологія VaR не дає абсолютної оцінки можливих втрат, як, втім, і більшість відомих методологій і методик. Однак вона є дійсно універсальним підходом до оцінки ринкових ризиків.

VaR — це виражена в грошових одиницях базової валюти оцінка величини, яку не перевищать очікувані протягом даного періоду часу втрати з заданою ймовірністю. (рівень довіри). Базою для оцінювання VaR виступає динаміка курсів і цін інструментів за встановлений період часу в минулому. Часовий період зазвичай обирають виходячи з терміну перебування фінансового інструменту в портфелі або його ліквідності, виходячи з мінімального реального терміну, протягом якого можна реалізувати на ринку цей інструмент без істотного збитку. Часовий період вимірюють як число робочих або торгових днів.

Зазвичай, значення VaR розраховують трьома методами: параметричним; методом історичного моделювання; методом Монте-Карло. Загалом, VaR відображає мінімальний збиток інвестиції, який буде понесений на певному рівні ймовірності (довірчому інтервалі) або максимальний збиток, який буде реалізований на рівні ймовірності. Однак, чим менш стабільна інвестиція, тим більша ймовірність того, що VaR не дасть повної картини ризиків, оскільки ця міра байдужа до всього, що виходить за межі її власного порогу. Саме тому для більш адекватної оцінки екстремальних подій також розраховують показник очікуваних втрат Expected Shortfall, найбільш відомий як CVaR або Conditional Value at Risk (Рис.1).

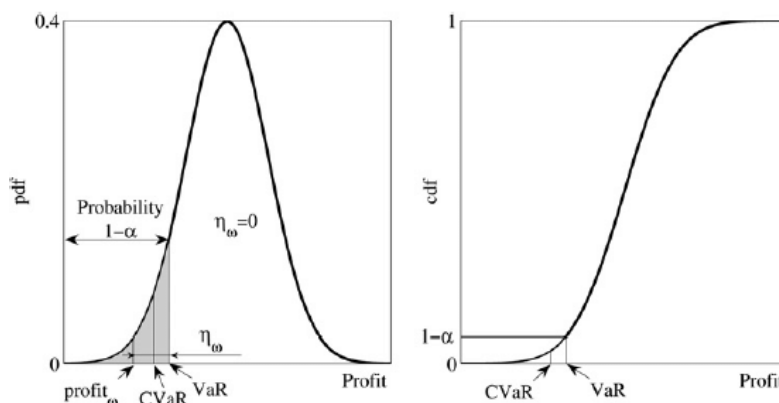


Рисунок 1. Візуалізація VaR та CVaR на графіку розподілу портфелю

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для дослідження було сформовано інвестиційний портфель з цін акцій компаній Apple, Google, Microsoft та Amazon. У якості часового горизонту було взято період у 7 років (1764 дні). Початкова інвестиція в кожен з акцій була обрана у розмірі 1000 \$, а в якості періоду для прогнозування обрано один торговельний рік – 252 дні.

Для обчислення VaR необхідно розглянути розподіл доходностей складових портфелю (Рис. 2).

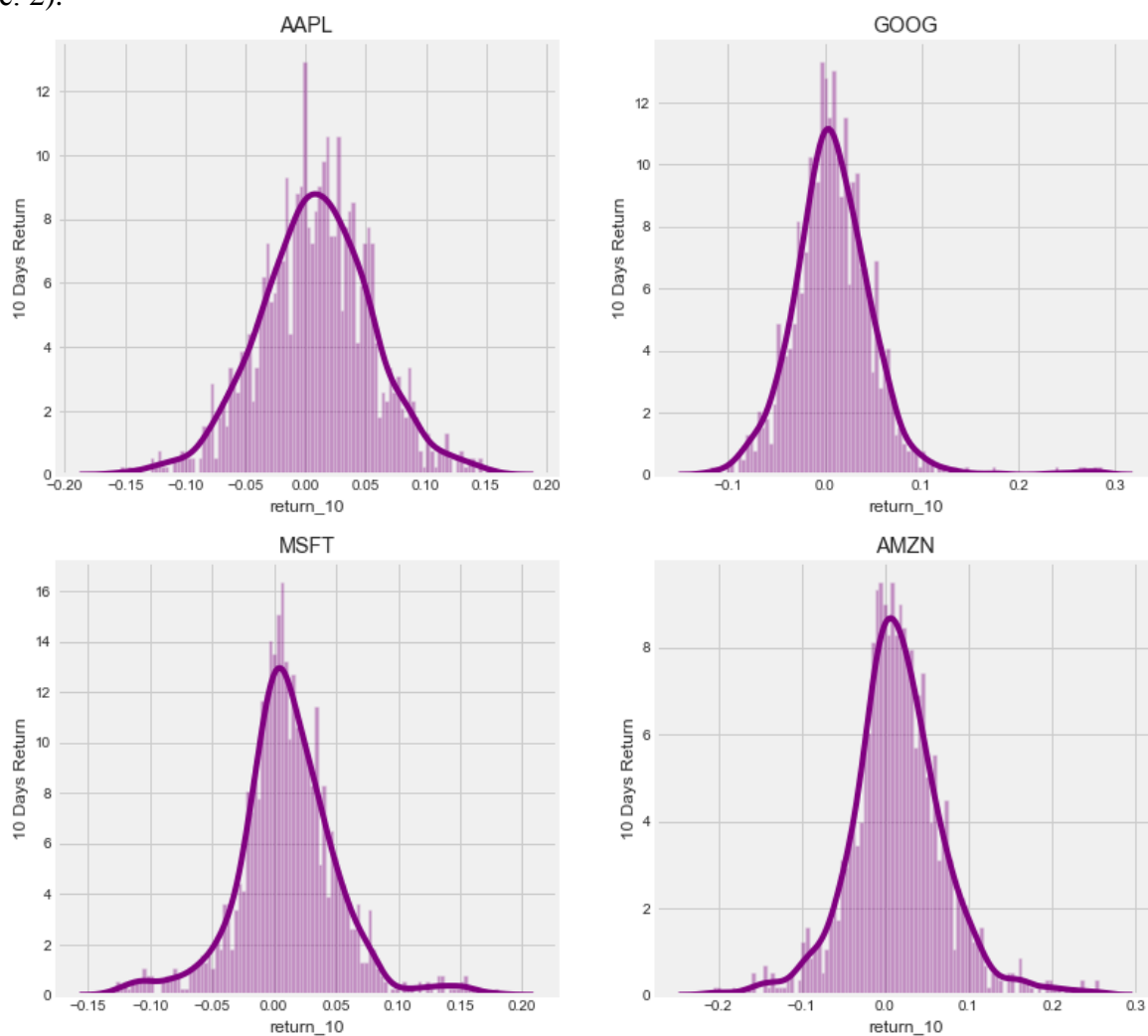


Рисунок 2. Розподіл доходу за 10 днів

Для розрахунку VaR припущення полягає в тому, що доходи розподілені нормально. Перший погляд на прибутковість показує, що щоденні доходи для портфелю розподіляються приблизно нормально, але хвости більш товсті. Це натякає на доцільність використання поряд із нормальним розподілом розподілу Стьюдента під час реалізації параметричного методу оцінки VaR. Використання розподілу Стьюдента-t для оцінки VaR може значно покращити оцінку VaR [7].

Найпростішим методом розрахунку VaR є історичний метод, для якого припущення про розподіл доходів не робиться, а моделюється на основі історичних емпіричних даних.

Для параметричного методу робиться припущення щодо розподілу доходів і використовуються історичні доходи портфелю та стандартні відхилення (оцінки) для визначення параметрів моделі.

Результати обчислень VaR та CVaR на довірчому інтервалі 99% наведені у таблиці нижче (Табл. 1).

Таблиця 1. Значення VaR та CVaR для різних рівнів довіри та різних методів обчислення

Акція	Історичний VaR	Історичний CVaR	Параметричний VaR N(0,1)
AAPL	42.205269	56.004843	35.550474
GOOG	36.072944	45.365946	32.723821
MSFT	38.523547	58.363429	35.370255
AMZN	53.040366	77.967922	45.032854

Зазначимо, що величини VaR та CVaR, оцінені за параметричним методом менші за значення, отримані історичним методом. Історичний метод припускає, що майбутні зміни у курсі акцій будуть подібні до минулих, у той час як параметричний метод припускає певний вид розподілу. Загалом, VaR надає хороше уявлення про втрати, які відбуваються менше заданого довірчого інтервалу, але не про ті, що лежать за його межею.

#### 4. ВИСНОВКИ

В усіх випадках для прогнозування VaR використовувались певні апроксимації, зокрема щодо розподілу вибірки. Ці апроксимації показують непоганий результат на низьковолатильних активах, такі як акції топ-компаній або біржеві індекси. Однак для будь-яких інших часових рядів такий підхід не дасть належних результатів, або може давати хибні оцінки для прогнозованих даних. В наступних дослідженнях планується розглянути динамічний підхід до оцінки розподілу та його параметрів, зокрема перевірка стаціонарності ряду та прогнозування його волатильності гетероскедастичними моделями. Для кращого порівняння вищезазначені методи будуть використані на більш чутливих даних.

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кузнецова Н.В., Бідюк П.І. Теорія і практика аналізу фінансових ризиків: системний підхід, монографія. Київ 2020, 400 с.
2. Куцик П.О., Васильців Т.Г., Сороківський В.М., Стефаняк В.І., Сокорівська М.В. . Управління фінансовими ризиками навч. посібник. Львів: Растр-7, 2016. 318 с.
3. Башкіров О.В. Порівняльний аналіз VAR-методів оцінки ризику активів банку О.В. Башкіров, Проблеми і перспективи розвитку банківської системи України : зб. наук. праць ДВНЗ «УАБС НБУ». – Вип. 14. –С. 302–309.
4. Moraux F. How valuable is your VaR? Large sample confidence intervals for normal VaR. F. Moraux. Journal of risk management in financial institutions. – 2011. – № 4.2. – P. 189–200. – [Електронний ресурс]. – Режим доступу : <https://perso.univ-rennes1.fr/franck.moraux/research/JRMFI.pdf>
5. Longerstaey J. Risk Metrics TM – Technical Document . J. Longerstaey, M. Spencer. Morgan Guaranty Trust Company of New York: NewYork, 1996. [Електронний ресурс]. – Режим доступу : <http://yats.free.fr/papers/td4e.pdf>
6. Theodore Barnhill and William Maxwell. Modeling correlated market and credit risk in fixed income portfolios. Journal of Banking and Finance, 26:347–374, 2002
7. Lin, C. and Shen, S. (2006), "Can the student- t distribution provide accurate value at risk?", Journal of Risk Finance, Vol. 7 No. 3, pp. 292-300. <https://doi.org/10.1108/15265940610664960>

# СИСТЕМНИЙ АНАЛІЗ ІНДЕКСУ АКЦІЙ З ВИКОРИСТАННЯМ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

Данилов В.Я.<sup>1</sup>, Деменкова В.В.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>2</sup>lera145dem@gmail.com

**Необхідною складовою будь-якої інвестиційної діяльності є прогнозування фінансових часових рядів. Кожен інвестор, вкладаючи кошти, намагається максимізувати прибуток та мінімізувати будь-які ризики для отримання невдалої спроби. Аби уникнути втрати значної кількості коштів необхідно приділити певний час для ретельного розрахунку та ймовірного прогнозування. Метою даної роботи є проведення комплексного огляду основних архітектур штучних нейронних мереж для аналізу та прогнозування індексу акцій. Результатом розробленого продукту є визначення найкращої архітектури з трьох обраних нейронних мереж та досліджень про їх покращень. Робота включає в себе використання як теоретичних так і емпіричних методів дослідження.**

**Ключові слова:** індекс акцій, фондова біржа, ринок цінних паперів, багат шарові нейронні мережі, згорткові нейронні мережі, нейронні мережі довгої короткочасної пам'яті, метрики, точність.

## 1. ВСТУП

Відомості про зміну біржового котирування мають значний вплив на економічне проектування. Саме інформація про стан попиту та пропозицій цінних паперів, передусім таких топ компаній як Apple, Alphabet, Microsoft, Amazon, Facebook, Coca-cola і т.д, слугують показниками загального стану світової економіки та поточних тенденцій їх розвитку. Прогнозування майбутніх змін зазвичай використовують саме для цих змінних та розраховані закономірності на їх даних. Найбільш значне значення при дослідженні ринку цінних паперів є визначення та оцінка динаміки біржового котирування, яка полягає у підрахунку біржових індексів фондовими біржами світу. На сьогоднішній день саме біржові індекси є обов'язковою складовою фондових ринків. З «розмаїттям» даних в базі, на основі яких проводиться цей підрахунок, можна отримати прогнозні значення різноманітних індексів.

Метою цієї роботи є проведення комплексного огляду і аналізу різних архітектур штучних нейронних мереж на даних однієї з найпопулярніших компаній, для подальшого вибору найкращої моделі та подальших досліджень про її вдосконалення.

## 2. СУЧАСНІ МЕТОДИ ПРОГНОЗУВАННЯ ІНДЕКСУ АКЦІЙ

Сучасна фінансова та економічна система не має сенсу без постійного використання індексу акцій. На сьогоднішній день, досить вагома кількість індексів акцій вимірюється біржами, міжнародними інвестиційними банками, інформаційними представництвами.

Сьогоднішня література описує різні методи прогнозування фондового ринку, які можуть бути використані інвесторами для їх досліджень. Основними такими методами є: метод експертних оцінок, економіко-математичні методи, статистичні методи, технічний аналіз, фундаментальний аналіз.

Експертні методи включають в себе процес оцінки проблеми на основі думки експертів в обраній галузі з ціллю подальшого прийняття рішення в умовах невизначеності. При прийнятті рішення недостатньо кваліфікованою особою, для розв'язання проблеми доцільно буде звернутися до компетентного експерта. Для ефективного застосування експертного методу використовують дані, отримані за допомогою консолідації думок групи експертів [1].

Метод експертів, так званий наприклад метод Делфі, зазвичай розділяють ще на індивідуальні колективні та експертні оцінки. Індивідуальні експертні оцінки спираються на думку лише одного спеціаліста, коли колективну експертну оцінку необхідно аналізувати з використанням математичної статистики [1].

Економіко-математичні методи можна розподілити на підгрупи: методи кореляційно-регресійного аналізу, методи математичного програмування та матричні методи та моделі [1].

Зазвичай методи кореляційно-регресійного аналізу використовують в процесі формування репрезентативної статистичної вибірки, що допоможе усунути взаємозалежні зв'язки, вплинувши на розмірність матриці із статичними даними. Як результат це надасть можливість застосувати адекватну модель без перевантажень вхідними даними.

Методи математичного програмування зазвичай застосовують для розв'язання проблеми таких задач як планування номенклатури, задачі Комівояжера, мінімізації залишків виробництва, тощо. Основною ідеєю цього методу є побудова відповідної математичної моделі, що являє собою штучну систему, яка спростовує відображення структури та основних закономірностей розвитку реального об'єкту, тим самим надаючи інформацію про стан та поведінку об'єкта, який досліджується (на основі вимірів).

Матричні методи та моделі економічного аналізу надають перевагу в розв'язанні складних і великорозмірних економічних структур [2]. В економіці прийнято використовувати модель міжгалузевого балансу в процесі підрахунку технологічних нормативів, управління ресурсами та запасами сировини.

В основі статистичного методу лежить гіпотеза випадкових блукань. Іншими словами, ціна акцій змінюється випадковим чином. Головним в цьому методі є те, що динаміка кожної наступної зміни не залежить від попередньої, і залежить лише від нових відомостей. Таким чином зміни індексу акцій формують послідовність з незалежних і однаково розподілених випадкових величин [3]. Найпоширенішим та найбільш дослідженим є нормальний розподіл.

Технічний аналіз дозволяє прогнозувати поведінку ринку базуючись на попередній динаміці цін та даних про об'єми торгів. Таким чином сенс технічного аналізу полягає в вивченні поточних та попередніх цін активу, при невідповідних коливаннях цих цін та отриманні ідентифікованих трендів. Тому саме цей аналіз використовують - для фондових індексів.

Передумовою технічного аналізу є те, що ціни на фондовому ринку відображають всі фундаментальні чинники, що мають вплив на визначений актив. Тому фундаментальний аналіз більш розрахований на використання стратегії дослідження з акцентом на якісні фактори. В цьому аналізі головною ідеєю ж те, що наступна ефективність активу буде залежати не тільки від історичних даних, а й від різних мікро та макроекономічних умов.

Моделювання економічних та фінансових процесів може бути реалізовано використовуючи як традиційні математичні методи так і сучасні. Крім того, останні здатні розв'язати задачі, які традиційним методам не вдається розв'язати. Доцільно буде



використати нейронні мережі для розв'язання погано формалізованих завдань з вагомим обсягом обчислень таких як прогнозування обсягів продаж, чи для задачі класифікації об'єктів економічного аналізу таких як класифікація клієнтів для надання кредиту.

З розвитком технологій на сьогоднішній день є досить широкий спектр різних архітектур нейронних мереж. Популярність в використанні нейронних мереж пояснюється тим, що цей інструмент має досить просту схему, яка містить в собі багато обчислювальних нейронів, що мають повні зв'язки між собою, а також більш складніші топології, в яких нейрони групуються в певні групи (шари), де виходи нейронів одного шару нейронів взаємозв'язані з входами іншого шару нейронів прямим та зворотними зв'язками (багатошарові персиптрони). Після визначення кількості шарів і числа елементів в кожному, відбувається процес навчання мережі з визначенням ваг мережі.

Штучні нейронні мережі прийнято відносити до технічного аналізу. Це обумовлюється тим, що вони здатні на пошук та знаходження відмінних закономірностей в поведінці часового ряду, в основі якої лежать історичні дані [3].

Сьогодні створено досить широкий спектр інших архітектур нейронних мереж. В даній статті будуть розглянуті одні з найпоширеніших структур, такі як багатошарові нейронні мережі та мережі довгої короткочасної пам'яті. Результати аналізу допоможуть побудувати найкращу нейронну мережу для прогнозування саме індексу акцій з найпоширеніших мереж.

На прикладі багатошарової нейронної мережі (Рис.1) можна побачити, що структура даної мережі полягає у тому, що кожний елемент попереднього шару має певний зв'язок з кожним елементом наступного шару. Завдяки цьому, процес навчання проводить сигнали через кожен обчислювальний елемент до вихідного шару, і далі порівнюється з заданим значенням (метод зворотнього поширення похибки) [4].

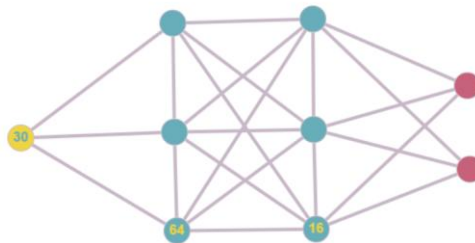


Рисунок 1. Схема багатошарової нейронної мережі

Згорткові нейронні мережі (ТСN) теж може бути потужним інструментом у розв'язанні задач при прогнозуванні індексу акцій. Першим кроком необхідно привести вхідний часовий ряд до відповідної матриці вхідних даних. Тепер наступним кроком буде використання ідеології згорткових нейронних мереж [5].

Крім того, в роботі розглядається ще один тип з найпопулярніших рекурентних нейронних мереж - мережа довгої короткочасної пам'яті [6]. Вона значно відрізняється від інших тим, що вона складається так звані LSTM-модулі, що здатні запам'ятовувати певні значення як на короткі, так і на довгі проміжки часу.

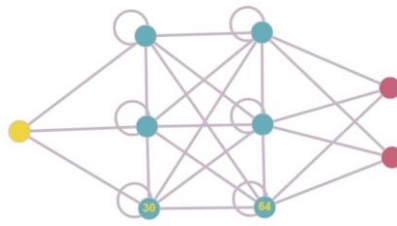


Рисунок 2. Схема мережі довгої короткочасної пам'яті

LSTM-модулі також можуть групуватися в певні блоки, що схоже на приклад глибокої багатошарової нейронної мережі, однак ці блоки містять зазвичай по три чи чотири вентиля, що контролюють потоки вхідних даних і вихідні збережені дані блоків [6]. Реалізація цих вентилів здійснена за допомогою використання логістичної функції для обчислення значень в діапазоні  $[0;1]$ . Ваги цієї нейронної мережі налаштовані на задання величин і напрямку оперування цих вентилів. Але для цієї нейронної мережі характерний один недолік, що може вплинути на тривалість навчання - це наявність вагової кількості обчислювальних модулів.

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Дослідження індексу акцій проведено на даних однієї з топ компаній, що у 2020 році визнали найдорожчим брендом світу за версією Forbes – Apple. В якості вхідних даних використано набір даних з цінами акцій за 10 років (з 2010 року до 2020 року), що включає інформацію про ціну акції на початку та на кінець календарного дня. Також набір даних відображає відомість про найвищу ціну впродовж дня та найнижчу ціну впродовж дня, значення кількості акцій, які були продані в відповідних день. Ціна акцій у наборі даних наведена у доларах США (Рис. 4).

Date	Close/Las	Volume	Open	High	Low
2/28/2020	\$273.36	1.07E+08	\$257.26	\$278.41	\$256.37
2/27/2020	\$273.52	80151380	\$281.10	\$286	\$272.96
2/26/2020	\$292.65	49678430	\$286.53	\$297.88	\$286.50
2/25/2020	\$288.08	57668360	\$300.95	\$302.53	\$286.13
2/24/2020	\$298.18	55548830	\$297.26	\$304.18	\$289.23
2/21/2020	\$313.05	32426420	\$318.62	\$320.45	\$310.50
2/20/2020	\$320.30	25141490	\$322.63	\$324.65	\$318.21
2/19/2020	\$323.62	23495990	\$320	\$324.57	\$320
2/18/2020	\$319	38190550	\$315.36	\$319.75	\$314.61
2/14/2020	\$324.95	20028450	\$324.74	\$325.98	\$322.85

Рисунок 3. Приклад використаного набору даних

Першим кроком в задачах аналізу та прогнозування індексу акцій є побудова графіку даних про зміну ціни у кінці календарного дня для кожного класу нейронних мереж (Рис. 4).

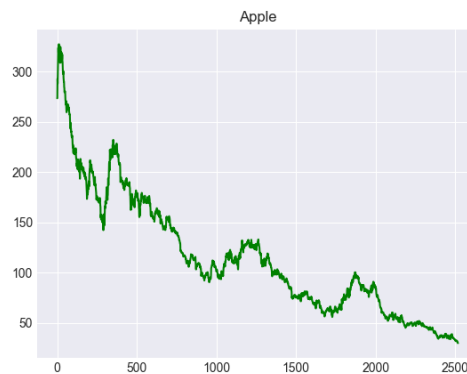


Рисунок 4. Динаміка ціни акцій компанії Apple

В процесі навчання використані проміжки у 30 днів в момент закриття ринку для прогнозування ціни на наступних день. Обраний набір даних розділено на навчальну та тестову дату у співвідношенні 80 до 20 відповідно. У результаті навчання кожною мережею отримано такі пари  $X$  та  $Y$  (залежності), які характеризують ціну під час закриття біржі за 30 днів та  $[0;1]$  або  $[1;0]$  в залежності від динаміки зросту чи спадання ціни.

Тепер для реалізації прогнозування за допомогою багатoshарової нейронної мережі спершу необхідно обрати модель. Для даної статті використано багатoshаровий персептрон, а саме тришарову нейронну мережу (достатньо одного прихованого шару). Вхідний шар складається з 30 нейронів, прихований шар з 64 нейронів. Після вхідного та прихованого шару застосовано *BatchNormalization* як метод регуляризації, аби нормалізувати значення у кожному з наборі даних та мінімізувати ризик перенавчання нейронної мережі. Активаційною функцією використано *LeakyReLU*, що являє собою *ReLU* з витоком та допоможе зменшити нестабільність і проблему виходу з ладу *ReLU* (Рис. 5). На останньому вихідному шарі використаємо функцію активації *Softmax*.

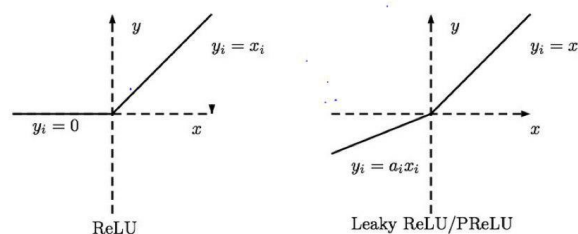


Рисунок 5. Зображення *ReLU* та *LeakyReLU*

Для регуляризації використано *Dropout* для запобігання перенавчання мережі. Крім того, під час навчання змінюються кількість нейронів прихованих шарів, завдяки яким отримано кілька базових, для даної задачі, структур персиптронів (на виході два персиптрони). Завдяки цьому покращено ефективність навчання і чіткість результату, оскільки менш навчені нейрони будуть мати меншу вагу. Для функції помилки та алгоритму оптимізації варто використати один з найпопулярніших оптимізаторів – *Adam* з початковим кроком 0.002. В задачі класифікації використано категоріальну кроссентропію. Кількість епох для даної задачі змінювались від 100 до 150.

Для реалізації задачі прогнозування індексу акцій з використання згорткової нейронної мережі необхідно буде також додати два гіперпараметри, такі як кількість фільтрів та розмір фільтра. Кількість має певну залежність від розміру обраних вхідних даних, але в даному

випадку обидва параметри будуть обрані випадково. Для функції активації варто застосувати LeakyReLU. Як функцію втрат обрано бінарну кроссентропію. Отримано і проаналізовано набір згорткових мереж для даної задачі.

Для реалізації прогнозування за допомогою методології довгої короткочасної пам'яті побудовано набір з LSTM-модулями рекурентних штучних нейронних мереж. Ці нейронні мережі дозволяють отримати «моделі» завдяки довгостроковим залежностям, що притаманні для обраної задачі. Всі параметри оцінюються аналогічно попереднім архітектурам.

Наведемо деякі результати моделювання динаміки (Рис. 6, 7).

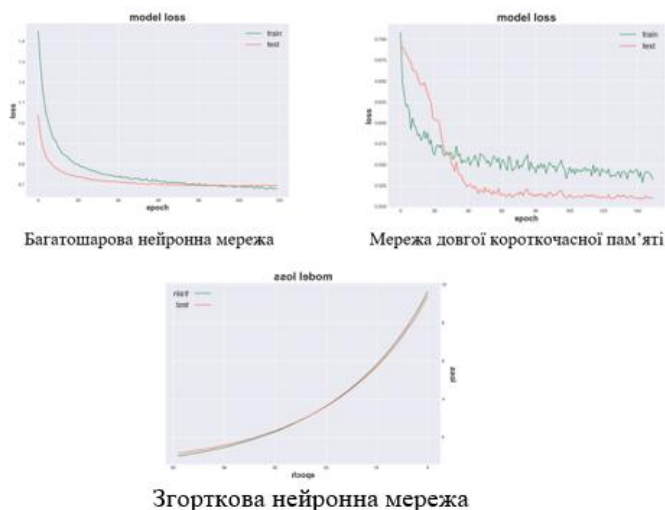


Рисунок 6. Динаміка значення похибки моделі для навчальної та тестової вибірки

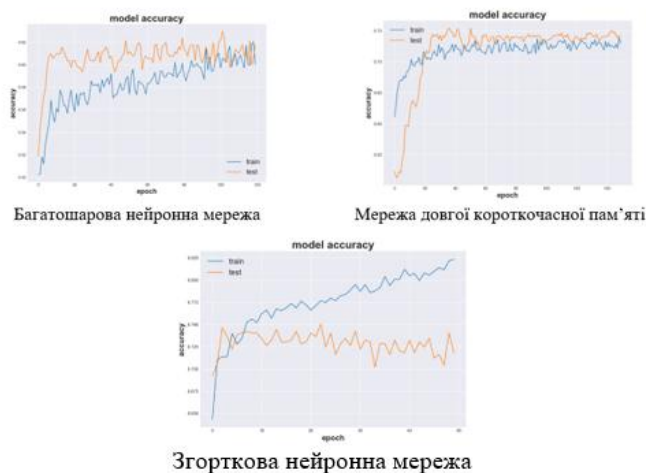


Рисунок 7. Динаміка значення точності моделі для навчальної та тестової вибірки

Для якісної оцінки в даній роботі також використано метрики precision, recall та f1-score. За допомогою Precision виявлено наскільки отримані позитивні прогнози правдиві за формулою (1) [7]:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

Recall — це метрика, яка показує яку частку з дійсно позитивних значень наша мережа передбачила правильно, та обраховується за наступною формулою (2) [7]:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

Для отримання балансу між precision та recall використано F1-score (3) [7]:

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision} \quad (3)$$

Результат відносять до доброго результату, якщо всі метрики наближені до 1.

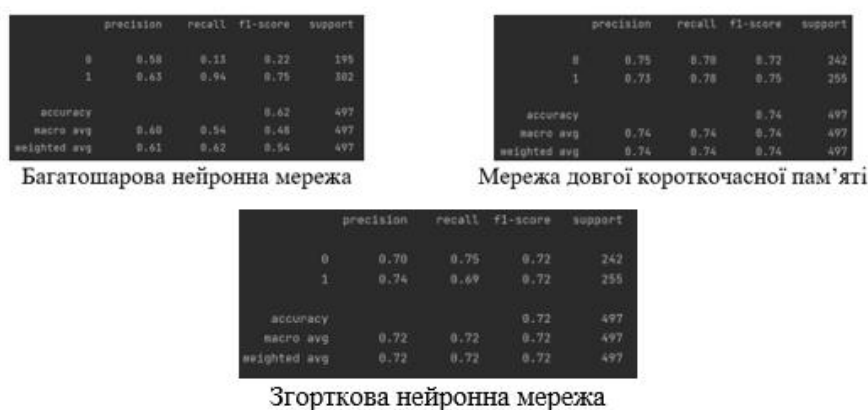


Рисунок 8. Результати метрик по використаним нейронним мережам

Звідси можна зробити висновок, що присутня незбалансованість у передбаченнях, оскільки метрика recall більша для одного з випадків для багатшарової нейронної мережі.



Рисунок 9. Матриця неточностей для всіх трьох нейронних мереж

За отриманими результатами (Рис. 8, 9) можна побачити, що найефективнішою нейронною мережею серед трьох представлених є згорткова нейронна мережа. Вона показала кращі результати у метриках та найшвидше «по часу» провела процес навчання. Гірші результати отримано багатшарова нейронна мережа.

#### 4. ВИСНОВКИ

У даній статті було зроблено комплексний огляд трьох класів нейронних мереж: багатшарової нейронної мережі, згорткової нейронної мережі та мережі довгої короткочасної пам'яті. При проведенні аналізу було використано дані з інформацією про індекси акцій на основі набору даних компанії Apple за останні 10 років.

Отримані результати свідчать про доцільність використання штучних нейронних мереж для аналізу цін акцій. У підсумку найефективнішою виявилась згортова нейронна мережа. Вона надала найкращі результати поміж інших, та мала найшвидші результати з навчання нейронних мереж. Друге місце займають мережі довгої короткочасної пам'яті. Вони також мають кращі метрики і результати. Багатошарова мережа продемонструвала гірші результати, що свідчить про те, що використання цієї архітектури непридатне (для заданої кількості прихованих шарів і кількості нейронів) для прогнозування індексу акцій для даного набору даних.

На нашу думку, результати могли б бути точнішими якби ми врахували фрактальні розмірності множини даних (Фрактальна розмірність наших даних дорівнює 1,06).

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мних Є. В. Економічний аналіз: підруч. / Київ: Знання, 2011. 630 с.
2. Бачевський Б. Є. Потенціал і розвиток підприємства: навч. посіб. / Київ: Центр учбової літератури, 2009. 400 с.
3. Мандельброт Б., Хадсон Р. Л. (Не)послушне рынки. Фрактальная революция в финансах: підруч. / Москва: Вильямс, 2006. 408с.
4. Минский М., Пейперт С. Перцептроны: підруч. / Москва: Мир, 1971.261 с.
5. Терехов В.А., Ефимов Д.В., Тюкин И.Ю. Нейросетевые системы управления: навч. посіб. / Москва: Высшая школа, 2002. 183 с.
6. LSTM – нейронная сеть с долгой краткосрочной памятью. URL: <https://neurohive.io/ru/osnovy-data-science/lstm-nejronnaja-set/> (дата звернення 10.09.2022)
7. Kevin Gurney. An introduction to neural networks. London : UCL Press, 1997.

# СИСТЕМНИЙ ПІДХІД У НАВЧАННІ МАТЕМАТИЦІ

Дроздова А.А.<sup>1</sup>, Данилов В.Я.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>tarasevych00@gmail.com,

<sup>2</sup>dani1ov1950@ukr.net [0000-0003-3389-3661]

**Система освіти України потребує негайних змін та нових підходів. Системний аналіз є невід'ємною частиною для побудування нової технології для навчання майбутніх вчителів математики. Впровадження штучного інтелекту зможе не тільки облегшити викладання, а й вивести його на новий персоналізований рівень для кожного учня.**

**Ключові слова: системний аналіз, методології навчання, штучний інтелект, парадигма.**

## 1. ВСТУП

У нашому сучасному світі прогрес розвивається стрімкими кроками вперед. Завдяки комп'ютеризації математизація доторкнулася до всіх областей науки. Сьогодні людство широко використовує методи математики. Наприклад, археологія, економіка, медицина, мовознавство, біологія. А ще кілька десятиліть вони вважались ніяк не пов'язані з математикою.

Сучасна система освіти потребує комплексного оновлення. Це можна зробити тільки за допомогою системного підходу.

## 2. СИСТЕМНЕ БАЧЕННЯ МЕТОДОЛОГІЇ НАВЧАННЯ

Системний аналіз – сукупність понять, методів, процедур і технологій для вивчення, опису, реалізації явищ і процесів різної природи і характеру, міждисциплінарних проблем; це сукупність загальних законів, методів, прийомів дослідження таких систем. Системний підхід у навчанні – це інноваційний напрямок, що досліджує сутність та закономірності виховання та навчання як єдину систему педагогічного процесу, як комплекс взаємопов'язаних заходів щодо формування світогляду та системи понять основ наук, системного мислення рідною та однією або кількома іноземними мовами як механізмів творчого розвитку особистості.

В основі такого підходу лежить метод, який спирається на такі науки, як системотехніка, кібернетика, математика, логіка, філософія, психологія, нейропсихологія та ін. життя системної лінгвістики.

Метод дозволяє в надзвичайно стислий термін вивчити будь-який предмет у середній та вищій школі. Особливі результати він дає при навчанні математиці, а оскільки є основою формування мислення. Це унікальний логічний метод вивчення математики, що дозволяє за допомогою системи типових базисних та ситуативних моделей, системи алгоритмів, особливим чином систематизованої навчальної дисципліни, зняти майже всі проблеми навчання і досягти автоматизму виконання різноманітних задач.

Сформуємо системне бачення цілей в шкільній математиці (Рис. 1).

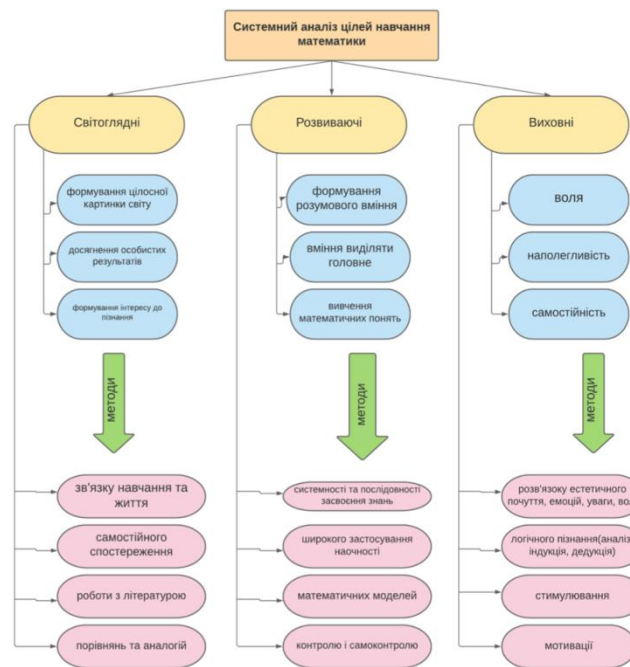


Рисунок 1. Системний аналіз цілей навчання

Системний підхід є інноваційною системною технологією навчання та виховання, впровадження в практику освіти якої дозволяє здійснити значні зміни у освіті: підвищити його результативність, різко знизивши у своїй терміни навчання.

Завдання, що висуває системний підхід у навчанні математики:

- формування системного мислення; досконале знання основ досліджуваних предметів та наук; володіння системою математичних понять.

Всі зазначені завдання досягаються за допомогою формування «правильного» свідомості та мислення - через оволодіння системою базисних моделей пропозицій.

Провідними принципами системного навчання є:

- опора на свідомість, мислення, мова; формування системного мислення; систематизація навчального матеріалу;

- використання системних понять, що вивчаються; визнання пріоритету у навчанні;

- суворе логічність у вивченні матеріалу на уроці, у підручнику, в організації навчального процесу;

- використання для навчання до межі стиснутої інформації (формули, моделі, системні схеми, алгоритмічна таблиця і т.д.);

- орієнтація на досконале знання, на чітке володіння навчальним матеріалом, на досягнення ефективності навчання, прагнення скорочення термінів навчання з усіх дисциплін на всіх ступенях освіти;

- використання блочного методу навчання як основного в подачі матеріалу; розширення знань з спіралі як усередині теми, предмета, і усередині всієї системи освіти;

- впровадження телекомп'ютерної системи навчання та освіти; міжнародний обмін, практика, контакти;

- висока духовність та ідейність освіти; моральність освіти, виховання високих моральних підвалин;

- використання для навчання моделей думки – базисних моделей пропозицій як найкоротшого шляху до знань у будь-яких областях; впровадження окрилюючого,



надихаючого навчання (за рахунок впровадження системно-модельного методу, що дозволяє швидко побачити результати);

Основні методики викладання представлені у схемі (Рис. 2).

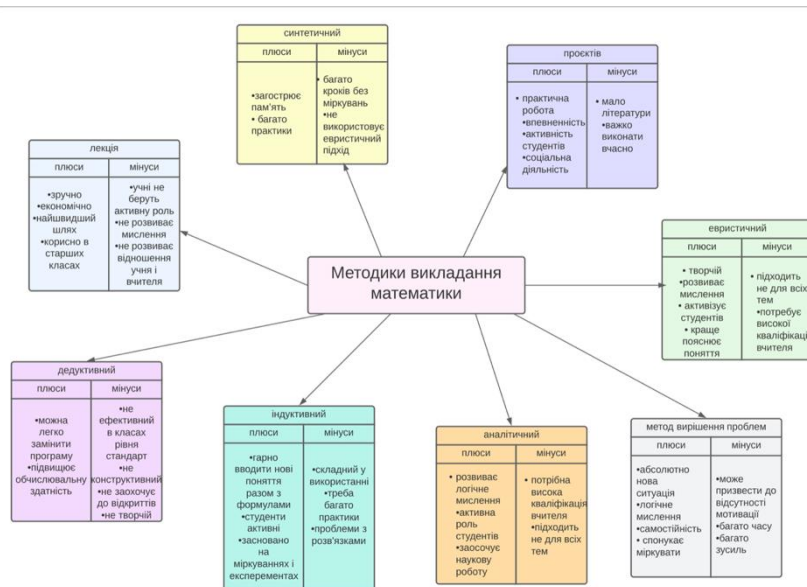


Рисунок 2. Методики викладання

Компетентнісний підхід у викладанні математики – це сукупність загальних принципів: визначення цілей математичної освіти, відбору змісту освіти, організації освітнього процесу та оцінки освітніх результатів. Сенс математичної освіти полягає у розвитку в учнів здатності самостійно вирішувати проблеми у різних видах діяльності на основі використання соціального досвіду. Організація процесу математичної освіти полягає у створенні умов для формування у студентів досвіду самостійного вирішення пізнавальних, комунікативних, організаційних, моральних та інших проблем, що становлять зміст освіти.

Інтерактивний метод – це взаємодія студента та викладача, їх спільна робота, творчі діяльності та процес взаємодії. Початок історії розвитку та використання інтерактивних методів навчання припадає на 20-ті роки XX століття, а в Киргизстані з 1997 року. З цього моменту використовувалися багато технологій інтерактивного методу, ми привели деякі з них, які в даний час значно застосовуються, але в той же час, говорити це межа розвитку інтерактивного методу не об'єктивно, так, як завжди технології оновлюються і тому, щоразу змінюється методи навчання.

Перехід до системного навчання, тобто впровадження в практику спочатку української освіти, а потім інших країн системної педагогіки дозволить створити спільноту інтегрованих держав щодо здійснення спільних зусиль з поглиблення інтеграції в освіті, тобто створення спільного освітнього простору.

Парадигма це комплекс наукових даних чи аксіом, прийнятих за правду, з урахуванням яких розгортається наукова, філософська чи релігійна теорія. Простіше кажучи, парадигма - той світ стопудових істин, який для вченого не підлягає сумніву. У перекладі з грецької слово парадигма означає «зразок, приклад».

У новій парадигмі освіти наголошується на здатності суспільства до трансформацій з урахуванням різних модифікацій на основі знань. Ця ідея лежить в основі інформаційного суспільства. Тому роль і значення знання, особливо теоретичного знання, зростає щорічно. Процес прийняття рішення, оцінки ризиків, контроль над власним життям, уміння управляти

процесами політики та економіки тощо зводять потенціал теоретичного знання до можливостей впливати на майбутнє людини.

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Технологія штучного інтелекту (ШІ) нещодавно привернула увагу своїм потенціалом у вдосконаленні різних сфер, зокрема освіти. Коротше кажучи, доведено, що штучний інтелект здатний підвищувати якість освіти шляхом удосконалення звичайних підходів до навчання та викладання. Зокрема, завдяки автоматизації деяких базових дій, таких як виставлення оцінок, це може надати вчителям більше часу для взаємодії зі своїми учнями або зосередитися на завданнях професійного розвитку. Корируючи навчання відповідно до конкретних потреб, здібностей або інтересів кожного учня, це може дозволити учням різних рівнів працювати разом в одному класі, а вчителі пропонують підтримку та допомогу, коли це необхідно. ШІ в освіті може забезпечити ефективне інтелектуальне навчання, розуміючи здібності, емоції та рівень досягнень учнів.

Цей процес включає такі основні етапи:

- I. Збір даних: збір ключових даних про учнів, які мають відношення до деяких або до всіх з наступних основних показників навчальних досягнень: академічна успішність, відвідуваність, робота на уроці, поведінка, послідовність, здатність до адаптації, наполегливість або «твердість характеру», впевненість у собі, задоволеність чи зацікавленість, емоційний розвиток, соціальний розвиток, етичний розвиток, фізичний розвиток.
- II. Подання даних: відображення даних у форматі, що дозволяє переглядати і запитувати інформацію, а також візуалізувати її.
- III. Історичний аналіз: ретроспективний аналіз показників успішності та досягнень учнів на поточний момент, що дозволяє зробити висновки та виявити області, які вимагають коригування або зміни курсу.

Всі наведені переваги можна побачити у схемі (Рис. 3).

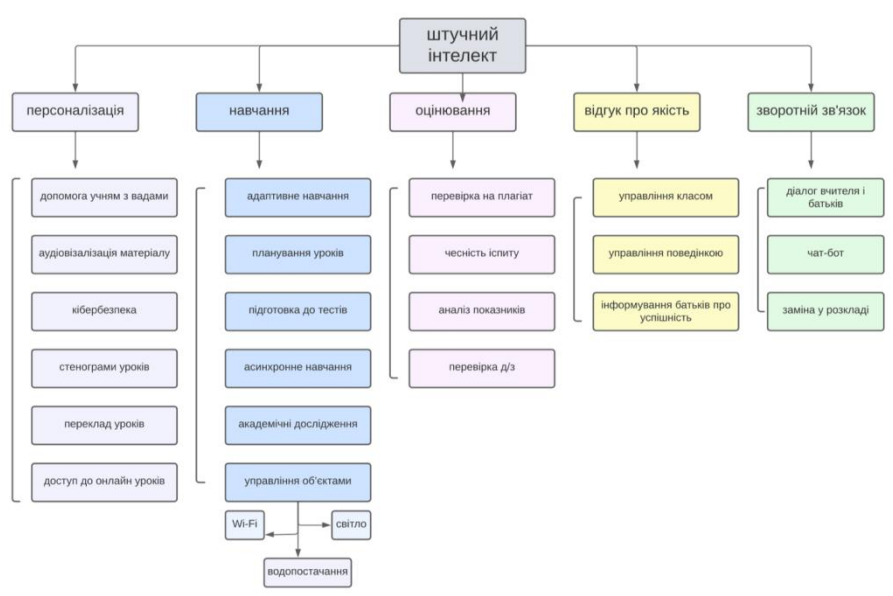


Рисунок 3. Штучний інтелект

## 4. ВИСНОВКИ

Отже, системний підхід лежить в основі якісної наскрізної методичної підготовки майбутніх учителів математики у педагогічних закладах вищої освіти, результатом якої є формування методичної компетентності студентів. На засадах системного підходу має відбуватися формування як методичних, так і ключових компетентностей майбутнього вчителя математики. З метою формування у майбутніх учителів математики методичної компетентності у навчанні учнів математики, система методичної підготовки має реалізовуватися на трьох етапах навчання: пропедевтичному, фактичному і праксеологічному.

У подальшому передбачається на основі детального вивчення функціонування окремих складових системи створити модель системи методичної підготовки майбутніх учителів математики.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Антонова О. Є. Формування у майбутніх учителів базових знань з педагогіки : дис. кандидата пед. наук : 13.00.01 / Антонова Олена Євгенівна. – Житомир, 1999. 189 с.
2. Акуленко І. А. Аксиологічний компонент методичних компетентностей Пленарні виступи 11 майбутніх учителів математики / І. А. Акуленко, Н. А. Тарасенкова // Вісник Черкаського університету. Серія: Педагогічні науки. – Черкаси, 2008. – Вип. 139. – С. 3 – 10.
3. Качество образования сильно мешает его количество. Главная Главная Математика в школе 2015, № 7 Математика в школе 2015, № 7 ОСОБЫЕ ТОЧКИ ОСОБЫЕ ТОЧКИ. Електронне джерело. URL: [http://www.schoolpress.ru/products/rubria/index.php?ID=71332&SECTION\\_ID=42](http://www.schoolpress.ru/products/rubria/index.php?ID=71332&SECTION_ID=42).
4. Аммосова Н. В. Методико-математическая подготовка студентов педагогических факультетов к развитию творческой личности школьника при обучении математике : дисс. ... д-ра пед. наук : спец. 13.00.02 «Теория и методика обучения математике» / Надежда Васильевна Аммосова ; Астраханский гос. пед. ун-т. – Астрахань, 1999. – 420 с.
5. Дубасенюк О. А. Професійна підготовка майбутнього вчителя до педагогічної діяльності: [монографія] / О. А. Дубасенюк, Т. В. Семенюк, О. Є. Антонова. – Житомир : Житомир. держ. пед. ун-т, 2003. – 193 с.
6. Жалдак М. И. Система подготовки учителя к использованию информационной технологии в учебном процессе : автореф. дис. ... д-ра пед. наук / Мирослав Иванович Жалдак. – М. : НИИ СИМО АПН СССР, 1989. – 48 с.
7. Луканкин Г. Л. Профессионализация фундаментальной естественно-математической подготовки учителя в высшем учебном

# **ПРОГНОЗУВАННЯ ОПЕРАЦІЙНИХ РИЗИКІВ ПІДПРИЄМСТВА (ЕКОНОМЕТРИЧНИЙ АНАЛІЗ)**

Д'яченко А.С.<sup>1</sup>, Бідюк П.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>aniyamaniya1@gmail.com [0000-0003-0036-4405],

<sup>2</sup>pbidyuke\_00@ukr.net [0000-0002-7421-3565]

**Для визначення прогнозного рівня втрат від операційних ризиків (ОР) за принципом економетрики у роботі використані регресори, які пов'язані з типами ОР підприємства. Прогнозування реалізовано в пакеті Eviews. За результатами досліджень ризику «втрати від збоїв у постачанні обладнання» з використанням прийомів регресійного аналізу та статистики Дарбіна-Уотсона, створені адекватні авторегресійні прогнозні моделі 1-5 порядку. Висока точність та якість прогнозу у побудованих моделях визначені високим коефіцієнтом детермінації та низькою середньою абсолютною похибкою.**

**Ключові слова: операційний ризик підприємства, EVIEWS, тренд, регресійний аналіз, коефіцієнт Тейла, авторегресія.**

## **1. ВСТУП**

Фінансові кризи примушують сектори виробничої діяльності переглядати підходи, створювати нові способи збільшення рівня прибутків, впроваджувати сучасні програмні засоби прогнозування. Особливо гостро це питання стосується виробничих, видобувних та інших підприємств. Прикро, але на виробничих підприємствах поняття ОР недостатньо поширено. Такий підхід треба змінювати, бо ОР може привести до найбільших збитків. Для забезпечення ефективного реагування на кризові ситуації компаніям необхідно визначитися з управлінням, аналізом та прогнозом ОР.

## **2. ПОСТАНОВКА ПРОБЛЕМИ ТА МЕТОДИ ДОСЛІДЖЕНЬ**

Мета дослідження полягає в аналізі та прогнозуванні операційних ризиків виробничого сектору економіки та практичному застосуванні інформаційних технологій – спеціалізованого економетричного пакету EVIEWS для вирішення задач математичного моделювання та прогнозування операційного ризику за допомогою кореляційного аналізу вихідних тимчасових даних. Виявлення стохастичного зв'язку та оцінка його сили – важливе завдання економетрики.

Методи дослідження стосовно розв'язання задач математичного моделювання та прогнозування операційного ризику: – а) казуальні (пошук зв'язку між певними критеріями і виявленою проблемою, застосування відповідної бази статистичних критеріїв для забезпечення високої якості даних і результатів); б) прогнозні статистичні (побудова адекватної моделі та генерування високоякісних прогнозів, передбачення майбутньої ситуації).

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

У поданому дослідженні аналіз ризиків проводився з урахуванням управлінської звітності за 2012 - 2021 рр. підприємства, що видобуває калійні солі. Проведений нами аналіз рівнів пріоритету ОР у формі рейтингування, показав, що основна причина настання ОР - погрози, пов'язані з втратами «від збоїв у постачанні систем та обладнання». Саме тому, цей тип ОР прийнятий для обчислювальних експериментів з побудови моделей прогнозування.

Обраний у дослідженні статистичний підхід ґрунтується на історичних даних ризикових подій підприємства. Кількісна оцінка ОР проводиться на підставі статистичних даних які мають поквартальні характеристики: величини втрат від ОР, кількість (од) і вартість обладнання (грн) та кількості зареєстрованих випадків що спричинили ризик. Прогнозування виконане на кінець 2022 року. Оскільки саме пакет Eviews [1] дозволяє працювати з часовими типами даних – це найкращий вибір для вирішення поставленого завдання та досягнення мети дослідження.

Для коректного проведення регресійного аналізу досліджено наявність можливих лінійних зв'язків між змінними, які в подальшому пояснюватимуть поведінку змінної (табл. 1), та визначити взаємозв'язок ризику з іншими змінними та змінних між собою.

Таблиця 1. Кореляція між змінними

	AMOUNT_O	AVERAGE_	COEFFICIENT	NUMBER_O	NUMBER_O	THE_NUMB
AMOUNT_O	1.000000	0.923658	0.161305	0.603975	0.970459	0.753462
AVERAGE_	0.923658	1.000000	0.197908	0.602374	0.970747	0.722572
COEFFICIENT	0.161305	0.197908	1.000000	0.121026	0.153841	0.090565
NUMBER_O	0.603975	0.602374	0.121026	1.000000	0.643044	0.396771
NUMBER_O	0.970459	0.970747	0.153841	0.643044	1.000000	0.764347
THE_NUMB	0.753462	0.722572	0.090565	0.396771	0.764347	1.000000

Визначено, що згідно з кореляційними показниками найбільше на ризик впливають середня вартість та кількість випадків, які спричиняють ризик. Це незалежні змінні, на основі яких зроблено прогноз для ОР на 2022 рік.

Один з етапів створення прогнозу моделі – розрахунок описових статистик (ОС) вибраних змінних (табл. 2), які для зручності ми перейменуємо так: кількість втрат від ризику – Y, середня вартість – X1, а кількість випадків, які зумовлюють ризик – X2. ОС необхідні для того, щоб більш детально чисельно охарактеризувати змінні, що вивчаються.

Таблиця 2. Описова статистика по усіх змінних

Date: 10/26/22 Time: 21:52 Sample: 2012Q1 2021Q4						
	AMOUNT_O	AVERAGE_	COEFFICIENT	NUMBER_O	NUMBER_O	THE_NUMB
Mean	2166.953	7225.875	0.023800	13.97500	111.5750	4.125000
Median	2000.860	7582.000	0.026000	14.00000	110.5000	4.000000
Maximum	4020.900	10138.00	0.047000	19.00000	153.0000	8.000000
Minimum	732.1200	4589.000	-0.018000	8.000000	79.00000	2.000000
Std. Dev.	862.5884	1723.606	0.014879	2.759482	21.72661	1.712511
Skewness	0.551121	-0.200245	-1.268780	-0.277718	0.256783	0.548236
Kurtosis	2.548818	1.882380	4.450743	2.393544	1.907649	2.505043
Jarque-Bera Probability	2.364172 0.306638	2.349111 0.308956	14.23978 0.000809	1.127164 0.569167	2.428301 0.296962	2.412052 0.299385
Sum	86678.10	289035.0	0.952000	559.0000	4463.000	165.0000
Sum Sq. Dev.	29018290	1.16E+08	0.008634	296.9750	18409.78	114.3750
Observations	40	40	40	40	40	40

**Примітка:** Mean, Median, Maximum, Minimum – середнє, медіанне, максимальне та мінімальне значення відповідно; Std. Dev. – стандартне відхилення (корінь із дисперсії); Skewness – асиметрія (0 у нормального розподілу); Kurtosis – ексцес (3 у нормального розподілу); Jarque-Bera - статистика Жака-Бера (для перевірки гіпотези про те, що вибірка

із нормального розподілу; *Probability* – ймовірність помилитися, відкинувши гіпотезу про те, що вибірка з нормального розподілу (у даному випадку ймовірність нульова, тому ми не помилимося, якщо скажемо, що вибірка отримана не з нормального розподілу); *Sum* – сума значень усіх спостережень, *Sum Sq. Dev.* – сума квадратів відхилень рівнів ряду від середнього; *Observations* – кількість спостережень.

Оскільки у лінійній моделі незалежних змінних дві, модель множинної регресії з коефіцієнтами розраховуємо методом найменших квадратів (Least Squares) (табл. 3).

Таблиця 3. Модель множинної регресії

Dependent Variable: Y Method: Least Squares Date: 10/26/22 Time: 22:04 Sample: 2012Q1 2021Q4 Included observations: 40				
Variable	Coefficient	Std. Error	t-Statistic	Prob.
X1	0.397134	0.043135	9.206663	0.0000
X2	90.70055	43.41496	2.089154	0.0436
C	-1076.827	226.1230	-4.762131	0.0000
R-squared	0.868639	Mean dependent var	2166.953	
Adjusted R-squared	0.861538	S.D. dependent var	862.5884	
S.E. of regression	320.9731	Akaike info criterion	14.45263	
Sum squared resid	3811879.	Schwarz criterion	14.57930	
Log likelihood	-286.0526	Hannan-Quinn criter.	14.49843	
F-statistic	122.3330	Durbin-Watson stat	0.393909	
Prob(F-statistic)	0.000000			

Оскільки ймовірність дуже мала, використовується альтернативна гіпотеза, яка говорить про нерівність коефіцієнтів нулю і їх статистичну значущість. Коефіцієнт детермінації (R-squared) на 87 % пояснює зміну Y в залежності від незалежних змінних. Таке значення є цілком прийнятним для побудованої моделі.

**Примітка:** *Std Error* – стандартна помилка; *t-статистика* (обраховується діленням коефіцієнту на стандартну помилку); *Prob* – вірогідність прийняття нульової гіпотези., *R-squared* – R-квадрат – це коефіцієнт детермінації. Метод оцінювання коефіцієнтів моделі – метод найменших квадратів.

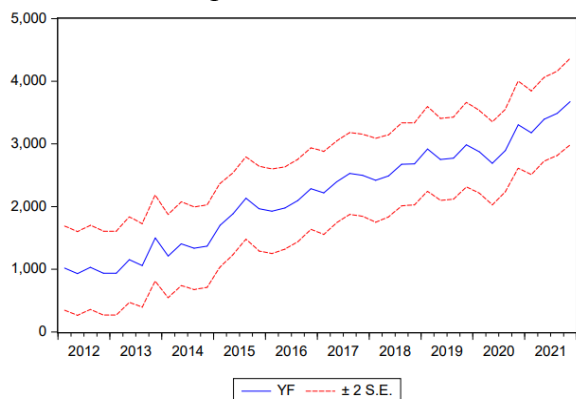
Наступний крок прогнозування – визначення статистичних характеристик (СХ) адекватності моделі або однокрокове прогнозування. Для цього обчислюємо коефіцієнти множинної регресії для X1 та X2 – табл. 4 та отримуємо результати розрахунку СХ оцінок прогнозу і їх якості з оцінкою адекватності моделі (задовільний коефіцієнт детермінації).

Таблиця 4. Статистика та коефіцієнти множинної регресії

Dependent Variable: Y Method: Least Squares Date: 10/26/22 Time: 22:04 Sample: 2012Q1 2021Q4 Included observations: 40				
Variable	Coefficient	Std. Error	t-Statistic	Prob.
X1	0.397134	0.043135	9.206663	0.0000
X2	90.70055	43.41496	2.089154	0.0436
C	-1076.827	226.1230	-4.762131	0.0000
R-squared	0.868639	Mean dependent var	2166.953	
Adjusted R-squared	0.861538	S.D. dependent var	862.5884	
S.E. of regression	320.9731	Akaike info criterion	14.45263	
Sum squared resid	3811879.	Schwarz criterion	14.57930	
Log likelihood	-286.0526	Hannan-Quinn criter.	14.49843	
F-statistic	122.3330	Durbin-Watson stat	0.393909	
Prob(F-statistic)	0.000000			



За статистикою Дарбіна-Уотсона (показник DW, яка стверджує, що похибки такої моделі є невисоко корельованими, приймаємо побудовану модель задовільною, але розуміємо, що модель можна покращити. Результат СХ оцінок прогнозу і їх якості в графічному вигляді (рис. 1, а) дозволяє визначити значення середньої абсолютної похибки (MAPE), середньоквадратичної похибки прогнозу (MSE) та коефіцієнту невідповідності Тейла (Theil) – рис. 1, б.



Forecast: YF	
Actual: Y	
Forecast sample: 2012Q1 2021Q4	
Included observations: 40	
Root Mean Squared Error	308.7021
Mean Absolute Error	266.4764
Mean Abs. Percent Error	13.20551
Theil Inequality Coefficient	0.066586
Bias Proportion	0.000000
Variance Proportion	0.035192
Covariance Proportion	0.964808
Theil U2 Coefficient	2.327667
Symmetric MAPE	13.15960

*MSE = 308,7021;*  
*MAPE = 13,2%;*  
*Theil = 0,066586*

а)

б)

Рисунок 1. Результат СХ оцінок прогнозу і їх якості в графічному вигляді (а) та показники, що характеризують якість виконаного прогнозу (б)

Отриманий результат: прогнозні значення добре співвідносяться. Але значення «MAPE» свідчать про бажане покращення моделі, тому надалі обчислення робимо покрокове та підвищуємо порядок регресії. Проводимо тест на серію корельованих залишків Бройша-Годфрі, який дозволяє нам перевірити гіпотезу про те, що інші моделі описуються моделлю авторегресії. Це Serial Correlation LM Test - автокореляційна функція (АКФ) для змінної X1 у вигляді корелограми (рис. 2).

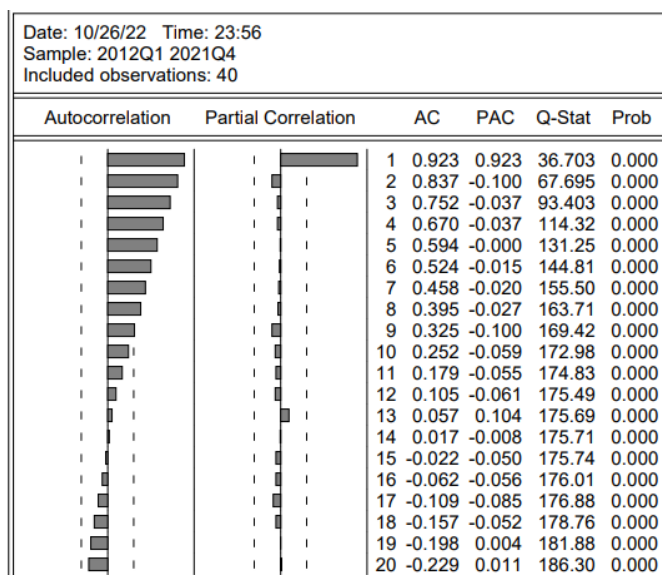


Рисунок 2. Корелограма значень автокореляційної та часткової АКФ

Аналіз отриманих даних дозволяє зробити висновок, що порядок авторегресії не перевищує 5. Тому аналізуємо авторегресію з покроковим підвищенням порядку - від першого до можливо максимального п'ятого. Результати розрахунку авторегресії першого та

п'ятого порядку подані в табл. 5. Спостерігається суттєве покращення оцінок прогнозів. Надалі застосовуємо апроксимацію моделі п'ятого порядку трендом (табл. 6). Рівняння побудованих автокореляційних моделей вказані в таблицях 5, 6. Отримана кінцева модель при порівнянні її з вхідними даними показала прийнятний результат: СХ прогнозу моделі з трендом –  $R^2=0.99$ ,  $DW=0.68797$ ,  $\Sigma e^2=259668.8$ ; СХ прогнозу моделі без тренду –  $R^2=0.8686$ ,  $DW=0.3937$ ,  $\Sigma e^2=3811879.1$ . Дві статистичні характеристики моделі покращились, показник «МАРЕ» є цілком задовільним.

Таблиця 5. Результати розрахунку авторегресії першого та п'ятого порядків

Dependent Variable: X1 Method: Least Squares (Gauss-Newton / Marquardt steps) Date: 10/27/22 Time: 21:08 Sample (adjusted): 2012Q2 2021Q4 Included observations: 39 after adjustments X1=C(1)+C(2)*X1(-1)					Dependent Variable: X1 Method: Least Squares (Gauss-Newton / Marquardt steps) Date: 10/27/22 Time: 00:00 Sample (adjusted): 2016Q2 2021Q4 Included observations: 23 after adjustments X1=C(1)+C(2)*X1(-1)+C(3)*X1(-2)+C(4)*X1(-9)+C(5)*X1(-13)+C(6)*X1(-17)					
	Coefficient	Std. Error	t-Statistic	Prob.						
C(1)	143.6036	147.7438	0.971977	0.3374	C(1)	737.6053	570.8157	1.292195	0.2136	
C(2)	0.999815	0.020126	49.67690	0.0000	C(2)	1.439980	0.214028	6.728005	0.0000	
R-squared	0.985228	Mean dependent var	7293.487		C(3)	-0.591378	0.267005	-2.214859	0.0407	
Adjusted R-squared	0.984829	S.D. dependent var	1691.546		C(4)	-0.000758	0.091580	-0.008274	0.9935	
S.E. of regression	208.3481	Akaike info criterion	13.56622		C(5)	0.056877	0.079673	0.713887	0.4850	
Sum squared resid	1606130.	Schwarz criterion	13.65153		C(6)	0.034543	0.077061	0.448248	0.6596	
Log likelihood	-262.5412	Hannan-Quinn criter.	13.59683		R-squared	0.979564	Mean dependent var	8476.870		
F-statistic	2467.794	Durbin-Watson stat	1.628978		Adjusted R-squared	0.973554	S.D. dependent var	848.9132		
Prob(F-statistic)	0.000000				S.E. of regression	138.0532	Akaike info criterion	12.91261		
					Sum squared resid	323997.7	Schwarz criterion	13.20883		
					Log likelihood	-142.4951	Hannan-Quinn criter.	12.98711		
					F-statistic	162.9743	Durbin-Watson stat	1.986286		
					Prob(F-statistic)	0.000000				
перший порядок					п'ятий порядок					

Таблиця 6. Результати оцінювання моделі для прогнозування Y з трендом

Dependent Variable: Y Method: Least Squares (Gauss-Newton / Marquardt steps) Date: 10/28/22 Time: 00:03 Sample: 2012Q1 2021Q4 Included observations: 40 Y = C(1)+C(2)*X1+C(2)*X2+C(3)*K+C(4)*K^2+C(5)*K^3				
	Coefficient	Std. Error	t-Statistic	Prob.
C(1)	1180.600	183.4514	6.435492	0.0000
C(2)	-0.158483	0.044878	-3.531432	0.0012
C(3)	191.7649	15.81718	12.12384	0.0000
C(4)	-7.452820	0.715458	-10.41685	0.0000
C(5)	0.138378	0.011377	12.16298	0.0000
R-squared	0.991052	Mean dependent var	2166.953	
Adjusted R-squared	0.990029	S.D. dependent var	862.5884	
S.E. of regression	86.13426	Akaike info criterion	11.86616	
Sum squared resid	259668.8	Schwarz criterion	12.07727	
Log likelihood	-232.3232	Hannan-Quinn criter.	11.94249	
F-statistic	969.0725	Durbin-Watson stat	0.687973	
Prob(F-statistic)	0.000000			

#### 4. ВИСНОВКИ

Послідовність виконаних під час дослідження кроків по розрахунку прогнозних показників, дозволяє зробити деякі висновки, а саме: втрати від збоїв у постачанні систем та обладнання» на 4 квартали 2012 р. та прогнозних розрахунків 2022 р. показали, що показник втрат ОР збільшився майже в 4,68 разів, тобто зростання втрат складає практично 47 % на



рік. Випробувані методика оцінки та прогнозування має як переваги, так й недоліки. Залежно від цілей і наявності ПЗ, або при недостатньому статистичному полі треба комбінувати різні методики, а підсумкову оцінку операційних втрат ОР розраховувати як середньозважене.

### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. *Построение AR моделей для Индекса отребительских цен в пакете Eviews 5* (2022)  
Студенческая библиотека онлайн: [studbooks.net](http://studbooks.net). Режим доступа:  
[https://studbooks.net/2273206/informatika/postroenie\\_modeley\\_indeksa\\_potrebitelskih\\_pakete\\_evie](https://studbooks.net/2273206/informatika/postroenie_modeley_indeksa_potrebitelskih_pakete_evie)  
[ws](https://studbooks.net/2273206/informatika/postroenie_modeley_indeksa_potrebitelskih_pakete_evie) (дата звернення 18.09.22).

# АНАЛІЗ РИНКОВОГО РИЗИКУ ЙМОВІРНІСНО-СТАТИСТИЧНИМИ МЕТОДАМИ

Затірка В.В.<sup>1</sup>, Бідюк П.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> lerazatirka@gmail.com,

<sup>2</sup> pbidyuke\_00@ukr.net [0000-0002-7421-3565]

**Робота будь-яких фінансових установ тісно пов'язана з виконанням різноманітних операцій з валютними ресурсами. У зв'язку з цим виникає необхідність високоякісного аналізу та управління ринковими ризиками. Багато широкоживаних моделей управління ризиками не підходять для використання українським фінансовим ринком, а отже необхідно розглянути та вдосконалити менш звичні методології. Метою цієї роботи є побудова адекватних моделей гетероскедастичних фінансових процесів для прогнозування дисперсії та оцінювання ринкових ризиків. Результатом дослідження є побудована модель, придатна для оцінювання ринкових ризиків. У роботі використано теоретичні та системні емпіричні методи дослідження.**

**Ключові слова:** ринковий ризик, гетероскедастичний процес, дисперсія, VAR, метод Монте-Карло, аналіз ризиків.

## 1. ВСТУП

Будь-яким фінансовим рішенням завжди властива наявність ризику. Однією з основних проблем фінансових установ є оцінювання ринкових ризиків, які виникають в результаті випадкових змін цін, процентних ставок, курсів обміну валют та інших ринкових факторів. Більшість фінансово-економічних процесів економіки перехідного періоду мають нестаціонарний характер та містять нелінійності і через це виникає задача їх коректного моделювання, прогнозування та оцінювання супутніх ризиків.

Український фінансовий ринок в умовах перехідної економіки не є стабільним, таким, що динамічно розвивається і швидко реагує на зміни ринкових факторів. Отже, задача застосування і удосконалення методологій оцінки ризиків в Україні ставиться досить гостро.

## 2. МЕТОДИ ОЦІНЮВАННЯ РИНКОВОГО РИЗИКУ

Таксономія ризиків є важливим інструментом управління ризиками, оскільки вона полегшує процес ідентифікації ризиків і прискорює вибір того інструменту, як з ними боротися. Знаючи категорію ризику, можна чітко зрозуміти основний метод управління цим ризиком, оскільки для багатьох категорій ризику вже існують ефективні методи та принципи управління. НБУ запропонована класифікація ризиків, яка враховує їх залежність від доходів:

- Фінансові ризики – тобто ризики, що піддаються кількісній оцінці, а управління ними полягає в їх оптимізації.
- Нефінансові ризики – тобто ризики, що не піддаються кількісній оцінці, а управління ними зводиться до їх мінімізації [1].

Фінансові ризики в свою чергу поділяють на 4 категорії: кредитний ризик, операційно-технологічний ризик, ризик ліквідності та ринковий ризик [1]. Ринкові ризики включають такі види ризиків: ризик дефолту, процентний ризик торгової книги, ризик кредитного спреда, фондовий ризик, валютний ризик, товарний та ризик волатильності [2]. В роботі розглядаються саме ринкові ризики та методи його аналізу, оскільки вони є невід'ємною складовою діяльності будь-яких фінансових установ.

Ринкові фактори, а саме постійні випадкові зміни цін, курси обміну валют, процентні ставки та інші, є причиною виникнення ринкових ризиків. Оцінка ринкових ризиків є однією з головних проблем фінансових установ, і саме тому для цього існує множина методів. Єдиного рішення для ризик-менеджменту не існує, а тому різні фінансові організації намагаються знайти метод, який би краще підходив для їх конкретних запитів.

Кількісною мірою оцінки ринкового ризику є величина value-at-risk (VaR). Основою для оцінювання цієї величини є зміни курсу та цін за певний період часу в минулому. VaR – це величина збитків, яка з певною ймовірністю не буде перевищена. Цю ймовірність називають рівнем довіри. Загалом прийнято робити оцінку з рівнем довіри 95% або 99%. Простіше кажучи, величина VaR – це максимально ймовірна втрата. Її можна вимірювати у грошовому вираженні та за відсотковою ставкою.

Для того, щоб переконатися в адекватності моделі, необхідно верифікувати її, тобто провести «бек-тестування». Верифікація дозволяє встановити ступінь адекватності використаної моделі оцінки ризику у вигляді показника VaR реальним умовам ринку [3].

Існують дві групи методів обчислення VaR: параметричні та непараметричні. Широко відомими непараметричними методами є метод історичного моделювання та метод імітаційного моделювання Монте-Карло. В свою чергу, серед параметричних методів популярністю користуються дельта-нормальний метод та метод дельта-гама наближення.

Поширеною мірою ризику є дисперсія, або стандартне відхилення очікуваної доходності. Вона є доволі точною й широко визнаною мірою ризику, яку використовують у теоретичних моделях оцінювання фінансових інструментів. Це показник розпорошення доходності навколо очікуваної величини. Чим менше розпорошені ставки доходів навколо очікуваної величини, тим вищою є впевненість в одержанні цієї доходності у майбутній період [3]. Пошук волатильності доходності фінансових інструментів є однією з головних задач при використанні дельта-нормального методу для визначення величини VaR.

Дисперсія фінансових процесів є змінною у часі величиною, а отже нам необхідно побудувати математичну модель, яка б коректно описувала динаміку дисперсії та могла б робити прогнози на кілька кроків вперед. Для аналізу фінансових часових рядів часто використовують модель авторегресії з умовною гетероскедастичністю (АРУГ) [4]. Гетероскедастичність означає, що дисперсія процесу є змінною в часі на інтервалі дослідження і при створенні моделі процесу потрібно знайти закон зміни дисперсії. Побудувавши адекватну модель, ми можемо значно покращити якість рішень в управлінні ризиком.

### **3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ**

Для того щоб застосувати методологію оцінювання ринкового ризику на основі гетероскедастичних моделей динаміки його волатильності у роботі використовуються дані стосовно цін акцій, представлені на світових біржах. Для детального опису процесу побудови моделей динаміки дисперсії та оцінки ринкового ризику на фондовому ринку було обрано дані компанії Qualcomm (QCOM). Вхідні дані – це зміна динаміки цін закриття активів за період з 1 листопада 2021 року по 31 жовтня 2022 року (Рис.1). Загальна кількість значень у вибірці – близько 250.

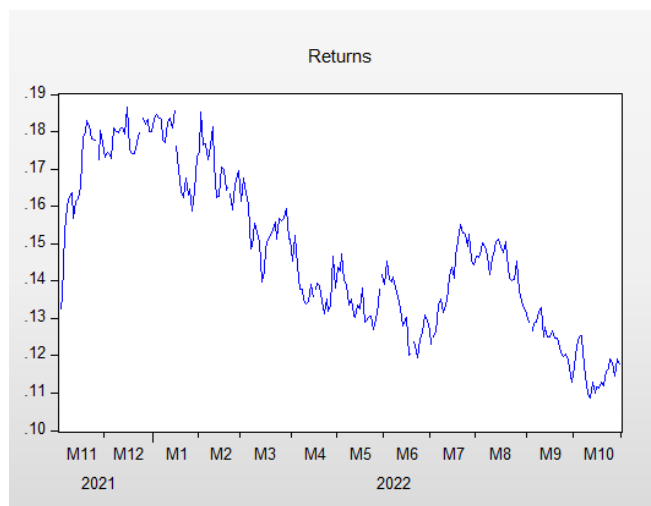


Рисунок 1. Значення доходностей акцій компанії Qualcomm.

Для того щоб побудувати модель авторегресії (АР) необхідно спочатку побудувати автокореляційну та часткову автокореляційну функцію (АКФ та ЧАКФ відповідно). Модель АР першого порядку має такий вигляд:  $y(k) = c(1) + c(2) * y(k-1)$ . Після оцінювання рівняння АР(1), отримуємо модель такого вигляду:  $y(k) = 0.002712 + 0.981243 * y(k-1) + e(k)$ .

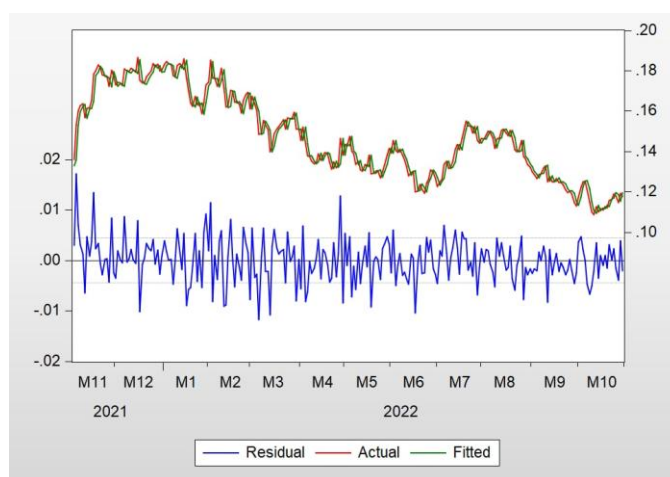


Рисунок 2. Графіки вихідного та оціненого рядів

Важливо підкреслити, що процес є гетероскедастичним, що підтверджується тестом Уайта щодо перевірки на гетероскедастичність залишків моделі. Наступний крок – це побудова моделі динаміки дисперсії. Для цього нам спершу потрібно визначити порядок моделі дисперсії, побудувавши корелограму квадратів залишків, а також корелограму вибіркової умовної дисперсії. Результати побудови корелограм дають нам зрозуміти, що порядки моделей для опису динаміки дисперсії рівні  $p=1$  та  $q=1$ . Побудуємо такі моделі: ARCH(1), GARCH(1,1), EGARCH(1,1) та GJR(1,1).

Результати порівняння якості моделей зведено до Таблиці 1. Результати верифікації моделей для оцінювання ризику подано у Таблиці 2.

Загалом, за результатами моделювання можна зробити висновок, що кожна модель є задовільною для оцінювання ризиків. Втім, зважаючи на те, що інформаційні критерії моделей GARCH та EGARCH приймають мінімальні значення, вони є найкращими серед усіх оцінених.

Таблиця 1. Результати моделювання волатильності для вибраної акції

Тип моделі	LogL	AIC	SC	HQ
ARCH (1)	1806.562	-14.41250	-14.34207	-14.38415
GARCH(1,1)	2744.473	-21.90779	-21.82327	-21.87377
EGARCH(1,1)	2768.532	-22.09226	-21.99365	-22.05257
GJR(1,1)	1204.215	-9.577720	-9.479119	-9.538036

Таблиця 2. Результати процедури бек-тестування для оцінювання VaR

Тип моделі	VaR 95%	VaR 99%
ARCH (1)	(0) 100%	(0) 100%
GARCH(1,1)	(3) 98.81%	(0) 100%
EGARCH(1,1)	(8) 97.8%	(4) 98.43%
GJR(1,1)	(11) 95.58%	(3) 98.81%

Тепер порівняємо моделі на якість прогнозування. Нехай розмір навчальної вибірки нараховує 240 значень. Спрогнозуємо 15 наступних значень. Результати прогнозування подані у Таблиці 3.

Таблиця 3. Результати прогнозування волатильності за кожною з моделей

Тип моделі	MAE	MAPE	Theil Inequality
ARCH(1)	0.000728	36.96	0.0994
GARCH(1,1)	0.000547	1.68	0.0923
EGARCH(1,1)	0.000539	1.64	0.0983
GJR(1,1)	0.001551	4.84	0.0969

Порівняємо результати оцінювання VaR з результатами, отриманими за допомогою методу Монте-Карло, який є одним з найпопулярніших методів для оцінювання VaR.

Таблиця 4. Результати процедури бек-тестування для оцінювання VaR

Тип моделі	VaR 95%	VaR 99%
ARCH (1)	(0) 100%	(0) 100%
GARCH(1,1)	(3) 98.81%	(0) 100%
EGARCH(1,1)	(8) 97.8%	(4) 98.43%
GJR(1,1)	(11) 95.58%	(3) 98.81%
Монте-Карло	(7) 97.20%	(1) 99.96%

Загалом, можемо бачити, що усі моделі є у прийнятній мірі адекватними. Найкращими, як за статистичними критеріями, так і за практичною значимістю є моделі GARCH(1,1) та EGARCH(1,1). Модель експоненційної узагальненої авторегресійної умовно гетероскедастичності є найбільш адекватною для поставленої задачі, оскільки вона не завищує оцінки ризикової вартості та максимально правдоподібно відображає динаміку процесу. Порівнюючи результати, отримані за допомогою побудованих моделей з методом Монте-Карло, ми можемо бачити, що вони є достатньо близькими. Метод Монте-Карло є одним з найточніших для оцінювання ризиків, але складним в побудові. Використання гетероскедастичних процесів для оцінювання ризиків є простішою альтернативою.

#### 4. ВИСНОВКИ

Фінансова організація чи виробниче підприємство за умов сучасної ринкової економіки зіштовхується з фінансовими ризиками, які можуть негативно позначитися на фінансову ефективність його діяльності. У зв'язку з цим велике значення мають наукові дослідження в галузі математичного моделювання процесів оцінювання фінансових ризиків та управління ними, що ґрунтуються на методах імовірнісної фінансової математики.

У ході дослідження побудовано адекватні моделі для опису динаміки поведінки та прогнозування дисперсії, яка використовується для оцінювання ринкового ризику. Для порівняння було побудовано кілька моделей – ARCH (1), GARCH(1,1), EGARCH(1,1) та GJR(1,1). В усіх випадках моделювання найгірші результати дає проста модель ARCH. Найкращі результати отримали за моделлю EGARCH.

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ризики [Електронний ресурс] – Режим доступу до ресурсу: <https://a-bank.com.ua/uploads/media/default/0001/01/0dee1c4388ffda54e69e5fb7a98f5f948477edff.pdf>.
2. Про затвердження Положення про організацію системи управління ризиками в банках України та банківських групах: постанова НБУ від 11.06.2018 р. № 64. Дата оновлення: 06.12.2021. Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/v0064500-18#Text>
3. Портфельне інвестування [Електронний ресурс] / А. А.Пересада, О. Г. Шевченко, Ю. М. Коваленко, С. В. Урванцева. – 2004. – Режим доступу до ресурсу: <https://buklib.net/books/21928/>.
4. Визначення величини ризику VaR на основі оцінок параметрів моделі стохастичної волатильності [Електронний ресурс] / П.І. Бідюк, М.М. Коновалюк // Систем. дослідж. та інформ. технології. — 2012. — № 3. — С. 85-94. – Режим доступу до ресурсу: <http://dspace.nbuv.gov.ua/handle/123456789/50180>
5. Taylor S.J. Modelling stochastic volatility: a review and comparative study // Mathematical Finance. — 1994. — No 4. — P. 183–204.

# **МОДЕЛЮВАННЯ СИСТЕМИ ПРОГНОЗУВАННЯ ЕФЕКТИВНОСТІ РЕКЛАМНИХ НОСІЇВ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ**

Каніовська І.Ю.<sup>1</sup>, Бойко П.О.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>ikaniovska@gmail.com,

<sup>2</sup>pjoidrivedisk@gmail.com

**Продуктові компанії розміщують величезну кількість реклами і кожна реклама має свій рекламний носій. Ці носії були створені на основі досвіду дизайнерів і потреб компанії. Як наслідок, немає способу оцінити оголошення під час виробництва та дізнатися, наскільки вони добре працюють при використанні. Створення системи, яка буде прогнозувати ефективність рекламних носіїв суттєво допоможе правильно витратити рекламні бюджети та аналізувати основні складові успішних рекламних матеріалів. Метою є аналіз рекламних носіїв для розуміння їх складової ефективності. Результатом дослідження є розуміння основних елементів креативу, які впливають на ефективність та побудова системи прогнозування ефективності рекламних носіїв.**

**Ключові слова: рекламний носій, креатив, машинне навчання, ком'ютерне бачення, задача прогнозування.**

## **1. ВСТУП**

Продуктові компанії весь час витрачають мільйони на рекламу в інтернеті чи фізичному світі. В багатьох великих компаніях постає запитання, як зменшити неефективне використання рекламного бюджету. Однією із проблем “зливання” бюджету - маркетологи не можуть оцінювати ефективність рекламних носіїв до того, як вони використовують їх у рекламі.

Своєчасне припинення показу або не використання рекламних матеріалів є однією з найважливіших операцій з рекламою, яка може мати значний вплив на рекламні бюджети. Після попереднього аналізу 30 000 рекламних оголошень виявилось, що 50% рекламних креативів не принесли взагалі ні одної покупки. Це приблизно 5% рекламного бюджету, що були використані неефективно.

## **2. МЕТОДИ ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ РЕКЛАМНИХ НОСІЇВ**

Рекламний креатив — це зображеннями, або відео, які відображаються користувачам на веб-сторінці, у додатку чи іншому цифровому середовищі [1]. Є три типи оцінювання ефективності по часу: до запуску реклами, після закінчення транслявання рекламних носіїв та в реальному часі. На даний момент майже не досліджено перший тип - до запуску реклами.

Основні метрики по яким оцінюють ефективність рекламних носіїв – це коефіцієнт конверсії, коефіцієнт залучення, CTR, загальна вартість реклами.

Коефіцієнт конверсії: під час планування цифрової маркетингової кампанії коефіцієнт конверсії є одним із найважливіших KPI, на якому слід зосередитися. Цей показник відображає кількість відвідувачів, які перетворилися на клієнтів. Існує кілька різних типів коефіцієнтів конверсії, і можна використовувати будь-який із них, щоб визначити успіх рекламних кампаній.

Загальна вартість реклами: окрім коефіцієнтів конверсії, також можна переглянути загальну вартість кожного потенційного клієнта, що дасть вам уявлення про вартість ваших маркетингових зусиль.

CTR: відстеження того, скільки разів клієнт натискає на певне оголошення або відкриває електронний лист, може допомогти маркетологам визначити, наскільки ефективними є їхні стратегії цифрового маркетингу. Наявність точного звіту про ключові показники ефективності вашої кампанії може допомогти вам визначити, які кроки потрібно зробити, щоб збільшити продажі та покращити ефективність [2].

Залучення: вимірює кількість вподобайок, розповсюджень і коментарів, отриманих вашою рекламою. Цей показник показує, наскільки ваша реклама подобається людям. Це важливий показник, який слід перевіряти, бо реклами стає все більше, тому потрібно створювати таку рекламу, яка буде захоплювати користувачів взаємодіяти з нею.

Ключові показники ефективності рекламних носіїв можна покращити за допомогою виділення основних об'єктів в рекламних матеріалах, які є успішними, та застосування цих знань при розробці нових рекламних носіїв.

Основні об'єкти рекламних носіїв – це колір, текст, логотипи, предмети на зображенні або відео.

Для того щоб оцінювати рекламні кретиви до запуску можна застосувати методи комп'ютерного зору на основі глибокого навчання [3]. Це дозволить виділяти та проводити аналіз по основних складових рекламного носія:

- текст поданий у рекламі,
- основні кольори, текстури, яскравість,
- об'єкти (люди, тварини, тощо),
- розташування логотипу або кнопки, яка викликає дію, наприклад, покупка.

Для побудови моделі, яка буде обраховувати ефективність рекламних матеріалів, можна використовувати алгоритми машинного навчання. Один із прикладів – це використання навчання з вчителем. Тобто прогнозування числового значення, наприклад, CTR рекламних носіїв, маючи набір характеристик або ознак (проект, колір, текст тощо). Найбільш важливі алгоритми навчання з вчителем – це лінійна регресія (linear regression), логістична регресія (logistic regression), дерева прийняття рішень (decision trees), випадковий ліс (random forest), нейронні мережі (neural networks) [4].

### **3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ**

Для дослідження було отримано доступ до даних по рекламних носіїв однієї з продуктових компаній України. В даних було отримано:

- рекламні носії,
- дані про дохід з рекламних носіїв,
- дані про витрати на показ рекламних носіїв.

Для отримання даних було написано два програмних кода на мові програмування Python 3. Перший на щоденній основі збирає дані витрат та доходів. Другий завантажує з Google Drive рекламні носії та зберігає локально на машину для подальшого використання.

Для розробки моделі, яка буде розраховувати ефективність рекламних носіїв було розроблено дві архітектури (Рис. 1 та Рис. 2).



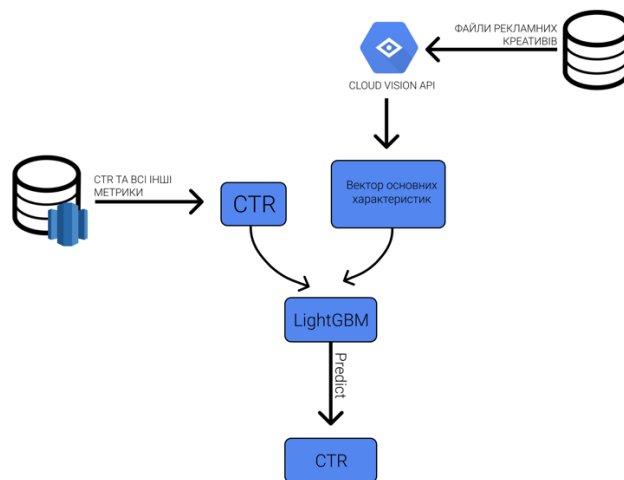


Рисунок 1. Архітектура №1

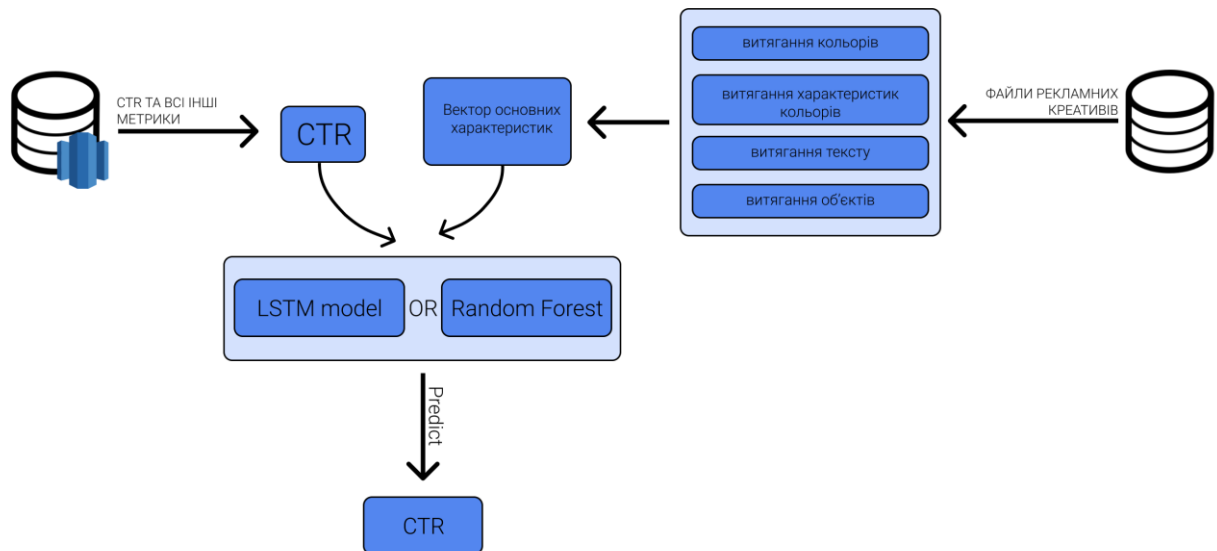


Рисунок 2. Архітектура №2

Архітектуру №1 було не реалізовано в повній мірі, бо один з елементів - Cloud Vision Api не безкоштовний сервіс. Тому в Архітектурі №2 було замінено на 4 основні блоки: витягання кольорів з рекламних носіїв, розпізнавання тексту, розпізнавання об'єктів та отримання більш детальних характеристик по кольору.

Маючі рекламні креативи спочатку отримуємо інформацію про колірну композицію зображень. За це відповідають дві функції: домінуючі кольори та показник барвистості. Для цього застосовувалась бібліотека extcolors. Також для розуміння барвистості було використано підхід, який Hasler and Sússtrunk розробили в своїй статті (шляхом опитування людей для оцінки барвистості за шкалою), бо люди можуть сприймати зображення як кольорове, дуже барвисте, не дуже кольорове або зовсім небарвисте (чорно-біле) [5].

Далі за допомогою бібліотеки `pytesseract` вилучаємо текст із рекламних носіїв. Та проводимо сентимент-аналіз для розуміння розподілу настрою в рекламному тексті. В результаті побачили більшість “позитивних” текстів.

Фінальний етап аналізу рекламних креативів був для вилучення типу об’єктів та його розташування на зображеннях за допомогою `Yolo`. Але не вийшло довести до успішного результату цей етап, бо алгоритм завжди виокремлював тільки людей (на рекламних носіях, які аналізувались в роботі, в більшості випадків були зображені люди), а всі інші елементи ігнорував на рекламних матеріалах.

Весь аналіз до цього зберігався в окремих для кожного етапу текстовому файлі. Далі в єдиний `Pandas DataFrame` збираються основні характеристики та метрики рекламних матеріалів і передаються на наступний етап машинного навчання. Основні колонки в фінальному `DataFrame` зображені на Рис. 3.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2209 entries, 0 to 2208
Data columns (total 29 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   color_1                               2209 non-null   object
1   color_1_occurrence                    2209 non-null   int64
2   color_2                               2209 non-null   object
3   color_2_occurrence                    2209 non-null   int64
4   color_3                               2209 non-null   object
5   color_3_occurrence                    2209 non-null   int64
6   color_4                               2209 non-null   object
7   color_4_occurrence                    2209 non-null   int64
8   color_5                               2209 non-null   object
9   color_5_occurrence                    2209 non-null   int64
10  click                                  2209 non-null   int64
11  impression                             2209 non-null   int64
12  post_engagement                       2209 non-null   int64
13  ctr                                    2205 non-null   float64
14  ID                                     2209 non-null   object
15  Text                                   2077 non-null   object
16  Text_Char_Length                      2209 non-null   int64
17  Text_Word_Length                      2209 non-null   int64
18  sentiment_score                       2209 non-null   float64
19  sentiment                              2209 non-null   object
20  Ad_color_diversity                    2209 non-null   object
21  Ad_color_valence                      2209 non-null   float64
22  Ad_color_arousal                      2209 non-null   float64
23  Ad_color_dominance                    2209 non-null   float64
24  Ad_hue_dof                            2209 non-null   float64
25  Ad_saturation_dof                     2209 non-null   float64
26  Ad_value_dof                          2209 non-null   float64
27  Ad_edge_pixel                          2209 non-null   object
28  Ad_s_mean                              2209 non-null   float64
dtypes: float64(9), int64(10), object(10)
memory usage: 517.7+ KB
```

Рисунок 3. Колонки в фінальному `DataFrame`

Було використано алгоритм випадковий ліс для оцінки ефективності рекламних кампаній по CTR. Результати Random Forest: Оцінка RMSE - 0.08, R2\_Squared - 0.25, оцінка MAE - 0.06.

Для покращення роботи мережі Yolo треба самостійно відмітити наші об'єкти та контури, та натренувати на декількох кроків мережу, щоб покращити в результаті вилучення об'єктів та збільшити вектор ознак на вхід у моделі.

В подальшому можна використовувати більш новіші моделі, які основані на архітектурі трансформера. Також в поточній реалізації додали можливість отримання текстових даних з зображення і в наступних кроках можна використовувати ці тексти додаючи до поточних ознак. Та за допомогою конкатенації всіх ознак та подаючи їх на вхід на нейронну мережу або моделі машинного навчання отримати якісніший результат, оскільки ми матимемо більше інформаційних чинників з нашого ресурсу та покращенні вектори ознак, які можливо будуть краще лінеаризувати простір в якому ми будемо отримувати результати.

#### 4. ВИСНОВКИ

На сьогоднішній день маємо одну з головних цілей маркетингових команд – це ефективне використання рекламних бюджетів. Виникає проблема – частина рекламного бюджету витрачається на неефективну рекламу. Тому є необхідність створення системи, яка буде обчислювати ефективність рекламних носіїв до того, як їх будуть використовувати у рекламних кампаніях.

У ході дослідження було реалізовано таку систему, яка дозволяє виконувати оцінку ефективності рекламних носіїв. Для цього було використано дані реальної продуктової компанії, написана системи отримання цих даних, були застосовані алгоритми комп'ютерного зору та машинного навчання.

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. О.І. Вікарчук, О.О. Калініченко, С.М. Ніколенко, І.О. Поїнта Рекламний креатив, навч. посібник. Київ 2018, 7 с.
2. Jin Chen, Ju Xu, Gangwei Jiang, Tiezheng Ge, Zhiqiang Zhang, Defu Lian, Kai Zheng. Creative Optimization for E-Commerce Advertising. – 2021. Р. 1–3. – [Електронний ресурс]. – Режим доступу : <https://arxiv.org/pdf/2103.00436.pdf>
3. Ted Vrontas. 5 Examples That Show How Machine Learning is Changing Digital Advertising, 2020. [Електронний ресурс]. – Режим доступу : <https://instapage.com/blog/machine-learning-in-advertising>
4. Aurelien Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems. Sebastopol; O'Reilly Media, 2019. 25-290 p.
5. David Hasler, Sabine E. Suesstrunk, Measuring colorfulness in natural images, 2003. [Електронний ресурс]. – Режим доступу : <https://www.spiedigitallibrary.org/proceedings/Download?urlId=10.1117%2F12.477378>

# АНАЛІЗ РОЗПОВСЮДЖЕННЯ COVID-19 НА ТЕРИТОРІЇ УКРАЇНИ З ВИКОРИСТАННЯМ МОДИФІКОВАНОЇ SEIRD-МОДЕЛІ

Клименко А.І.<sup>1</sup>, Подколзін Г.Б.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>asja653@gmail.com [0000-0001-9595-8155],

<sup>2</sup>podkolzin.gleb@iit.kpi.ua [0000-0002-7120-2772]

**Метою роботи є розширення базової SEIRD-моделі за рахунок використання додаткових факторів. Метод дослідження програмна реалізація математичної моделі SEIRD на базі мови програмування: Python. Результатом дослідження є аналіз отриманих результатів з прогнозування розвитку епідемії COVID-19 із використанням статистичних даних перебігу захворювання на території України. Новизна дослідження полягає в урахуванні додаткових факторів при моделюванні таких, як народжуваність, смертність, вакцинованість так невакцинованість населення.**

**Ключові слова: епідеміологічні моделі, COVID-19, математичне моделювання, пандемія.**

## 1. ВСТУП

Протягом усієї історії людське суспільство жило з періодичними епідеміями та пандеміями. Численні спалахи захворювань призводили до смерті, суспільних потрясінь і економічних розладів. Таким чином, прогнозування того, як спалах може прогресувати, важливо для пом'якшення його наслідків, чим і займається область епідеміологічного моделювання.

В епідеміології для опису епідемічних процесів і розуміння динаміки інфекційних захворювань використовуються різні математичні моделі. Багато сучасних математичних моделей, які використовуються для опису передачі інфекційних захворювань, базуються на компартментальних моделях Кермака та Маккендріка.

Епідеміологічні моделі відіграли центральну роль у пандемії COVID-19, особливо коли потрібні були термінові рішення, а доступних доказів було мало. Вони використовувалися для прогнозування еволюції захворювання та для інформування при розробці алгоритму дії запобіжних заходів.

## 2. МОДИФІКОВАНА SEIRD-МОДЕЛЬ

Пандемія COVID-19 вплинула на життя людей у всьому світі. Дослідники пропонують різні моделі та підходи для вивчення того, як можна пом'якшити розповсюдження хвороби. Найпоширенішими у епідеміологічному моделюванні є SIR, SEIR та SEIRD моделі. Найбільш актуальною вважається модель, що охоплює перебіг захворювання протягом усіх його етапів: період здоров'я людини, інкубаційний період, період клінічного перебігу захворювання, одужання та смерті. SIR модель не містить параметра, що відповідає за інкубаційний період захворювання, SIR та SEIR не можуть бути використані, якщо епідемія передбачає смертність та народжуваність. Саме тому для модифікації була обрана SEIRD-модель, що враховує всі необхідні параметри для дослідження всіх етапів захворювання.

Класична SEIRD-модель не розглядає наявність вакцинованих осіб у популяції. В даній роботі була запропонована модифікована SEIRD-модель, яка враховує цей фактор:

$$\left\{ \begin{array}{l} \frac{dS_{unvac}}{dt} = l - \mu S_{unvac} - \frac{S_{unvac}(\beta_{uu}I_u + \beta_{uv}I_v)}{N} \\ \frac{dS_{vac}}{dt} = -\mu S_{vac} - \frac{S_{vac}(\beta_{vu}I_u + \beta_{vv}I_v)}{N} \\ \frac{dE_{unvac}}{dt} = \frac{S_{unvac}(\beta_{uu}I_u + \beta_{uv}I_v)}{N} - (\mu + \alpha_{unvac})E_{unvac} \\ \frac{dE_{vac}}{dt} = \frac{S_{vac}(\beta_{vu}I_u + \beta_{vv}I_v)}{N} - (\mu + \alpha_{vac})E_{vac} \\ \frac{dI_{unvac}}{dt} = \alpha_{unvac}E_{unvac} - (\gamma_{unvac} + \mu + \theta_{unvac})I_{unvac} \\ \frac{dI_{vac}}{dt} = \alpha_{vac}E_{vac} - (\gamma_{vac} + \mu + \theta_{vac})I_{vac} \\ \frac{dR_{unvac}}{dt} = \gamma_{unvac}I_{unvac} - \mu R_{unvac} \\ \frac{dR_{vac}}{dt} = \gamma_{vac}I_{vac} - \mu R_{vac} \\ \frac{dD}{dt} = \theta_{vac}I_{vac} + \theta_{unvac}I_{unvac} + \mu(S_{vac} + S_{unvac} + E_{unvac} + E_{vac} + I_{unvac} \\ + I_{vac} + R_{unvac} + R_{vac}) \end{array} \right.$$

де  $S_{unvac}$  — сприятливі для вірусу невакциновані особи населення, які не інфіковані, але можуть заразитися при контакті з інфікованою особою (невакцинованою або вакцинованою);  $S_{vac}$  — сприятливі для вірусу вакциновані особи населення, які не інфіковані, але можуть заразитися при контакті з інфікованою особою (невакцинованою або вакцинованою);  $E_{unvac}$  — кількість невакцинованих людей із захворюванням у латентному режимі (контактували з інфікованою особою);  $E_{vac}$  — кількість вакцинованих людей із захворюванням у латентному режимі (контактували з інфікованою особою);  $I_{unvac}$  — кількість невакцинованих хворих людей, що передають вірус невакцинованим і вакцинованим сприятливим особам;  $I_{vac}$  — кількість вакцинованих хворих людей що передають вірус невакцинованим і вакцинованим сприятливим особам;  $R_{unvac}$  — кількість невакцинованих перехворівших, що сприйнятливі до повторного зараження, хоча ймовірність менша;  $R_{vac}$  — кількість вакцинованих перехворівших, що сприйнятливі до повторного зараження, хоча ймовірність менша;  $D$  — люди, що померли і від вірусу, і з інших причин;  $\theta_{unvac}$  — смертність від вірусу інфікованих невакцинованих;  $\theta_{vac}$  — смертність від вірусу інфікованих вакцинованих;  $\beta_{uu}$  — ймовірність передачі вірусу від інфікованих невакцинованих до невакцинованих;  $\beta_{uv}$  — ймовірність передачі вірусу від інфікованих невакцинованих до вакцинованих;  $\beta_{vu}$  — ймовірність передачі вірусу від інфікованих вакцинованих до невакцинованих;  $\beta_{vv}$  — ймовірність передачі вірусу від інфікованих вакцинованих до вакцинованих;  $\alpha_{unvac}$  — ймовірність переходу захворювання із латентної фази у відкриту у невакцинованих;  $\alpha_{vac}$  — ймовірність переходу захворювання із латентної фази у відкриту у вакцинованих;  $\gamma_{unvac}$  — одужання інфікованих невакцинованих від вірусу;  $\gamma_{vac}$  — одужання інфікованих вакцинованих від вірусу;  $\mu$  — смертність не від інфекції;  $l$  — народжуваність.

Основна відмінність цієї моделі від класичної SEIRD-моделі - це поділення популяції на вакцинованих та невакцинованих. Ймовірність зараження вакцинованих осіб ( $S_{vac}$ ) набагато нижча, ніж у невакцинованих осіб ( $S_{unvac}$ ). Хворі вакциновані особи ( $I_{vac}$ ) менш заразні, у них менше ймовірність померти, ніж у невакцинованих інфекційних осіб ( $I_{unvac}$ ).

### 3. РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ З УКРАЇНСЬКИМИ ДАНИМИ

Змоделюємо ситуацію з COVID-19 в Україні. Для розрахування коефіцієнтів скористуємося інформацією за 2021 рік.

Кількість українців за 2021 (без урахування окупованих територій) становить 41 588 354 людей [1, 3]. Смертність за 2021 рік становить 714 263 людей [2, 3], з яких COVID-19 становить 86 015 випадків [2]. Народжуваність за 2021 рік становить 271 983 дитини [2, 3].

Кількість усіх смертей за день у середньому становить  $714\,263/365 = 1\,956.88$  людей. Тобто, за день від загальної кількості населення помирають  $1\,956.88/41\,588\,354 = 0.0000470537$ . У цю цифру також входить смертність від COVID-19. Смертність від COVID-19 становить  $(86\,015/365)/41\,588\,354 = 0.0000056664$ .

На жаль, статистики смертності вакцинованих та невакцинованих саме по Україні не було знайдено. Проте можна розрахувати цей коефіцієнт, якщо врахувати, що дані, що були зазначені в інтерв'ю з професоркою Школи громадської охорони здоров'я Леанни Вен [4], правдиві: що вакциновані люди мають у шість разів менше шансу заразитися, ніж у невакцинованих, і мають ймовірність померти від коронавірусу у 11 разів менше. Леанна Вен посилається на дослідження CDC (Centers for Disease Control) [5].

У цьому випадку смертність вакцинованих за день становить 0.0000004722, а невакцинованих 0.0000051942. Відповідно, маємо  $\Theta_{unvac} = 0.0000051942$ ,  $\Theta_{vac} = 0.0000004722$ .

Відповідно, смертність за день не від COVID-19 становить  $0.0000470537 - 0.0000056664 = 0.0000413872$ , тобто  $\mu = 0.0000413872$ . Народжуваність за день становить  $(271\,984/365)/41\,588\,354 = 0.0000179176$ , тобто,  $l = 0.0000179176$ .

Так як для модулювання треба і вакцинована, і невакцинована популяція, треба навести дані стосовно цього. Станом на 2021 рік в Україні вакциновано двома вакцинами 15 201 112 людей [6]. Тобто,  $15\,201\,112/41\,588\,354 = 0.3655136724$ . У даному дослідженні моделюється популяція, що складається зі ста людей, серед яких відповідно є і вакциновані, і невакциновані.

З них буде два хворих (один вакцинований та один невакцинований), 35.5279289582 — це ще не хворілі вакциновані, а 61.8132941437 — ще не хворілі невакциновані.

Інкубаційний період — це кількість днів між моментом зараження чимось і моментом появи симптомів. Для розрахунку коефіцієнту, що відповідає за швидкість переходу вірусу із латентного періоду у повноцінний інфікований, наступною формулою:

де  $\alpha$  — швидкість переходу захворювання із латентної фази у відкрити;  $T_{inc}$  — середній час інкубаційного періоду вірусу.

Віруси постійно змінюються, що іноді призводить до появи нових штамів. Різні штами COVID-19 можуть мати різні інкубаційні періоди. У середньому симптоми з'явилися у новоінфікованої людини приблизно через 5.6 дня після контакту [7]. Тобто,  $T_{inc} = 5.6$ , відповідно,  $\alpha = 1/5.6 = 0.17858$ . Відповідно до досліджень, інкубаційний період у вакцинованих та невакцинованих складає однакову кількість днів [8], тобто  $\alpha_{vac} = \alpha_{unvac} = 0.17858$ .

Дослідження показали, що організму може знадобитися 2 тижні, щоб подолати легку хворобу, або до 6 тижнів у важких або критичних випадках [9]. Отже, візьмемо час одужання за два тижні у середньому.

Розрахуємо  $\gamma$  відповідно до формули розрахунку коефіцієнту швидкості одужання:

$$\gamma = \frac{1}{T_{rec}}$$

де  $\gamma$  — коефіцієнт одужання інфікованих людей від вірусу;  $T_{rec}$  — середній час одужання.

Тобто,  $T_{rec} = 14$  днів. Відповідно,  $\gamma = \gamma_{unvac} = 1/14 = 0.0714$ . У випадку, коли населення поділяється на вакцинованих та невакцинованих, час одужання буде різний. Дослідження 2021 року, що було проведено Центрами з контролю та профілактики захворювань, зазначено, що вакциновані учасники проводили в ліжку в середньому від двох до шести днів хвороби менше, ніж невакциновані [10]. Візьмемо в середньому, що вакциновані хворіють на шість днів менше, тобто  $\gamma_{vac} = 1/8 = 0.125$ .

Розрахуємо тепер ймовірності передачі захворювання. Для розрахування даних для вакцинованих людей, знадобиться статистика по ефективності вакцини (візьмемо вакцину Pfizer). Відповідно до даних [11], Pfizer має захист від легкої форми COVID-19 95%. Тобто, ймовірність захворіти, якщо ти вакцинований,  $p_{заразитися, якщо ти вакцинований} = 0.05$ .

Також треба зазначити, що вакциновані люди менш схильні до передачі хвороби, навіть якщо вони інфікуються. На листопадовій прес-конференції Тедрос Гебреесус, генеральний директор ВООЗ, заявив, що вакцини на 60 відсотків захищають від поширення вірусу до появи дельта-варіанту [12][13]. Тобто  $p_{передати хворобу, якщо ти вакцинований} = 0.4$

Відповідно до того ж дослідження [12][14], у вакцинованих людей у десять разів менше шансів заразитися [13], а ймовірність заразити мене вдвічі менша, судячи з наведених вище цифр.

Тобто,  $p_{заразитися, якщо ти невакцинований} = p_{захворіти, якщо ти вакцинований} \times 10 = 0.5$ , а  $p_{передати хворобу, якщо ти невакцинований} = 0.4 \times 2 = 0.8$ .

Тепер можемо розрахувати коефіцієнти передачі захворювання.

$$\beta_{uu} = p_{передати хворобу, якщо ти невакцинований} \times p_{заразитися, якщо ти невакцинований} = 0.8 \times 0.5 = 0.4$$

$$\beta_{uv} = p_{передати хворобу, якщо ти невакцинований} \times p_{заразитися, якщо ти вакцинований} = 0.8 \times 0.05 = 0.04$$

$$\beta_{vu} = p_{передати хворобу, якщо ти вакцинований} \times p_{заразитися, якщо ти невакцинований} = 0.4 \times 0.5 = 0.2$$

$$\beta_{vv} = p_{передати хворобу, якщо ти вакцинований} \times p_{заразитися, якщо ти вакцинований} = 0.4 \times 0.05 = 0.02$$

Змоделюємо з цими параметрами модель. На рис. 1 можна побачити результати для ста днів:

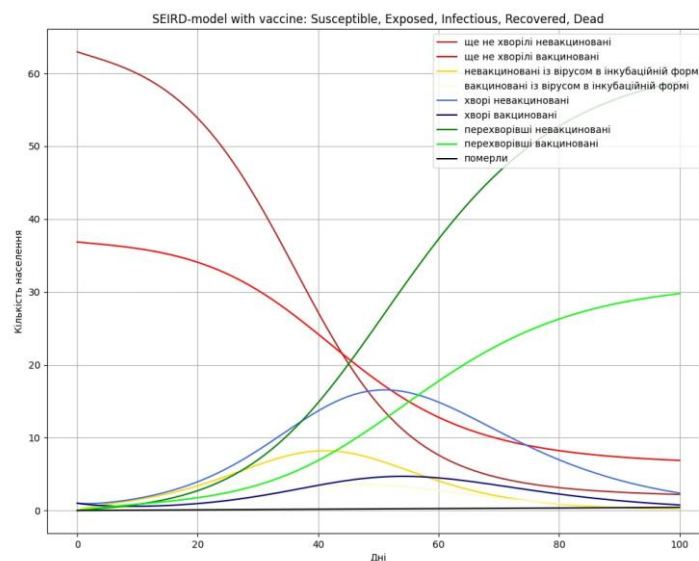


Рисунок 1. Модифікована SEIRD-модель для 100 днів

З рис. 1 можна побачити, як кількість ще не хворівших людей зменшується і одночасно кількість перехворівших зростає протягом часу (100 днів). Кількість перехворівших невакцинованих зростає швидше, і з урахуванням вакцинації [10] результат є достовірним. Кількість померлих відносно повільно зростає. Кількість не хворівших невакцинованих спадає швидше, бо у них ймовірність підхопити хворобу вище, у той час як вакциновані хворіють повільніше.

Якщо ж розглянути більший часовий проміжок (наприклад, 8 000 днів), то з рис. 2 можна побачити, як зростає кількість померлих, і зменшується для всіх інших груп дослідження. Наприклад, кількість перехворівших людей зменшується, бо модифікована SEIRD-модель враховує також природну смертність, у яку у даному випадку не входить смертність від COVID-19.

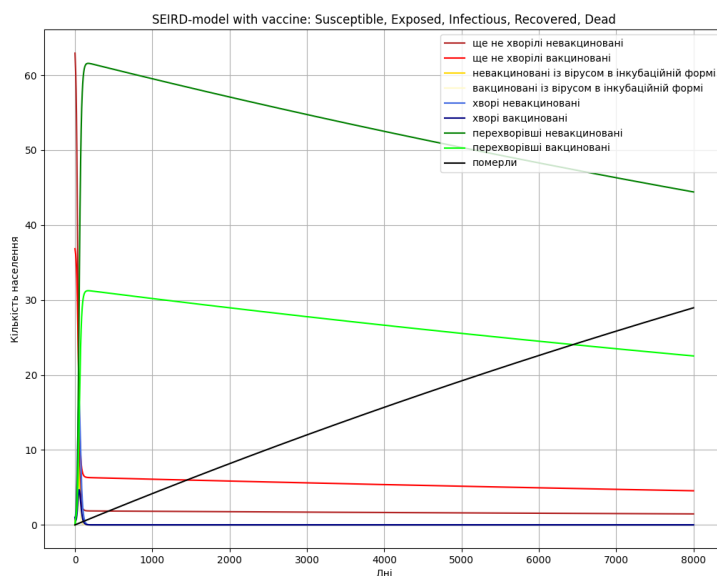


Рисунок 2. Модифікована SEIRD-модель для 8000 днів

Таке зростання смертності відносно всіх інших показників на рис.2 обумовлюється тим, що в Україні народжуваність менше, ніж смертність [15]. Якщо ж збільшити коефіцієнт народжуваності, щоб він перевищив смертність, то ситуація буде виглядати інакше (Народжуваність  $l = 0,0500181008$ ) (Рис. 3):

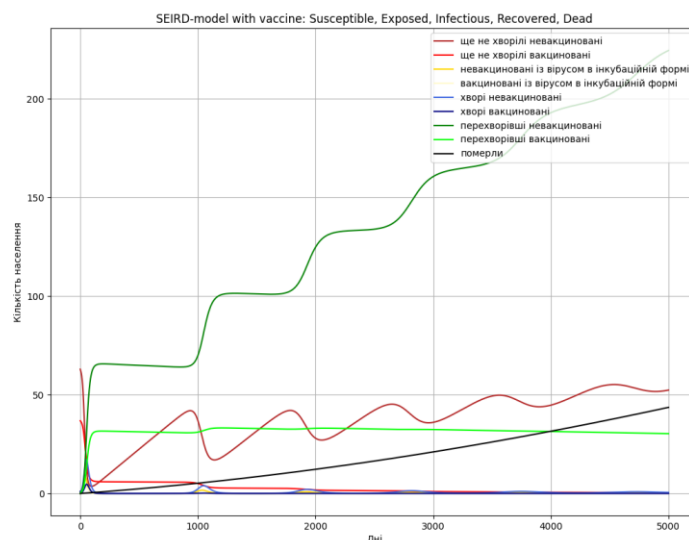


Рисунок 3. Модифікована SEIRD-модель для 5000 днів з народжуваністю  $l = 0,0500181008$



Згідно рис. 3, зростає лише кількість ще не перехворілих невакцинованих, бо дана модель не передбачає вакцинацію і відповідно перехід від невакцинованих до вакцинованих. Уся популяція, що народжується, за замовчуванням є невакцинованою. З рис. 3 ще можна побачити, як періодично йде процес захворювання: на тисячному і двохтисячному дні можна побачити хвилі з хворих невакцинованих, і відповідно періодами зменшуються ще не перехворілі невакциновані.

#### 4. ВИСНОВКИ

За результатами даної роботи було проаналізовано та досліджено модифіковану SEIRD-модель, на базі якої було зроблено наступні висновки, що корелюються з реальними статистичними даними України, а саме:

- Кількість захворілих невакцинованих в період піку епідемії приблизно в 3.5 рази більше, ніж захворілих вакцинованих;
- Смертність в Україні перевищує народжуваність;
- Розвиток епідемії має періодичний характер з тенденцією до затухання з часом.

Отримані результати підтверджують коректність роботи розробленої моделі, яка дозволяє моделювати ситуацію розповсюдження захворювання COVID-19 з урахуванням таких додаткових факторів, як розподілення населення на вакцинованих та невакцинованих, та використання додаткових коефіцієнтів, які дозволяють також врахувати вплив народжуваності та смертності населення. Використання перелічених додаткових факторів при моделюванні дозволяє прогнозувати більш реалістичну картину як розповсюдження захворювання, так і прорахування піків епідемії та її спадів. Результати даного дослідження можуть використовуватися в якості прогнозу розвитку епідемії для організації відповідних запобіжних заходів в медичних закладах, громадських місцях та ін.

Представлені розрахунки на базі модифікованої SEIRD-моделі використовували статистичні дані з українських джерел.

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Населення України. URL: <https://index.minfin.com.ua/ua/reference/people/> (дата звернення: 18.11.2022)
2. Смертність України за 2021. URL: <https://index.minfin.com.ua/ua/reference/people/deaths/2021/> (дата звернення: 18.11.2022)
3. Держстат України. Демографічний щорічник населення України за 2021 рік: статистичний збірник. Київ. 2022. с.27, 53.
4. Why vaccinated people dying from Covid-19 doesn't mean the vaccines are ineffective. URL: <https://edition.cnn.com/2021/10/18/health/covid-19-vaccination-colin-powell-death-wellness/index.html> (дата звернення: 19.11.2022)
5. COVID-19-associated hospitalizations among vaccinated and unvaccinated adults  $\geq 18$  years – COVID-NET, 13 states, January 1 – July 24, 2021. URL: <https://www.medrxiv.org/content/10.1101/2021.08.27.21262356v1.full#T3> (дата звернення: 13.11.2022)
6. 6 Вакцинація в Україні. URL: <https://index.minfin.com.ua/ua/reference/coronavirus/vaccination/ukraine/> (дата звернення: 13.11.2022)
7. Coronavirus Incubation Period. URL: <https://www.webmd.com/lung/coronavirus-incubation-period#1> (дата звернення: 18.11.2022)
8. SARS-Cov-2 incubation period according to vaccination status during the fifth COVID-19 wave in a tertiary-care center in Spain: a cohort study. URL:

<https://bmcinfectdis.biomedcentral.com/articles/10.1186/s12879-022-07822-4>

<https://www.webmd.com/lung/coronavirus-incubation-period#1> (дата звернення: 18.11.2022)

9. Coronavirus Recovery. URL: <https://www.webmd.com/lung/covid-recovery-overview#2> (дата звернення: 15.11.2022)

10. It's not just severity—the types of Covid symptoms you get depend on the vaccines you've received, new data says. URL: <https://www.cnbc.com/2022/10/26/covid-symptoms-differ-based-on-vaccination-status-zoe-health-study.html> (дата звернення: 14.11.2022)

11. COVID-19 vaccine efficacy explained. URL: <https://www.nebraskamed.com/COVID/covid-19-vaccine-efficacy-explained> (дата звернення: 12.11.2022)

12. Modelling of COVID-19. URL: <https://www.doherty.edu.au/our-work/institute-themes/viral-infectious-diseases/covid-19/covid-19-modelling/modelling> (дата звернення: 10.11.2022)

13. Vaccinated NSW residents to be allowed into Victoria as the state records 2,179 COVID-19 cases and six deaths. URL: <https://www.skynews.com.au/australia-news/coronavirus/watch-live-vic-health-officials-to-provide-covid19-update/news-story/863cdc24d57dd787251a8cfff26b5ec5> (дата звернення: 10.11.2022)

14. Your unvaccinated friend is roughly 20 times more likely to give you COVID. <https://theconversation.com/your-unvaccinated-friend-is-roughly-20-times-more-likely-to-give-you-covid-170448> (дата звернення: 12.11.2022)

15. Народжуваність в Україні продовжує падати і готується поставити новий антирекорд останніх 30-ти років. URL: <https://opendatabot.ua/analytics/deropulation-2021> (дата звернення: 10.11.2022)

# ЗАСТОСУВАННЯ ФРАКТАЛЬНИХ МОДЕЛЕЙ НЕОБМЕЖЕНОЇ ВАРІАЦІЇ ДО АНАЛІЗУ ЧАСОВИХ РЯДІВ

Кравченко А.А.<sup>1</sup>, Бондаренко В.Г.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>kravchenkoann1602@gmail.com, <sup>2</sup>bondareng@gmail.com

**Прогнозування майбутнього завжди було актуальною проблемою, не тільки через невизначеність, але й через те що для багатьох компаній прогнозування майбутніх цін акцій чи цінних паперів є невід’ємною складовою бізнес процесу. Метою роботи є аналіз часових рядів за допомогою моделі фрактального броунівського руху, їх основних властивостей та методів прогнозування. Результатом роботи є порівняльний аналіз прогнозу, побудованого за допомогою фрактального броунівського руху та класичної моделі для роботи з часовими рядами. У роботі було використано теоретичні та емпіричні методи дослідження.**

**Ключові слова:** часовий ряд, фрактальний броунівський рух, прогноз, авторегресійні моделі, аналіз.

## 1. ВСТУП

Математичні моделі національної, регіональної та глобальної економіки є дуже потужними інструментами для теоретичних досліджень реальних макроекономічних процесів. Важливою перевагою макроекономічних моделей є їхня доступність для детального математичного аналізу та можливість їх вивчення за допомогою невеликої кількості вхідних даних [1].

Особливо цікавими є моделі що описують процеси з довгою пам’яттю, в багатьох випадках це моделі, що основані на дробовому численні, адже якщо брати моделі на основі інтегральних або диференціальних функцій цілого порядку, вони будуть описувати економічні процеси, в яких агенти фактично мають тотальну амнезію. Процеси з довгою пам’яттю можна описати як статистичні методи, що дозволяють аналізувати поведінку часових рядів. Через те що процеси з довгою пам’яттю мають ряд нестандартних властивостей, застосування вже звичних моделей Маркова та АРКС в більшості випадків не будуть давати бажаних результатів [2].

## 2. ЗАСТОСУВАННЯ ФРАКТАЛЬНИХ МОДЕЛЕЙ ДЛЯ МОДЕЛЮВАННЯ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

Оскільки при побудові стохастичних моделей часових рядів в ряді випадків доцільно використовувати в якості базового процесу фрактальний броунівський рух, то він і є метою цього дослідження. Модель фрактального броунівського руху, це процес  $B_H t, t \geq 0, B_H 0 = 0$ , де  $H$  – параметр Харста, що визначається як гаусівський випадковий процес з нульовим середнім та кореляційною функцією (1).

$$R(t, s) = E B_H(t) B_H(s) = \frac{1}{2} (t^{2H} + s^{2H} - |t - s|^{2H}) \quad (1)$$

Варто також зауважити, що значення параметру Харста визначає поведінку часового ряду. Цей скаляр представляє відносну тенденцію часового ряду або до сильного регресу до

середнього, або до групування у певному порядку, а також визначає фрактальні властивості процесу. Оскільки параметр Харста є мірою довгострокової пам'яті часового ряду, то на основі його значення часовий ряд можна класифікувати як:

-  $H < 0.5$  – антиперсистентний ряд. Чим ближче значення до 0, тим сильнішою є реверсія до середнього. На практиці це означає, що за високим значенням слідує назьке значення, і навпаки.

-  $H = 0.5$  – геометричне випадкове блукання, тобто стандартний вінерівський процес.

-  $H > 0.5$  – трендовий, або ж стійкий ряд. Чим ближче значення параметру Харста до 1, тим сильніший тренд, тобто на практиці, за високим значенням йде ще більше.

В силу автономності фрактального броунівського руху [5], параметр Харста можна знайти за допомогою однокрокового коефіцієнта кореляції приростів, визначених рівністю (2).

$$\rho_1 = 2^{2H-1} - 1 \quad (2)$$

Також варто відмітити, що прогноз значень фрактального броунівського руху можливий лише для персистентного випадку, тобто коли параметр Харста є більшим за 0.5. З основними характеристиками моделі фрактального броунівського руху детальніше можна ознайомитися в роботах [3,4].

Наступною задачею є оцінка параметрів фрактального броунівського руху. Тут варто зазначити, що основними об'єктами дослідження і оцінки є не значення фрактального броунівського руху, а саме їх прирости  $y_k = X \frac{k}{n} - X \frac{k-1}{n}$ ,  $k = 1, 2, \dots, n$ , де  $X t = \sigma B_H(t)$ , що утворюють гаусівську послідовність. Постає гіпотеза: чи можна вважати побудовані раніше прирости, приростами фрактального броунівського руху? Перевірка цієї гіпотез полягає у визначенні значень наступних статистик (3):

$$\begin{aligned} A_n &= \frac{1}{n} v_k y_k^3, \text{ при } H \in 0, \frac{1}{2}, \\ B_n &= \frac{1}{n^{1+H}} v_k^2 y_k^3, \text{ при } H \in 0, \frac{1}{2}, \\ D_n &= \frac{1}{n^{2H}} v_k y_k^3, \text{ при } H \in \frac{1}{2}, 1, \\ F_n &= \frac{1}{n^H} y_k^3, \text{ при } H \in \frac{1}{2}, 1, \end{aligned} \quad (3)$$

де  $v_k$  визначаються як сума приростів  $v_k = \sum_{j=1}^{k-1} y_j$ .

Оскільки в даному дослідженні розглядаються ряди з довгою пам'яттю, то гіпотеза вважається прийнятною, якщо статистики  $D_n$  та  $F_n$  задовольняють наступним умовам:

$$0 < D_n < 6, -8.4 < F_n < 8.4 \quad (4)$$

Після того, як ми прийняли гіпотезу, що прирости  $y_k$  дійсно є приростами фрактального броунівського руху, тепер можемо побудувати прогноз на їх основі. Як відомо, найкращим прогнозом, або ж екстраполяцією траєкторії випадкового процесу  $X(t)$  у точці  $t$  є величина  $X(t + \tau)$ , що задовільняє наступній умові  $X t + \tau = E(X(t + \tau)|D_\tau)$ , де  $D_\tau$  -  $\sigma$ -алгебра, утворена випадковими величинами  $X t, 0 \leq t \leq \tau$ . Загалом у випадку фрактального броунівського руху визначення цього умовного середнього є доволі складною задачею, але в деяких випадках допускаються спрощення. В нашому випадку розглянемо ряд, побудований на основі гаусівського вектора:

$$\begin{aligned} \xi &= \xi_1, \dots, \xi_n, \xi = \xi_{tr}, \xi_{est}, \\ \dim \xi_{tr} &= m, \dim \xi_{est} = n - m, \\ \xi &\approx \aleph 0, S, \xi_{tr} \approx \aleph 0, A, \xi_{est} \approx \aleph 0, D \end{aligned}$$

Тобто в такому випадку кореляційний оператор  $\xi$  це блочна матриця наступного вигляду:

$$S = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

де елементи матриці  $B, C$  є взаємною кореляцією координат  $\xi_{tr}$  та  $\xi_{est}$ . Якщо ж розглядати перший вектор як вектор, що спостерігається, а другий вектор – той, що оцінюється відповідно, то за рахунок гаусовості найкращий прогноз збігатиметься з лінійною оцінкою:

$$\xi_{est} = E \xi_{est} \xi_{tr} = CA^{-1}\xi_{tr} \quad (5)$$

Тобто в нашому випадку, оскільки

$$\xi_k = y_k = B_H \frac{k+1}{n} - B_H \frac{k}{n},$$

Маємо змогу розбити прирости  $y_k$  на навчальну  $y_1, \dots, y_m$  та прогнозовану  $y_{m+1}, \dots, y_n$  вибірки, та за рухонек гіпотези, що ці прирости є приростами фрактального броунівського руху з відомою кореляційною функцією (1), формула (5) набуває наступного вигляду:

$$y_{m+j} = \sum_{k=1}^m \sum_{i=1}^m \left( \frac{m+j-k+1}{2} {}^{2H} + \frac{m+j-k-1}{2} {}^{2H} - \frac{m+j-k}{2} {}^{2H} \right) a^{ki} y_i \quad (6)$$

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для дослідження було використано дані щодо цін акцій відомої соціальної мережі “Twitter Inc” розмірністю 372 значення, що відповідає відрізку ледь більшому за календарний рік. Для застосування моделі фрактального броунівського руху була проведена необхідна первинна обробка даних, а саме приведення рядку динаміки зміни цін акцій з нестационарного ряду до стаціонарного, шляхом визначення та віднімання тренду, для досягнення нового ряду з нульовим середнім, а також переходом до приростів оброблених значень, аби досягнути однієї з умов стаціонарності – коваріація між будь-якими двома точками на однаковому проміжку є констатним значенням. Задля того, аби підтвердити гіпотезу, що побудований «новий» ряд можна описати за допомогою фрактального броунівського руху за допомогою формули (2) був обрахований параметр Харста, що показав, що даний ряд є часовим рядом з довгою пам'яттю, а також значення статистик (3), що належать довірчому інтервалу (4). Після цього припускаючи, що значення ряду утворюють гаусівський вектор(що було визначено за допомогою середнього значення та гістограми приростів), за рахунок його лінійності було побудовано прогноз, який в силу гаусовості збігається з лінійною оцінкою, за формулою (6). За допомогою цього були отримані результати прогнозу (Рис. 1).

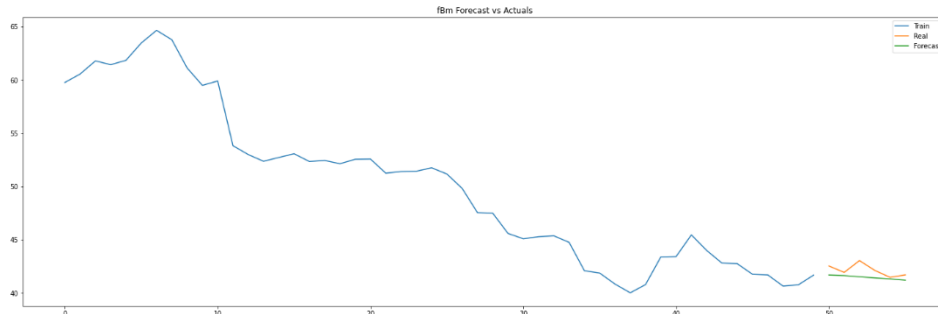


Рисунок 1. Графічне зображення прогнозу майбутніх значень за допомогою фрактального броунівського руху

Аби оцінити якість побудованого прогнозу, в якості порівняльного методу, також було побудовано прогноз для того самого вікна спостережень за допомогою моделі АРІКС [6], а саме АРІКС(1, 1, 1) (Рис. 2).

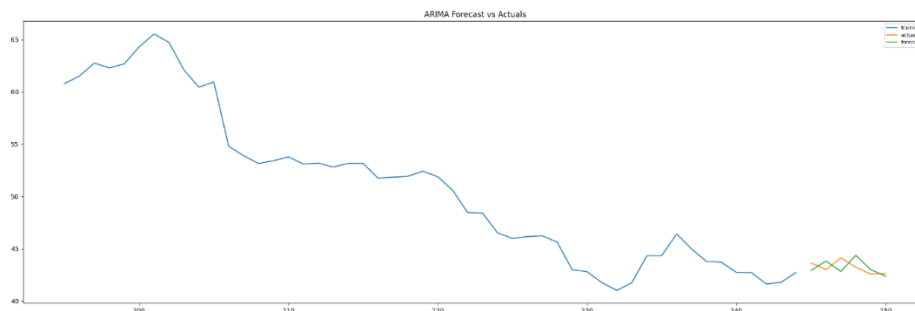


Рисунок 2. Графічне зображення прогнозу майбутніх значень за допомогою моделі АРІКС(1, 1, 1)

Результати порівняльного аналізу якості роботи прогнозування обома методами наведено у таблиці нижче (Табл. 1).

Таблиця 1. Характеристики адекватності моделей та оцінки якості прогнозів

Тип моделі	Характеристики моделі					
	$R^2$	$SSE$	$DW$	$MSE$	$MAE$	$Theil\ index$
ARIMA(1, 1, 1)	0.763	4.419	3.118	0.736	0.017	0.0005
fBm	-	3.882	-	0.647	0.015	-

Можна помітити, що за характеристиками, що відповідають за оцінку якості моделі, тобто середньою квадратичною похибкою, середньою абсолютною похибкою та накопиченням квадратів похибок ( $MSE$ ,  $MAE$  та  $SSE$ ) більш якісним прогнозом є прогноз за допомогою фрактального броунівського руху. Інші ж характеристики коефіцієнт детермінації  $R^2$ , індекс Дарбіна-Вотсона  $DW$  та індекс Тейла, що відповідають за оцінку адекватності побудованої моделі, обраховані лише для моделі АРІКС, адже вони застосовуються лише для лінійних моделей і були наведені для того, аби показати, що модель АРІКС, з якою порівнюється робота моделі фрактального броунівського руху, дійсно є дійсно валідною моделлю для порівняння.

#### 4. ВИСНОВКИ

З кожним днем економічні та фінансові системи стають все більш невизначеними та залежними від подій у зовнішньому світі, адже 21 сторіччя це вік персоналій, а не продуктів, тому багато економічних процесів, таких як ціна акцій чи цінних паперів, залежать не лише від успіхів самої компанії, але й від дій та висловлювань облич, що представляють цю компанію, а отже мають великий і доволі довгостроковий вплив на успіхи компанії.

Тому одним з найбільш ефективних методів у цій галузі є аналіз, оснований на моделях часових рядів із довгою пам'яттю. У ході дослідження на практичному прикладі реальних цін акцій протягом календарного року, було проведено аналіз за допомогою моделі фрактального броунівського руху. А також для порівняльного аналізу для того самого набору даних була використана одна з класичних лінійних моделей - АРІКС.

Результати дослідження показали, що навіть при точно підібраних порядках моделі АРІКС, та побудованого на її основі прогнозу, більш ефективним на заданій вибірці вхідних

даних все ж виявилася апроксимація майбутніх значень за допомогою моделі фрактального броунівського руху.

### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. R.G.D. Allen, *Mathematical Economics*, 2nd edn., Macmillan, London, 1960, 812p., ISBN 978-1-349-81549-4, 10.1007/978-1-349-81547-0.
2. W.Palma *Long-Memory Time Series: Theory and Methods*, Willey-InterScience, Hoboken, 2007, 304p., ISBN 978-0-470-11402-5.
3. F. Biagini, Y. Hu, B. Øksendal, and T. Zhang, “Stochastic Calculus for Fractional Brownian Motion and Applications”, Springer, 329 p., 2013. doi: 10.1007/978-1-84628-797-8.
4. R.F. Peltier and J. Levy Vehel, “A new method for estimating the parameter of fractional Brownian motion”, *Rapport de recherché de l’INRIA*, no. 2396, 27 p., 1994.
5. I. Nourdin, “Noncentral convergence of multiple integrals”, *Ann. Probab.*, vol. 37,no. 4, pp. 1412–1426, 2009. doi: 10.1214/08-AOP435.
6. Brockwell, P. J., & Davis, R. A. (2016). *Introduction to time series and forecasting* (3rd ed). New York, USA: Springer

# ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ ПОКАЗНИКІВ СМЕРТНОСТІ ВІД ЗАБРУДНЕННЯ ПОВІТРЯ

Кулина А.П.<sup>1</sup>, Данилов В.Я.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>kulyna.anisiia@lil.kpi.ua

<sup>2</sup>danilov1950@ukr.net [0000-0003-3389-3661]

**Задача побудови моделей та прогнозу є важливою складовою розв'язання проблеми забруднення та смертності у зв'язку із нею. Маючи модель, що здатна оцінити ризик смертності, можна зрозуміти доцільність покращення природних умов, відстеження та контролю викидів у атмосферне повітря. Для побудови таких моделей краще за все використовувати методи машинного навчання. Метою роботи є порівняння методів системи прогнозування показників смертності від забруднення повітря. Об'єкт дослідження - набір реальних даних, що описує рівень забруднення та рівень смертності від забруднення.**

**Ключові слова:** машинне навчання, показник смертності, прогноз, забруднення повітря, PM<sub>2.5</sub>, CO<sub>2</sub>.

## 1. ВСТУП

Глобальне потепління, вирубка лісів, деградація екосистем і втрата життєво важливих середовищ – це лише деякі з багатьох проблем, які постали перед людством через широкомасштабну урбанізацію, економічний та промисловий розвиток. Немає сумніву, те, що відбувається сьогодні, визначатиме, наскільки ефективно ми будемо справлятися з ризиками та потрясіннями в майбутньому. І все ж забруднення повітря все ще залишається однією з найбільших проблем сьогодення, яка є наслідком вищеперерахованих чинників. Ці чинники безумовно полегшують умови існування людини, але є «але». Навколишнє середовище об'єднує у собі велику різноманітність важливих складових, однією з яких є атмосферне повітря [1]. Його забруднення підвищує ризик розвитку неінфекційних захворювань. Ці захворювання поділяються на три основні категорії: серцево-судинні захворювання, респіраторні захворювання та рак, які є основними причинами глобальної смертності. Окрім цього, це також збільшує ризик захворювань від інфекцій нижніх дихальних шляхів, передчасних пологів та інших причин смерті дітей і немовлят [2].

Враховуючи серйозність негативного впливу забруднення повітря, задача пошуку, аналізу та порівняння методів прогнозування показників смертності є актуальною проблемою. Маючи модель, що здатна оцінити ризик смертності, можна зрозуміти доцільність покращення природних умов, лікування хворих, розширити спектр прогнозу в дослідженнях для того, щоб зрозуміти важливість відстеження та контролю викидів у атмосферне повітря, а також способи їх зниження. Оскільки використання машинного навчання для прогнозування смертей від конкретних захворювань викликає все більший інтерес у науковій літературі [3], метою даного дослідження є порівняльний аналіз різних моделей машинного навчання (МН) для прогнозування показників смертності від забруднення повітря.



## 2. ПОСТАНОВКА ЗАДАЧІ

Метою роботи є розробка програмного забезпечення системи прогнозування показників смертності від забруднення повітря. Буде виконано аналіз і обробку набору даних та порівняння обраних методів машинного навчання для прогнозування показників смертності від забруднення повітря. Методи машинного навчання будуть застосовуватися на наборі реальних даних. За вхідними параметрами необхідно буде визначити рівень смертності. Об'єктом дослідження є набір реальних даних, що описує рівень забруднення (показник PM2.5 та CO2) та рівень смертності від забруднення.

## 3. ОПИС НАБОРУ ДАНИХ

У задачах машинного навчання вибір та правильна попередня обробка набору даних є вкрай важливими для отримання точних результатів. У даній роботі використовуються наступні набори даних: «Death Due to Air Pollution» (зібрано організацією Our World in Data), «PM2.5 Global Air Pollution» (The World Bank Group), «CO2 Emission by countries» (зібрано організацією Our World in Data). Ці дані було об'єднано в один спільний набір даних. Кінцевий датасет містить 6468 записів. Параметрами є країна, рік, рівень CO2 та PM2.5. Вихідним параметром є показник смертності населення. Дані продемонстровано на рис. 1.

	Entity	Year	Air pollution (total) (deaths per 100,000)	CO2 emission (Tons)	PM25
0	Afghanistan	1990	299.477309	59182404.0	0.000000
1	Afghanistan	1991	291.277967	61609770.0	0.000000
2	Afghanistan	1992	278.963056	62989232.0	0.000000
3	Afghanistan	1993	278.790815	64322287.0	0.000000
4	Afghanistan	1994	287.162923	65603889.0	0.000000
...	...	...	...	...	...
6463	Zimbabwe	2013	143.850145	701597967.0	23.111088
6464	Zimbabwe	2014	138.200536	713503654.0	22.805950
6465	Zimbabwe	2015	132.752553	725729475.0	21.733709
6466	Zimbabwe	2016	128.692138	736467042.0	21.726947
6467	Zimbabwe	2017	125.028843	746048675.0	22.251671

6468 rows × 5 columns

Рисунок 1. Вигляд датасету

Також нижче представлено на рис. 2 рівень смертності в усьому світі від забруднення повітря.

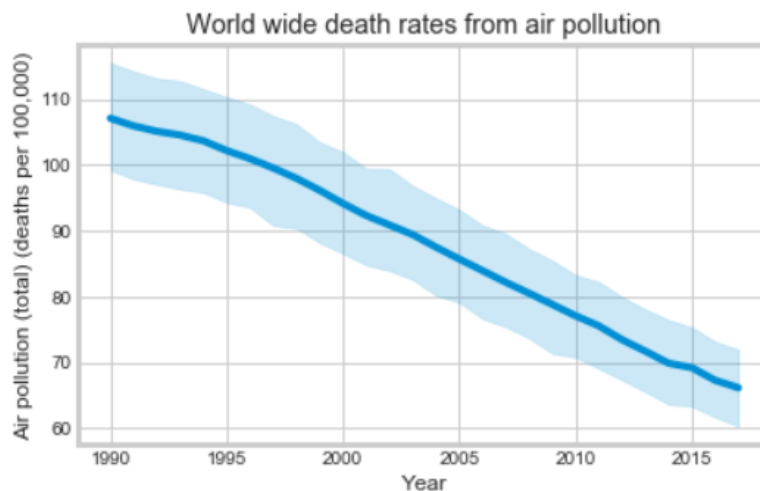


Рисунок 2. Рівень смертності в усьому світі від забруднення повітря

Для оцінки роботи моделей дані були випадковим чином розділені на дві вибірки. Поділ даних на навчальну та тестову вибірки було виконано у співвідношенні 70/30.

#### 4. ОПИС АЛГОРИТМУ

Для виконання прогнозу смертності від забруднення повітря було обрано методи машинного навчання. Машинне навчання (МН) — це область штучного інтелекту, заснована на обчислювальних алгоритмах, які дозволяють комп'ютерам навчатися безпосередньо з даних, без явного програмування [4]. Алгоритми МН за прийнятний час аналізують великі обсяги даних, що представлені багатьма характеристиками (прогнозуючими змінними), і можуть виявляти та моделювати складні зв'язки між даними [5]. Таким чином, МН можна визначити як набір інструментів і методів для ідентифікації шаблонів у даних. Ці шаблони можна використовувати для покращення розуміння конкретної проблеми, прогнозування майбутнього та сприяння прийняттю рішень. Кажуть, що алгоритм навчається на даних. Використовуючи різні конфігурації, мета полягає в тому, щоб знайти модель, яка краще описує набір даних [6].

Моделі прогнозування в медицині історично спиралися на логістичну регресію [7]. Однак більш гнучкі методи машинного навчання призвели до розробки високоточних моделей. Наприклад, нещодавні дослідження продемонстрували, що алгоритми машинного навчання, такі як випадкові ліси та глибокі нейронні мережі, можуть передбачати гострі стани у пацієнтів точніше, ніж традиційна логістична регресія [8,9]. Тим не менш, важко заздалегідь визначити, який метод буде найкращим для конкретної проблеми [10].

Саме тому з метою виявлення оптимального методу для даної проблеми, порівняно кількох алгоритмів машинного навчання, які коротко описані нижче. Проведено порівняння

Лінійна регресія.

Лінійна регресія – це алгоритм машинного навчання, заснований на керованому навчанні, виконує завдання регресії. Регресія моделює цільове прогнозоване значення на основі незалежних змінних. Здебільшого використовується для з'ясування зв'язку між змінними та прогнозом.

eXtreme Gradient Boosting.

Метод посилення градієнту використовує дерева рішень як базові оцінювачі. Деревя будуються з використанням залишків, а не фактичних міток класу. Тобто, базовими оцінювачами в цих алгоритмах є дерева регресії, а не дерева класифікації. Це тому, що залишки неперервні, а не дискретні. Метод посилення градієнту будує дерева послідовно,

щоб кожне дерево покращувало модель [11]. У даній роботі використовувалася популярна реалізація GBM під назвою eXtreme Gradient Boosting (XGBoost).

#### Випадковий ліс.

Випадковий ліс подібний до XGBoost тим, що він створює ансамбль дерев рішень, але замість того, щоб будувати їх послідовно, він будує кожне дерево окремо на основі випадкової вибірки навчальних даних [12]. У кожному дереві доступна лише випадкова кількість змінних предикторів для кожного розділення «так-ні», що призводить до того, що дерева відрізняються одне від одного. Загальний результат враховує рішення усіх дерев. У регресійних задачах використовується середнє значення серед прогнозів по всіх деревах.

#### Метод опорних векторів.

Метод опорних векторів (SVM – support vector machine) проектує дані в N-вимірний простір на основі значень змінних, що є входними параметрами (N-кількість параметрів); а потім створює межу, яка чітко класифікує точки даних [12].

#### Метод k-найближчих сусідів.

Алгоритм k-найближчих сусідів, також відомий як KNN або k-NN, — це непараметричний метод навчання з учителем. Використовує сусідство для класифікації або прогнозування щодо групування окремої точки даних, виходячи з припущення, що подібні точки можна знайти поруч. Тобто, якщо k найближчих сусідів належать до певного класу, то і точка, яка підлягає класифікації, буде відноситися до цього ж самого класу. Даний метод можна використовувати як для регресії, так і для задач класифікації [12].

## **5. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ**

Методи, що були описані вище, застосовано на наборі даних. Результати обчислення точності на тренувальній та тестовій вибірках представлено нижче в таблиці 1.

Таблиця 1. Результати обчислення точності для реалізованих методів

Метод	Точність тренувальної вибірки	Точність тестової вибірки
Лінійна регресія	0.9579	0.7980
eXtreme Gradient Boosting	0.9943	0.9372
Випадковий ліс	0.9967	0.9262
Метод опорних векторів	0.2925	-0.1378
Метод k-найближчих сусідів	0.9878	0.9219

Відповідно до таблиці 1 найкращий результат показує метод посилення градієнту (eXtreme Gradient Boosting). Але метод випадкового лісу та метод k-найближчих сусідів не сильно поступаються у точності моделі. Трохи гірший результат на тестовій вибірці показала лінійна регресія. Метод опорних векторів для даного набору даних зовсім не підходить, адже точність тренувальної вибірки досить низька, а точність тестової менше нуля. Від'ємне значення вказує на те, що лінія регресії не відповідає тренду даних, іншими словами, це означає, що модель погана.

## **6. ВИСНОВКИ**

З кожним роком проблема забруднення повітря стає все більш важливою та необхідною для вирішення, щороку помирають мільйони людей через високу концентрацію небезпечних для живих організмів речовин. Необхідно мати системи що здатні оцінити ризик смертності та побудувати прогноз. У даній роботі було розглянуто низку методів машинного навчання для прогнозування показників смертності від забруднення повітря. Методи машинного навчання було застосовано на наборі реальних даних. Результати

показали, що для прогнозу смертності найкращою моделлю є метод посилення градієнту. Точність прогнозу складає 0.94.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Борисенко А. В. Проблематика громадського здоров'я [Електронний ресурс] / А. В. Борисенко // ДУ «Вінницький ОЛЦ МОЗ України». – 2020. – Режим доступу до ресурсу: [http://cgz.vn.ua/problematika-gromadskogo-zdorovya/problematika-gromadskogo-zdorovya\\_455.html](http://cgz.vn.ua/problematika-gromadskogo-zdorovya/problematika-gromadskogo-zdorovya_455.html).
2. WHO global air quality guidelines. Particulate matter (PM2.5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide. Geneva: World Health Organization; 2021. Licence: CC BY-NC-SA 3.0 IGO.
3. Ludermir TB. Inteligência Artificial e Aprendizado de Máquina: estado atual e tendências. *Estudos Avançados*. 2021, v. 35, n. 101, pp. 85–94.
4. Minchole A, Rodriguez B. Artificial intelligence for the electrocardiogram. *Nature medicine*. 2019;25(1):22–3. Epub 2019/01/09. pmid:30617324
5. Deo, Rahul C. Machine learning in medicine. *Circulation*, v. 132, n. 20, p. 1920–1930, 2015. pmid:26572668
6. Wiens Jenna; Shenoy Erica S. Machine learning for healthcare: on the verge of a major shift in healthcare epidemiology. *Clinical Infectious Diseases*, v. 66, n. 1, p. 149–153, 2018. pmid:29020316
7. Goldstein BA, Navar AM, Pencina MJ, et al. Opportunities and challenges in developing risk prediction models with electronic health records data: A systematic review. *J Am Med Inform Assoc*. 2017; 24:198–208
8. Churpek MM, Yuen TC, Winslow C, et al. Multicenter comparison of machine learning methods and conventional regression for predicting clinical deterioration on the wards. *Crit Care Med*. 2016; 44:368–374
9. Tomašev N, Glorot X, Rae JW, et al. A clinically applicable approach to continuous prediction of future acute kidney injury. *Nature*. 2019; 572:116–119
10. Christodoulou E, Ma J, Collins GS, et al. A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. *J Clin Epidemiol*. 2019; 110:12–22
11. Saini A. Gradient Boosting Algorithm: A Complete Guide for Beginners [Електронний ресурс] / Anshul Saini // *Analytics Vidhya*. – 2021. – Режим доступу до ресурсу: <https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/>.
12. Hastie T, Tibshirani R, Friedman JH: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, Springer, 2009, pp xxii, 745

# МЕТОДИ ОЦІНКИ І ПРОГНОЗУВАННЯ ФІНАНСОВИХ РИЗИКІВ ДЛЯ БАНКІВСЬКОГО СЕКТОРА

Мельник І.А.<sup>1</sup>, Гуськова В.Г.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> Imelnik406@gmail.com, <sup>2</sup> guskovavera2009@gmail.com

**Метою дослідження є створення адекватних математичних моделей та розробка ефективних методів вирішення питань моделювання та прогнозування фінансових процесів, а також визначення ризиків потенційних втрат на основі використання методів інтелектуального аналізу даних, прикладної статистики та аналізу часових рядів. Внаслідок того, що управління фінансовими ризиками є проблемою, з якою стикаються всі учасники ринку ця проблема є надзвичайно актуальною станом на зараз.**

**Ключові слова:** орієнтування доходів менеджменту; бюджетування; планування; прогнозування; управління ризиками

## 1. ВСТУП

Людство завжди намагалося зрозуміти навколишній світ, шукаючи в усьому зв'язки та закономірності, накопичуючи величезні обсяги даних. Технології інтелектуального аналізу (ІАД) даних допомагають аналітику зрозуміти інформативність та повноту даних, виявити їх зв'язок та створити модель для вирішення реальних проблем[1]. Інтелектуальний аналіз даних спрямований на виявлення раніше невідомих закономірностей у масивах даних, реалізацію практичної та доступної інтерпретації знань, а також побудову аналітичних моделей, оптимізованих для вирішення певних класів прикладних задач при прийнятті рішень у різних сферах людської діяльності.

Стандартні технології аналізу даних базуються на підходах математичної статистики, що не у всіх випадках описують реальний стан речей та стандартні статистичні методи відкидають нетипові спостереження — так звані піки та сплески. За основу інтелектуальний аналіз даних бере концепцію шаблонів, щоб знайти широкоаспектний зв'язок між даними. Саме ці шаблони дозволяють зобразити неочікувані зв'язки та закономірності і залучити «приховані знання» («hidden knowledge») в наборах даних[2].

## 2. МЕТОДИ ОЦІНКИ ФІНАНСОВИХ РИЗИКІВ БАНКІВСЬКОГО СЕКТОРУ

Для застосування методологій та моделей ІАД, машинного навчання моделі та досягнення бажаних успішних результатів необхідно підготувати та сформувати набір даних. Визначення очікуваного фінансового ризику буде можливим за допомогою створеної моделі та доступу користувача до економічних даних. Також, важливо порівняти алгоритми та підходи ІАД та дослідити їх вплив на точність прогнозу моделі, відсоток розподілу масиву даних, що використовується для навчання та тестування, а також вплив пропусків у масиві даних.

Постановка та визначення задачі прогнозування полягають у наступному:

- 1) сформувати набори даних, використовуючи дані з відкритих сайтів (наприклад kaggle.com), які включають економічні показники та певний фінансовий ризик для кожного набору;
- 2) оцінити сучасні інструменти та програми для проведення ІАД;
- 3) провести огляд методологій, підходів та моделей інтелектуального аналізу даних для обробки набору даних та виконання моделювання і прогнозування фінансового ризику;
- 4) надати обґрунтування вибору середовища та мови програмування для вирішення питання створення програмного комплексу;
- 5) виконати аналіз даних (перший аналіз даних, візуалізація даних);
- 6) виконати обробку набору даних (приведення категоріальних стовпців до числових, заповнення пропусків, нормалізація даних);
- 7) створення математичних моделей для прогнозування фінансового ризику;
- 8) для навчання моделі прогнозування ризиків використовувати необхідні методи, наприклад лінійна регресія;
- 9) потім слід розглянути результати навчання та дослідження.

При створенні математичної моделі для представлення об'єкта використовуються теорія множин, теорія графів, теорія ймовірностей та інші математичні методи. До моделі застосовуються такі умови:

1. Модель повинна належним чином відповідати процесу або об'єкту.
2. Якщо аналітичне рішення є неможливим, для кожного рівняння в моделі повинно існувати чисельне рішення.
3. Модель повинна бути достатньо адаптивною, щоб її можна було використовувати для пояснення групи пов'язаних процесів або функціонування процесу за різних обставин.
4. Вимога робастності (robust - міцний, довговічний). Під робастністю розуміється здатність моделі забезпечувати відповідний прогноз вихідної змінної для будь-якого відрізка часового ряду, що відповідає обраному режиму роботи, а не тільки для того відрізка, на якому вона побудована.
5. Необхідність адаптивності. Внаслідок цього критерію принаймні частина параметрів моделі (хоча б один) може бути покращена в міру надходження свіжих даних з об'єкта.

Придатність регресійних моделей може бути визначена шляхом аналізу послідовності залишків (похибок моделі), при цьому розрахункові значення отримують шляхом заміни реальних значень усіх компонентів, що входять до моделі, на розрахункові. Послідовність залишків досліджується на відповідність критеріям випадкової складової економічного часового ряду, зокрема близькості математичного сподівання до нуля, випадковості відхилень, відсутності автокореляції та нормальності закону розподілу.

В основі статистичних методів аналізу та прогнозування, як правило, лежить глибока обробка статистичних даних, що стосуються досліджуваного процесу. В результаті можна виділити наступні етапи прогнозування.

**Етап 1** - постановка завдання - визначається перелік факторів і показників (змінних), опис зв'язків між ознаками, що цікавлять, а також ролі цих факторів і показників (тобто які з них в рамках конкретного завдання можна вважати вхідними (тобто повністю або частково регульованими або, принаймні, такими, що легко піддаються реєстрації і прогнозуванню)[3].

**Етап 2** - апріорний, передмодельний аналіз - ґрунтується на первинному вивченні змістовної сутності досліджуваного явища або процесу. На цьому етапі розробляються і формалізуються різноманітні гіпотези і припущення, які представляють собою апріорні знання про ці явища.

**Етап 3** - інформаційно-статистичний - відомий також як збір даних, передбачає реєстрацію значень використовуваних у дослідженні компонентів і показників на різних часових і (або) географічних циклах функціонування модельованої системи.

**Етап 4** - Специфікація моделі базується на передумовах і перших припущеннях, зроблених на другому етапі. Вона також передбачає створення фундаментальних зв'язків моделі, що з'єднують вхідні та вихідні змінні.

**Етап 5** - Для того, щоб "підігнати" значення невідомих параметрів під вихідні статистичні дані, а саме - ідентифікація моделі - проводиться статистичне дослідження даних. Для реалізації цього етапу спочатку необхідно визначити, чи можливо теоретично відновити значення невідомих параметрів моделі за наявними вихідними статистичними даними для обраної на четвертому етапі структури моделі.

**Етап 6** - верифікація моделі - порівняння результатів, оцінок і наслідків моделі з реальністю з використанням різних підходів. Інша назва цього етапу - статистичний аналіз точності та достатності моделі.

Об'єктивна оцінка якості прогнозу є важливим кроком у процесі прогнозування. Життєво важливо оцінити очікувані значення, оскільки вони є випадковими величинами. Обґрунтованість і точність прогнозів є двома важливими складовими прогнозування. На цьому етапі для оцінки якості прогнозу використовується набір стандартів, методів і процесів[4].

### **3. АНАЛІЗ РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯ ФІНАНСОВИХ РИЗИКІВ**

У зв'язку з тим, що визначення кредитоспроможності є погано структурованою проблемою з об'єктивною наявністю як якісних, так і кількісних показників, моделювання фінансових ризиків, зокрема кредитного ризику, ускладнюється. Крім того, багато банківських даних можуть бути представлені у нечіткому вигляді, а не піддаватися точному, чіткому визначенню[5].

**Крок 1** - База знань та пояснення даних експерименту. Для первинного аналізу та розробки математичних моделей визначення кредитоспроможності було обрано вибірку з 3350 клієнтів. Ці клієнти були ідентифіковані за такими ознаками:  $x_1$  - стать позичальника;  $x_2$  - вік;  $x_3$  - сімейний стан;  $x_4$  - кількість дітей;  $x_5$  - наявність власного житла;  $x_6$  - працює чи не працює;  $x_7$  - освіта;  $x_8$  - вид діяльності;  $x_9$  - наявність поручителя-фізичної особи;  $x_{10}$  - сума кредиту. Вихідна змінна - оцінка кредитного ризику - позначена у (результат аналізу).

**Крок 2** - Заповнення пропусків у даних. Для застосування до генеральної сукупності дані, отримані в результаті вибіркового обстеження, мають бути оброблені. Значення узагальнюючих показників встановлюються на основі зведеного набору всіх даних, зібраних під час обстеження.

Виключення часткових спостережень з прогалинами і подальший аналіз отриманих "повних" даних є найпростішим методом інтерпретації неповних даних. Штучні пропуски у співвідношенні 5% до 10% були введені у вихідних даних дослідження кредитоспроможності.

Для заповнення пропусків було використано регресійну модель та середнє значення вибірки. Значення похибки MAPE для різних методів заповнення пропусків даних про кредитоспроможність наведено нижче (Табл. 1).

Таблиця 1. Значення похибок MAPE при заповненні пропусків

Метод заповнення пропусків	Часовий ряд	Пропущено значень, %	
		5	10
Модель регресії	Заробітна плата	16,12	17,16
	Сума кредиту	4,92	5,01
Середнє за вибіркою	Заробітна плата	22,15	24,19
	Сума кредиту	28,35	27,87

Як видно з табл.1, в експерименті використовуються часові ряди з 5% та 10% фіктивних пропусків даних відносно загальної кількості значень часового ряду. Як видно з таблиці 1, похибка MAPE значно варіюється для розглянутих часових рядів. Однак найбільш успішними виявилися методи, які використовували для заповнення пропусків прогнози регресійної моделі (MAPE = 4,92%).

**Крок 3** - у табл. 2 наведено точність моделей, створених на основі критеріїв інформативності оцінювання змінних. В якості моделей використовувалися модель LOGIT та нечіткі системи на основі правил з трикутною та трапецієподібною функціями належності.

Таблиця 2. Показники якості моделі для всіх змінних та меншого набору атрибутів

Тип моделі		Точність (загальна кількість)
Logit model		0,89
FRBS	TRIANGLE	0,851
	TRAPEZOID	0,940

З таблиці бачимо, що найвищу точність результатів показала система на основі нечітких правил з трапецієподібною функцією належності (0,940).

**Крок 4** - співвідношення (%) навчальної та тестової вибірок, що використовувалися для прогнозування, становило 70/30, 80/20 та 90/10. Результати наведені в таблиці 3.

Таблиця 3. Інструменти прогнозування атрибутів

Співвідношення навчальної та перевіркової вибірок, %		70/30	80/20	90/10
Modeltype		Значення MSE		
Logitmodel		0,76658	0,76838	0,7612
FRBS	TRIANGLE	0,6104	0,61050	0,6191
	TRAPEZOID	0,23062	0,23001	0,2303



Системи, що використовують нечіткі правила з трапецієподібною функцією належності та співвідношенням навчальної та тестової вибірки 80/20, показали найкращий результат MSE - 0,23001.

#### **4. ВИСНОВКИ**

Початкова інформація та характеристики, що визначають кредитоспроможність позичальника, можуть бути суперечливими або відсутніми. На кредитоспроможність позичальника, як правило, впливає його фінансовий стан, який включає власні кошти, ліквідні активи та навички управління грошовими коштами. Під час двох етапів оцінки кредитоспроможності необхідно враховувати бізнес-ризик, а також фінансовий стан позичальника, для чого розглядаються фінансові коефіцієнти та грошові потоки. За цих обставин бажано обирати стратегію, яка поєднує в собі низку методів аналізу даних та прогнозування.

#### **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Методи інтелектуального аналізу даних [Електронний ресурс] – Режим доступу до ресурсу: <https://buklib.net/books/24506/>
2. Statistical Modeling [Електронний ресурс] – Режим доступу до ресурсу: <https://www.heavy.ai/technical-glossary/statistical-modeling>
3. Theodore M. Barnhill and William F. Maxwell Journal of Banking & Finance, 2002, vol. 26, issue 2-3, pp. 347-374.
4. Forecasting in manufacturing management [Електронний ресурс] – Режим доступу до ресурсу: [https://link.springer.com/referenceworkentry/10.1007/1-4020-0612-8\\_363](https://link.springer.com/referenceworkentry/10.1007/1-4020-0612-8_363)
5. Qualitative and quantitative analysis of creditworthiness of the companies [Електронний ресурс] – Режим доступу до ресурсу: <https://js.ugd.edu.mk/index.php/JE/article/view/2736>

# АНАЛІЗ ПРОЦЕСІВ ТА ПРИЙНЯТТЯ РІШЕНЬ НА РИНКУ КРИПТОВАЛЮТ

Мельников А.А.<sup>1</sup>, Бідюк П.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> melnykovandrew@gmail.com

**Об'єктом дослідження даної роботи є ринок криптовалют. Предметом дослідження є часові ряди фінансового ринку та, зокрема, ринку криптовалют, а також методи машинного навчання, які спрямовані на більш якісне та ефективне прийняття рішень. Метою роботи є розробка системи прийняття рішень (СППР), що базується на методах штучного інтелекту, та відтворення схеми прийняття рішень трейдера відповідними моделями. Результатом дослідження є модель побудована на базі методу штучного інтелекту та машинного навчання, а також висновок про придатність нейронних мереж до вирішення поставленої проблеми за умови постійного додаткового навчання. В роботі використано емпіричні й теоретичні методи дослідження.**

**Ключові слова:** криптовалюта, машинне навчання, згорткові нейронні мережі, трейдинг, глибоке навчання, бінарна класифікація.

## 1. ВСТУП

На сьогодні криптовалютні операції стають дедалі більш поширеним методом розрахунків між фізичними особами. Також зростає інтерес до операцій з деривативами. Зокрема вже зараз для українців доступні два типи ф'ючерс контрактів: безстроковий та кварталний [1]. У розріз із типовим біржовим трейдингом для торгівлі криптовалютами або для заключення контрактів на їх купівлю продаж не потрібно спеціальної ліцензії, а правове регулювання в цій сфері доволі м'яке, що робить проведення операцій доступним для будь кого у віці від 16 років.

Однак є і перепони на шляху популяризації біржових операцій на основі криптовалют, зокрема відсутність прив'язки до фізичних активів. Це призводить до важко прогнозованості поведінки ринку криптовалют – ціна формується лише очікуваннями користувачів. Оскільки поведінка ринку залежить від рішень користувачів, то в роботі було вирішено відтворити процес прийняття рішень за допомогою класифікації деякого відрізка часового ряду - аналогічно до роботи трейдера.

Під час виконання роботи поставлено та виконано такі задачі:

1. Розробити систему прийняття рішень, що базувалась на методах штучного інтелекту;
2. Відтворити схему прийняття рішень трейдера відповідними моделями;
3. Провести порівняння з загально відомими моделями.

Побудована система повністю відповідає поставленим цілям і відтворює процес прийняття рішень трейдером (тобто підхід до задачі іде з боку задачі класифікації а не прогнозування часового ряду). В якості ядра системи було обрано нейронну мережу та логістичну функцію втрат.

Також виконано аналіз моделей на старіння та ефективність підходу на 16 криптовалютах. В якості візуалізації ефективності роботи моделі було побудовано РОК криві для моделей в періоді навчання та на ринкових даних після зміни загальної тенденції.

## 2. РОЗГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ТА РОЗКРИТТЯ ЗАДАЧІ

Розробка моделі спирається на факт домінування на ринку візуального сприйняття ситуації трейдерами. Часткове розкриття структури ринку криптовалют було розглянуте в попередніх роботах [2]. Спираючись на попередні дослідження очікування користувачів є найбільш впливовим фактором для ринку криптовалют.

Однак з іншого боку одним з важливих факторів що впливають на формування ціни криптовалют також виступає ціна підтримки мережі як платіжної системи. Ця величина може бути оцінена і закладена як мінімальна ціна активу. Відповідно, з кожним роком вона зростає через ускладнення мережі, контрольованої інфляції цін на електроенергію та залучення більшої кількості користувачів до мережі. Загалом це створює тенденції до зростання ціни криптовалют що робить прибутковими навіть стратегії з нульовим математичним середнім (частково використання цих стратегій призводить до зростання капіталу що знаходиться безпосередньо в криптовалюти як інвестиція – провокуючи подальше зростання). Однак основною ціллю даної роботи є прогнозування змін курс відповідно до першого чинника - очікування користувачі, звісно в майбутньому цей чинник буде зменшувати свій вплив (рис. 1).

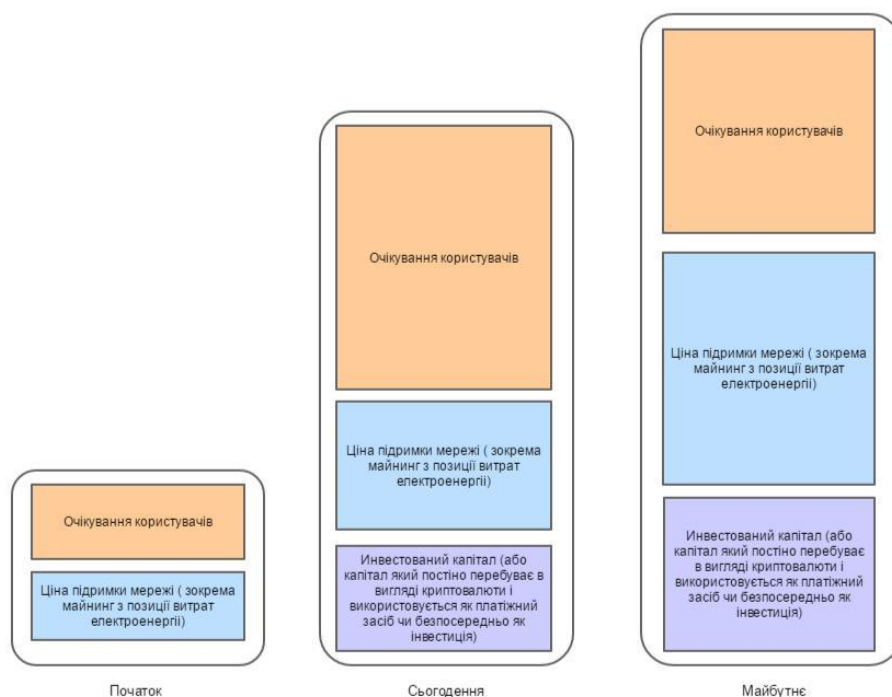


Рисунок 1. Основні ціноутворюючі чинники для криптовалют (представлено динаміку зростання ціни підтримки системи та інвестиційної привабливості криптовалют)

Через неможливість оцінити сподівання безпосередньо було вирішено використовувати нейронну мережу як засіб що дозволяє зробити висновки аналогічні до висновків трейдера та знайти приховані закономірності в їх поведінці.

Оскільки ситуація на ринку не може характеризуватись постійно одними величинами, то побудована модель матиме тенденції до старіння. Як пізніше буде розглянуто в розділі 3. Цей проміжок займатиме лише 3 дні. Після цього часу модель необхідно перевірити на якість та прийняти рішення про додаткового навчання чи продовження роботи (рис. 2).



Рисунок 2. Життєвий цикл моделі

Модель безпосередньо відповідає за прийняття рішень. Користувач моніторить загальну роботу системи лише за тижневими звітами. Відсторонення користувача від роботи системи є свідомий крок. На жаль, людина не може бути абсолютно неупередженим, а з вигляду на ефективність прийняття рішень упередженість користувача може бути фатальною (оскільки мова йде безпосередньо про гроші, людина схильна до ризику при ймовірності високої винагороди, навіть при незначному шансі на виграш).

Також революційним є відмова від безпосереднього прогнозування курсу. Саме перехід на навчання системи приймати рішення дозволяє збільшити точність. Адже з практичної точки зору не має сенсу наскільки зростає курс активу, якщо це приносить прибуток то є успіх. Таким чином модель безпосередньо навчається діям аналогічним до дій користувача.

Отже, у розділі було розглянуто основні характеристики розробленої системи та її особливості. Було наведено характеристику досліджуваного об'єкта, а саме ринків криптовалют. Розглянуто складові ціноутворення.

На основі викладених фактів було прийнято рішення про розробку системи, що базується на нейронній мережі і відповідає безпосередньо за прийняття рішень.

### 3. РОЗГЛЯД МЕТОДІВ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

Трейдинг доволі консервативна сфера прийняття рішень. Хоча на сьогодні на будь-якому ринку трейдеру доводиться обробляти величезні масиви інформації та шукати приховані залежності. Однак прийняття рішень все відповідальний процес який неможливо перекласти з людини через правове регулювання сфери. Відповідно до законів алгоритм прийняття рішень повинен бути чітким і зрозумілим, однак нейронні мережі працюють за принципом чорного ящика і не дають чіткого розуміння роботи в критичних ситуаціях. Однак криптовалюта не має чітких норм щоб регулювали операції з нею, а отже в немає жодних обмежень на модель.

З іншого боку доволі масове використання криптовалют почалось лише 4-6 років тому - а отже сфера має незначну вибірку даних, що вже накладає обмеження на методи побудови моделей.

Одним з найважливіших моментів при побудові дослідження є обрання метрики для оптимізації роботи моделі, тобто формалізація поставленої задачі за допомогою функції на обраному просторі (для формалізації очікувань користувача/дослідника). Було розглянуто наступні метрики:

1. Binary cross entropy (бінарна перехресна ентропія):

$$H(y, \underline{y}) = \frac{1}{n} \sum_{i=1}^n y_i \log(y_i) + (1 - y_i) \log(1 - y_i),$$

де  $y$  – вектор дійсних значень розмірності  $n$ , де  $n$  – кількість тренувальних даних,

$\underline{y}$  – вектор отриманих прогнозів моделі,

$y_i, \underline{y}_i$  – відповідні координати векторів.

Дозволяє враховувати як правильно класифіковані об'єкти так і неправильно з однаковою вагою і тих і тих класів. Призводить до більш чіткого розділення класів. Зокрема на рисунку 3 подано зміну значення похибки при дійсному значенні 1 та 0.

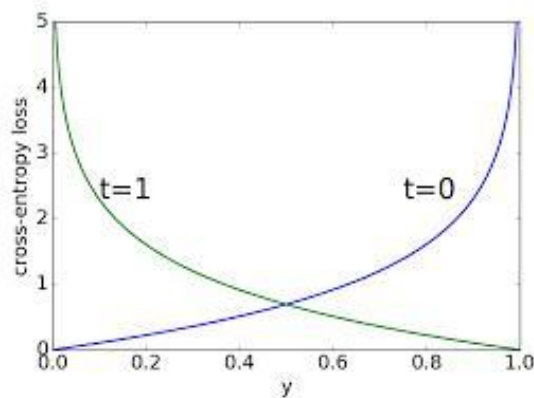


Рисунок 3. Зміни значення втрат при в залежності від значення що було визначено моделлю ( $y$ ) та реальних значень ( $t$  - реальне значення належності до класу, синім кольором - немає приналежності до класу, зеленим - приналежність є)

Це працює лише у випадку бінарного індикатора класу. У випадку не бінарної класифікації результат може бути представлено як середнє значення по кожному з векторів класу. Метрику можна використовувати для задач с несиметричною кількістю об'єктів класів;

2. ROC AUC (Area Under the Receiver Operating Characteristic Curve) площа під кривою помилок, побудованій на різних порогах відсікання рисунок 4 [3]. Може використовуватись лише для визначення якості бінарного класифікатора. Головною особливістю є врахування кількості пар об'єктів одночасно класу що правильно класифіковані та які були відбраковані (тобто помилку першого та другого роду). Також за формою кривої можна визначити оптимальний рівень для прийняття рішень. Оскільки метрика враховує ефективність розділення класів моделлю то у випадку незначної кількості об'єктів класу буде давати хибний результат (точніше призведе до навчання моделі ігнорувати цей клас).

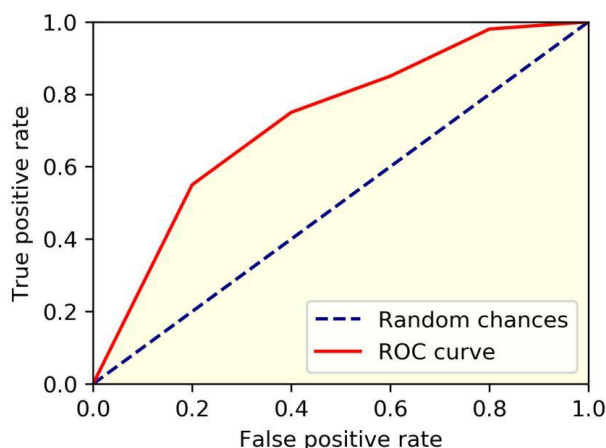


Рисунок 4. Типова РОК крива. Синім зазначено випадкові значення. Червоним - робота моделі: чим більше червона лінія відрізняється від діагоналі тим краща якість моделі. Аналогічно чим більша площа під графіком тим краще.

3. F-score. Метрика враховує кількість правильно класифікованих об'єктів до всіх класифікованих та кількість некласифікованих. Обидва критерії мають однакову вагу.

$$Fscore = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

де *precision* – відношення кількості правильно класифікованих об'єктів до кількості всіх класифікованих (рисунок 5),

*recall* – відношення кількості правильно класифікованих об'єктів до кількості всіх об'єктів (рисунок 5).

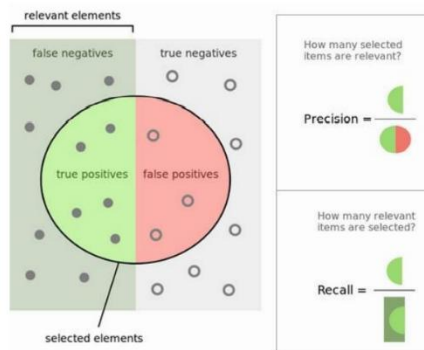


Рисунок 5. Схема до пояснення термінів

Однак при незначній кількості об'єктів класу є схильність до їх занулення. Метрику не можна використовувати для незначної кількості даних. Через несиметричність даних та вимогу в автоматизації було обрано бінарний перехресний критерій як якість моделі (точніше його мінімізація). Однак для оцінки моделей на старіння було вирішено використовувати РОК криву - так як вона має доволі просту інтерпретацію і гарно візуалізує якість моделей. Також може показати різні варіанти помилок моделі та порівняти їх із випадковими результатами.

Отже, в третьому розділі було розглянуто типові метрики для проведення оптимізації та типи моделей. Було докладно розглянуто причини та передумови обрання конкретних засобів.

## 4. РЕЗУЛЬТАТИ ЗАСТОСУВАННЯ МОДЕЛЕЙ

Поведінка моделі була досліджена на трьох відрізках:

1. Навчання базової моделі.
2. Тестування навчальної моделі та донавчання.
3. Тестування моделі отриманої після навчання на попередніх відрізках.

Для кожного активу використовувались моделі одного типу, що відрізняються лише процесом тренування. В процесі навчання точність моделі (крок динамічно змінюються, якщо точність почне коливатись і не буде зменшуватись).

Для первинної моделі було обрано інтервали у 10000 вимірів (через технічні обмеження системи) Зокрема нижче наведено навчання моделі на проміжку з 2020-10-10 по 2020-10-15.

Виміри відбуваються кожні 30 секунд. В векторі вимірювань присутні змінні що характеризують: об'єм торговельних операцій, ціна на 2 біржах, зміна ставки фінансування для деривативів. Для прийняття рішень використовувався інтервал що відповідає максимальному торговельному вікні що може спостерігати трейдер - 6 годин (або 720 вимірів). В якості умови прийняття рішень було обрано підвищення ціни активу на 2% і не зниження її менш ніж на 1%.

Нижче наведено лог навчання моделі:

```
start=2020-10-10 16:55:14.182461 end=2020-10-15 11:55:00.786176 long:70 short:137
train leng 15873, test leng 1500
long:0 short:33
Epoch 1/10
249/249 - 156s - loss: 0.0465 - val_loss: 0.0636
Epoch 2/10
249/249 - 154s - loss: 0.0377 - val_loss: 0.0650 Epoch 3/10
249/249 - 156s - loss: 0.0330 - val_loss: 0.0637 Epoch 4/10
249/249 - 156s - loss: 0.0302 - val_loss: 0.0625 Epoch 5/10
249/249 - 155s - loss: 0.0273 - val_loss: 0.0621 Epoch 6/10
Epoch 00006: ReduceLROnPlateau reducing learning rate to 1.0000001111620805e-07.
249/249 - 156s - loss: 0.0255 - val_loss: 0.0611 Epoch 7/10
249/249 - 155s - loss: 0.0246 - val_loss: 0.0612 Epoch 8/10
249/249 - 155s - loss: 0.0248 - val_loss: 0.0608 Epoch 9/10
249/249 - 155s - loss: 0.0240 - val_loss: 0.0614 Epoch 10/10
Epoch 00010: ReduceLROnPlateau reducing learning rate to 1.000000082740371e-08.
249/249 - 156s - loss: 0.0245 - val_loss: 0.0608.
```

З логів програми видно, що модель показує доволі непогані результати. Однак потрібно врахувати, що у валідаційній вибірці відсутні представники одного з класів, а отже можливе несиметричне навчання. Однією з переваг обраної моделі є можливість донавчання моделі в процесі її використання. Таким чином можна вирішити проблему старіння моделі без перенавчання по всій вибірці. Так через деякий час моделі починають псуватись.

Для ілюстрації цього факту було обрано ще один 5 денний проміжок часу через 6 днів після навчання моделі. Навчання моделей було проведено на доволі тихому ринку, були відсутні тенденції зростання. Очікувано, що після зміни тенденцій ринку моделі показують гірші результати або будуть потребувати перенавчання.

Однак по невдачах можна теж судити про внутрішні якості моделі та потенціал моделі для подальшого навчання. Зокрема, можна побачити доволі непогані результати, якщо взяти обернену модель (оскільки модель де-факто бінарний класифікатор, то достатньо змінити клас). Це свідчить про недостатнє навчання моделі (потрібно збільшити кількість епох).

Також це свідчить про те що дані та модель підібрані правильно і є потенціал для продовження дослідження.

Отже в розділі було розглянуто основні особливості обраної моделі. Було виявлено тенденції до старіння моделі. Навіть у випадку старіння модель на поданому активі дала задовільні результати точності. Це дозволяє зробити висновки що:

1. Поставлена проблема може бути розв'язана за допомогою обраних засобів.
2. Обрана модель дозволяє отримати непогані результати, однак наявне старіння моделі і потрібне донавчання моделі.
3. Модель може працювати без донавчання протягом 1-3 днів. На 5 день ефективність роботи моделі значно зменшується.
4. Модель та кількість даних було обрано правильно і подальший розвиток СППР на її основі мають потенціал до подальшого розвитку.
5. На 25% криптовалютах було досягнуто комерційний рівень точності на тестових даних.

## 5. ВИСНОВКИ

Отже, у поданій роботі розглянуто варіант трейдингової системи прийняття рішень що базується на класичному підході оперуванням з активами. Було розглянуто модель нейронної мережі, на базі якої приймалось рішення на

ринку. Було доведено можливість її застосування до поданої задачі. Також результати засвідчили її ефективність для окремих пар навіть після значного проміжку часу, а отже з постійним донавчанням розглянута система може бути цілком автоматичною. В роботі було досліджено періоди донавчання. Також було проведено тестування роботи моделі на реальних даних, що переконливо свідчать про правильний вибір типу моделі для даної проблеми, однак для деяких валютних пар необхідно провести додаткове донавчання моделі (а саме збільшити кількість епох навчання).

Успішність всіх попередніх кроків свідчить про правильно обраний напрям роботи, а саме застосування моделей класифікації для ринків з метою прогнозування напряму торговельних операцій інших трейдерів (уникаючи прогнозування власне значень часового ряду, прогнозуючи лише тенденцію до росту чи до падіння). Даний підхід є унікальним, однак застосування його, на мою думку, обмежене лише сферами де ціноутворення залежить більшою мірою від очікувань користувачів (інших трейдерів), що і є характерним для ринку криптовалют.

Наведено повний лог програми навчання моделі, дані що використовувались для навчання їх візуалізацію та ефективність моделі на цих проміжках. Дослідження охоплювало як стабільний ринок (з умовно незмінним трендом) так і швидко зростаючий. Така зміна умов мала б жахливі наслідки для моделей, що сфокусовані на прогнозуванні ціни. Однак з погляду прийняття рішень трейдером це має незначний ефект, що і було показано моделлю на прикладі ринку EOS. Однак для інших ринків спостерігалась значна зміна поведінки, що вплинуло на модель, зокрема обернене рішення моделі було більш наближеним до розміченого – це свідчить про недостатня навченість моделі та існування глобальних патернів поведінки користувачів.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Романенко В.Д., Милявский Ю.Л., Канцедал Г.О. Стабилизация неустойчивого курса криптовалюты на основе модального управления импульсным процессом когнитивной карты. XXVI Міжнародна конференція з автоматичного керування. Автоматика 2020: матеріали 26 наукової конференції, м. Київ, 13-15 жовтня 2020р. С. 75-76.



2. Бінанс відкриває гривневий шлюз через Advcash. URL: <https://www.binance.com/ru/support/articles/360035986971> (дата звернення: 22.11.2020).
3. Nielsen. M. Neural Networks and Deep Learning: free online book, 2019. chapter 3. URL: <https://static.latexstudio.net/article/2018/0912/neuralnetworksanddeeplearning.pdf> (дата звернення: 22.11.2020).
4. Fawcett T. An introduction to ROC analysis: Pattern Recognition Letters / USA, 2006, pages 861-874.
5. Xiao-Yuan Jing, Fei Wu, Dong Xiwei, Baowen Xu. An Improved SDA Based Defect Prediction Framework for Both Within-Project and Cross-Project ClassImbalance Problems. IEEE Transactions on Software Engineering. 2016. P. 43. DOI: <http://dx.doi.org/10.1109/TSE.2016.2597849>.
6. Heorhii Kantsedal. USE OF DECISION TREES IN THE LIMITED CRYPTOCURRENCY MARKETS FOR ADAPTIVE DECISION MAKING. XX International Students R&D Conference 2019 “Science and Technology of the XXI century”: Kyiv, Ukraine, 22 October 2019. P. 72-73.
7. Канцедал Г.О. Градієнтний бустинг. Травень 2019. URL: [https://www.researchgate.net/publication/335567928\\_gradientnij\\_busting](https://www.researchgate.net/publication/335567928_gradientnij_busting) (дата звернення: 22.11.2020).
8. Канцедал Г.О. Проміжні результати. Ризики при проведенні торговельних операцій в криптовалюти. 2020. DOI: <http://dx.doi.org/10.13140/RG.2.2.14023.85922>.

# МОДЕЛІ І ПРОГНОЗИ ВОЛАТИЛЬНОСТІ ФІНАНСОВИХ ПРОЦЕСІВ З ВИКОРИСТАННЯМ ЙМОВІРНІСНО-СТАТИСТИЧНИХ МЕТОДІВ

Морозов Р.Д.<sup>1</sup>, Бідюк П.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>morozov.roman@lil.kpi.ua

**Практично всі процеси, у будь-якій сфері діяльності – економічні, соціальні, екологічні, технічні, фінансові потрібно розглядати як нелінійні та нестационарні. Моделі фінансового ринку являються типовим представником нелінійних нестационарних процесів, адже вони, як правило, нестабільні, тобто мають досить складну динаміку процесу. Відповідно, для того щоб врахувати всі параметри, які впливають на розвиток нелінійного нестационарного процесу, тривіальних моделей буде недостатньо, оскільки отриманий результат буде некоректним. Тому потрібно використовувати нові, складніші (адекватні) моделі для того щоб забезпечити високу якість прогнозу.**

**Ключові слова:** нелінійні процеси, нестационарні процеси, волатильність, мережа байєса, нейронна мережа.

## 1. ВСТУП

Моделювання цін акцій є основою фінансової економіки і, зокрема, теорії управління інвестиційним портфелем і оцінки фінансових інструментів.

Прогнозування і моделювання волатильності – важливе питання дослідження фінансових ринків. Велика емпірична робота була пророблена для поліпшення моделей волатильності, оскільки більш точні прогнози приводять до кращого ціноутворення фінансових активів і кращого управління ризиками. З іншого боку, волатильність фондового ринку інтенсивно вивчалася в останні три десятиліття, і було пророблено великий обсяг емпіричної роботи.

Статистично-ймовірнісні методи є одними з найпопулярніших для прогнозування розвитку економічних процесів, а також об'ємів торгових операцій, об'ємів виробництва, формування бюджетів підприємств та держави, оцінювання альтернативних економічних стратегій, прогнозування та менеджменту економічних і фінансових ризиків. Сьогодні без економічних прогнозів не обійтись також для визначення цілей розвитку суспільства, обґрунтування напрямів розвитку економіки або передбачення наслідків прийнятих рішень.

## 2. ОПИС ОБРАНИХ МЕТОДІВ

Більшість процесів у сучасному нестабільному світі є нелінійними та нестационарними. Останнім часом виникає все більше нових, непередбачуваних процесів, що негативно впливають на діяльність суспільства.

У різних напрямках досліджень встановлено, що часові ряди, які описують поведінку економічних і фінансових змінних у сучасній економіці, показують нерегулярні закономірності, які часто характеризуються коливаннями, що передбачають різке зниження та збільшення деяких параметрів [1].

З тих пір як економісти почали працювати над цими питаннями, основна увага зосередилася на спробі зрозуміти причини коливання економічних змінних, а також природу та причини цієї нерегулярної поведінки. Потім особливий акцент було зроблено на з'ясуванні того, чи є нестабільність явищем, що виникає ендогенно в економічній системі, чи вона продиктована екзогенними стохастичними збуреннями. Ці дослідження спрямовані на те, щоб пояснити ринкову динаміку, і мають на меті показати, що поява хаосу узгоджується з поведінкою оптимізуючих агентів, широко поширених у процесі прийняття рішень як фірмами, так і споживачами. Ці розробки природним чином привели до побудови нелінійних динамічних моделей [2].

Причини виникнення волатильності – перехідні процеси в економіці, суспільних формаціях, внутрішні та зовнішні військові конфлікти (втрати), глобальні та локальні фінансові кризи, недостатньо кваліфікований менеджмент на всіх рівнях управління економікою, застарілі технології у промисловості та сільському господарстві, зміна режимів функціонування соціально-економічних систем (наприклад, закінчення перехідного процесу і перехід до усталеного), природні катаклізми (вулкани, повені, потепління, ...).

Останнім часом мережі Байєса почали набирати популярність серед способів моделювання процесів різної природи та стали окремим напрямом в області інтелектуального аналізу даних. Мережі Байєса широко застосовуються для обробки часових рядів (статистичні дані), та інших все можливих типів представлення даних, наприклад, експертні оцінки, лінгвістичні змінні тощо [3].

Можливість врахування водночас кількісних та якісних показників, правильно використати надходження нової інформації у динаміці, використання явної залежності між існуючими факторами (що в свою чергу впливають на фінансові показники) є, мабуть, основною перевагою мережі Байєса. При використанні мережі Байєса часто виникають труднощі при знаходженні методу формування точного імовірнісного висновку базуючись на навчальній вибірці: для того щоб порахувати ймовірності вершин вузлів, потрібно використати матрицю спільних ймовірностей замість значень умовних ймовірностей емпіричних значень. Основною метою є те, що це дозволяє не розробляти структуру мережі заздалегідь. Також мережу просто реалізувати і при цьому досягти бажаної обчислювальної швидкості (це залежить від розміру вибірки) [4]; також не можна нехтувати складністю (чи навіть проблемою) вибору змінних для побудови моделі.

Нейронні мережі є одним із найпопулярніших методів (математичних моделей) вирішення задачі прогнозування часових рядів, вони включають в себе розвинену методологію структурного моделювання й методів навчання, що базуються на добре розвинутій теорії нелінійного програмування. Метою прогнозування є зменшення ризику при прийнятті рішень. Прогноз зазвичай виходить помилковим, але похибка залежить від прогнозуючої системи, яка використовується. Безумовно програмні реалізації розглянутих алгоритмів повинні мати дружній інтерфейс для користувача, дозволяти фахівцеві конкретної предметної області моделювати процес самостійно, без участі фахівця-математика.

Експоненційне згладжування є одним з методів прогнозування часових рядів для одновимірних даних. Цей метод створює прогнози, які є середньозваженими показниками минулих спостережень, де вагові показники старих спостережень експоненційно зменшуються. Форми експоненційного згладжування розширюють аналіз до даних моделі з тенденціями та сезонними компонентами [2].

Методи прогнозування експоненційного згладжування подібні тим, що вони прогнозують ряд на основі попередніх значень. Проте дані моделі використовують

експоненційно спадні вагові коефіцієнти. Це означає, що минулі значення впливають менше, ніж останні.

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для порівняння оцінок двокрокового прогнозу використано ряд значень індексу NYSE Composite. Для прогнозування методом групового врахування аргументів (Рис. 1) було використано дворядний алгоритм із розміром навчальної вибірки 45 значень (60% від загальної кількості вимірів), за даними отриманої моделі відновлено 2975 значень.

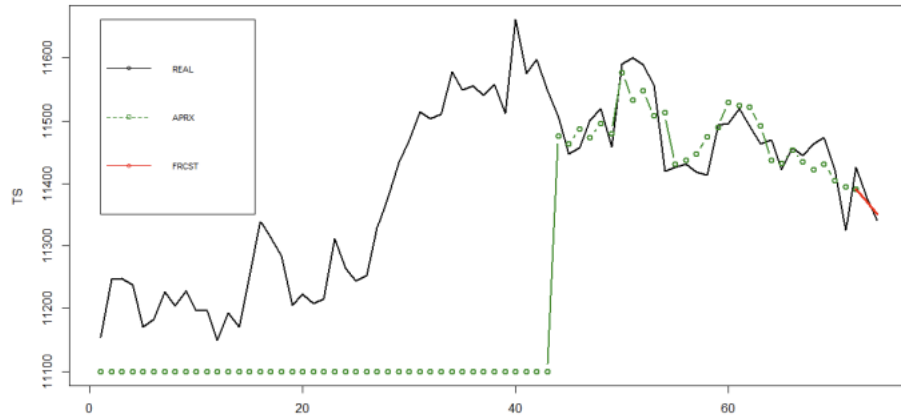


Рисунок 1. Двокроковий прогноз МГВА

На (Рис. 2, 3) зображено результати прогнозування методом експоненційного згладжування та за допомогою нейронної мережі відповідно.

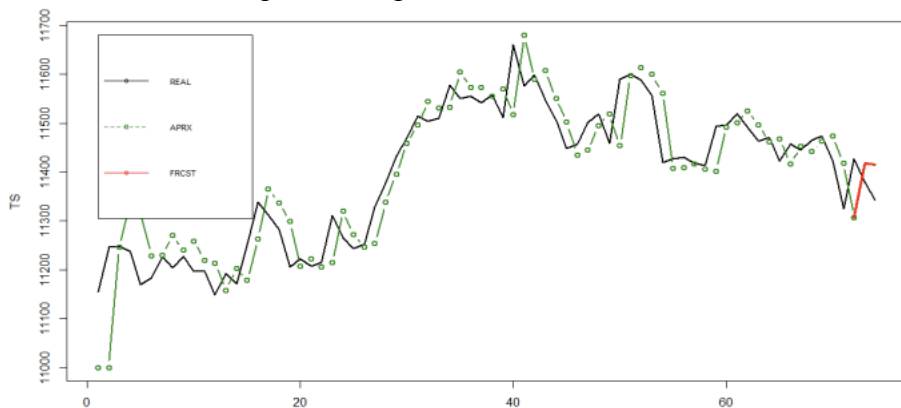


Рисунок 2. Двокроковий прогноз методом експоненційного згладжування

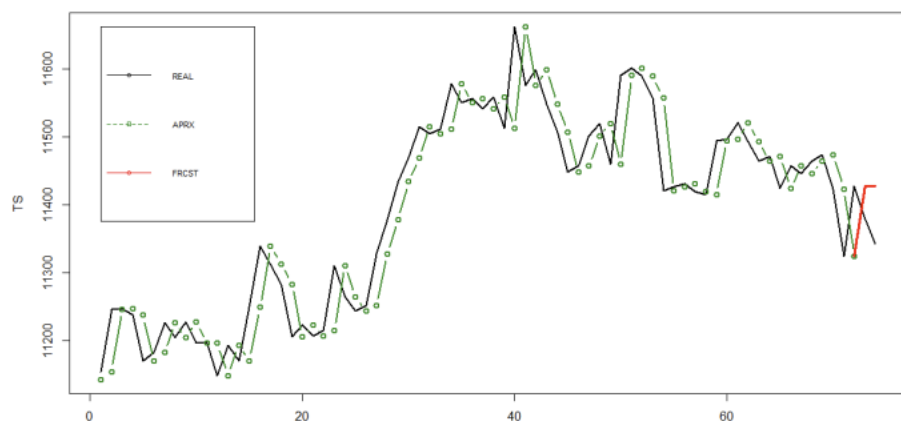


Рисунок 3. Двокроковий прогноз, отриманий за допомогою нейронної мережі

Результати експерименту подані нижче (Табл. 1). Як видно з таблиці, найкращі результати на ділянці прогнозування показав поліноміальний дворядний МГВА: прогнозоване значення відрізняється від реального лише на 0.7%, що є дуже хорошим результатом, при цьому, варто відмітити, що при відновленні даних метод показав значно гірші результати: СКП для відновленої моделі – 37,11; СКП для прогнозу – 8,57.

Таблиця 1. Результати прогнозування

NYSE Composite	модель			прогноз			
	СКП (RMSE)	МАП	КД	СКП (RMSE)	САПП (MAPE)	МіАПП	МАПП (MAE)
ЕЗ	56.740	73.14	0.86	58,84	4,97	3,49	6,45
МГВА	37.110	92.70	0.91	8,57	0,74	0,62	0,87
НМ	52.065	148.83	0.86	68,99	5,86	4,26	7,46

Діаграма зміни прогнозного, з використанням мережі Байєса, і реального відхилень індексу на останніх 50 часових інтервалах подана нижче (Рис. 4).



Рисунок 4. Відхилення індексу від значення на попередньому кроці

Після аналізу результатів бачимо, що у понад 90% випадків було точно передбачено зростання або падіння значення обраного індексу. У 86% випадків прогнозне і реальне значення відхилення індексу від значення в попередній момент часу мають відхилення менше 10%. І лише 15% експериментів мають відхилення більше 10% – така ситуація характерна для малих різниць та може бути пов’язана з варіантом дискретизації змінної.

Перш за все слід підкреслити, що метод експоненційного згладжування дає прогноз лише на 1 крок, тому його результати в цьому аналізі враховуватись не будуть. Експоненційне згладжування є найпростішим з розглянутих методів. Цей метод дає достатньо якісний для практичного використання прогноз на 1 крок за умови стаціонарності ряду або чіткої тенденції без суттєвих коливань. Тому область можливого використання метода досить обмежена. Прогнозування за допомогою нейронної мережі в цілому дало хороший результат. Проте при прогнозуванні на більшу кількість кроків якість суттєво погіршується. Це можна пояснити використанням простої мережі- з одним прихованим шаром. Метод групового врахування аргументів видає дуже якісні прогнози при короткостроковому прогнозуванні. Також зазначимо, що порівняно з усіма іншими методами МГВА значно програє в швидкості роботи, однак дозволяє отримати високоякісну модель при малій кількості вихідних даних, що дозволяє побудувати якісний прогноз.

При побудові мережі Байєса встановлено, що у 86% випадків прогнозне і реальне значення відхилення індексу від значення в попередній момент часу мають відхилення менше 10%. При цьому, при прогнозуванні на більшу кількість кроків якість прогнозу не погіршується. Для покращення результату варто звернути увагу на вибір кількості проміжків дискретизації та, за можливості, збільшити навчальну вибірку для мережі.

#### **4. ВИСНОВКИ**

При прогнозуванні фінансово-економічних показників необхідно раціонально обрати метод, який якісно прогнозує необхідний показник. При цьому слід також визначити параметри метода відповідно до значень ряду, що прогнозується. Таким чином найбільш "вузким" місцем є вибір методу та параметрів моделі в залежності від особливостей вхідних даних для знаходження найкращого варіанту. Перевагами методу експоненційного згладжування є швидкість обчислень та обґрунтованість результатів. Використання МГВА має перевагу у відсутності потреби створювати структуру моделі. Мережа Байєса допомагає описати процес шляхом статистичного опису ймовірностей та взаємодії між ними. Метод групового врахування аргументів дозволяє вирішувати широке коло задач, з якими звичайні регресійні моделі не справляються. Перевагою є якість прогнозу на довгих рядах без ресурсоємних обчислень. До недоліків можна віднести побудова нової моделі при зміні якогось параметра. Застосування нейронних мереж для прогнозування процесів в економіці, фінансах, екології, медицині вирішує багато актуальних задач. Перевагами такого підходу є незалежність дослідника від поведінки моделі, гнучкість і точність обчислень. До недоліків можна віднести те, що мережа сама приймає рішення і вивести явну модель поведінки ми не можемо.

#### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Кулявець В. О. Прогнозування соціально-економічних процесів. Київ: Кондор, 2009. – 194 с.
2. Зайченко Ю.П. Основи проектування інтелектуальних систем: навч. посіб. Київ: Слово, 2004. – 352 с.
3. Бідюк П.І., Терентьев О.М. та Коновалюк М.М. Байєсівські мережі в технологіях інтелектуального аналізу даних. Київ: Штучний інтелект, 2010. – 168 с.
4. Левченко Л.О. Огляд методів прогнозування фінансового стану підприємства на основі економетричних моделей / Л.О. Левченко, Д.С. Белова // Управління розвитком складних систем. - 2013. - Вип. 14. – С. 164-169.

# СИМУЛЯЦІЯ МОНТЕ КАРЛО СИСТЕМ КЕРУВАННЯ ЗАПАСАМИ

Наконечна В.А.<sup>1</sup>, Тимошук О.Л.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>nakonechnavlada@gmail.com,

<sup>2</sup>o.tymoshchuk@kpi.ua [0000-0003-1863-3095]

**Запаси є невід’ємною частиною будь-якого підприємства, і вміння керувати ними безперечно грає велику роль в успіху та розвитку фірми, тому важливо розрізняти основні системи керування запасами, їх переваги та недоліки за тих чи інших обставин і параметрів. Метою роботи є аналіз систем керування запасами за допомогою імітаційного моделювання методом Монте Карло. Результатом дослідження є демонстрація переваг та недоліків найрозповсюдженіших систем керування запасами. У роботі було використано теоретичні та емпіричні методи дослідження.**

**Ключові слова:** логістика запасів, керування запасами, система керування запасами, симуляція Монте Карло.

## 1. ВСТУП

Запаси різних видів продукції створюються для досягнення різноманітних цілей, які визначаються керівництвом промислової або торгової фірми (підприємством, організацією). До найбільш поширених цілей створення запасів належить підвищення ефективності підприємства, ефективне обслуговування споживачів, страхування перебоїв у постачанні, захист від підвищення цін закупівлі, економія на оптових знижках та транспортуванні [1].

Питання управління запасами займає важливе місце у системі управління поточними активами кожного підприємства. Воно є ключовою логістичною функцією, на долю якої, згідно з декількома джерелами, разом з транспортуванням приходиться від 60 до 80% загальних логістичних витрат [2]. За даними консалтингової компанії Deloitte, у 2020 році в умовах пандемії COVID-19, показник запасоемності в загальному по всіх індустріях США перевищив історичний максимум за останні 20 років. Аналогічний тренд простежується в усьому світі, у тому числі в Україні. [3] За останні два роки ситуація не покращилася, з повномасштабним вторгненням навряд чи залишилися підприємства, чії логістичні ланцюги не постраждали від нових перешкод. Але навіть незважаючи на це можна впевнено сказати що система управління запасами низького рівня з часом стає слабкістю підприємства – воно починає зазнавати збитків від поганого обслуговування через незадовільне накопичення важливих запасів і підвищені інвестиції, що зумовлене затриманням строку виконання замовлень.

## 2. СИСТЕМИ КЕРУВАННЯ ЗАПАСАМИ

Основним призначенням функції управління запасами є забезпечення збалансованості між обслуговуванням споживачів та інвестиціями в товарно-матеріальні запаси, особливо з точки зору концепції оптимізації запасів. Виділяють багато моделей керування запасами. Деякі джерела [4] розрізняють моделі за кількістю товару в системі, за постійністю таких параметрів як, наприклад, попит, за часом, за стратегією керування, за динамічністю

закупівлі, а інші виділяють ще більше категорій: за попитом, за поповненням запасів (за затримкою і за об'ємом), за витратами, обмеженнями та в залежності від стратегії керування. [1] З такого величезного розмаїття на практиці використовується лише обмежена кількість моделей - в основному ті, які надають відносно прості способи регулювання параметрів замовлення, поставок і рівня запасів на складі, а також не вимагають великих обсягів вихідної інформації і складних методів контролю.

Управління запасами передбачає організацію контролю їх фактичного стану. У теорії управління запасами було розроблено основні системи управління ними, з яких найпоширенішими є модель з фіксованим обсягом (названа також модель економічного розміру замовлення) і модель з фіксованим періодом (названа також періодичною моделлю, моделлю періодичного контролю) [5, 6]. Розглянемо ці системи детальніше.

*Система управління з фіксованим розміром замовлення* (Рис. 1). У даній системі розмір замовлення на поповнення запасу є постійною величиною і здійснюється за умови зменшення наявного на складах запасу до встановленого мінімального рівня, який називають "точкою замовлення". Залежно від інтенсивності споживання матеріальних ресурсів інтервали між поставками можуть бути різними, а величина замовлення зафіксована і не змінюється ні за яких умов роботи системи [5, 7]

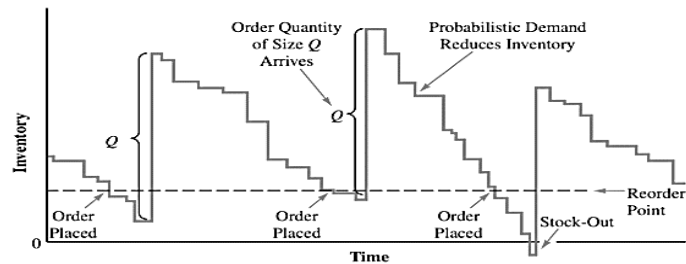


Рисунок 1. Шаблон інвентаризації для системи управління з фіксованим розміром замовлення з стохастичним попитом [8]

*Система управління з фіксованим інтервалом часу між поставками* (Рис. 2). У цій системі вхідним параметром є період часу між замовленнями. Вона зорієнтована на роботу за умови значних коливань споживання. Щоб запобігти завищенню обсягів запасів, які знаходяться на складі, або їхньому дефіциту, замовлення подаються не тільки у встановлені моменти часу, але і за умови досягнення запасом граничного рівня. Таким чином, рівень матеріального запасу регулюється як зверху, так і знизу. У тому випадку, якщо розмір запасу знижується до мінімального рівня раніше настання терміну подачі чергового замовлення, то робиться позачергове замовлення. В інший час система функціонує як система з фіксованою періодичністю замовлення. [5, 7]

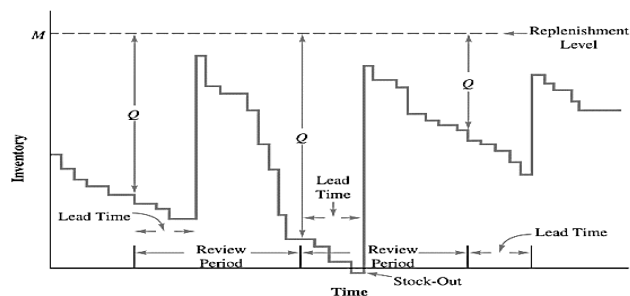


Рисунок 2. Шаблон інвентаризації для системи управління з фіксованим інтервалом часу між поставками з стохастичним попитом [8]



### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

#### 3.1 Вхідні дані

Для дослідження було сформовано вибірку з чотирьох продуктів. Параметри цих продуктів представлено у Табл 1.

Таблиця 1. Відомі параметри продуктів

№	Час виконання замовлення, дні	Об'єм, м3	Ціна закупівлі, грн	Ціна продажу, грн	Початковий запас
1	9	0.57	12	16.01	2750
2	6	0.052	7	8.6	22500
3	16	0.53	6	6	5200
4	22	1.05	37	37	1400

Попри те що змодельовати поведінку споживача досить важко, можна дати приблизну оцінку того, що клієнт має ймовірність  $p$  розмістити замовлення в будь-який день. Це  $p$  можна просто обчислити, поділивши кількість замовлень минулого року на кількість робочих днів.

Якщо у підприємця немає контракту з конкретним клієнтом виникає ще одна невизначеність – розмір замовлення. Для цього прикладу було зроблено припущення, що розмір замовлення відповідатиме логнормальному розподілу, параметри розподілу якого невідомі (що часто трапляється). Тому є важливим фіксувати історичні продажі продукту. Метою є максимізація очікуваного прибутку. За допомогою XYZ аналізу даних кожного товару можна побачити чітку різницю у попиті (Рис. 3). Наприклад, продукт 2 – це продукт великого обсягу, тобто його купують щодня ( $p_2 = 1$ ), а середній розмір його замовлення становить 649. В порівнянні продукт 4 купують лише у 24% випадків, а його середній розмір замовлення становить близько 150.

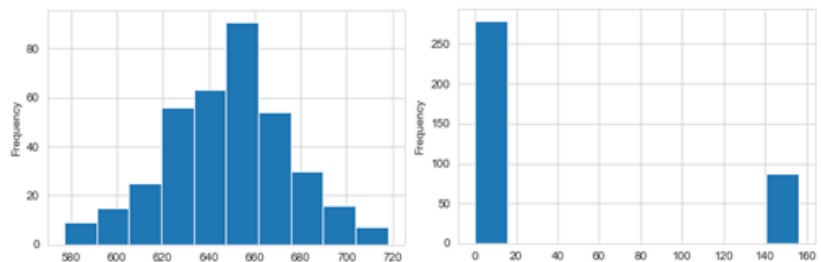


Рисунок 3. Графіки попиту продукту 2 (ліворуч) та 4 (праворуч)

У Табл. 2 надано короткий опис параметрів кожного продукту, які можна розрахувати виключно на основі минулих даних про продажі

Таблиця 2. Короткий опис параметрів кожного продукту

Продукт	1	2	3	4
Середнє значення	103.5	648.6	201.7	150
Стандартне відхилення	37.32	26.45	31.08	3.22
Очікувана пропорція днів	0.76	1	0.7	0.24
Очікуваний попит (Під час виконання замовлення)	705	3891.3	2265.8	784.8
Стандартне відхилення (Під час виконання замовлення)	165	64.8	383.3	299.7

Для моделювання поведінки попиту та розрахунку прибутку для однієї реалізації було проведено моделювання за методом Монте-Карло. Під час моделювання алгоритм повторює кожен день, намагаючись визначити рівень запасів продукту. Це необхідно для створення попиту на продукт на цей день. Це моделювання проводилося 10000 разів, щоб отримати кілька реалізацій прибутку та частки втрачених замовлень кожного разу.

### 3.2 Результати симуляції системи з фіксованими інтервалами часу між замовленнями

За результатами симуляції було побудовано гістограми та розраховано середнє значення і середнє відхилення прибутку та частки втрачених замовлень для конкретної точки замовлення (M). Так само цей експеримент можна виконати для діапазону значень M, щоб визначити значення, яке дає нам найбільший очікуваний прибуток. Як ви можете бачити на малюнку нижче для продукту 1, ми змодельовали результати для діапазону значень від 1000 до 3000. Це дало оптимальний очікуваний прибуток у розмірі 87 863 грн. Рисунок 4 нижче демонструє результати цих симуляцій для продукту 1 до точки 2071.

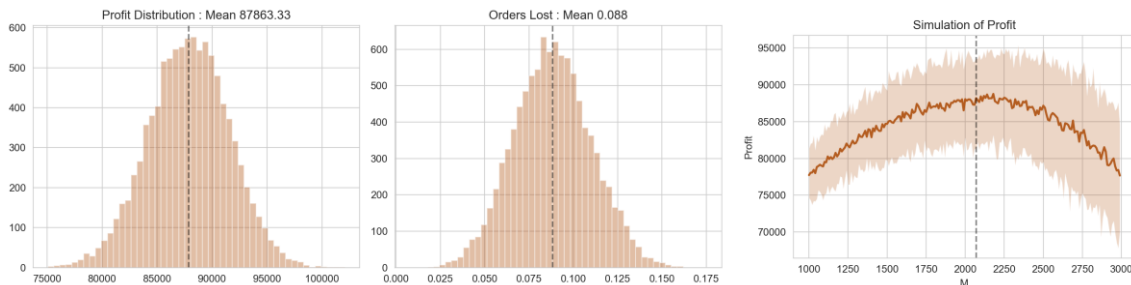


Рисунок 4. Гістограми для прибутку (ліворуч) та частки втрачених замовлень (посередині), графік моделювання для M (праворуч) для продукту 1

Використовуючи місячний інтервал перевірки, у таблиці 3 нижче наведено оптимальні точки замовлень, очікуваний річний прибуток і частку втрачених замовлень протягом року для кожного з продуктів.

Таблиця 3. Результати симуляції системи з фіксованим інтервалом замовлень

	1	2	3	4
Оптимальна точка замовлення	2 071	18 424	4 154	1 305
Очікуваний річний прибуток, грн	87 992	372 188	165 959	320 507
Очікуване стандартне відхилення, грн	5 035	1 454	6 561	29 864
Очікувана частка втрачених замовлень	0.077	0.01	0.05	0.074

### 3.3 Результати симуляції системи з фіксованими розміром замовленнями

За результатами симуляції було побудовано гістограми та розраховано середнє значення і середнє відхилення прибутку та частки втрачених замовлень для кількості замовлення 2002 і точки повторного замовлення 812 (Рис. 5) Виконуючи Grid Search між діапазоном значень для кожного продукту, можна розробити комбінацію кількості замовлення та точок повторного замовлення, які оптимізують функцію прибутку. Як можна бачити на рисунку 5 для продукту 1, значення були змодельовані для кількості замовлень від 1000 до 3000 і точки повторного замовлення між 500 і 1100. Схоже, що на графіку відображається опукла вгору функція, що означає максимальну точку прибутку.



Рисунок 5. Гістограми для прибутку (ліворуч) та частки втрачених замовлень (посередині), результати Grid Search (праворуч) для продукту 1

Використовуючи цю політику, у таблиці 4 нижче наведено оптимальні точки повторного замовлення, оптимальну кількість замовлення, очікуваний річний прибуток і частку втрачених замовлень протягом року для кожного продукту.

Таблиця 4. Результати симуляції системи з фіксованим розміром замовлень

	1	2	3	4
Оптимальна точка замовлення	2002	21150	3318	1099
Оптимальна точка повторного замовлення	812	4036	2877	1221
Буферний запас	107	145	611	436
Очікуваний річний прибуток, грн	110,174	501,621	196,235	378,896
Очікуване стандартне відхилення, грн	8,814	25,860	10,439	40,724
Очікувана частка втрачених замовлень	0.028	0.01	0.004	0.019

### 3.4. Фінальний огляд результатів

В завершення огляду результатів складемо таблицю 5, в якій можна чітко помітити, що система з фіксованим розміром замовлення перевершує систему з фіксованим інтервалом з точки зору очікуваних прибутків для кожного продукту.

Таблиця 5. Порівняння прибутку між Системою з фіксованим інтервалом замовлення та Системою з фіксованим розміром

	Фіксований інтервал	Фіксований розмір
<b>1</b>	87,992.00	110,174.00
<b>2</b>	372,188.00	501,621.00
<b>3</b>	165,959.00	196,235.00
<b>4</b>	320,507.00	378,896.00
<b>Разом</b>	<b>946 646.00</b>	<b>1,186,926.00</b>

Загальний очікуваний прибуток від системи з фіксованим розміром замовлення на 25% вищий, ніж від системи з фіксованим інтервалом часу між замовленнями.

## 4. ВИСНОВКИ

Порівняння двох найрозповсюдженіших систем управління запасами призводить до висновку про наявність у них недоліків та переваг.

Система з фіксованим розміром замовлення потребує безперервного обліку поточного запасу складі. Це призводить до підвищення витрат. Однак максимальний бажаний запас у цій системі завжди буде меншим, ніж у системі з фіксованим інтервалом часу між замовленнями. Це призводить до економії витрат утримання запасів складі з допомогою

скорочення площ, займаних запасами, а також забезпечує гнучкість щодо часу розміщення замовлення. Оскільки в системі з фіксованим розміром є дві важливі змінні тоді, коли в періодичній системі є лише одна, простір рішень для системи з постійним розміром потенційно буде більшим.

Система з фіксованим інтервалом часу між замовленнями потребує лише періодичного контролю рівня запасу. Це призводить до підвищення рівня максимального бажаного запасу в порівнянні з системою з фіксованим розміром замовлення, але скорочує витрати у зв'язку з використанням системи. Крім того вона дозволяє підприємству краще прогнозувати замовлення, зроблені протягом певного періоду часу.

Якщо вибір системи полягає суто в максимізації очікуваного загального прибутку, за результатами дослідження можна зробити висновок що система з фіксованим розміром замовлення є найкращим вибором з двох.

### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Корпоративная логистика. 300 ответов на вопросы профессионалов / В. І. Сергіїв та ін. ; ред. В. І. Сергіїв. Москва :ИНФРА-М, 2005. 975 с. URL: <https://www.zagorskaya.info/wp-content/uploads/2016/09/Сергеев-логистика-300-ответов.pdf> (дата звернення: 25.10.2022).

2. Тюнін Д. К. Эволюция развития систем управления запасами. Управление экономическими системами: электронный научный журнал. 2013. URL: <https://cyberleninka.ru/article/n/evolyutsiya-razvitiya-sistem-upravleniya-zapasami/viewer> (дата звернення: 26.06.2022).

3. Найпоширеніші помилки в управлінні запасами, які припускають більшість підприємств - KLS. KLS. URL: <https://kyivlogisticsschool.com/news/najposhirenishi-pomilki-v-upravlinni-zapasami/> (дата звернення: 19.11.2022).

4. Бродецкий Г. Л. Управление запасами. Москва : Эксмо, 2007. 282 с.

5. Зінченко Є. В. Переваги та недоліки основних систем управління запасами підприємства / Є. В. Зінченко // Розвиток європейського простору очима молоді: економічні, соціальні та правові аспекти : матеріали Всеукраїнської науково-практичної конференції докторантів, молодих учених та студентів, м. Харків, 28 квітня 2017 року. – Х. : ФОП Панов А. М., 2017. – С. 538 – 543.

6. Ткаченко Н. М. Бухгалтерський фінансовий облік на підприємствах України. Київ : А.С.К., 2001. 784 с.

7. Управління запасами. Міжнародний університет бізнесу та нових технологій. URL: <http://www.studfiles.ru/preview/3000222/page:2/>(дата звернення: 25.10.2022).

8. An introduction to management science: quantitative approaches to decision making / D. R. Anderson та ін. 14-те вид. Cengage Learning, 2015. 912 с.

# ІНТЕГРУВАННЯ ПУАССОНІВСЬКИХ ВИПАДКОВИХ МІР ТА ЗНАХОДЖЕННЯ ЇХ АСИМПТОТИК ТА ОЦІНКА ПАРАМЕТРІВ ПУАССОНІВСЬКИХ ВИПАДКОВИХ ПОЛІВ

Пилипенко А.Ю.<sup>1</sup>, Горбенко Н.О.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>2</sup>sobolonia@gmail.com

**Багато прикладних задач, які зустрічаються сьогодні застосовують моделі процесів Пуассона. Проте розробка багатьох важливих моделей і методів ще не завершена до кінця. Мета цієї роботи полягає в розв'язанні наступних задач для пуассонівських точкових випадкових мір: знаходженні максимальних значень, побудові консистентних оцінок параметрів.**

**Ключові слова: пуассонівські точкові випадкові міри, рекорди, статистика випадкових процесів.**

Теорія екстремальних значень – це елегантна та математично захоплююча теорія, а також наука, який пронизує величезну різноманітність застосувань. Розглянемо такі приклади: станції моніторингу забруднення повітря, які розташовані в різних місцях міста, урядові постанови вимагають, щоб концентрація забруднення, виміряна на кожному місці, була нижче певного рівня, тому якраз ці максимальні значення нас і цікавлять; гірничодобувна компанія відбирає зразки керна в певному регіоні, продовження буріння відбуватиметься в напрямку максимального зосередження руди; в спортивних змаганнях нецікаві другі, треті місця, а цікавлять в першу чергу рекорди України, світові і т.п. [1].

Загальною рисою цих ситуацій є те, що найбільший інтерес представляють найбільші або найменші значення спостережень. Рішення в прикладних задачах іноді приймаються на основі поведінки екстремальних значень [2].

Сам процес Пуассона є одним із найважливіших стохастичних процесів, і тому його часто вважають першою математичною моделлю в багатьох випадках. Існує велика кількість літератури про застосування моделей процесів Пуассона в різних областях (астрономії, біології, аналізі зображень, медицині, фізиці, теорії надійності тощо). Водночас ідентифікація багатьох важливих пуассонівських моделей (як і загальна теорія оцінювання) ще недостатньо розроблена, і така спроба допомогла б покрити цю прогалину [3].

## 1. ПОСТАНОВКА ПРОБЛЕМИ

Мета роботи полягає в дослідженні пуассонівських випадкових мір, а саме: знаходження асимптотичної поведінки екстремальних значень, побудові консистентних оцінок параметрів полів, тощо.

## 2. ПРИКЛАДИ ДОСЛІДЖЕННЯ

Наведемо приклад знаходження розподілу максимуму від випадкової кількості доданків.

### Приклад 1.

Нехай  $\xi_k$  і  $\eta_k$  – випадкові незалежні величини.

$\{\xi_k\} \sim \text{Geom}(\alpha)$ , а  $\{\eta_k\} \sim \text{Geom}(\beta)$ .

Знайти функцію розподілу  $\max_{1 < k \leq \eta} \xi_k$ .

Розв'язання

$$\begin{aligned}
 P \max_{1 < k \leq \eta} \xi_k \leq x &= \sum_{n=0}^{\infty} P \eta = n, \max_{1 < k \leq n} \xi_k \leq x = \sum_{n=0}^{\infty} P(\eta = n) P(\max_{1 < k \leq n} \xi_k \leq x) \\
 &= \sum_{n=0}^{\infty} P(\eta = n) F_{\xi}^n x \\
 P \eta = n &= q^{n-1} p, n \geq 1; \\
 P \xi = k &= \beta^{k-1} \alpha, k \geq 1.
 \end{aligned}$$

$$\begin{aligned}
 F_{\max_{1 < k \leq \eta} \xi_k}(x) &= P \max_{1 < k \leq \eta} \xi_k \leq x = \sum_{n=1}^{\infty} q^{n-1} p F_{\xi}^n x = \begin{matrix} \text{геометрична} \\ \text{прогресія} \end{matrix} = \frac{p F_{\xi} x}{1 - q F_{\xi} x} \\
 &= \frac{p(1 - \beta^x)}{1 - (1-p) 1 - \beta^x}.
 \end{aligned}$$

Відповідь:  $F_{\max_{1 < k \leq \eta} \xi_k}(x) = \frac{p(1 - \beta^x)}{1 - (1-p) 1 - \beta^x}$ .

### 3. ОСНОВНІ ЗАДАЧІ

#### Задача 1 (Про забруднення).

Пуассонівське випадкове поле на  $[0; T] \times R$  з інтенсивністю  $\frac{c}{|x|^\alpha} dt dx, \alpha > 1, t \in [0; T], x \in R$ . Нехай  $t$  – моменти забруднення,  $x$  – величина забруднення.

Побудувати консистентну оцінку для  $c$  та  $\alpha$ , тобто якусь функцію, яка прямує до  $c$  та  $\alpha$  при  $T \rightarrow \infty$ .

Розв'язання

Для кращої візуалізації наведемо рисунок 1.

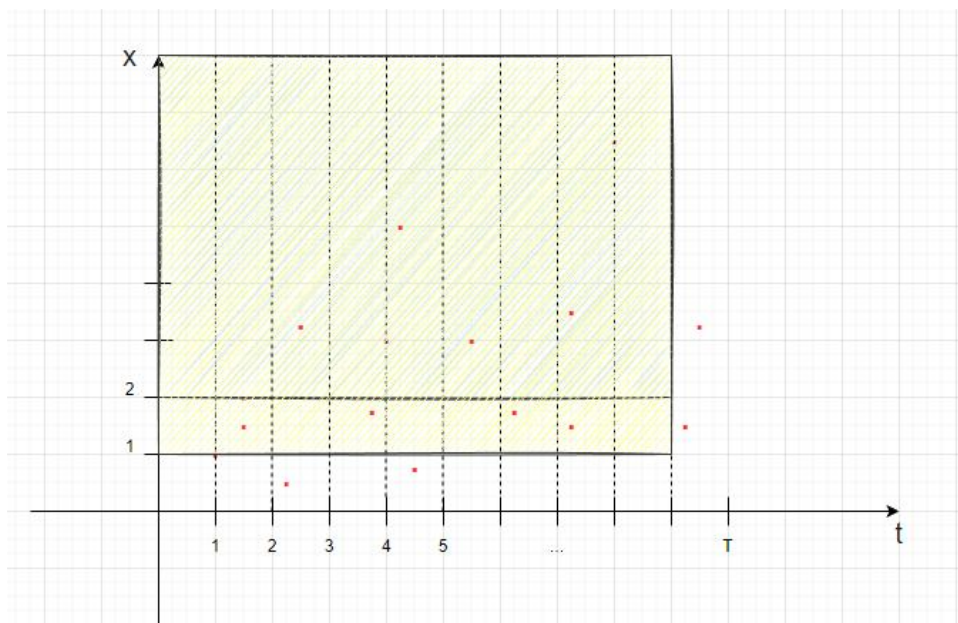


Рисунок 1. Ілюстрація розв'язку

Позначимо через  $\nu_1$ - інтенсивність на полосі  $0; 1 \times [1; \infty)$  та  $\nu_2$  - інтенсивність на полосі  $0; 1 \times [2; \infty)$  та знайдемо їх:

$$\nu_1 = c \int_0^1 \int_1^\infty \frac{1}{|x|^\alpha} dx dy = c \int_0^1 \frac{x^{-\alpha+1}}{-\alpha+1} \Big|_1^\infty dx = \text{при } \alpha > 1 = c \int_0^1 \frac{1}{\alpha-1} dx = \frac{cx}{\alpha-1} \Big|_0^1 = \frac{c}{\alpha-1}.$$

$$\nu_2 = c \int_0^1 \int_2^\infty \frac{1}{|x|^\alpha} dx dy = c \int_0^1 \frac{x^{-\alpha+1}}{-\alpha+1} \Big|_2^\infty dx = \text{при } \alpha > 1 = c \int_0^1 \frac{2^{-\alpha+1}}{\alpha-1} dx = \frac{c2^{-\alpha+1}x}{\alpha-1} \Big|_0^1 = \frac{c2^{-\alpha+1}}{\alpha-1}.$$

Нехай  $E_1 = E_1(n)$  - це кількість точок, які потрапляють в полосу  $0; 1 \times [1; \infty)$ ,  $E_2 = E_2(n)$  - це кількість точок, які потрапляють в полосу  $0; 1 \times [2; \infty)$ .

$$\text{За законом великих чисел } \frac{E_1}{n} \rightarrow \frac{c}{\alpha-1}, \frac{E_2}{n} \rightarrow \frac{c2^{-\alpha+1}}{\alpha-1}.$$

$$\frac{E_2}{E_1} \rightarrow 2^{-\alpha+1},$$

$$\log_2 \frac{E_2}{E_1} \rightarrow -\alpha + 1,$$

$$\alpha^* \rightarrow 1 - \log_2 \frac{E_2}{E_1}, (1.1)$$

$$c^* \rightarrow \frac{(\alpha-1)E_1}{n}, \text{ де } \alpha \text{ беремо з (1.1).}$$

Відповідь:  $\alpha^* \rightarrow 1 - \log_2 \frac{E_2}{E_1}, c^* \rightarrow \frac{E_1 \log_2 \frac{E_2}{E_1}}{n}$  - консистентні оцінки.

## Задача 2.

Спостерігається Пуассонове випадкове поле на  $[0; T] \times \mathbb{R}$  з щільністю

- 1)  $\frac{1}{x^\alpha}, \alpha > 1,$
- 2)  $e^{-|x|},$

криві  $x = t^\beta, x = a \ln t.$

Яка кількість екстремальних точок скінченна чи нескінченна буде потрапляти в заштриховану зону кривої? Позначимо  $X(T)$  - максимальна точка пуассонівського поля на відрізьку  $[0, T]$ . Знайти  $\lim_{T \rightarrow \infty} \frac{X(T)}{x(T)}$ .

### Розв'язання

Для кращої візуалізації наведемо рис.2 та рис. 3:

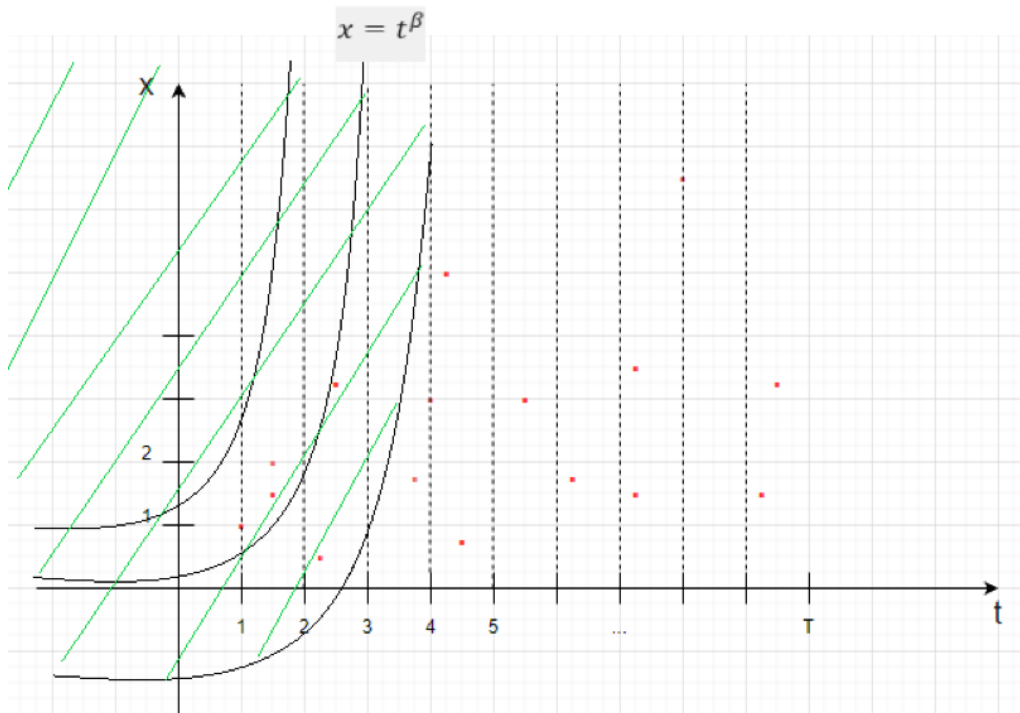


Рисунок 2. Ілюстрація для кривої  $x = t^\beta$

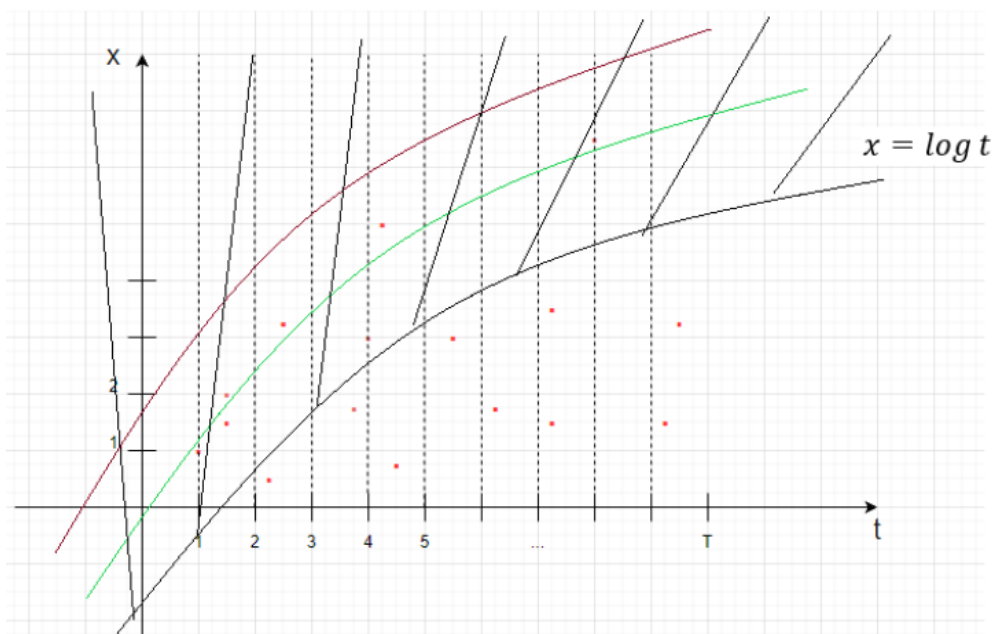


Рисунок 3. Ілюстрація для кривої  $x = a \log t$ .

Наведемо два розв'язки з чотирьох, де буде скінченна і нескінченна кількість точок.

$$1) \int_1^\infty \int_{a \ln t}^\infty e^{-|x|} dx dt = \int_1^\infty -e^{-x} \Big|_{a \ln t}^\infty dt = \int_1^\infty \frac{1}{t^a} dt = \frac{t^{a-1}}{a-1} \Big|_1^\infty = \frac{1}{1-a}, \quad \text{при}$$

$a < 1$  скінченна кількість точок лежить в цій зоні, отже max теж буде скінченна кількість точок і нескінченна при  $a > 1$ ;



2) 
$$\int_1^{\infty} \int_{t^{\beta}}^{\infty} \frac{1}{x^{\alpha}} dx dt = \int_1^{\infty} \frac{x^{-\alpha+1}}{-\alpha+1} \Big|_{t^{\beta}}^{\infty} dt = \int_1^{\infty} \frac{t^{(-\alpha+1)\beta}}{-\alpha+1} dt = \frac{t^{-\alpha+1} \beta+1}{(-\alpha+1)(-\alpha+1) \beta+1} \Big|_1^{\infty} = \frac{1}{(\alpha-1)(-\alpha+1) \beta+1}$$
, при  $-\alpha\beta > \beta + 1$ . Отже, скінченна кількість точок лежить в цій зоні, отже max теж буде скінченна кількість точок.

Відповідь: 1) при щільності  $e^{-|x|}$  та кривій  $\ln t$  скінченна кількість точок лежить вище від кривої та  $\lim_{T \rightarrow \infty} \frac{X(T)}{\ln t} = 1$  при  $a < 1$  і нескінченна кількість при  $a > 1$ ; 2) при щільності  $\frac{1}{x^{\alpha}}$  та кривій  $t^{\beta}$  скінченна кількість точок лежить вище від кривої для  $\beta > \frac{1}{1+\alpha}$  та  $\lim_{T \rightarrow \infty} \frac{X(T)}{x(T)} = 0$ , а для  $\beta < \frac{1}{1+\alpha}$  – нескінченна кількість точок  $\lim_{T \rightarrow \infty} \frac{X(T)}{x(T)} = \infty$ .

#### 4. ВИСНОВКИ

Наведені методи досліджень найбільших значень є дуже важливими, адже ця галузь не є дослідженою до кінця. Саме ця тема застосовується дуже часто для забезпечення високоякісних результатів, щоб застосувати їх в багатьох сферах людської діяльності, наприклад для дослідження рівня забруднень, як було наведено вище, а це може бути дуже корисним для збереження нашої екології.

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Yu. A. Kutoyants *Statistical Inference for Spatial Poisson Processes*. – 233 Spring Street, New York: Springer, 1998, 278 p.
2. Sidney I. Resnick *Extreme Values, Regular Variation, and Point Processes* – 233 Spring Street, New York: Springer, 2007, 320 p.
3. Sidney I. Resnick *Heavy-Tail Phenomena. Probabilistic and Statistical Modeling* – , 233 Spring Street, New York: Springer, 2007, 404 p.

# ДОСЛІДЖЕННЯ СТОХАСТИЧНИХ СИСТЕМ ЗА ЧАСТКОВИМ СПОСТЕРЕЖЕННЯМ

Пилипенко А.Ю.<sup>1</sup>, Ковальчук О.О.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> pilipenko.ay@gmail.com [0000-0002-8786-1617],  
<sup>2</sup> olelchik1770@gmail.com

**Об'єктом дослідження є параметри, які описують популярність блогера. Ми припускаємо, що популярність блогера модулюється ланцюгом Маркова народження і загибелі. Популярність ми не можемо спостерігати, на відмінну від кількості лайків та дизлайків, що мають залежний від популярності пуассонівський розподіл. Результатом роботи є побудовані консистентні оцінки для параметрів популярності. Отримані результати можна використовувати при роботі з маркетингом впливу.**

**Ключові слова: ланцюги Маркова, умовні розподіли, консистентна оцінка, закон великих чисел.**

## 1. ВСТУП

Кількість бізнесів постійно зростає, і усі вони конкурують за увагу людей. Маркетинг є однією з ключових організацій будь-якого бізнесу, і саме вони відповідальні за залучення нових клієнтів. Для цього вони використовують різноманітні джерела трафіку, але кожен з них, на жаль, не безмежний. Тому з появою соціальних мереж і популярністю блогерів, маркетологи побачили потенціал для створення нового типу реклами. Цей спосіб залучення клієнтів отримав назву маркетинг впливу.

Працюючи з маркетингом впливу маркетолог повинен обрати правильного блогера серед тисячі інших для розміщення реклами бренду. Зазвичай для цього маркетологи збирають певну статистику соціальних мереж блогера, а саме, кількість підписників, лайків, коментарів, охоплення публікації. Усі ці статистики є випадковими і постійно змінюються у часі, тому важко спрогнозувати, які ж будуть показники на момент виходу реклами бренду.

Для оцінки блогера ми вирішили досліджувати саме його популярність, і з якою інтенсивністю вона змінюється, спостерігаючи таку статистику як кількість лайків та дизлайків під останніми публікаціями у соціальних мережах.

## 2. ПОСТАНОВКА ЗАДАЧІ

Розглянемо наступну задачу: потрібно обрати найкращого блогера для реклами бренду. Кожен блогер має певну популярність – кількість людей, які знають даного блогера, зацікавлені у ньому і потенційно готові відвідати його сторінки у соціальних мережах. Чим більше людей відвідає соціальні мережі блогера, тим більше буде охоплення у реклами бренду.

Припустимо, що популярність блогера випадковим чином змінюється у часі. Наприклад, певний час популярність зростає, а потім спадає.

Дані про блогера, які ми можемо спостерігати, є кількість переглядів, лайків чи дизлайків минулих його публікацій. Маючи ці дані, потрібно прийняти рішення, чи купувати рекламу бренду у певного блогера.

Припустимо, що популярність блогера – це випадковий процес  $\xi(t)$ , що описується ланцюгом Маркова народження і загибелі з параметром  $p$ . Параметр  $p$  визначає, з якою інтенсивністю зростає популярність блогера,  $q = 1 - p$  – інтенсивність спадання. Різниця лайків та дизлайків у момент  $t$  – це випадкова величина  $\eta(t)$ , що має розподіл Пуассона з випадковим параметром  $\xi(t)$ :  $\eta \sim \text{Pois}(\xi(t))$ . Якщо  $\xi(t)$  набуває від’ємне значення  $-n$ , вважатимемо, що  $\eta \sim -\text{Pois}(n)$  (якщо кількість дизлайків більша ніж кількість лайків). При додатніх  $\xi$  величина  $\eta$  відображає наскільки кількість лайків більша за кількість дизлайків.

Зауважимо, що пуассонівський розподіл випадкової величини  $\eta$  є цілком природнім явищем і відповідає граничній теоремі Пуассона, якщо кількість лайків і дизлайків не дуже багато. При великій кількості лайків, скоріше за все,  $\eta$  матиме нормальний розподіл:  $\eta \sim \mathcal{N}(\xi(t), \xi(t))$ .

Потрібно, спостерігаючи випадкові величини  $\eta \sim \text{Pois}(\xi(t))$ , оцінити популярність блогера  $\xi(t)$ , а саме параметр  $p$ , за яким популярність  $\xi(t)$  збільшується у часі.

Розглянемо три різні варіанти розподілів популярності  $\xi(t)$ :

- a) Ланцюг Маркова народження і загибелі на  $\{0, \dots, m\}$  – зі скінченною кількістю станів (Рис. 1);

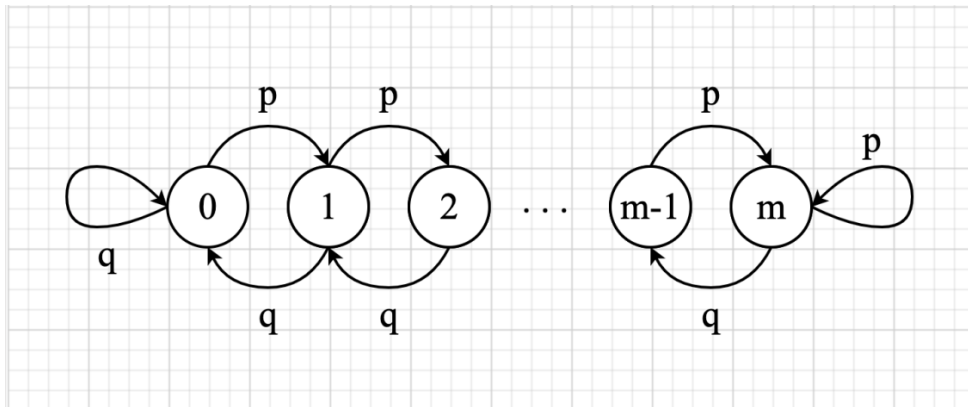


Рисунок 1. Ланцюг Маркова народження і загибелі на  $\{0, \dots, m\}$

- b) Ланцюг Маркова народження і загибелі на  $\{0, \dots, +\infty\}$  – з необмеженою кількістю станів, якщо  $p < \frac{1}{2}$  (Рис. 2);
- c) Ланцюг Маркова народження і загибелі на  $\{-\infty, \dots, +\infty\}$  – з необмеженою кількістю станів, з притяганням до середнього значення популярності  $\alpha = 0$  (Рис. 3).

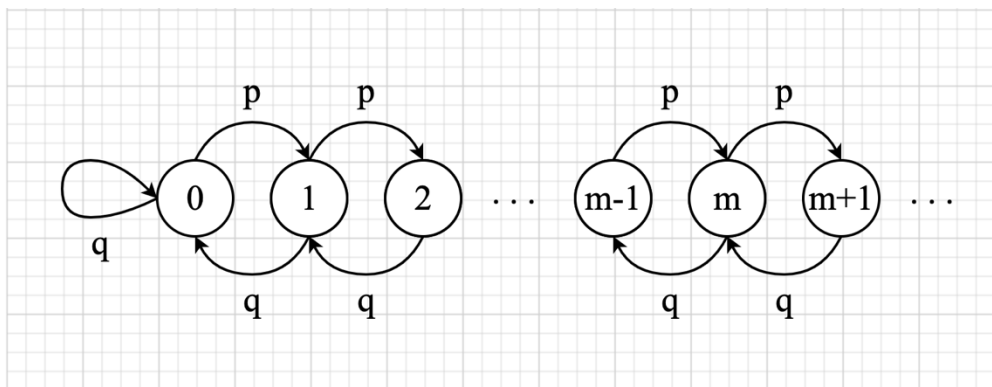


Рисунок 2. Ланцюг Маркова народження і загибелі на  $\{0, \dots, +\infty\}$

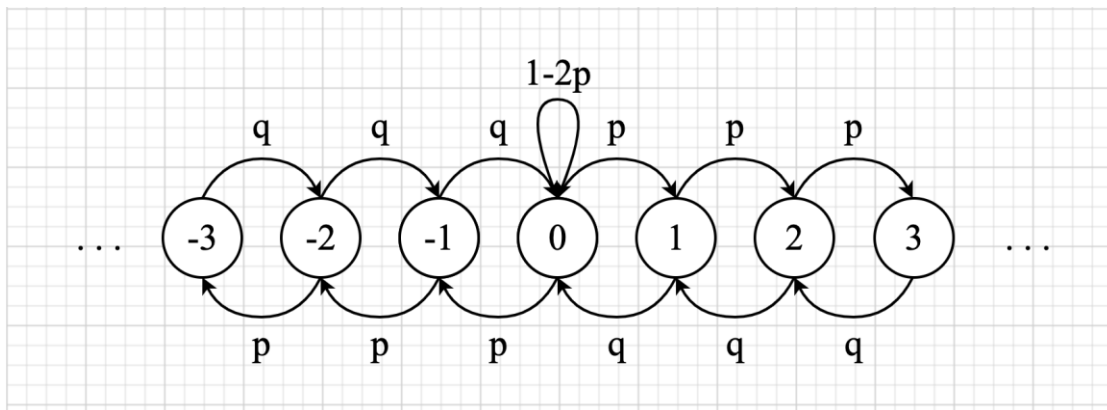


Рисунок 3. Ланцюг Маркова народження і загибелі на  $\{-\infty, \dots, +\infty\}$  – з притягненням до значення популярності  $\alpha = 0$

### 3. РОЗВ'ЯЗОК ЗАДАЧІ

Було проведено дослідження випадкової величини  $\eta \sim Pois \xi(t)$  за допомогою умовних розподілів. Для цього нехай у деякий момент  $t$   $\xi(t) = i$ . Тоді умовний розподіл:

$$P \eta = k \mid \xi = i = \frac{e^{-i} i^k}{k!}.$$

Виразили розподіл  $\eta$  через формулу повної ймовірності:

$$P \eta = k = \sum_i P \xi = i P \eta = k \mid \xi = i.$$

Розписали математичне очікування  $\eta$ :

$$\begin{aligned} \mathbb{E}_\eta = \sum_k k \cdot P \eta = k &= \sum_k k \cdot \sum_i P \xi = i P \eta = k \mid \xi = i = \\ &= \sum_i P \xi = i \cdot \sum_k k \cdot P \eta = k \mid \xi = i = \sum_i \pi_i \cdot \mathbb{E}_\eta \mid \xi = i, \end{aligned}$$

де:

$\pi_i$  – стаціонарний розподіл Ланцюга Маркова, що відповідає розподілу  $\xi(t)$ ,

$\mathbb{E}_\eta \mid \xi = i$  – умовне математичне сподівання  $\eta$ .

Знайшли, що  $\mathbb{E}_\eta \mid \xi = i = \mathbb{E}_\eta \mid \eta \sim Pois i = i$ .

Отже,  $\mathbb{E}_\eta = \sum_i \pi_i \cdot i$

Дослідили збіжність середнього арифметичного випадкових величин  $\eta$ . Зауважимо, що ми не можемо застосувати закон великих чисел, оскільки  $\eta_k$  розподілені неоднаково.

Нехай кількість випадкових величин  $\eta$ , що ми спостерігаємо, рівна  $n$ :

$$\frac{\sum_k \eta_k}{n} = \frac{\sum_k \eta_k \cdot \mathbb{1}_{\{\xi=i\}}}{n} = \frac{\sum_i \sum_k \eta_k \cdot \mathbb{1}_{\{\xi=i\}}}{n}.$$

Нехай  $N_i$  – кількість разів відвідування стану  $i$  за  $n$  кроків:

$$\frac{\sum_i \sum_k \eta_k \cdot \mathbb{1}_{\{\xi=i\}}}{n} = \frac{\sum_i \frac{N_i \sum_k \eta_k \cdot \mathbb{1}_{\{\xi=i\}}}{N_i} \cdot N_i}{n} = \sum_i \frac{N_i}{n} \frac{\sum_k \eta_k \cdot \mathbb{1}_{\{\xi=i\}}}{N_i}.$$

Для ланцюгів Маркова, у яких усі стани сполучні і період рівний 1, справедливе твердження  $\frac{N_i}{n} \rightarrow \pi_i$  майже напевно (м.н.). Також за законом великих чисел  $\frac{N_i \sum_k \eta_k \cdot \mathbb{1}_{\{\xi=i\}}}{N_i} \rightarrow$

$\mathbb{E}_\eta \mid \xi = i$  майже напевно. Отже,

$$\sum_{i=0}^{N_i} \frac{\eta_k \cdot \mathbb{I}_{\{\xi=i\}}}{N_i} \cdot \frac{N_i}{n} \cdot \mathbb{E}_{\eta} \xi=i \cdot \pi_i$$

Загальний результат:

$$\frac{k \eta_k}{n} \cdot \mathbb{E}_{\eta} \xi=i \cdot \pi_i = i \cdot \pi_i = \mathbb{E}_{\pi_i}(\xi)$$

Для розв'язання задачі знайшли стаціонарний розподіл ланцюга Маркова народження і загибелі, підставили його у рівняння  $\mathbb{E}_{\eta} \xi=i \cdot \pi_i$  і виразили консистентну оцінку  $\rho$  для параметру  $\rho$ .

- а) Стаціонарний розподіл для ланцюга Маркова народження і загибелі на  $\{0, \dots, m\}$  знаходиться з системи рівнянь:

$$\begin{aligned} \pi_1 &= \pi_0 \cdot \frac{p}{q} \\ \pi_2 &= \pi_0 \cdot \left(\frac{p}{q}\right)^2 \\ &\vdots \\ \pi_i &= \pi_0 \cdot \frac{p^i}{q^i} \\ \sum_{i=0}^m \pi_i &= 1 \end{aligned}$$

Після заміни  $\frac{p}{q} = \beta$  і розв'язання системи, знайшли, що стаціонарний розподіл:

$$\pi_i = \frac{\beta^i - 1}{\beta^{m+1} - 1} \cdot \beta^i$$

Для знаходження оцінки  $\rho$  потрібно розв'язати рівняння відносно параметра  $\rho$ :

$$\mathbb{E}_{\pi_i} \xi = \sum_{i=0}^m i \cdot \pi_i = \frac{\beta-1}{\beta^{m+1}-1} \cdot \sum_{i=0}^m i \cdot \beta^i = \frac{\beta^{m+1} - m + 1 - \beta}{\beta-1 \cdot \beta^{m+1}-1}, \text{ де } \beta = \frac{p}{1-p}.$$

Це можна зробити за допомогою чисельних методів обчислення.

- б) Стаціонарний розподіл для ланцюга Маркова народження і загибелі на  $\{0, \dots, +\infty\}$  – з необмеженою кількістю станів, при  $\rho < \frac{1}{2}$  знаходиться з системи рівнянь:

$$\begin{aligned} \pi_1 &= \pi_0 \cdot \frac{p}{q} \\ \pi_2 &= \pi_0 \cdot \left(\frac{p}{q}\right)^2 \\ &\vdots \\ \pi_i &= \pi_0 \cdot \frac{p^i}{q^i} \\ \sum_{i=0}^{\infty} \pi_i &= 1 \end{aligned}$$

Після заміни  $\frac{p}{q} = \beta$  і розв'язання системи, знайшли, що стаціонарний розподіл:

$$\pi_i = (1 - \beta) \cdot \beta^i$$

Для знаходження оцінки  $\rho$  для параметр  $\rho$  потрібно розв'язати рівняння:

$$\mathbb{E}_{\pi_i} \xi = \sum_{i=0}^{\infty} i \cdot \pi_i = 1 - \beta \sum_{i=0}^{\infty} i \cdot \beta^i = \frac{\beta}{(1 - \beta)}$$

Отримали, що  $\beta = \frac{\mathbb{E}_{\pi_i} \xi}{1 + \mathbb{E}_{\pi_i} \xi}$ . Нехай  $\frac{\mathbb{E}_{\pi_i} \xi}{1 + \mathbb{E}_{\pi_i} \xi} = M$ .

Тоді консистентна оцінка для  $\rho$ :

$$\rho = \frac{M}{1 + M}, \text{ де } M = \frac{\mathbb{E}_{\pi_i} \xi}{1 + \mathbb{E}_{\pi_i} \xi}.$$

с) Стационарний розподіл для ланцюг Маркова народження і загибелі на  $\{-\infty, \dots, +\infty\}$  – з необмеженою кількістю станів, з притягненням до значення популярності  $\alpha = 0$  знаходиться з системи рівнянь:

$$\begin{aligned} \pi_1 &= \pi_0 \cdot \frac{p}{q} \\ \pi_2 &= \pi_0 \cdot \left(\frac{p}{q}\right)^2 \\ &\vdots \\ \pi_i &= \pi_0 \cdot \frac{p^i}{q^i} \\ &\dots \\ \sum_{i=-\infty}^{+\infty} \pi_i &= 1 \end{aligned}$$

Після заміни  $\frac{p}{q} = \beta$  і розв'язання системи, знайшли, що стационарний розподіл:

$$\pi_i = \frac{1 - \beta}{1 + \beta} \cdot \beta^i$$

Для знаходження оцінки  $\rho$  для параметр  $\rho$  потрібно розв'язати рівняння:

$$\mathbb{E}_{\pi_i} \xi = \sum_i |i| \cdot \pi_i = 2 \sum_{i=0}^{\infty} i \cdot \beta^i = \frac{2\beta}{(1 - \beta^2)}$$

Це квадратне рівняння, додатним коренем якого є  $\beta = \frac{-2 - 2 \sqrt{1 + \mathbb{E}_{\pi_i} \xi^2}}{2\mathbb{E}_{\pi_i} \xi}$ . Нехай  $\frac{-2 - 2 \sqrt{1 + \mathbb{E}_{\pi_i} \xi^2}}{2\mathbb{E}_{\pi_i} \xi} = M$ . Тоді консистентна оцінка для  $\rho$ :

$$\rho = \frac{M}{1 + M}, \text{ де } M = \frac{-2 - 2 \sqrt{1 + \mathbb{E}_{\pi_i} \xi^2}}{2\mathbb{E}_{\pi_i} \xi}.$$

#### 4. ВИСНОВКИ

У роботі було проведено дослідження 3 видів ланцюгів Маркова народження та загибелі, які моделюють популярність блогера. Параметр  $p$  відображає, з якою інтенсивністю зростає популярність. Спостерігаючи такі випадкові величини як кількість лайків та дизлайків, різниця яких розподілена за залежним від популярності пуассонівським розподілом, було знайдено консистентні оцінки  $\rho$  для параметра  $p$ .

Для дослідження були використані знання з тем ланцюги Маркова народження та загибелі, стационарні розподіли, гранична теорема Пуассона, закон великих чисел та умовні розподіли.

Отриманні результати можна використовувати в маркетингу, а саме при роботі з маркетингом впливу.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gut A. An intermediate course course in probability, 2nd edn. NewYork: Springer New York, 2009. 319 с.
2. Гринь А.Г. Цепи Маркова: учеб. пособ. Омск: Омский гос. ун-т им. Ф.М. Достоевского, 2019. 42 с.
3. Феллер В. Введение в теорию вероятностей и её приложения: в 2-х т. / пер. с англ. Ю.В. Прохоров. Москва: Мир, 1967. Т. 2. 752 с.
4. Brémaud P. Recurrence of markov chains. *Discrete probability models and methods*. Cham, 2017. С. 145–183.
5. Турчин В. М. Теорія ймовірностей і математична статистика. Дніпропетровськ: ІМА-прес, 2014. 556 с.
6. Чжун К.Л. Однородные цепи Маркова / ред. С.Х. Сираждинов; пер. В.Ф. Колчин. Москва: Мир, 1964. 426 с.

# ВИЯВЛЕННЯ ШАХРАЙСЬКИХ ОПЕРАЦІЙ МЕТОДАМИ МАШИННОГО НАВЧАННЯ

Позняк Д.І.<sup>1</sup>, Кузнєцова Н.В.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>pozniakdi@gmail.com,

<sup>2</sup>n.kuznietsova@kpi.ua [0000-0002-1662-1974]

**Виникнення шахрайських операцій з платіжними картками є одним з небажаних сценаріїв використання платіжних систем, тому для їх своєчасного виявлення доцільно використовувати методи машинного навчання, одним із результатів застосування яких може стати інформаційна технологія з класифікації транзакцій. Метою роботи є дослідження і розробка методів виявлення шахрайських операцій. Результатом дослідження є інформаційна технологія з виявлення шахрайств. У роботі було використано теоретичні та емпіричні методи дослідження.**

**Ключові слова:** шахрайські операції, машинне навчання, інформаційна технологія, налаштування гіперпараметрів.

## 1. ВСТУП

Шахрайство - це заволодіння чужим майном або придбання права на майно шляхом обману чи зловживання довірою. Із розповсюдженням електронних грошових переказів та оплати товарів банківськими картками, на жаль, зростає також і кількість шахрайських операцій у фінансовій сфері [1]. Своєчасне виявлення та протидія шахрайським операціям є значущим питанням серед сучасних надавачів платіжних послуг по всьому світу, адже це один із ключових аспектів підтримки роботи систем гарантування платіжних операцій.

Зважаючи на це, актуальним є дослідження транзакцій у платіжних системах для виявлення факторів впливу на появу шахрайських операцій, моделювання систем виявлення шахрайств, і відповідно зниження ризику проведення шахрайських операцій у платіжних системах. При цьому, є важливим врахування усіх особливостей вихідних даних під час їх аналізу та ретельної підготовки їх до використання у алгоритмах машинного навчання. Цього можна досягнути шляхом комбінування відомих технік аналізу даних із емпіричними дослідженнями. Використання даних підходів дозволяє збільшувати прогностну здатність моделей, що будуються, а отже - створювати ефективні інформаційні технології для виявлення шахрайських операцій.

## 2. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Основною задачею дослідження є розробка інформаційної технології для виявлення шахрайських транзакцій. Дане питання містить в собі ряд підзадач, серед яких: розвідковий аналіз даних, підготовка даних до навчання моделей, навчання моделей наряду з налаштуванням їх гіперпараметрів та аналіз результатів моделювання. Варто зазначити, що виявлення шахрайських транзакцій у масиві даних методами машинного навчання являє собою задачу бінарної класифікації. Тобто приймаючи на вхід вибірку з записів-транзакцій, результатом роботи моделі буде спрогнозоване значення змінної-маркера шахрайства



(назвемо її isFraud) для кожної транзакції. При цьому змінна isFraud є бінарною - такою, що приймає значення з множини  $\{0,1\}$ .

### 3. ОГЛЯД ОСНОВНИХ МЕТОДІВ ДОСЛІДЖЕННЯ

Виявлення шахрайських операцій, будучи актуальною проблемою сучасної банківської та фінансової сфери, за останній час було досліджено у багатьох роботах. Одними із найбільш використовуваних методів у даному питанні є інтелектуальний аналіз даних, наряду із активним залученням автоматизації процесів [2]. Зокрема деякі дослідники демонструють ефективність методів машинного навчання з учителем [3].

Навчання з учителем має на увазі алгоритми, що навчаються на спостереженнях типу “стимул-відповідь” таким чином, щоб потім визначати “відповідь” для “стимулів”, що не входять до навчальної множини спостережень. Популярними алгоритмами навчання з учителем для вирішення задачі класифікації є випадковий ліс та варіації градієнтного бустингу. Дані алгоритми, також відомі як ансамблеві, дозволяють боротись із проблемою перенавчання моделей - явищем, за якого модель не здатна здійснювати надійні передбачення на загальних даних, крім тих, що використовувались для навчання моделі. Ідея роботи ансамблевих алгоритмів полягає у комбінуванні декількох алгоритмів машинного навчання, що прийнято називати базовими алгоритмами, або слабкими моделями. Так, бустинг та випадковий ліс в якості базових алгоритмів зазвичай використовують дерева рішень. Основною відмінністю бустингу від випадкового лісу є послідовне навчання слабких моделей таким чином, щоб вони виправляли помилки попередніх і підвищували якість усього ансамблю, в той час як у випадковому лісі дерева рішень навчаються незалежно, після чого класифікуючи об’єкт шляхом голосування.

Варто зазначити, що дані, на яких навчаються алгоритми виявлення шахрайських операцій, у різних дослідженнях на цю тему є досить схожими - містять схожі набори відомостей про транзакції [4]. Це пояснюється тим, що дані, зібрані під час операції з кредитною картою, мають відповідати міжнародним стандартам фінансової звітності. Наприклад, моделі для виявлення шахрайств з картками у банківських установах часто є чутливими до зміни такого параметру як збільшення кредитного ліміту а також ліміту оплати в інтернеті та дозвіл на платежі за кордоном [5]. На жаль, у наборах даних у відкритому доступі дана інформація зазвичай не представлена, оскільки це суперечило б договору про захист персональних даних у банківських установах. Для побудови більш точних моделей дослідники зазвичай використовують feature engineering - конструювання нових ознак на основі існуючого датасету. Так, наприклад, знаючи час транзакції можна створити додаткові категоріальні змінні, що вказуватимуть на належність транзакції до певного дня тижня, місяця, свята, тощо. Метою агрегації змінних є виявлення зв’язків між вихідними змінними, які може бути важко виявити іншим чином. Конструювання змінних потрібно проводити з обережністю, щоразу валідуючи роботу моделі. До feature engineering також відносять стандартизацію даних, label та one-hot encoding.

Щоб порівняти роботу навчених класифікаторів між собою, за прогнозами моделей на тестових вибірках можна обчислити різноманітні метрики. Одним із показників якості роботи моделі є криві ROC (Receiver operating characteristic), за якими можна оцінити відношення істинно позитивних прогнозів моделі до хибно позитивних [6]. На графіку за віссю абсцисс відкладається  $FPR \left( \frac{\text{істинно позитивні}}{\text{хиб.негат.} + \text{іст.позт.}} \right)$ , а за віссю ординат -  $TPR \left( \frac{\text{хибно позитивні}}{\text{істинно негат.} + \text{хибно позит.}} \right)$ . Крива ROC може слугувати надійним методом ідентифікації класифікаторів, що потенційно є оптимальними [7]. При цьому можна також обчислити

метрику AUC, яка власне є площею під кривою ROC (Area Under Curve). Показник AUC, що дорівнює одиниці, свідчитиме, що модель класифікує дані ідеально вірно.

Ефективна робота алгоритмів машинного навчання залежить зокрема від налаштування гіперпараметрів моделей. Розглянемо  $A$  - алгоритм з  $N$  гіперпараметрів, що належать простору  $\theta = \theta_1 \times \theta_2 \times \dots \times \theta_h$ , при цьому конкретний набір гіперпараметрів для  $A$  має наступний вигляд (1). Функція втрат  $L(Y, f(X, \theta))$  дає можливість оцінити роботу алгоритму із заданим набором гіперпараметрів  $\theta$ , навченого на вибірці  $X$  з цільовою змінною  $Y$ . Таким чином можна формалізувати задачу до пошуку  $\theta^*$  (2).

$$\theta = \langle \theta_1, \theta_2, \dots, \theta_h \rangle, \theta_i \in \Theta_i \quad (1)$$

$$\theta^* = \arg \min L(Y, f(X, \theta)) \quad (2)$$

Дана процедура потребує залучення методів нелінійної оптимізації. Одним із широко використовуваних методів прямого пошуку є метод симплексного спуску, також відомий як метод Нелдора-Міда [8]. Сутність методу полягає у послідовному переміщенні та деформуванні (зовнішньому та внутрішньому скороченні, стягуванні) симплексу навколо точки екстремуму.

#### 4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для дослідження було використано набір даних з архіву компанії-надавача платіжних послуг Vesta Corporation, доступний для завантаження з ресурсу Kaggle. Навчальний набір початково складався з 590540 рядків-транзакцій та 434 змінних-інформації по кожній транзакції.

Перш ніж навчати моделі, було проведено розвідковий аналіз даних. Завдяки йому зокрема було встановлено часовий проміжок, у який відбувались усі транзакції. Так, завдяки аналізу моделей пристроїв, що було використано для здійснення транзакцій та відкритих даних про купівельну активність від компанії Google, було прийнято рішення вести відлік транзакцій у датасеті, починаючи із 1 листопада 2017 року. Це в свою чергу дало можливість сконструювати нові змінні для датасету - день тижня та час транзакції.

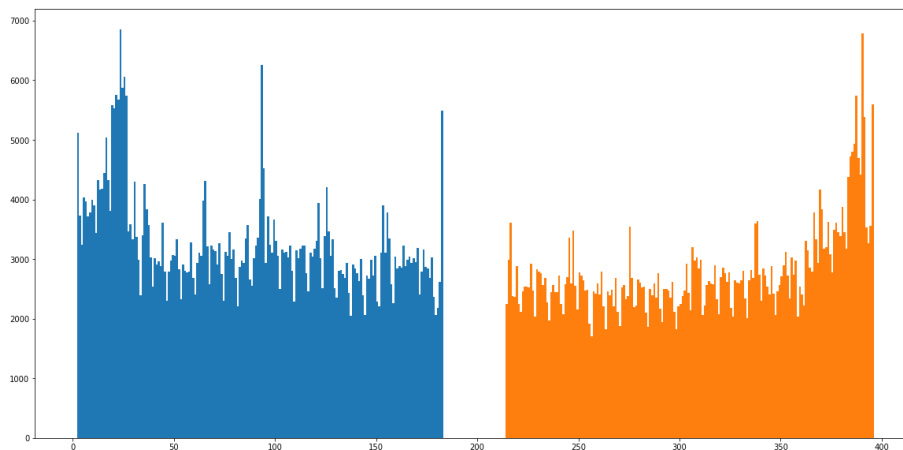


Рисунок 1. Розподіл транзакцій за днями у навчальній та тестовій вибірках

Розвідковий аналіз датасету також показав наявність мультиколінеарності серед ряду змінних. За допомогою їх групування на основі розподілу пропущених значень з вихідних 339 змінних з цією особливістю було отримано 128, майже на 60% менше. Враховуючи великий обсяг даних, крім можливості подолати проблему незначущості параметрів моделі, це позитивно вплинуло також і на швидкість навчання алгоритмів.

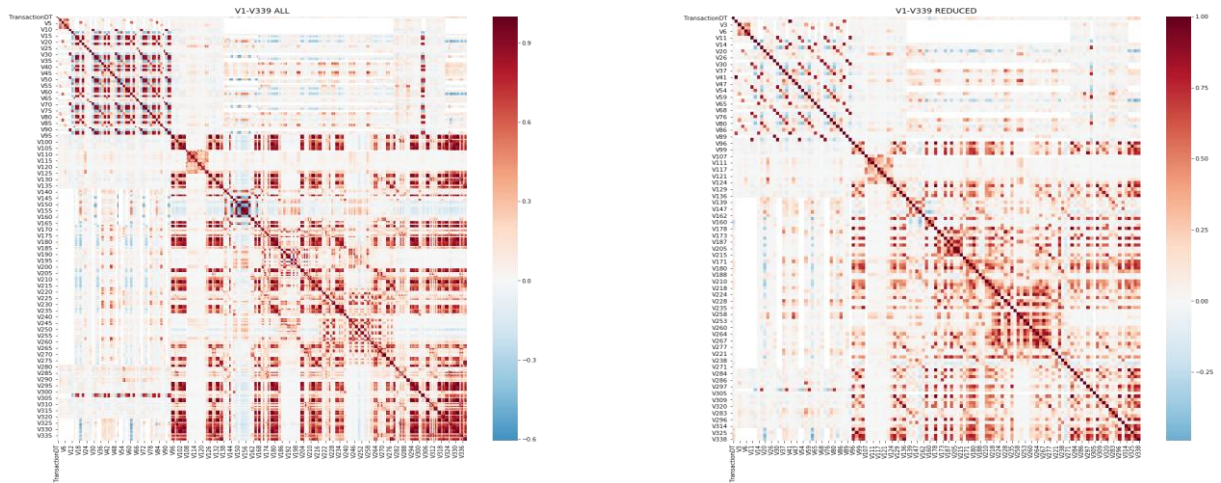


Рисунок 2. Кореляційні матриці вихідної і згрупованої вибірок

Також перш ніж навчати алгоритми, було заповнено пропуски у датасеті. Вибіркове середнє було використане для неперервних змінних, у той час як для категорійних було використано позначку NA як окрему категорію. Через виявлену незбалансованість класів у вибірці, для випадків шахрайств було використано метод генерації додаткових екземплярів мінорного класу SMOTE. Дані також було закодовано за допомогою label encoding, щоб категорійні змінні були представлені числовими значеннями. Під час навчання було застосовано стратегію 5-кратної крос-валідації. При внесенні змін у навчальні дані або модель, ефективність тих чи інших дій було оцінено за середнім значенням AUC по усіх фолдах крос-валідації.

У ході дослідження було навчено наступні моделі: випадковий ліс та модифікації градієнтного бустингу XGBClassifier та LGBM. Для градієнтного бустингу було виконано налаштування гіперпараметрів за допомогою пошуку оптимальних параметрів на заданому просторі з використанням алгоритму Нелдора-Міда. Результати роботи моделей на тестовій вибірці подано у таблиці 1.

Таблиця 1. Результати роботи моделей

№	Алгоритм	AUC
1	Random Forest Classifier	0.887
2	XGBoost Classifier	0.897
3	LGBM	0.919

Як видно з таблиці, Light Gradient Boosting Machine найкраще працює на тестових даних з-поміж трьох розглянутих алгоритмів. У подальшому можна працювати над покращенням моделей, проводячи експерименти з конструювання змінних та, за можливості, працюючи над агрегацією транзакцій за користувачами, щоб таким чином розглядати саме поведінку користувача платіжною картою і знаходити аномалії у ній, а не у кожній окремій транзакції.

Використовуючи результати проведеного аналізу та моделювання, було створено інформаційну технологію з виявлення шахрайських транзакцій. Ідея роботи даної системи полягає в попередній обробці транзакції (послідовність дій, використану також і для навчання моделей), поданні її на вхід до попередньо навченої моделі та, як результат, її класифікації. Інформаційна технологія надає можливість обирати для класифікації

транзакцій одну із моделей: випадкового лісу, екстремального градієнтного бустингу або модифікацію стохастичного градієнтного бустингу LGBM, а також виконує порівняння результатів, отриманих під час застосування вищевказаних моделей. Провівши експериментальні обчислення у спроектованій СППР, можна не лише визначити розмір можливих втрат, а й зробити висновок про те, який з методів обчислення є найкращим для використання на практиці.

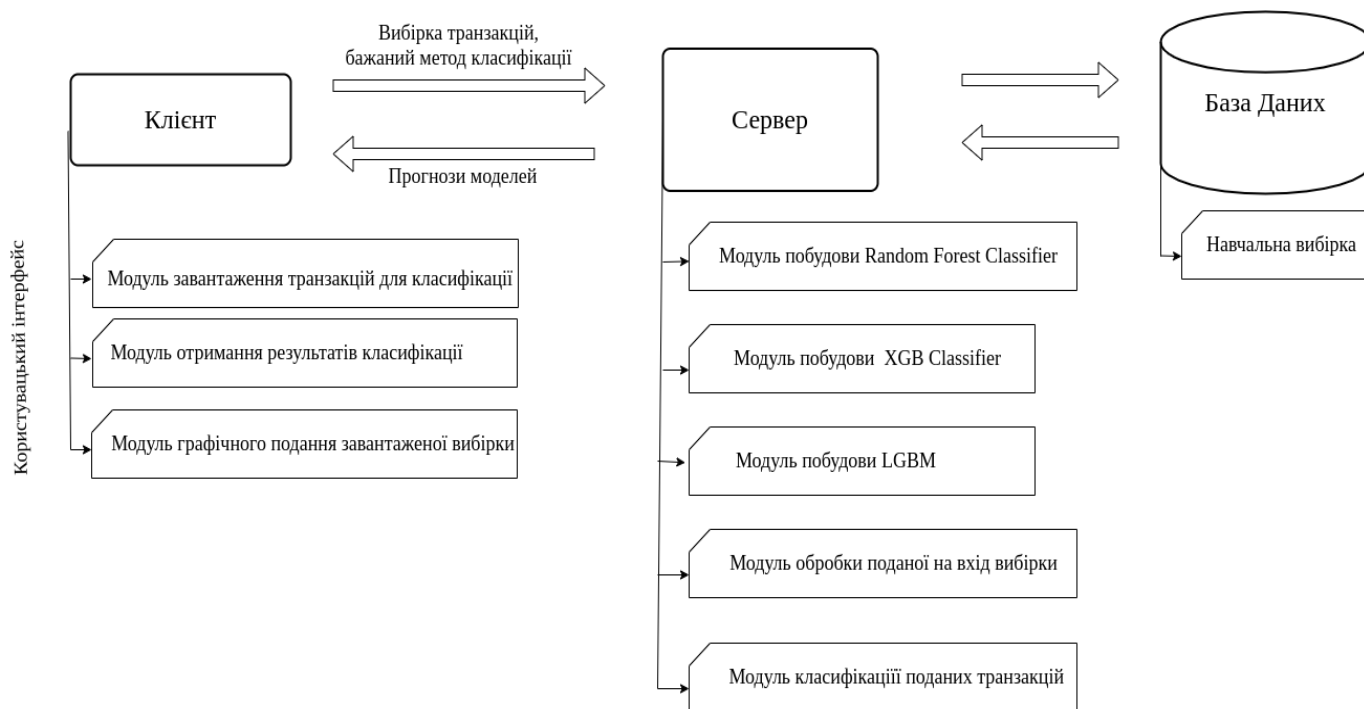


Рисунок 3. Клієнт-серверна архітектура інформаційної технології

## 5. ВИСНОВКИ

На сьогоднішній день шахрайські операції становлять загрозу безпеці платіжних систем, тому повсякчас виникає необхідність знаходження нових ефективних підходів до їх виявлення. Одним із перспективних підходів у вирішенні цього питання є інтелектуальний аналіз даних. Для ефективної класифікації транзакцій методами ІАД доцільним є створення інформаційної технології.

У ході дослідження було проведено розвідковий аналіз даних, підготовку вихідного датасету до навчання моделей - зокрема проведено feature engineering, а також label encoding та балансування класів. На основі підготованих даних було навчено ряд моделей машинного навчання, що порівнювались між собою за показником AUC. Таким чином, у результаті досліджень було створено інформаційну технологію з виявлення шахрайських операцій, у якій використано отримані у ході експериментів бінарні класифікатори.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інтернет-представництво Національного банку України [Електронний ресурс]. Доступно: <https://bank.gov.ua/ua/news/all/startuye-informatsiy-na-kampaniya-natsionalnogo-banku-z-platijnoyi-bezpeki-shahraygudbay>. Дата звернення: Жов. 2022.
2. С. Phua, V. Lee, K. Smith, and R. Gayler “A Comprehensive Survey of Data Mining-based Fraud Detection Research”, School of Business Systems, Faculty of Information Technology, Monash University, Clayton, Australia, 2010 [Online]. Available: [https://www.researchgate.net/publication/302557906\\_A\\_Comprehensive\\_Survey\\_of\\_Data\\_Mining-based\\_Fraud\\_Detection\\_Research](https://www.researchgate.net/publication/302557906_A_Comprehensive_Survey_of_Data_Mining-based_Fraud_Detection_Research). Accessed on: Oct 2022.

3. W.F. Yu, and N. Wang “Research on Credit Card Fraud Detection Model Based on Distance Sum”, in *International Joint Conference on Artificial Intelligence*, Hainan, China, 2009, pp. 353-356, DOI: 10.1109/JCAI.2009.146.

4. A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, “Feature engineering strategies for credit card fraud detection”, *Expert Systems with Applications*, vol. 51, pp. 134-142, June 2016, DOI: 10.1016/j.eswa.2015.12.030.

5. N. V. Kuznietsova “Scoring Technology for Risk Assessment of Fraud in Banking”, *Selected Papers of the XVI International Scientific and Practical Conference "Information Technologies and Security" (ITS 2016)*, CEUR Workshop Proceedings, vol. 1813, pp. 54–61, 2016.

6. F. Provost., and T. Fawcett, “The case against accuracy estimation for comparing induction algorithms”, In *Proc. 15th Intl. Conf. On Machine Learning*, pp. 445–453, Madison, WI, USA, 1998.

7. F. Provost., and T. Fawcett, “Robust Classification for Imprecise Environments”, *Machine Learning* vol.42, pp. 203–231, 2001, DOI: 10.1023/A:1007601015854.

8. J.A. Nelder, and R. Mead, “A simplex method for function minimization” *The Computer Journal*, vol. 7, pp. 308–313, 1965.

# СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ АНАЛІЗУ РОЗВИТКУ ФІНАНСОВИХ ПРОЦЕСІВ

Радіо О.В.<sup>1</sup>, Бідюк П.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>oliaradi007@gmail.com

**В** нинішній час фондовий ринок стає популярним інвестиційним майданчиком як для інституційних, так і для індивідуальних інвесторів. Волатильність ринку та інновації у фінансовій торгівлі зробили прийняття інвестиційних рішень все більш складним та ризикованим. У таких умовах системи підтримки прийняття рішень (СППР) грають важливу роль та сприяють підвищенню якості рішень. Метою роботи є аналіз фінансових процесів фондового ринку з метою покращення прийнятих рішень. Результатом дослідження є СППР, що виконує прогноз волатильності та оцінює ринковий ризик із застосуванням методології VAR. У роботі використані експериментальні дослідження та методи математичного моделювання.

**Ключові слова:** волатильність, прогнозування, математичні моделі, VAR, ринковий ризик.

## 1. ВСТУП

В нинішній час фондові ринки широко розповсюджені та в певній мірі впливають на макроекономічні показники будь-якої сфери чи в цілому країни та їх можна віднести до ефективних. Тобто фондовий ринок – це ринок, на якому активно конкурує велика кількість інвесторів, що прагнуть до максимізації прибутку компаній, кожна з яких намагається передбачити майбутню ринкову вартість окремих цінних паперів, і де важлива поточна інформація майже вільно доступна для всіх учасників. Іншими словами, на ефективному ринку будь-якої миті часу фактична ціна цінного паперу буде гарною оцінкою її внутрішньої вартості. Денний оборот фондового ринку налічується мільярдами доларів та мільйонами торгових доручень. З кожним роком учасників ринку стає дедалі більше, серед них фінансові організації, хеджфонди, банки та приватні інвестори. Трейдери використовують різні стилі торгівлі та системи прийняття рішень, кожен стиль чи система мають різну економічну ефективність та ступінь ризику.

Однак, доступність надійних даних фінансового ринку та потужних програмних інструментів є найвищою. У таких умовах системи підтримки прийняття рішень (СППР) грають найважливішу роль і підвищують якість рішень, прийнятих як професійними, так і індивідуальними інвесторами. Отже, користувачам, які прагнуть підвищити ефективність ухвалення рішень на фондовому ринку, необхідне застосування СППР, що й пояснює актуальність обраного дослідження.

## 2. МАТЕМАТИЧНІ МОДЕЛІ ДЛЯ АНАЛІЗУ ФІНАНСОВИХ ПРОЦЕСІВ

Система підтримки прийняття рішень, або СППР, являє собою комп'ютерну інформаційну систему, яка організовує, збирає й аналізує дані, які можуть

використовуватися під час прийняття рішень для управління, експлуатації та планування [1]. Системи підтримки прийняття рішень призначені для того, щоб прискорено і високоякісно допомогти людям аналізувати великі обсяги даних для прийняття обґрунтованих рішень. Іншими словами, коли потрібно ухвалити рішення, можна звернутися до системи підтримки прийняття рішень і встановити, що «кажуть» дані про це рішення або вибір, який необхідно зробити [2,3]. СППР можуть створювати складні моделі або уявлення того, що відбувається з даними [4]. Приклади таких систем включають ручні системи, гібридні системи, різноманітні типи аналітики, а також складне програмне забезпечення для підтримки прийняття рішень. Фактором, який відрізняє новіші комп'ютерні системи від ранніх систем підтримки ухвалення рішень, є їхня здатність аналізувати надзвичайно великі набори даних, надаючи засновані на даних рекомендації, що виключають здогади під час ухвалення рішень.

Традиційний підхід до розробки СППР полягає в тому, щоб визначити математичну модель, встановити, як система приймає рішення, і використовувати програмне забезпечення для оптимізації, щоб визначити та порівняти результати різних сценаріїв. Цей метод ґрунтується на розпорядчій аналітиці і є надзвичайно ефективним.

Як зазначається в роботі [5], комп'ютеризація фондових ринків і доступність детальних електронних записів потоку заявок і динаміки цін на фондових ринках за останнє десятиліття вивільнила терабайти високочастотних даних про транзакції, потоки заявок і динаміку портфеля заявок на фондових ринках, які дають детальне уявлення про високочастотну динаміку попиту, пропозиції та ціни на цих ринках. Ці дані можуть бути використані для дослідження природи механізму ціноутворення, який описує, як ринкові ціни реагують на коливання попиту та пропозиції. На високому рівні "механізм ціноутворення" - це карта, яка відображає взаємозв'язок між ринковою ціною та такими змінними, як ціна, історія та потік замовлень [12]:

$$Price_{t + \Delta t} = F(Price\ history\ 0 \dots t, Order\ Flow\ 0 \dots t, Other\ Information) + F(X_t, \varepsilon_t) \quad (1)$$

де  $X_t$  - множина змінних стану (наприклад, лагові значення ціни, волатильності та потоку заявок), наділених певною динамікою;  $\varepsilon_t$  - випадковий "шум" або інноваційний член, що представляє надходження нової інформації та інші ефекти, які не повністю охоплюються змінними стану.

Ціна на фондовому ринку формується на основі рівноваги пропозиції та попиту, а також на основі залежності кількості угод від ціни, як показано на рис. 1.

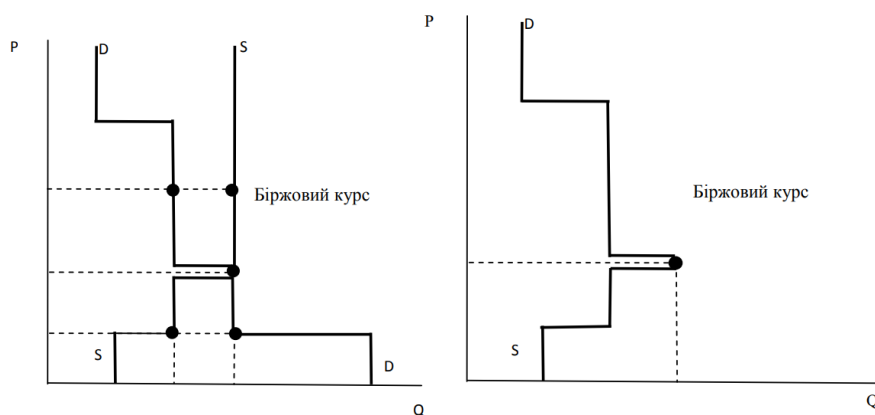


Рисунок 1. Формування ціни на фондовому ринку: а) на основі рівноваги пропозиції та попиту; б) на основі залежності кількості угод від ціни

Існує низка методів оцінювання ціноутворення на фондовому ринку, метою яких є побудова максимально адекватної математичної моделі вибраного процесу. Якщо розглядати прогнозування волатильності та оцінку ринкового ризику до таких моделей

можна віднести ARCH/GARCH моделі. Загалом, ARCH/GARCH методологію можна охарактеризувати як методологію моделювання дисперсії досліджуваного показника. Оскільки дисперсія – момент другого порядку, модель дисперсії нелінійна, отже, її не можна оцінювати методами, розробленими для лінійних моделей, таких як ARIMA. Також під час досліджень велику увагу привернула нейронна мережа LSTM, саме своїми результатами. Основними компонентами мережі LSTM є рівень введення послідовності та рівень LSTM. Шар введення послідовності вводить дані послідовності або часового ряду в мережу. Шар LSTM вивчає довгострокові залежності між тимчасовими кроками даних послідовності.

Великої популярності на сьогоднішній день набула методологія Value at Risk (VaR). Це зумовлено тим, що така методологія володіє рядом безсумнівних переваг, а саме:

- дає змогу оцінити ризик в межах можливих втрат, співвіднесених з ймовірністю їх виникнення;
- вимірювати ризики на різних ринках універсальним чином;
- агрегувати ризики за окремими позиціями в єдину величину (інтегральний ризик) для всього портфеля активів, враховуючи при цьому інформацію про кількість позицій, і волатильності на ринку.

Волатильність або абсолютна/відсоткова дисперсія цін часто вважається гарним заходом для оцінювання ринкового ризику. Професійні аналітики також схильні використовувати такі методи, як моделювання вартості під ризиком (VaR) для виявлення потенційних збитків за допомогою статистичного управління ризиками.

### **3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ**

У процесі виконання роботи розроблено і реалізовано систему підтримки прийняття рішень, яка призначена для аналізу ціноутворення на фондовому ринку. Вона виконує такі задачі: побудова моделей гетероскедастичних процесів для прогнозування волатильності, оцінювання ринкових ризиків методом VaR та аналіз якості отриманої прогнозованої моделі. Головною метою створення даної програми було підвищення рівня ефективності прийняття рішень на фондовому ринку, а також проілюструвати роботу методів моделювання фінансових процесів, застосувати їх на реальних даних.

Для моделювання у даній роботі обрано процес ціноутворення акцій міжнародної компанії TESLA, взятий з інтернет ресурсу *finance.yahoo.com*. Обраний датасет складається з таких колонок: «High», «Low», «Open», «Close», «Volume», «Adj Close» та містить 1259 записів, а саме з 13.11.2017 – 11.11.2022. Для подальшої роботи потрібно обрати характеристику, яка буде найбільш коректною для моделювання. Після дослідження значення кожної колонки було визначено взяти саме колонку «Adj Close», адже вона показує скориговану ціну закриття, тобто ціну закриття після всіх застосованих коригувань. Двома основними з них є розбиття та дивіденди. Після візуального аналізу графіків бачимо, що розподіл цін схожий на нормальний, проте являється трохи вищим, а от розподіл Стюдента краще описує ряд, особливо на хвостах розподілу (Рисунок 2, 3).



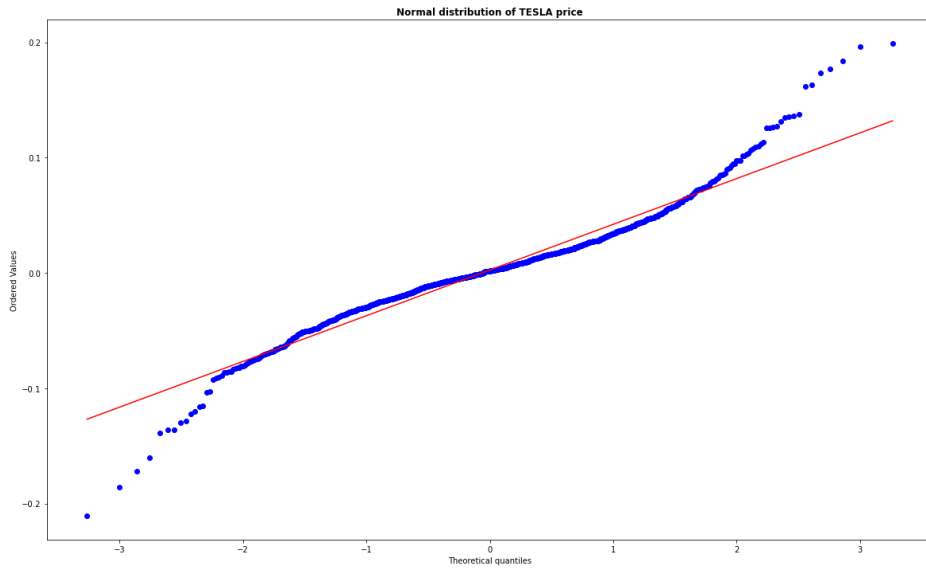


Рисунок 2. Графік нормального розподілу даних компанії TESLA

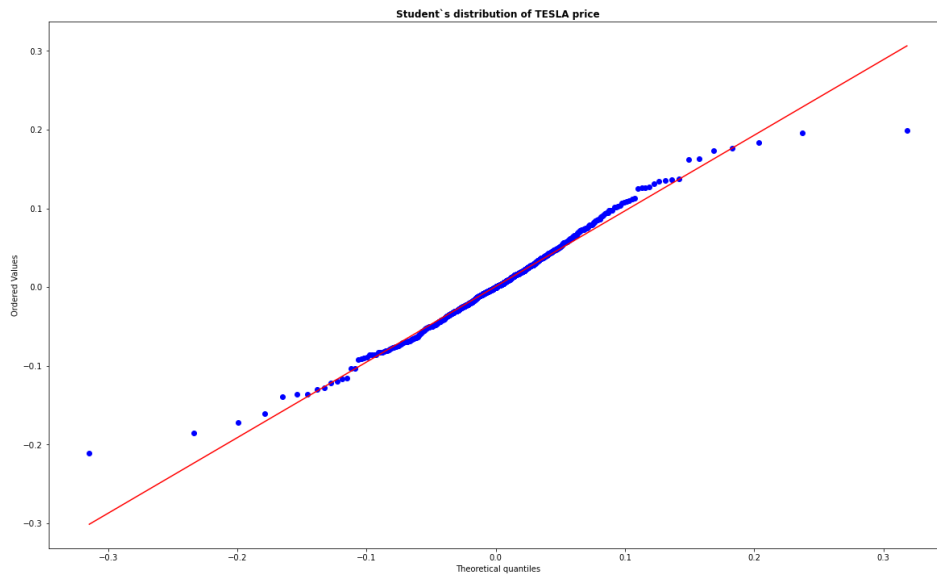


Рисунок 3. Графік розподілу Стьюдента компанії TESLA

Для побудови відповідних математичних моделей було проведено підготовку даних. Спочатку розбиваємо вибірку на тестову й тренувальну, для очистки ряду від випадкових шумів застосовуємо фільтр Калмана, а також проведено тест на гетероскедастичність. Можна звернути увагу, що ймовірність, отримана в результаті, явно менша за поріг 0,05 (Рис. 4). Це свідчить про те, що у даному випадку необхідно відхилити нуль-гіпотезу, яка припускає незмінність дисперсії у часі. Отже для залишків даної моделі характерна наявність ефекту ARCH.

```

=====
Dep. Variable:          Adj_Close    R-squared:              0.000
Mean Model:           Constant Mean  Adj. R-squared:         0.000
Vol Model:            GARCH          Log-Likelihood:        -5505.34
Distribution:         Normal         AIC:                   11018.7
Method:              Maximum Likelihood  BIC:                   11039.1
                                           No. Observations:      1208
Date:                Wed, Nov 16 2022  Df Residuals:          1207
Time:                14:05:47         Df Model:               1
                                           Mean Model
=====
                coef    std err          t      P>|t|    95.0% Conf. Int.
-----+-----+-----+-----+-----+-----
mu                21.0683   9.064e-02    232.432    0.000 [ 20.891, 21.246]
Volatility Model
=====
                coef    std err          t      P>|t|    95.0% Conf. Int.
-----+-----+-----+-----+-----+-----
omega              0.1408   3.216e-02     4.377   1.201e-05 [7.774e-02, 0.204]
alpha[1]           1.0000   1.051e-02    95.116    0.000 [ 0.979, 1.021]
beta[1]            1.7790e-11  6.167e-06   2.885e-06    1.000 [-1.209e-05,1.209e-05]
=====

```

Рисунок 4. Описова статистика та результати тесту Engle

Наступним кроком для моделювання волатильності дисперсії процесу було визначення порядку самої моделі. Для цього побудували графіки АКФ та ЧАКФ для квадратів залишків моделі та побудували декілька моделей для порівняння. Результати подано в таблиці 1.

Таблиця 1. Результати застосування GARCH моделі

Model	MAE	MAPE	THEIL
GARCH (1,1)	45.503398699012	11.60310685149867	0.25971130284802
GARCH (2,4)	43.702794762397	11.51956147319662	0.25971123179284
GARCH (4,8)	43.259252938935	11.73202941090763	0.25708675806683
GARCH (4,12)	42.703393537471	11.60310954523587	0.25711302848347

Отримані результати прогнозів в таблиці 1 виявилися задовільними. В процесі дослідження також перевірялися моделі EGARCH і TARARCH, які показали майже такі ж оцінки, тому було вирішено не додавати їх у дану роботу. Як найкращу модель було обрано саме GARCH (4,12).

Також у поданій магістерській роботі проведено дослідження щодо прогнозування часових рядів за допомогою рекурентних нейронних мереж. Для моделювання справжньої волатильності було обрано саме нейронну мережу LSTM, яка працює по принципу системи довгої короткочасної пам'яті. Представимо результати роботи даного методу в таблиці 2.

Таблиця 2. Результати роботи LSTM моделі

Model	MAE	MAPE	THEIL
LSTM	7.3834419570547	0.42678149293463	0.06897247272352

Останній крок дослідження – це розрахунок ринкового ризику методом Value-at-Risk. Для розрахунку використано прогнози, отримані саме за допомогою моделі LSTM. Згадуємо, що на початку розділу було зроблено дослідження про схильність до розподілу Стюдента, тобто розраховуємо показник саме для цього випадку. Результати подано в таблиці 3.

Таблиця 3. Порівняння результатів застосування моделей

Показник	Результат
95%	478.834
99%	435.057

## 4. ВИСНОВКИ

Враховуючи складність фінансового ринку та нелінійну характеристику фінансових змінних, штучна нейронна мережа, як важлива форма штучного інтелекту, стала великим досягненням у багатьох фінансових галузях та може застосовуватися для аналізу фінансових процесів на фондовому ринку.

Згідно з літературними джерелами найбільшу ефективність при прогнозуванні цін на фондовому ринку продемонстрували нейронні мережі LSTM, саме тому вони обрані для моделювання в межах магістерської дисертації. Окрім цього, для моделювання дисперсії при аналізі фінансових показників на фондовому ринку обрано методологію ARCH/GARCH, яка є найбільш ефективною серед аналогів.

У ході дослідження було реалізовано таку систему підтримки прийняття рішень, яка дозволяє виконувати моделювання волатильності певного процесу та розраховувати ринковий ризик. Ця система використовує моделі GARCH та LSTM, а також методологію VaR для обчислення ринкового ризику. Результати дослідження показали, що найбільш ефективним методом виявилася саме нейронна мережа LSTM. Окрім того, було визначено значення показника VaR для 95% і 99% ймовірностей.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Конспект лекцій з курсу “Системи прийняття рішень” для студентів напряму підготовки 6.030502 “Економічна кібернетика”, спеціальності 051 “Економіка” / С. В. Гринчуцька – Тернопіль, ТНТУ імені І. Пулюя, 2017. – 130с.

2. Системи і методи підтримки прийняття рішень. [Електронний ресурс] навч. посіб. для здобувачів ступеня магістра за освітніми програмами «Системний аналіз та управління», «Системний аналіз фінансового ринку» спеціальності 124 «Системний аналіз» / П.І.Бідюк, О.Л.Тимощук, А.Є.Коваленко; Л.О.Коршевнік КПП ім. Ігоря Сікорського ;. Електронні текстові дані. Київ : КПП ім. Ігоря Сікорського, 2020. – 259 с.

3. Системи підтримки прийняття рішень [Текст]: навчальний посібник для самостійного вивчення дисципліни / [уклад.: С. М. Братушка, С. М. Новак, С. О. Хайлук] ; Державний вищий навчальний заклад “Українська академія банківської справи Національного банку України”. – Суми : ДВНЗ “УАБС НБУ”, 2010. – 265 с.

4. Проектування інформаційних систем підтримки прийняття рішень: навч. посіб. / [Бідюк П.І., Коршевнік Л.О., Коваленко А.Є., Гожий О.П.]. — К.: НТУУ «КПІ», 2013. – 608с.

5. Justin Sirignano, Rama Cont. Universal features of price formation in financial markets: perspectives from Deep Learning. 2018. – 21 p.

6. Башкіров О.В. Порівняльний аналіз VAR-методів оцінки ризику активів банку О.В. Башкіров, Проблеми і перспективи розвитку банківської системи України : зб. наук. праць ДВНЗ «УАБС НБУ». – Вип. 14. – С. 302–309.

7. What is a decision support system (DSS)? [Електронний ресурс]: Режим доступу: <https://www.techtarget.com/searchcio/definition/decision-support-system> (дата звернення: 15.10.2022) НТУУ «КПІ», 2010. – 310с.

8. Market Risk [Електронний ресурс]: Режим доступу: <https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/market-risk/> (дата звернення: 17.10.2022).

9. Long Short-Term Memory Networks [Електронний ресурс]: Режим доступу: <https://www.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html> (дата звернення: 19.10.2022).

# МЕТОДИ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ СТАНУ ПІДПРИЄМСТВА ЗА ДОПОМОГОЮ РЕГРЕСІЙНИХ МОДЕЛЕЙ.

Сандига Я.С.<sup>1</sup>, Бідюк П.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

**На сьогоднішній день кількість підприємств збільшується кожного дня. Для аналізу релевантності та напрямів розвитку підприємства потрібно постійно розуміти ступінь ризику банкрутства самого підприємства. Основною проблематикою в даній сфері прогнозування є вибір методу, який буде використано для вказаних цілей. Далі в роботі буде розглянуто дещо нестандартний підхід до аналізу стану підприємства за допомогою логістичної регресії. Метою роботи визначено створення програмного забезпечення (ПЗ), за допомогою якого користувач може подати на вхід початкові дані фінансової звітності і отримати оцінку ризику банкрутства підприємства.**

**Ключові слова:** фінансовий стан підприємства, логістична регресія, прогнозування ризиків банкрутства.

## 1. ВСТУП

Перетворення фінансового аналізу в статистичний починається з розрахунків великої кількості показників, що з часом не відображають актуальність проблеми (такі як коефіцієнти трансформації та іммобілізації). Зазвичай, аналіз фінансового стану підприємства зводиться до оцінювання одного коефіцієнта, який на думку користувача даних є найбільш впливовим в конкретній постановці задачі. Якщо ж перейти до правильної побудови зв'язків між економічними факторами та математичними методами, то можна зробити висновок, що в даному випадку нам не потрібна така велика кількість показників. Навпаки, основною метою побудови моделей аналізу стану підприємства є використання мінімальної кількості показників для максимально правильного розрахунку коефіцієнта, що характеризує стан підприємства. В залежності від значення цього коефіцієнту ми отримуємо відповідь на економічне запитання щодо того, наскільки близько підприємство знаходиться до банкрутства.

## 2. МЕТОДИ ОЦІНЮВАННЯ ФІНАНСОВОГО СТАНУ ПІДПРИЄМСТВА

Першим методом буде розглянуто побудову модель логістичної регресії. Логістична регресія – це статистичний регресійний метод, що використовується у випадку коли пояснювальна (залежна) змінна може набувати тільки двох значень, зазвичай 0 та 1 (бінарна логіт-регресія) або, більш загально, скінченну множину значень. Її призначення полягає у тому, щоб прокласти зв'язок між декількома незалежними змінними та однією залежною змінною. Таким чином, результатом роботи даного методу буде вірогідність тієї чи іншої події, а це якраз те, що нам потрібно, тобто необхідно оцінити вірогідність події банкрутства підприємства.

Для початку, розглянемо рівняння множинної регресії:

$$y = a + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

Якщо використовувати такий вигляд моделі для оцінки ймовірності події, ми можемо отримати проблему, коли значення  $y$  буде більшим за 1 або меншим за 0. Така ситуація можлива через те, що змінна відгуку бінарна за своєю природою при множинній регресії. Для вирішення задачі ми змінюємо завдання регресії і будемо прогнозувати не значення бінарної змінної, а безперервну змінну на інтервалі  $[0;1]$ . Таку заміну можна зробити за допомогою формули:

$$P = \frac{1}{1 + e^{-y}}$$

де  $P$  – ймовірність події;  $Y$  – визначається рівнянням регресії;  $e$  – основа натурального логарифма.

Оцінювання коефіцієнтів логістичної регресії може виконуватись різними методами, але зазвичай використовується метод максимальної правдоподібності (ММП). Він застосовується для оцінювання параметрів генеральної сукупності по даним вибірки. Основу метода виступає функція правдоподібності, яка виражає щільність імовірності сумісної появи результатів вибірки:

$$L(Y_1, Y_2, \dots, Y_k; \theta) = p(Y_1; \theta) \cdot \dots \cdot p(Y_k; \theta)$$

Згідно до методу, в якості оцінки невідомого параметра приймається таке значення  $\Theta = \Theta(Y_1, \dots, Y_k)$ , що максимізує функцію  $L$ . Максимізувати значення функції правдоподібності еквівалентно максимізації її логарифму:

$$L^*(Y, \theta) = \ln(L(Y, \theta)) \rightarrow \max$$

Так виглядає логарифмічна функція правдоподібності:

$$L^* = \sum_{i=1}^k \ln P_i(w) + \sum_{i=0} \ln(1 - P_i(w)) = \sum_{i=1}^k [Y_i \ln P_i(w) + (1 - Y_i) \ln(1 - P_i(w))]$$

де  $0, 1$  – множина спостережень, для яких  $Y$  може приймати значення 0 або 1.

### **3. ПРАКТИЧНЕ ВИКОРИСТАННЯ МОДЕЛІ ЛОГІСТИЧНОЇ РЕГРЕСІЇ В ОЦІНКАХ ФІНАНСОВОГО СТАНУ ПІДПРИЄМСТВА**

Розглянемо приклад використання логістичної регресії для визначення фінансового стану підприємства. Важливим недоліком бінарної логістичної регресії є те, що для обрахунків вона може отримувати обмежену кількість вхідних факторів. Тому для використання такого методу прогнозування потрібно залучити експерта, який визначить найважливіші фактори в фінансовій звітності підприємства. В прикладі використано такі показники фінансової звітності:

- коефіцієнт автономії;
- коефіцієнт абсолютної ліквідності;
- коефіцієнт поточної ліквідності;
- коефіцієнт оборотності;
- коефіцієнт рентабельності.

Маючи інформацію про показники 160 підприємств, перевіримо наскільки точно працює логістична регресія. На основі 154 підприємств побудуємо модель логістичної регресії та спробуємо спрогнозувати дані інших 6 підприємств. За допомогою системи EViews побудуємо модель (рис. 1):

Dependent Variable: CRITICAL\_CONDITION  
Method: ML - Binary Logit (Quadratic hill climbing)  
Date: 05/17/14 Time: 21:59  
Sample: 1 154  
Included observations: 154  
Convergence achieved after 7 iterations  
QML (Huber/White) standard errors & covariance

Variable	Coefficient	Std. Error	z-Statistic	Prob.
KOEF_ABS_LIKVID	-21.45350	7.413021	-2.894029	0.0038
KOEF_AUTONOMY	-8.887599	1.448821	-6.134365	0.0000
KOEF_OBOROT_AKTYVIV	-1.718040	0.375649	-4.573531	0.0000
KOEF_POTOCH_LIKVID	0.693932	0.223439	3.105683	0.0019
RENTAB_CAPITALU	-5.891127	1.983635	-2.969864	0.0030
C	5.340537	0.969530	5.508374	0.0000

McFadden R-squared	<u>0.663470</u>	Mean dependent var	0.500000
S.D. dependent var	0.501631	S.E. of regression	<u>0.284025</u>
Akaike info criterion	<u>0.544452</u>	Sum squared resid	11.93923
Schwarz criterion	0.662774	Log likelihood	-35.92277
Hannan-Quinn criter.	0.592514	Deviance	71.84554
Restr. deviance	213.4893	Restr. log likelihood	-106.7447
LR statistic	141.6438	Avg. log likelihood	-0.233265
Prob(LR statistic)	0.000000		

Obs with Dep=0	77	Total obs	154
Obs with Dep=1	77		

Рисунок 1. Основні характеристики якості моделі

Отримаємо таке рівняння логістичної регресії для нашої моделі:

$$P_{condition_{critical}} = 1 / (1 + \exp(-(5.340537 - 21.45350 * koef\_abs\_likvid - 8.887599 * koef\_autonomy - 1.718040 * koef\_oborot\_aktyviv + 0.693932 * koef\_potoch\_likvid - 5.891127 * rentab\_capitalu)))$$

На рисунку 2 показано прогноз за моделлю логістичної регресії та його порівняння з реальними значеннями по 6 підприємствах. Фактично отримано високу якість прогнозування стану підприємств.

Таблиця 1. Характеристики якості моделі

Назва	Формула	Значення
McFadden R-squared (коефіцієнт детермінації)	$= 1 - \frac{\log L^{max}(full)}{\log L^{max}(empty)}$	Відношення логарифмічної функції правдоподібності повної моделі до нульової Значення від 0 до 1 Чим більше значення, тим краще модель описує дані
Інформаційний критерій Акайке (AIC)	$= 2k - 2\ln(L)$ <ul style="list-style-type: none"> <li>• k – кількість параметрів статистичної моделі;</li> <li>• L – максимізоване значення функції правдоподібності</li> </ul>	Для кращої моделі $AIC \rightarrow \min$
S.E. of regression (Standard Error of the Regression)	$= \frac{\sqrt{\sum_{i=1}^N (y_i - \hat{y}_i)^2}}{N}$ <ul style="list-style-type: none"> <li>• N – розмір вибірки</li> <li>• <math>y</math> – фактичні значення</li> <li>• <math>\hat{y}</math> – значення оцінені за моделлю.</li> </ul>	Для кращої моделі $S.E. \text{ regression} \rightarrow \min$

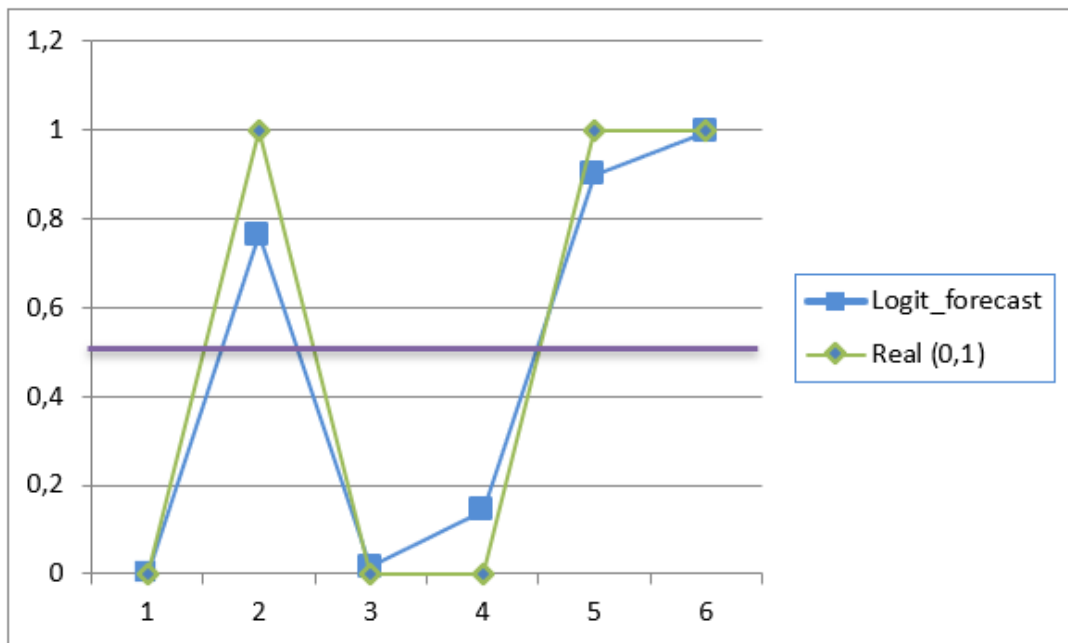


Рисунок 2. Прогноз за моделлю логістичної регресії та порівняння її з реальними значеннями по 6 підприємствах

## 4. ВИСНОВОК

Логістична регресія має високий ступінь точності визначення фінансового стану підприємства за допомогою правильно обраних коефіцієнтів. Однак, складність моделі заперечує її просте використання звичайним користувачем. Коректне використання прогнозуючої змінної допомогло нам використати модель для наших цілей та є дуже хорошим, що і намагались ми зробити в даній роботі. Вважаємо, що використання описаного методу знайде свого споживача та цілком може бути автоматизовано для кінцевого користувача, що спростить роботу та розуміння отриманих результатів і обов'язково допоможе в майбутньому.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Поддєрьогін А.М. Фінанси підприємств / Поддєрьогін А.М. – М.: Финансы и статистика, 1999. – 261с.
2. Ковалев В.В. Введение в финансовый менеджмент / Ковалев В.В. – М.: Финансы и статистика, 2000. – 234 с.
3. Sahakian С.Е. The Delphi Method [Text] / Sahakian С.Е. – The Corporate Partnering Institute, 1997. – 11 p.
4. А.Н.Борисов. Модели принятия решений на основе лингвистической переменной / А.Н.Борисов и др. – Рига: Зинатне, 1982. – 256 с.
5. Bollerslev T. Generalized Autoregressive Conditional Heteroskedasticity / Bollerslev T. // Journal of Econometrics. – 1986. - Vol. 31 – p. 307-327.
6. Engle Robert F. Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation / Engle Robert F. // Econometrica. - 1982. - Vol. 50 – p. 987-1007.



# ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПРОГНОЗУВАННЯ НЕЛІНІЙНИХ НЕСТАЦІОНАРНИХ ПРОЦЕСІВ В ЕКОНОМІЦІ

Селін Ю.М.<sup>1</sup>, Жук В.М.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>2</sup>volodymyr.zhuk.gp@gmail.com

**Вивчення та прогнозування часових рядів економічної природи потребує впровадження широкого кола засобів математичного апарату, інформаційних систем, систем керування сховищами даних тощо. Розробка та викроистання інтелектуальних систем (ІС) для прогнозування нелінійних нестационарних процесів в економіці дозволить частково автоматизувати створення прогнозів, підвищити якість прогнозів використаних алгоритмів, знизить планку необхідних знань і навичок для використання та розуміння поведінки того чи іншого економічного процесу. Метою роботи є створення ІС для прогнозування нелінійних нестационарних процесів в економіці із застосуванням системного підходу. Результатом дослідження є ІС, що виконує прогнозування часових економічних рядів з використанням алгоритму лінгвістичного моделювання.**

**Ключові слова:** інтелектуальна система, системний підхід, лінгвістичне моделювання, часові ряди.

## 1. ВСТУП

Зберігання та аналіз даних економічної природи – один з ключових кроків для детального розуміння виникнення та поведінки будь-якого економічного процесу. Правильна робота з цими даними дозволяє комплексно прогнозувати процеси економічної природи.

Саме тому створення інформаційної системи, що дозволяє комплексно проводити моделювання часових рядів є одним з найактуальніших завдань, яке дозволяє покращити точність прогнозів, зменшити обсяг витраченого часу, автоматизувати необхідні процеси тощо. Застосовуючи різні методології прогнозування, можливо покращити загальну точність інформаційної системи, в залежності від природи даних, які моделюються. Подальше порівняння результатів дасть змогу краще налаштувати необхідні параметри для отримання більш точного прогнозу.

## 2. МЕТОДИ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

Існує багато методів прогнозування часових рядів. Найбільш розповсюджені з них – методи регресійного аналізу. Ці методи зазвичай дозволяють побудувати точний прогноз із лінійними або експоненційними залежностями. Хоча існують і поліноміальні види регресії для побудови більш складних моделей. Підхід побудови регресії базується на знаходженні відношення між залежною змінною  $y$  та однією або декількома незалежними змінними  $x$ . Для збільшення ефективності моделі, рівняння може містити значення шуканої змінної, затриманої у часі на визначену кількість кроків(лаги). Тому, мішане рівняння регресії виглядає наступним чином :  $y(k) = a_0 + a_1y(k - 1) + a_2y(k - 2) + \dots + a_n(k)y(k - n) + b_1x_1(k) + b_2x_2(k) + \dots + b_px_p(k) + e(k)$  [1]. Також може бути застосований

сегментований підхід, коли для кожного окремого сегменту намагаються побудувати регресійну модель [2].

Метод групового урахування аргументів іноді вважають узагальненням підходу регресійного аналізу. Ця методика використовується у дуже різноманітних галузях аналізу даних (наприклад, розпізнавання образів, моделювання різноманітних систем, оптимізації тощо). У ході виконання алгоритму відбувається перебір різних варіантів моделювання вхідного процесу, при цьому автоматично обирається найкращий варіант моделі за спеціальними критеріями адекватності (коефіцієнт детермінації, суми квадратів похибки тощо). Постановка завдання для застосування методу повинна включати наступні елементи :

- вибір додаткових описів, які створюють основу для можливого остаточного вибору моделі;
- відбір та адаптація параметрів моделі для конкретного застосування;
- розробка нової або застосування відомої моделі оцінки;
- вибір критерії якості моделі для використання в проміжних кроках обчислень і для остаточного вибору моделі;

Моделі, побудовані з відповідним чином налаштованим МГУА, зазвичай забезпечують середню або високу якість короткострокових прогнозів [3].

Серед методів, що будуть використовуватися в ІС для прогнозування нелінійних нестационарних процесів слід зазначити лінгвістичного моделювання. Цей метод належить до родини статистичних методів прогнозування, що зумовлює широку можливість його використання з даними абсолютно різної природи. Даний тип методів прогнозування відрізняється універсальністю, проте трохи гіршою точністю.

Для побудови системи, необхідно розв'язати задачу пошуку лінгвістичного образу для досліджуваного часового ряду, яка складається з декількох етапів:

- обрахунок різницевого ряду;
- вибір значення критерію інтервального розбиття різницевого ряду;
- співставлення часовому інтервалу ряду певної літери алфавіту;
- побудова матриці переходу між будь-якими двома літерами;

Задача побудови першої різниці часового ряду – аналогу першої похідної [4]. Цей спосіб дозволяє прибрати тренд часового ряду та зробити його стаціонарним для подальшої побудови моделі для прогнозу, оскільки майже все методі прогнозування дають успішний результат при використанні їх зі стаціонарними часовими рядами. Слід зазначити, що використання перших різниць не завжди зробить ряд стаціонарним. Для визначення порядку тренду можна проаналізувати коефіцієнти автокореляції часового ряду. Маємо : вектор з цілих чисел  $X$  потужністю:  $n = X$ . Отримаємо : вектор з цілих чисел  $D$  потужністю:  $k = |D|$ .

Тобто,  $\forall d_i \in D: d_i = x_{i+1} - x_i$ .

Задача вибору оптимального значення критерія інтервального розбиття різницевого ряду. Розв'язок задачі дозволяє побудувати алфавіт користувача шляхом розділення відсортованого ряду першої (зазвичай, якщо порядок тренду лінійний) різниці на безліч інтервалів, в якому кожний елемент характеризує певну літеру заданого алфавіту [4]. Варто розуміти, що дуже велика або дуже маленька кількість інтервалів розбиття негативно вплине на отриманий результат, тому що при побудові матриці переходу або дуже багато значень будуть потрапляти в один проміжок (при малому розбитті) або значення ймовірності переходу від одної літери до іншої будуть майже однаковими (при великому розбитті).

Лінгвістична задача. Метою цієї задачі є формування лінгвістичного ланцюга шляхом співставлення відповідної букви алфавіту для кожного спостережуваного значення різницевого ряду. Кожна буква у визначеному алфавіті однозначно відповідає певному

інтервалу з набору інтервалів, які були отримані при розв'язанні минулої задачі розбиття ряду відповідними інтервалами. Лінгвістична задача. Метою цієї задачі є формування лінгвістичного ланцюга шляхом співставлення відповідної букви алфавіту для кожного спостережуваного значення різницевого ряду. Кожна буква у визначеному алфавіті однозначно відповідає певному інтервалу з набору інтервалів, які були отримані при розв'язанні минулої задачі розбиття ряду відповідними інтервалами[4].

Задача побудови матриці переходу між будь-якими двома літерами. Отримана послідовність аналізується на наявність граматичних конструкцій. На виході ми отримуємо список граматичних конструкцій з ймовірностями їх присутності в процесі, а також матрицю ймовірностей переходу від одного символу до іншого символу. Цей етап тісно пов'язаний із моделюванням (прихованих) процесів Маркова, а також методом подібних траєкторій.

ІС складається таких основних модулів: інтерфейсу користувача, підсистеми аналізу, системи прогнозування та модуля генерації звіту. Користувач завантажує файл з даними, система сама проводить оцінку часового ряду, визначає необхідні параметри для методів, що будуть використовуватися та будує прогнози. Далі отриманий прогноз звіряється з реальним, похибка зберігається у базу даних, а наступне обчислення робиться з урахуванням дійсного результату. Далі описана поверхнева блок-схема структури ІС (Рис. 1).

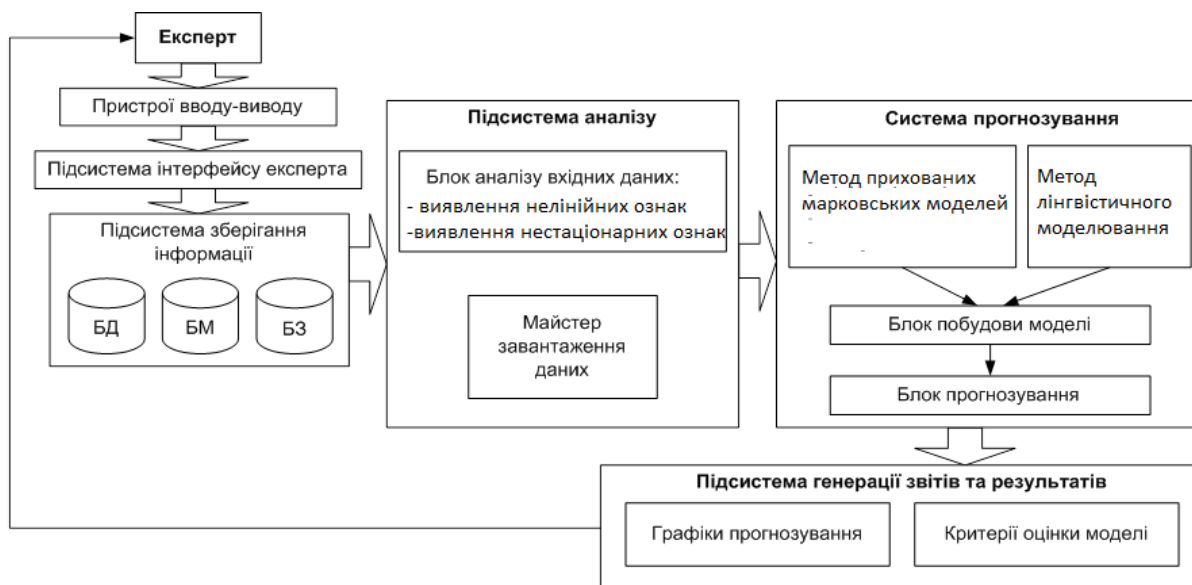


Рисунок 1. Структура інформаційної системи

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для цілей прогнозування був обраний часовий ряд акції компанії Netflix. У якості часового горизонту було обрано період в один торговельний рік – 252 дні. Період прогнозування було взято в 5 днів.

Візуалізація часового ряду дає інформацію щодо його нестационарності (Рис.2).

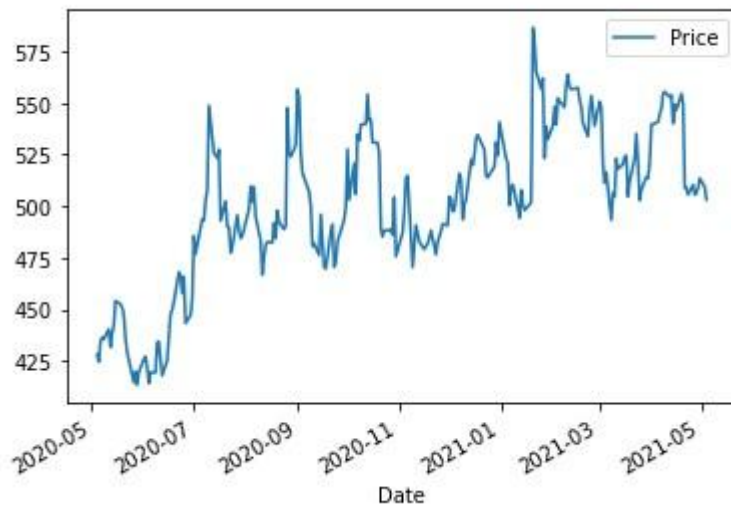


Рисунок 2. Графічне зображення часового ряду “Netflix”

Метод лінгвістичного моделювання дозволяє застосовувати мінімальну необхідну попередню обробку даних перед використанням даного алгоритму. Необхідність диференціювання ряду для зведення його до стаціонарного є єдиною необхідною умовою для подальшої побудови математичної моделі.

Після приведення ряду до стаціонарного ми визначаємо критерій розбиття, тобто на скільки інтервалів нам розділити значення часового ряду та крок з яким ми будемо це робити. Крок визначається як різниця максимального та мінімального значення диференційованого ряду, поділеного на кількість інтервалів. Зазвичай оптимальною є кількість 25-35 інтервалів, де добре підходять літери англійського алфавіту. Далі ми будемо відповідні інтервали, співставляючи кожній літері відрізок. Після чого ми перетворюємо наш числовий ряд у лінгвістичну послідовність. Далі побудуємо матрицю переходів яка містить ймовірнісні значення переходу від однієї літери до іншої. Після отримання прогнозу, ми рахуємо отриману помилку та використовуємо справжнє значення для створення наступного прогнозу.

Результати прогнозування часового ряду різними методами наведено нижче (Табл. 1).

Таблиця 1. Порівняння результатів статистичних тестів використаних методів

Модель часового ряду	$R^2$	Sum squared resid	Durbin – Watson
AP(1)	0.87447	43540.31	2.21994
AP(4)	0.872836	42237.67	1.95032
AP(5)	0.87669	40409.32	1.95289
APKC(4,4)	0.90509	39770.35	2.00993
ARIMA(4,1,4)	0.955309	40037.91	2.005514
ARIMA(5,1,1)	0.968889	41037.91	2.017372
Linguistic modeling	0.9348136	11.637035	1.425621

Також ми можемо побачити результати прогнозування, виконане за допомогою методу лінгвістичного моделювання на 5 кроків вперед (Рис. 3).

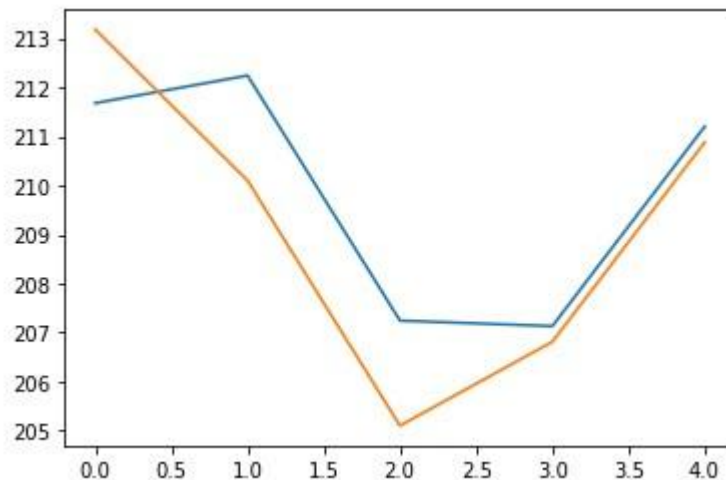


Рисунок 3. Прогноз на 5 кроків вперед з використанням методу лінгвістичного моделювання

#### 4. ВИСНОВКИ

Було створено інформаційну систему, яка дозволяє прогнозувати нелінійні нестационарні процеси в економіці. Окрім того, виконані наступні поставлені задачі:

- виконано аналіз деяких методів дослідження нелінійних нестационарних процесів;
- виявлено нелінійності і нестационарності у сучасних фінансово-економічних процесах;
- вибрано процеси для дослідження та зібрано необхідні статистичні дані;
- виконано порівняльний аналіз отриманих результатів і виробити рекомендації стосовно їх практичного застосування;

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Юрченко М. Є. Прогнозування та аналіз часових рядів. Методичні вказівки до практичних занять та самостійної роботи студентів спеціальності 051 “Економіка” освітня програма “Економічна кібернетика”, “Економічна аналітика”. 2018. 88с.

2. Бідюк П. І, Романенко В. Д., Тимошук. О. Л. Аналіз часових рядів: підручник / НТУУ КПІ ім. Ігоря Сікорського. Київ: ВПК "Політехніка", 2013. 599 с.

3. Зайченко Ю.П., Кебкел О.Г., Крачковський В.Ф. Нечіткий метод групового врахування аргументів та його застосування в задачах прогнозування макроекономічних показників. //Наукові вісті НТУУ «КПІ», №2, 2000р. с.18-26

4. Селін Ю.М. Системний аналіз екологічно небезпечних процесів різної природи / Ю.М. Селін // Системні дослідження та інформаційні технології. — 2007. — № 2. — С.22–32

# ТЕОРЕТИКО-ІГРОВИЙ ПІДХІД ДЛЯ ТРЕЙДИНГОВИХ СТРАТЕГІЙ НА КРИПТОРИНКУ

Семіконь Я.В.

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

[semikonur@gmail.com](mailto:semikonur@gmail.com)

**Мета дослідження - виявити кращу стратегію для отримання прибутку у криптовалюти, використовуючи фундаментальний та технічний аналіз для коректного визначення можливих вигравів та ймовірностей. Більше того, до дослідження залучається теорія ігор, а саме ігри з природою. Економіка або економічний ринок непередбачувані, тому у даній роботі такі ігри розглядаються з використанням декількох критеріїв: критерій Севіджа, Байєса, Лапласа.**

**Ключові слова: теорія ігор, трейдинг, криптовалюти, фундаментальний аналіз, ігри з природою.**

## 1. ВСТУП

Криптовалюта або електронна готівка – це віртуальна грошова одиниця, яка вимірюється у цифрових монетах або коїнах (від англ. coin). Ці монети складаються з зашифрованих пакетів даних, які неможливо скопіювати або підробити. Щоб згенерувати інформацію, яка лежить в основі цифрових монет, знадобиться потужний комп'ютер та нехитре програмне забезпечення. Емісія, або іншими словами виробництво монет, відбувається децентралізовано, тобто немає єдиного емітента. Технічний сенс майнінгу – добувати зашифровані коди за допомогою підбору числових варіацій. Сам процес майнінгу – це розрахунок ланцюжка даних для блоку інформації, який вносять у блокчейн. За розрахунок та формування нового блоку даних, який відповідає всім вимогам системи, майнер (власник цього або цих комп'ютерів) отримує певну кількість одиниць криптовалюти, які можуть після конвертуватися у звичні нам гроші (гривні, долари, євро тощо).

Першою криптовалютою у світі є Біткоїн (Bitcoin), саме він стоїть біля джерел формування ринку криптовалют і всього, що з ним пов'язано. Біткоїн, а точніше технологія його виробництва, з'явився на світ у 2008 році. Його «батьком-засновником» є Сатоші Накамото, який опублікував файл з описом протоколу та принципу роботи платіжної системи за допомогою однорангової мережі. Виробляти біткоїни може кожен, хто має необхідне обладнання, але при цьому ніхто не може самовільно змінювати кількість існуючих біткоїнів іншим шляхом, окрім як самостійно створювати їх за всіма правилами системи.

Торгівля криптовалютою – це купівля-продаж криптовалют з метою отримання прибутку. Торгова стратегія в торгівлі криптовалютою, сформульована інвестором, – це алгоритм, який визначає набір заздалегідь визначених правил для купівлі та продажу на ринках криптовалют.

Теорія ігор – це математична дисципліна, що сформувалася у середині минулого століття, за часів великих війн. Гра – частина нашого життя, і конфлікти, які виникають і вирішуються через воєнні дії, скандали, проблеми – це все є елементами теорії ігор. Коли виникла математична теорія ігор, математики показали, що завдання теорії ігор –

формалізувати реальну життєву ситуацію, що є грою; навчитися за допомогою математичних методів отримувати додаткові переваги у цій грі, вибирати оптимальні стратегії.

## **2. ТЕОРІЯ ІГОР З ВИКОРИСТАННЯМ АНАЛІЗУ КРИПТОРИНКУ**

Теоретичний ігровий підхід до фінансових ринків є не менш «науковим», ніж будь-який інший; це просто нешаблонно. Якщо гру на фінансових ринках можна розглядати як форму гри, то чому б не використати науку про ігри для їх аналізу?

Прийняття рішень в умовах невизначеності відбувається, коли кожне рішення призводить до конкретного результату, який відбувається з невідомою ймовірністю. Якщо ви йдете додому в годину пік і хочете мінімізувати час, витрачений на дорогу, чи варто вибирати «звичайний» маршрут чи спробувати альтернативний маршрут? Будь-яке рішення призводить до конкретного результату (прибуття додому), але ви можете не знати який маршрут є найшвидшим у конкретний день.

Щоб визначити ціну на акції або валюту, або на будь-який інший фінансовий інструмент, існує два основних типи аналізу фінансових ринків. Перший називається технічний аналіз (ТА). Технічний аналіз передбачає, що для прогнозування ціни на актив у майбутньому достатньо знати статистику як у минулому ринок реагував на якісь певні фактори і швидше за все з великою ймовірністю і в майбутньому поводитиметься так само. Другий – фундаментальний аналіз – це метод аналізу та прогнозування ринку з урахуванням усіх фундаментальних фактів: політичні, переговори, вибори, зміна влади, всі можливі форс-мажори, природні катаклізми, війни та революції, психологічні фактори, якісь чутки, очікування. Це все також може впливати на фінансові ринки. Інша група факторів – це економічні чинники і їх потрібно розділити на дві категорії. Перша, – якщо ми говоримо про економічні чинники, пов'язані з якоюсь конкретною компанією, де необхідно враховувати всі публікації, звітності, засідання ради директорів, можливо прийняття якихось важливих рішень. Другий, – якщо ми говоримо про фундаментальні фактори в рамках світової економіки або економіки конкретної держави.

У дослідженні розглядається практичне застосування знаходження найкращої стратегії у заданій задачі з критеріями Севіджа, Байеса і Лапласа, а також аналіз отриманих результатів. У роботі використовуються реальні дані, отримані з відкритих доступів сайтів Yahoo Finance та Trading view. Для задачі візьмемо три криптовалюти: одна, яка відома усім індивідам, що займаються трейдингом, або просто знайомі з цією темою; другою буде криптовалюта нової української біржі, що запустилась порівняно нещодавно та випустила свій криптокоїн; третя – теж відносно нова криптовалюта – токен управління проектом STEPN, грою, яка працює за принципом «ходи і заробляй»; GMT використовується для розрахунків в ігровому пулі, дозволяючи гравцям заробляти гроші просто ідучи або прогулюючись. Це валюти Ethereum, Qmall coin та STEPN (GMT) відповідно. Надалі будемо розглядати їх як потенційні криптовалюти для інвестування. Сама задача буде поставлена таким чином: трейдеру необхідно вибрати одну з трьох криптовалют, в яку він потенційно може вкластися з 10 тисячами гривень, якщо прибуток залежить від того, яким буде ринок: ріст, консолідація (стабілізація курсу) або падіння. Також важливо встановити рамки інвестування; в нашій задачі період інвестування буде місяць - з 28 березня 2022 року до 28 квітня 2022 року. Використовуючи критерій Байеса для встановлення ймовірностей, необхідно перш за визначити можливі стани ринку в найближчій час. Важливо розуміти, що Біткоїн є засновником, «батьком» усіх інших існуючих коїнів, тому треба враховувати, що якщо Біткоїн знаходиться на стадії падіння, то і всі інші монети не будуть рости (принаймні так швидко та з великими іксами, як при Біткоїні у стадії консолідації чи росту). По-друге,

використовуючи технічний аналіз необхідно проаналізувати графік Біткоїну за періоди: місяць, півроку та рік для бачення усїєї картини та спрогнозувати вірогідності стану ринку через місяць.

Необхідно проаналізувати графіки таких періодів:

1. 12 місяців (28.03.21 - 28.03.22)
2. 6 місяців (28.09.21 - 28.03.22)
3. Місяць (28.02.22 - 28.03.22)



Рисунок 1. Графік біткоїну за період 28.03.2021 - 28.03.2022, таймфрейм 1 тиждень

Бачимо головну картину, яку маємо з Біткоїном, а саме те, що на графіку є фігура «подвійна вершина», що виступає провідником розвороту тренду (Рис. 1). Тобто, з ймовірністю 90% тренд піде вниз. Тепер необхідно дивитись більш детально період за пів року (Рис. 2).



Рисунок 2. Графік біткоїну за період 28.09.2021 - 28.03.2022, таймфрейм 1 день

На даному графіку (Рис. 2) бачимо, що тренд дійсно йде все вниз та вниз, але з лютого спостерігаємо деяку стабільність, тобто маємо етап консолідації. Консолідація – це термін



для акцій або цінних паперів, які не продовжують і не змінюють більшу цінову тенденцію. Консолідовані акції зазвичай торгуються в обмеженому діапазоні цін і пропонують відносно мало торгових можливостей, доки не з'явиться інша модель. Наступний етап може бути будь-яким, але так як ми розуміємо ситуацію загалом (дивлячись на річний графік) можемо припустити, що графік піде вниз. У той час на 28 березня маємо відмітку у 47 152 долара за 1 біткоїн (Рис. 3).



Рисунок 3. Графік біткоїну за період 28.02.2022 - 28.03.2022, таймфрейм 4 години

На графіку (Рис. 3) за період у місяць бачимо, що нібито графік йде наверх, пробиває максимум за останній місяць, та, можливо, така ціна протримається або піде вгору на дуже короткий час. Та, дивлячись трохи далі та спираючись на загальний графік, можемо не передбачити значний ріст ціни. Тому маємо ймовірність того, що буде падіння цін протягом зазначеного періоду - 50%, консолідації - 35% та росту - 15% відсотків.

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Необхідно залучити не тільки технічний, а й фундаментальний аналіз. Наразі Qmall та STEPН (GMT) нові монети, тому важливо розуміти, що на нових проєктах на початку є багато новин стосовно них (наприклад безкоштовна комісія на вивід/залиття грошей або підключення нових функцій, розіграші і т.п.), а, отже, і запит та інтерес багатьох людей інвестувати дуже великий. Тому якщо буде ріст, то він буде стрімким, дасть багато іксів (але недовгим). Щодо Ethereum, то ця криптовалюта вже давно у використанні на ринку, та вона «йде» за біткоїном, тому якщо біткоїн піде вниз, то і ціни на Ethereum також підуть вниз.

Таблиця 1. Таблиця вигравів монет та ймовірностей на кожному стані ринку

State/ Cryptocurrency	Growth	Consolidation	Loss
Ethereum	2	1	- 3
Qmall	3	2	- 1
STEPN (GMT)	4	2	- 2
probability	0,15	0,35	0,5

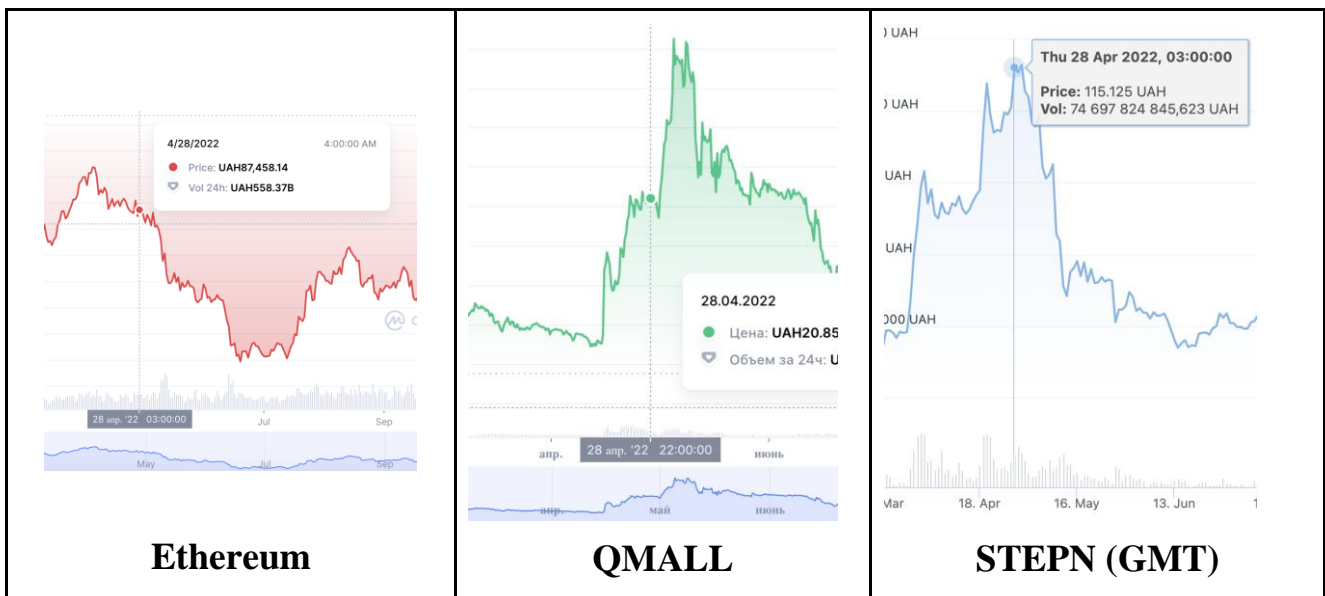
Таблиця 2. Порівняльна таблиця отриманих результатів з використанням критеріїв Лапласа, Байеса та Севіджа

Criterion	Laplace	Bayes	Savage
Result	Qmall, STEPН (GMT)	Qmall	Qmall, STEPН (GMT)

З результату бачимо, що непогано було б інвестувати в монети Qmall та STEPН (GMT), але є результат за критерієм Байеса, який показує лише один варіант інвестування – це монета Qmall.

Нижче наведена таблиця з вже відомими цінами на наш період з 28 березня по 28 квітня.

Таблиця 3. Таблиця даних з вартості криптовалют на заданий період



Для розрахунків була реалізована програма, яка знаходить кращу стратегію за матрицею виграшів за критеріями Лапласа, Севіджа та Байеса. Програма розроблялася в середовищі IntelliJ IDEA CE та була реалізована на мові Kotlin.

#### 4. ВИСНОВКИ

У даному дослідженні була описана задача про виявлення кращої стратегії для інвестування на крипторинку з завчасно обраних варіантів: монети QMALL, STEPН (GMT) та Ethereum. Також були обрані дані вартості біткоїну у графіках за різний період часу та потім проаналізовані, використовуючи технічний аналіз. Більше того, був задіяний фундаментальний аналіз для виявлення наскільки швидко або стрімко монета або токен може вирости.

Програма, що виявляє найкращий шлях для інвестування та заробляння гарного прибутку, була написана на мові JVM Kotlin у середовищі IntelliJ IDEA. Продукт ще може бути вдосконалений, а саме додаванням більшої кількості можливих монет для інвестування та реалізування юзер френдлі віконечка для вводу та виводу результату.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сергей Бондаренко. Теория игр Инструмент, благодаря которому блокчейн-революция стала возможной. URL: <https://www2.deloitte.com/ua/uk/pages/press-room/deloitte-press/2018/game-theory.html> (дата звернення: 16.11.2022).
2. Филиппов Е. Криптовалюта. М.: STFOREX, 2017. URL: [http://el.tfi.uz/images/Kriptoalyuta\\_ot\\_a\\_do\\_ua.pdf](http://el.tfi.uz/images/Kriptoalyuta_ot_a_do_ua.pdf) (дата звернення: 16.11.2022).
3. Круглик, Р. И., Брыкова, А. Л. Обзор бирж криптовалют для трейдерской деятельности. Постулат. 2018. № 1.
4. Shelton R. B. Gaming the market: Applying game theory to create winning trading strategies. John Wiley & Sons. 1997. Vol. 69.
5. Богоявленский С.Б. Критерий Лапласа. 2014. URL: [http://risking.ru/materials/risktheory/part2\\_9.html](http://risking.ru/materials/risktheory/part2_9.html) (дата звернення: 16.11.2022).

# СИСТЕМНИЙ ПІДХІД ДО АНАЛІЗУ КРЕДИТНОГО ТА РИНКОВОГО РИЗИКІВ

Смиковська Д.В.<sup>1</sup>, Бідюк П.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>smykovska.daryna@lil.kpi.ua

**Кредитний ризик і ринковий ризики є двома найбільш поширеними і критичними фінансовими ризиками. Для більш цілісного та ефективного аналізу цих типів ризику варто застосовувати системний підхід. Побудова системи підтримки прийняття рішень (СППР) є досить розповсюдженим фактичним впровадженням системного підходу, що значно полегшує процес аналізу ризиків. Метою роботи є аналіз ризику контрагента та ризику процентної ставки, а також їх кореляції. Результатом дослідження є побудована СППР, що моделює процентну ставку, оцінює ймовірність дефолту та взаємозв'язок між двома типами ризиків за допомогою моделювання CVA. У роботі використано теоретичні та емпіричні методи дослідження.**

**Ключові слова:** системний підхід, кредитний ризик, ринковий ризик, СППР, CVA, метод Монте-Карло.

## 1. ВСТУП

Слід зазначити, що будь-яка фінансова дія передбачає певний рівень ризику. Ризики є наслідком наявності невизначеності та є випадковими величинами, оскільки визначаються існуванням різноманітних факторів, вплив яких на результати фінансової діяльності неможливо достовірно передбачити заздалегідь.

Ринковий і кредитний ризики часто розглядалися як не пов'язані джерела ризику: види ризиків вимірювалися окремо та управлялися окремо. Проте кредитний і ринковий ризики переплітаються. По-перше, кредитний ризик залежить від факторів ринкового ризику, оскільки ймовірність дефолту, вартість застави та вартість вимог можуть залежати від процентних ставок, обмінних курсів або інших ринкових цін. По-друге, ринковий ризик залежить від факторів кредитного ризику, оскільки дефолт контрагента може відкрити раніше закрити позицію.

Неповне розуміння взаємозв'язку між кредитним і ринковим ризиком все ще є джерелом недоліків в управлінні ризиками фінансових установ. Інтегроване управління кредитними та ринковими ризиками може допомогти окремим установам досягти кращого розуміння їхньої загальної ризикової позиції, таким чином сприяючи покращенню стабільності фінансової системи в цілому.

Наразі досить критичною є тема якісних методів аналізу ризиків. Це пояснює чому популярним є використання методології, яка передбачає детальне вивчення усіх ознак того чи іншого типу ризику. Саме такою можна розглядати системну методологію, адже вона вимагає поетапності дій у вивченні ризиків та допомагає в удосконаленні процесів збору та обробки інформації, і оцінювання того, що використовується в процедурах моделювання, і розробці управлінських рішень. Тому можна зазначити, що саме системний підхід надасть змогу визначити рівень ризику та його кореляцію з іншим типом ризику.

## 2. МЕТОДИ ОЦІНЮВАННЯ КРЕДИТНОГО ТА РИНКОВОГО РИЗИКУ

Для того щоб ефективно керувати ризиком потрібно його ідентифікувати та провести аналіз для загальної структури розуміння його виникнення і трендів притаманних цьому ризику. Наступним етапом є оцінювання ризику, що дозволяє корпораціям, урядам та інвесторам оцінити ймовірність того, що несприятлива подія може негативно вплинути на бізнес, економіку, проект або інвестиції. Хоч ризик не завжди являє собою саме настання несприятливих подій при його оцінюванні завжди визначаються саме можливі втрати.

Істотним є не тільки оцінювання можливих втрат, а й структурний та цілісний підхід до огляду чинників виникнення ризику і його залежності від факторів, що впливають на подальші можливі відхилення від очікуваних результатів. Тому слід акцентувати увагу на важливості імплементації системного підходу.

Системний підхід до аналізу ризиків можна визначити як метод дослідження інтегральних залежностей і взаємодій факторів, що впливають на появу або прояв ризиків, прогнозування та оцінювання наслідків їх реалізації [1].

Як було зазначено раніше, одним із результатів імплементації системного підходу є саме побудова СППР. СППР генерує рішення за допомогою різноманітної множини алгоритмів і методів, включаючи алгоритми перетворення даних, ігрові методи пошуку та обробки даних, ситуаційний аналіз, методи прийняття рішень на основі нейронних мереж, імітаційне моделювання, метод когнітивної карти, байєсівські мережі та інше.

СППР може бути використана в діяльності малих підприємств з низьким ступенем забезпеченості та розвитку інформаційних технологій. Архітектура СППР розділена на такі рівні: - інтерфейс; - обчислювальний модуль (система обробки даних); - база знань і даних; - система отримання результатів (система прийняття рішень) [2, 3].

Аналіз кредитного і ринкового ризиків відбувається із застосуванням методів, заснованих на побудові певних математичних моделей. Існує множина таких методів і моделей, де серед найбільш популярних для аналізу ринкового та кредитного ризиків можна визначити моделі АРУГ, УАРУГ, VaR, лінійну регресію, дерева рішень, випадковий ліс, нейронні мережі та мережі Байєса, моделювання CVA (ринкова вартість кредитного ризику контрагента), модель Vasicek та модель Hull-White. Моделювання CVA для процентного свопу складається з: - моделювання процентної ставки; - обчислення ймовірності дефолту, очікуваного ризику; - підрахунку CVA.

Для прогнозування процентних ставок, які використовуються для процентного свопу використано модель Vasicek. Модель Vasicek вже давно використовується як техніка для прогнозування еволюції цін на фінансові інструменти. Ринковий ризик, час, волатильність і середній коефіцієнт повернення є компонентами цього підходу [4, 5].

Однофакторна модель – це така, яка має лише один фактор ризику, це свідчить про те, що зміни процентної ставки пояснюються одним компонентом (фактором ринкового ризику).

## 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Розрахунки дослідження базувалися на таких інструментах, як процентні свопи та кредитні дефолтні свопи (CDS). Для дослідження була обрана така інформація:

- 3-місячна середньозважена процентна ставка за міжбанківськими кредитами Libor;
- кредитний дефолтний своп контрагента JPMorgan Chase дефолтного свопу.

У цій роботі використано однорічний процентний своп з датою розрахунків від 17.09.2019 і терміном дії до 17.09.2020.

Процентні ставки прогнозувались з використанням моделі Vasicek. Усі симуляції починаються з періоду 0, який є датою початку обміну. Це вказує на те, що протягом усіх симуляцій процентна ставка періоду 0 буде встановлена на рівні 2,7859%. Довгострокове середнє значення становитиме 3,7337%, що є історичним середнім значенням 3-місячної ставки Libor у доларах США від початку до періоду 0. Волатильність визначається як історичне стандартне відхилення тримісячної ставки Libor у доларах США, яке становить 0,0427.

Завдяки моделюванню 10000 симуляцій процентної ставки методом Монте-Карло отримуємо наступний графік (Рис.1):

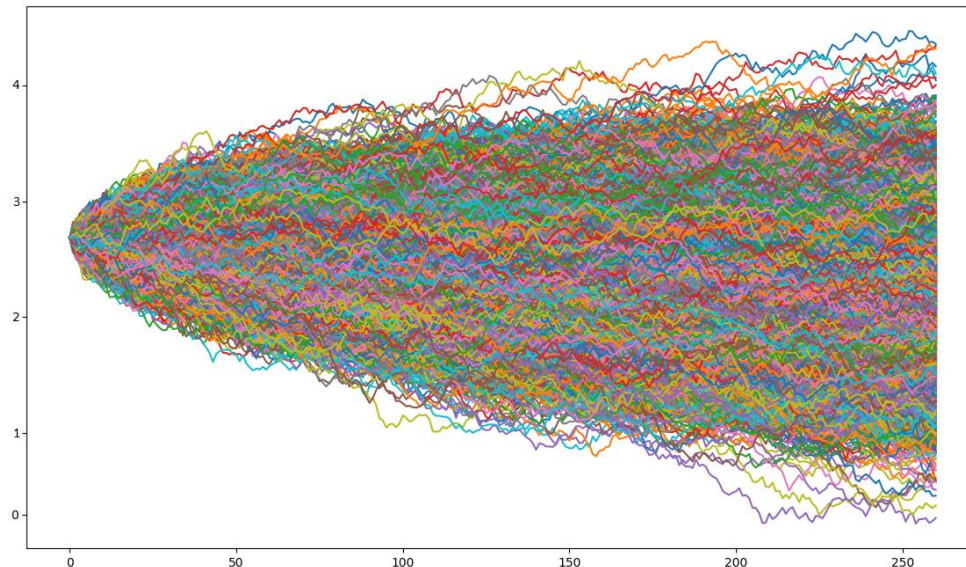


Рисунок 1. 10000 симуляцій процентної ставки

Найбільше значення у моделюванні становить близько 4,5%, тоді як найменше значення у моделюванні становить приблизно 0%. Модель Васічека враховувала від'ємні відсоткові ставки, і, як можна побачити, кілька симуляцій опускалися нижче 0% у деякі моменти часу.

Симуляції негативної ставки невеликі в основному через реверс середнього значення. Важко завершити моделювання негативним результатом, коли процентні ставки відхиляються в бік довгострокового середнього (що є позитивним). Однак, дивлячись на історичні 3-місячні ставки Libor в доларах США, це не обов'язково викликає занепокоєння, оскільки ці ставки ніколи не опускалися нижче 0% протягом тривалих періодів часу, але регулярно торкалися нульової нижньої межі під час Великої рецесії.

Далі ми розраховуємо щоденні шляхи впливу та моделюємо Expected Exposure (Рис. 2):

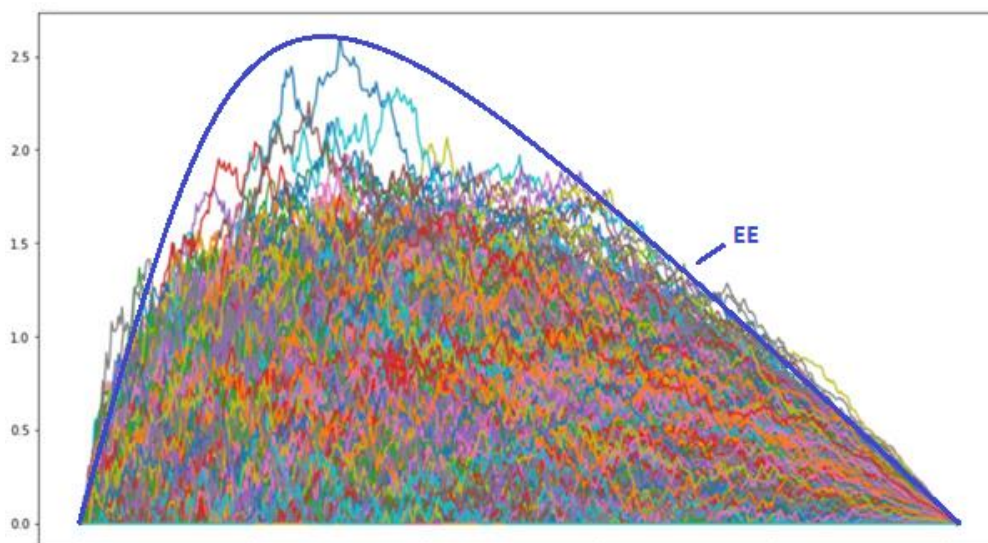


Рисунок 2. Шляхи впливу та Expected Exposure

LGD або втрата в разі дефолту є фіксованим параметром, і в цьому дослідженні дорівнює 61%.

JPMorgan Chase є контрагентом у цьому дослідженні, і оскільки це велика фірма, вона має відповідні CDS, які будуть використовуватися в наступних обчисленнях.

Розраховуємо ймовірність дефолту, як ймовірність дефолту між двома періодами. Отримані значення подані в таблиці 1.

Таблиця 1. Граничні ймовірності дефолту для JPM

Дата погашення	Гранична ймовірність дефолту
6 місяців	0.0024039308084044123
1 рік	0.0032476259533171154
2 роки	0.004158892130961667
3 роки	0.004967525197541266
4 роки	0.006329247322708333
5 років	0.007743749056949012
7 років	0.011023452687434012
10 років	0.014031132505716224

У таблиці нижче показано граничний ризик дефолту для термінів погашення на кривій спреда CDS. Однак нам потрібні ці цифри на щоденній основі для того щоб розрахувати оцінку CVA для процентних свопів. Щоденні значення будуть знайдені за допомогою інтерполяції поліномами методом Ерміта для того щоб гарантувати, що щоденні ймовірності за замовчуванням обчислюються на основі таблиці 4.

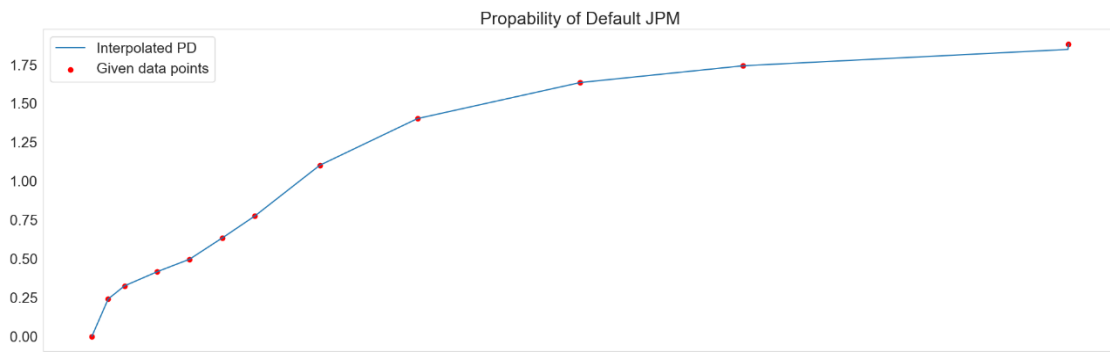


Рисунок 3. Интерпольована ймовірність дефолту

Ми бачимо, що ймовірність дефолту з часом зростає. Це цілком очікувано, оскільки компанія з хорошим кредитним рейтингом має низький ризик дефолту в короткостроковій перспективі, який з часом зростатиме.

Таблиця 2. Значення CVA

CVA	31479
SE для CVA	0.0251 %

Потім, для того щоб зафіксувати кореляцію між зміною спреда CDS та зміною процентної ставки, ми моделюємо фактичні зміни 3-місячної ставки Libor у доларах США протягом періоду свопу, а також обчислюємо ймовірність дефолту на основі фактичних спредів CDS.

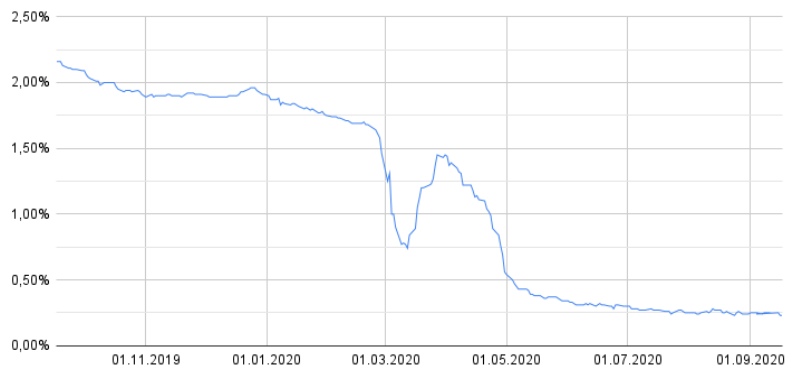


Рисунок 4. Ставка Libor

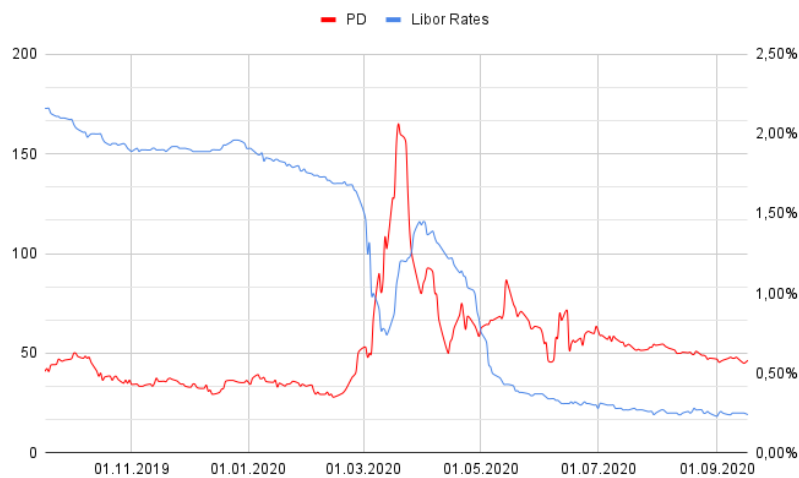


Рисунок 5. Ймовірність дефолту та ставка Libor



Рис. 5 показує, що існує певна відносно послідовна залежність між зміною спредів CDS та зміною процентних ставок. Процентні ставки знизилися в той же час, коли ймовірність дефолту різко зросла. У результаті, оскільки процентні ставки залишалися низькими, ймовірність дефолту залишалася високою. По суті, тенденція, візуалізована на Рис. 5, вказує на те, що зі зниженням процентних ставок знизилися і кредитна якість контрагента.

#### **4. ВИСНОВКИ**

На сьогоднішній день через величезну важливість деривативів для сучасної економіки всім, хто цікавиться міжнародним бізнесом, важливо якісно та ефективно аналізувати невизначеності та володіти знаннями про можливості цих інструментів. Як було зазначено раніше, системний підхід є ефективним інструментом методології досліджень для структурованого дослідження об'єкта як цілісної множини елементів в сукупності відношень і зв'язків між ними. Для ефективного аналізу кредитного та ринкового ризику із застосуванням системного підходу рекомендується СППР. У ході виконання дослідження реалізовано таку систему, яка дозволяє моделювати процентну ставку для подальшої імітації змін на основі єдиного фактору ринкового ризику. За допомогою моделювання значень процентної ставки та ймовірності дефолту ми мали змогу наочно побачити кореляцію між ризиком процентної ставки та ризиком контрагента.

#### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Згуровський М. З., Панкратова Н. Д. Основи системного аналізу: підруч. для студ. вищ. навч. закл. К. : Вид. група ВНУ, 2007. 544 с.
2. Белевець М. С., Бідюк П. І., Кузнецова Н. В. Розробка системного підходу до менеджменту ризиків. Системні науки та кібернетика. 2015. № 1. С. 30–48. URL: [http://mmsa.kpi.ua/sites/default/files/ssc/issues/ssc\\_1\\_2015.pdf](http://mmsa.kpi.ua/sites/default/files/ssc/issues/ssc_1_2015.pdf).
3. Бідюк П. І., Кузнецова Н. В., Терентьев О. М. Система підтримки прийняття рішень для аналізу фінансових даних. Наукові вісті НТУУ "КПІ". 2011. № 1. С. 48–61.
4. Brace, A., Gatarek, D., and Musiela, M., .The market model of interest rate dynamics. *Mathematical Finance*, 7(2), (1997), 127–47.
5. Theodore Barnhill and William Maxwell. Modeling correlated market and credit risk in fixed income portfolios. *Journal of Banking and Finance*, 26:347–374, 2002.

# МЕТОДИ ПОБУДОВИ ПРОГНОЗУ ФІНАНСОВИХ ДАНИХ. АНАЛІЗ ТА ПОРІВНЯННЯ

Собкович В.С.<sup>1</sup>, Тимошук О.Л.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> sobk2015@gmail.com, <sup>2</sup> o.tymoshchuk@kpi.ua [0000-0003-1863-3095]

**В дослідженні розглянуто задачу прогнозування ціни для акцій компаній, які тісно пов'язані із фінансовими ринками. Для аналізу методів побудови прогнозу часових рядів пропонуються методики для роботи з фінансовими даними. Побудована програмна система прогнозу дозволяє отримати дані прогнозування для фінансових рядів на кілька кроків вперед, що дає можливість безпосереднього використання результатів дослідження для роботи на торговельних фінансових біржах.**

**Ключові слова:** прогнозування фінансових даних, фінансові ринки, методи побудови прогнозу, фінансові часові ряди, порівняння даних.

## 1. ВСТУП

Економічна стійкість різних компаній залежить від різноманітних показників. Одними із найважливіших є фінансові характеристики. Можливість побудови прогнозу є важливою складовою для ефективної роботи компанії.

Для аналізу обираємо акції компаній JPYX, AABV, SFIX і PNC що мають різні тенденції на ринку. Відповідно вони утворюються на фондових ринках. Також дані ціни є волатильними та будуть постійно змінюватись. Залежить це від багатьох причин, наприклад сезонність, або загальні економічні процеси, світова криза, війна, пандемія, погодні чинники.

Сьогодні галузь торгівлі на фондових ринках залишилась інструментом, що не зазнав значних втрат та зберіг можливість працювати, тому є актуальною побудова прогнозу показників для фінансового ринку.

## 2. МАТЕМАТИЧНІ МОДЕЛІ ДЛЯ ПОШУКУ РІШЕНЬ В ЗАДАЧАХ ПРОГНОЗУВАННЯ

### 2.1 Алгоритм ARIMA

ARIMA це клас моделей, які «пояснюють» певний часовий ряд на основі його власних минулих значень, тобто особистих лагів та похибок прогнозу з лагом, щоб можна було використовувати рівняння [1]. Будь-який «несезонний» часовий ряд, який демонструє закономірності і не є випадковим білим шумом, можна моделювати за допомогою моделей ARIMA, що задані формулою (1):

$$\Delta^d y_t = a_1 \Delta^d y_{t-1} + \dots + a_p \Delta^d y_{t-p} + \varepsilon_t - \beta_1 \varepsilon_{t-1} - \dots - \beta_q \varepsilon_{t-q}, \quad (1)$$

де  $y$  - є деяким процесом;  $\Delta^d$  - оператор різниці часового ряду порядку  $d$ ; параметри  $a, \beta$  - параметри моделі;  $\varepsilon_t$  - білий шум. Першим кроком для побудови моделі ARIMA важливо зробити часовий ряд стаціонарним. Найпоширеніший підхід це віднімання. Тобто від поточного значення відняти попереднє значення. Наступним кроком буде

визначення, чи потрібні моделі в цілому лаги для ковзного середнього (КС). Ми можемо дізнатися потрібне значення лагів КС, перевіривши графік часткової автокореляції.

Часткову авто кореляційну функцію можна уявити як кореляцію між графіком та її лагом. Відповідна математична формула (2) для часткової авто кореляційної функції для третього лагу має вигляд:

$$Y_t = a_0 + a_1 Y_{t-1} + a_2 Y_{t-2} + a_3 Y_{t-3}. \quad (2)$$

Тобто у випадку, якщо  $Y_t$  — поточний ряд, а  $Y_{t-1}$  — це перший лаг від  $Y$ , то часткова автокореляція лагу 3 ( $Y_{t-3}$ ) — це коефіцієнт  $a_3$  у наведеному рівнянні (2).

Шукану авто кореляційну функцію в стаціонарному ряді можна виправити, додавши достатню кількість термінів авто регресії. Отже, спочатку ми приймаємо порядок лагів авто регресії рівним стільки, скільки перетинають межу достатності на графіку часткової автокореляційної функції.

Також можна знайти на графіку авто кореляційної функції відхилення для кількості лагів ковзного середнього. Це дозволить побудувати більш точне прогнозування.

## 2.2 Алгоритм LSTM

Проаналізуємо роботу мережі довгої короткочасної пам'яті [2], що показано на Рис. 1.

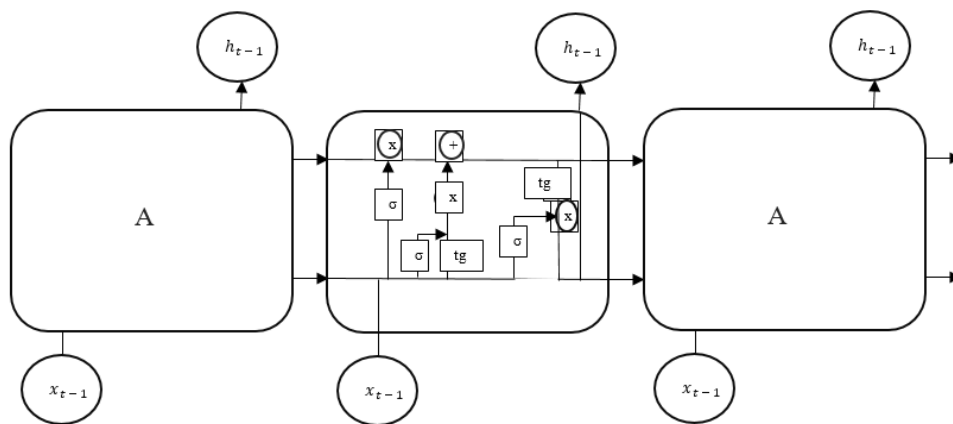


Рисунок 1. Принцип роботи алгоритму LSTM

Основною частиною в алгоритмі роботи є стан горизонтальної лінії, яка проходить через верхню частину блок-схеми. Вона може бути змінена сигмоподібним шаром, котрий передає значення від 0 до 1. Числа будуть нести інформацію про те, скільки попередніх даних ми візьмемо до наступного кроку [3].

Алгоритм роботи мережі. Першим кроком мережа з використанням сигмоїдального шару забирає чи відкидає частину значень. Відповідно, формула для роботи першого рівня описана виразом (3):

$$f_t = \sigma (w_f * [h_{t-1}, x_t] + b_f), \quad (3)$$

де  $f_t$  - це значення, котрі передаються до другого рівня роботи;  $\sigma$  - сигмоїдальна функція, що повертає значення від 0 до 1;  $w_f$  - вага відповідного нейрона;  $h_{t-1}$  - значення, отримані з попереднього кроку роботи мережі;  $x_t$  - дані для поточного значення часу;  $b_f$  - зміщення. На другому кроці мережа визначає нову інформацію, котру буде зберігати у стані комірки. Спершу сигмоїдальний рівень вирішить, які значення ми оновлюватимемо, а потім шар, що містить тангенс, створить вектор нових значень  $\hat{c}_t$ , котрі будуть додаватися до стану.

Значення, які передаються до наступного рівня, обчислюються відповідно до формули (4), а вектор нових значень показаний у формулі (5):

$$i_t = \sigma (w_i * [h_{t-1}, x_t] + b_i) \quad (4)$$

$$\hat{C}_t = \text{tg} (w_c * [h_{t-1}, x_t] + b_c) \quad (5)$$

Надалі буде оновлюватися останній стан  $C_{t-1}$ . Він передаватиметься до  $C_t$ . Попередній стан буде множитись на  $f_t$  (функція воріт, які забувають) та до знайденого числа додаватиметься  $i_t * \hat{C}_t$  (дані поточного значення часу помножені на отримані значення попереднього кроку роботи мережі). Отримані значення будуть станом комірки у даний проміжок часу. Формула (6) буде відображати алгоритм:

$$C_t = (f_t * C_{t-1} + i_t * \hat{C}_t). \quad (6)$$

На останньому кроці мережа буде визначати, які значення будуть шуканими. Ці дані будуть залежні від стану клітинки. Перш за все буде працювати сигмоїдальна функція. З її допомогою ми вирішимо, які частини стану однієї клітинки ми виведемо. Далі перевіриться стан клітинки (будуть значення в проміжку між -1 та 1). Тепер ці значення помножаться на результат сигмоїдального шару, це треба для виведення тих значень, які мали залишитися. Відповідні значення для вихідного шару зображені в формулі (7) та отримані результати роботи відповідно у формулі (8):

$$O_t = \sigma (W_o * [h_{t-1}, x_t] + b_o) \quad (7)$$

$$h_t = (O_t * \text{tg} (C_t)). \quad (8)$$

### 3. ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ

Початкові дані представляють собою вибірку часових рядів, акції різних компаній, які мають різні тенденції на ринку. Зокрема, ці тенденції розуміють під собою ведмежий, бичачий ринок, та горизонтальні тренди. Були вибрані акції компаній JPYX, AABV, SFIX і PNC.

Для роботи моделі ARIMA значення  $p$  та  $q$ , які відповідають за порядок авторегресії і порядок ковзного середнього, використовувалися від 1 до 12. Характеристика  $d$ , яка означала даний порядок диференціювання дорівнювала 1 та 2. Модель працює гарно на невеликих проміжках. Прогноз побудований для останніх 15 параметрів. Для результату було отримано значення критерію Акайке та статистика Дарбіна-Уотсона.

Проаналізуємо найкращі моделі для усіх часових рядів відповідно до критерію Акайке (для моделей АРІКС).

Наступним методом прогнозу є модель LSTM.

Для SFIX і JPYX число даних що подається на вхід дорівнює 260 значень (число робочих днів у році), також для AABV обрано 840 значень і для PNCД подано 1270 значень (взяті за останні 3 та 5 років відповідно). З них 50 значень будуть в перевірочній вибірці. Розмір партії = 50. Кількість епох дорівнює 25.

Порівняння результатів роботи для SFIX показано у таблиці 1.

Таблиця 2 демонструє порівняння результатів роботи для JPYX.

У таблиці 3 представлено порівняння результатів роботи для даних AABV.

Порівняння результатів для PNCД показано у таблиці 4.

Таблиця 1. Результати роботи для SFIX

	ARIMA(8,2,1)	ARIMA(8,1,4)	ARIMA(9,2,2)	ARIMA(6,2,1)	LSTM
R2	0,78228624	0,790338	0,773869	0,781441	0,559485
Sum squared Resid	11,36	11,73	11,72	11,26	13,85260
AIC	165,687634	197,6674	268,3034	262,6322	
DW	2,08357188	1,757118	2,014487	2,023688	
MAE	0,03793743	0,114221	0,080607	0,039669	0,695109

Таблиця 2. Результати роботи для JPYX

	ARIMA(4,1,4)	ARIMA(5,1,3)	ARIMA(5,1,5)	ARIMA(6,1,4)	ARIMA(6,1,5)	LSTM
R2	0,874897018	0,87812228	0,87897618	0,88848263	0,878978	0,654425
Sum squared Resid	45,49595539	44,28559090	44,25381031	40,46509896	44,13075778	201,7806
AIC	108,9511491	109,334338	110,460998	111,432053	113,073335	
DW	2,039194292	2,05487416	2,0020572	2,0266948	1,97412352	
MAE	0,02544935	0,02245648	0,0282241	0,02262397	0,02190825	0,596936

Таблиця 3. Результати роботи для AABV

	ARIMA(2,2,2)	ARIMA(10,2,2)	ARIMA(11,2,4)	LSTM
R2	0,775330093	0,75601934	0,74938959	0,821961
Sum squared Resid	0,00110278	0,00119473	0,00123537	0,02249
AIC	626,9193942	633,469462	634,631295	
DW	2,033178966	1,99047297	2,02201218	
MAE	0,000178533	0,00019655	0,00018912	0,00010725

Таблиця 4 – Результати роботи для PNCD

	ARIMA(9,1,8)	ARIMA(10,1,6)	ARIMA(10,1,10)	ARIMA(11,1,10)	ARIMA(11,1,11)	LSTM
R2	0,847332831	0,846372419	0,853333074	0,849185742	0,842075675	0,9308607
Sum squared Resid	578,49704726	574,00493475	547,0556718	557,37438677	592,903753	7437,0384
AIC	1512,748571	1468,7141449	1468,705809	1471,181233	1472,931583	
DW	2,006802687	1,990320241	1,993016397	1,98793682	1,988749209	
MAE	0,1090303	0,118914	0,114430183	0,092760017	0,099412983	0,020977

У фінансах коефіцієнт R2 вище 0,7 означає високий рівень кореляції. Сума квадратів залишків визначає наскільки добре регресійна модель пояснює або представляє дані в моделі. Інформаційний критерій Акайке буде використовуватись для порівняння

відповідності різних регресійних моделей. Він кращий для моделі де є найнижче значення AIC. В роботі для значень SFIX якщо порівнювати з моделями ARIMA, краще працює модель ARIMA (8, 2, 1). Для значень JPYX найкраща ARIMA (4, 1, 4). Для акцій AABV найкраще буде ARIMA (2, 2, 2). Для значень PNCD найкращою серед авторегресійних інтегрованих ковзних середніх є ARIMA (10, 1, 10). Наступною характеристикою показана Статистика Дарбіна Уотсона. Це є тест на автокореляцію залишків із статистичної моделі або регресійного аналізу. Статистика вимірюється від 0 до 4, також 2 означає, що у вибірці не виявлено автокореляції. Середня абсолютна похибка це показник оцінки моделі, який є середнім абсолютних значень окремих помилок передбачення для значень. Значення найменше для SFIX для ARIMA (8, 2, 1), для JPYX даних для ARIMA (5, 1, 3), для AABV а також для даних PNCD найменше для алгоритму LSTM.

#### **4. ВИСНОВКИ**

В роботі було досліджено нейронну мережу LSTM та метод ARIMA.

Порівнявши результати роботи моделей для часового ряду SFIX найкраще працює модель ARIMA (8, 2, 1). Вона дає найкращі значення у порівнянні з моделями ARIMA. Також вона має менші значення середньої абсолютної помилки. Відповідно для часового ряду JPYX найкращі значення дає модель ARIMA (4, 1, 4). Вона має найкращі значення критерію Акайке порівнюючи з іншими моделями та найменшу середню помилку в порівнянні з мережею LSTM. У AABV найкращою є LSTM. Її MAE менша за значення усіх моделей. Для PNCD найкраща модель також LSTM. Де значення середньої абсолютної помилки менші за значення інших моделей відповідно.

Порівнюючи результати зростаючого та спадного ринку, а також часового ряду без тренду можна зазначити що алгоритм ARIMA добре працює на невеликому розмірі даних, в той час, коли алгоритм LSTM дає хороші результати для великих часових рядів. Також точність роботи залежить від початкового набору значень. Ринок має бути зі чітко визначеним трендом, і також для правильності роботи не потрібно обирати нестабільний ринок. Ці тенденції дозволяють покращити отримані значення.

#### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Модель ARIMA [Електронний ресурс] – Режим доступу до ресурсу: <https://www.machinelearningplus.com/time-series/arma-model-time-series-forecasting-python/>.
2. Introduction to LSTM network [Електронний ресурс] – Режим доступу до ресурсу: <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>.
3. Алгоритм LSTM [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/company/wunderfund/blog/331310/>.

# СИСТЕМНИЙ ПІДХІД ДО АНАЛІЗУ ФІНАНСОВИХ РИЗИКІВ

Танчук М.

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

mariya.kovalenko.2000@gmail.com

**Ризики є елементами ведення фінансової діяльності та є наслідком невизначеності, тому для ефективного управління ними доцільно використовувати системний підхід, одним із результатів застосування якого може стати створення Системи підтримки прийняття рішень (СППР), яка дасть змогу суттєво полегшити аналіз ризиків. Метою роботи є аналіз фінансових ризиків для покращення їх оцінок із застосуванням системного підходу. Результатом дослідження є СППР, що виконує оцінку ринкового ризику фінансового портфелю із застосуванням методології VAR. У роботі було використано теоретичні та емпіричні методи дослідження.**

**Ключові слова: фінансові ризики, системний підхід, VAR, CVAR, аналіз ризиків, ринковий ризик.**

## 1. ВСТУП

Категорія ризику займає значне місце в економічній науці й практиці. Ризик - це елемент ведення будь-якої фінансової діяльності. Ризики є результатом невизначеності і являють собою випадкові величини, оскільки зумовлюються наявністю ряду факторів, вплив яких на результати фінансової діяльності не можна заздалегідь точно визначити.

Зважаючи на це, одним із найактуальніших завдань, що обумовлює доцільність дослідження фінансових ризиків та механізмів управління ними, є виявлення економічної сутності фінансових ризиків, їх моделювання та визначення впливу на ведення фінансової діяльності. При цьому, важливим є врахування усіх особливостей та властивостей ризику та розгляд його як цілісної системи взаємопов'язаних елементів, що визначають причини та наслідки появи ризиків [1]. Це можна досягнути шляхом застосування системної методології, яка, до того ж, дозволяє встановити міру і залежність фінансових ризиків від інших видів ризиків, тим самим дає можливість зробити оцінку ризику найбільш точною.

## 2. МЕТОДИ ОЦІНЮВАННЯ ФІНАНСОВИХ РИЗИКІВ

За одним із варіантів класифікації, фінансові ризики можна поділити на чотири великі категорії: ринковий ризик, кредитний ризик, ризик ліквідності та операційний ризик. Під час розрахунку ризикового капіталу портфелю традиційно розділяють ринковий та кредитний ризики та працюють з ними незалежно [6]. Для визначення рівня можливих втрат та забезпечення інвесторів від потенційних збитків при настанні несприятливих ситуацій на ринку надзвичайно важливим є запровадження ефективного процесу управління ризиками.

Загалом, процес управління ризиками можна поділити на такі етапи. Першим етапом є ідентифікація ризику. Цей етап має на меті виявити існуючі ризики, а також ті, що можуть потенційно виникнути у майбутньому. Наступним етапом є аналіз ризику, метою якого є одержання необхідної інформації про структуру та властивості ризику, що виникає. Оцінювання ризиків являє собою кількісний опис виявлених ризиків, у ході якого

визначаються ймовірність та розмір можливого збитку від конкретного ризику. Останнім етапом управління ризиками є контроль ризику та результатів процесу управління.

Оцінювання ризику полягає у визначенні втрат, пов'язаних з видами ризику, які були виявлені на першому етапі управління ризиком, шляхом побудови економіко-математичних моделей і проведення відповідних розрахунків [1].

Оскільки фінансовий ризик за своєю природою передбачає невизначеність щодо здійснення певного виду фінансової операції у майбутньому, то не завжди вдається отримати ідеальні оцінки [2]. Саме тому, на сьогодні, одним із найактуальніших напрямів розвитку економічної науки є пошук ефективних, науково обґрунтованих підходів та методів аналізу та оцінювання фінансових ризиків.

Одним із таких підходів є системний підхід, який можна визначити як напрям методології досліджень, що полягає у дослідженні об'єкта як цілісної множини елементів в сукупності відношень і зв'язків між ними, тобто розгляд об'єкта як модель системи.

Як зазначалось раніше, оцінювання ризиків відбувається із застосуванням методів, заснованих на побудові певних математичних моделей, які у подальшому використовуються для вимірювання визначених на попередніх етапах видів ризику.

Існує низка методів оцінювання фінансових ризиків, метою яких є побудова функціональної, ефективною та максимально точною моделі для певних категорій ризиків з метою створення точних прогнозів та визначення стратегії мінімізації ризику. До них можна віднести методологію VaR, мережі Байєса, а також деякі математичні моделі для прогнозування втрат, такі як ARCH та GARCH модель.

Великої популярності на сьогоднішній день набуває використання методології Value at Risk (VaR). Це зумовлено тим, що ця методологія володіє рядом безсумнівних переваг, а саме:

- дає змогу оцінити ризик в межах можливих втрат, співвіднесених з ймовірністю їх виникнення;
- вимірювати ризики на різних ринках універсальним чином;
- агрегувати ризики за окремими позиціями в єдину величину (інтегральний ризик) для всього портфеля активів, враховуючи при цьому інформацію про кількість позицій, і волатильності на ринку [4].

Загалом, VaR відображає мінімальний збиток інвестиції, який буде понесений на певному рівні ймовірності (довірчому інтервалі) або максимальний збиток, який буде реалізований на рівні ймовірності. Однак, чим менш стабільна інвестиція, тим більша ймовірність того, що VaR не дасть повної картини ризиків, оскільки ця міра байдужа до всього, що виходить за межі її власного порогу. Саме тому для оцінки екстремальних подій (найгірших випадків) поруч із показником VaR також розраховують показник очікуваних втрат Expected Shortfall, найбільш відомий як CVaR або Conditional Value at Risk (Рис. 1).

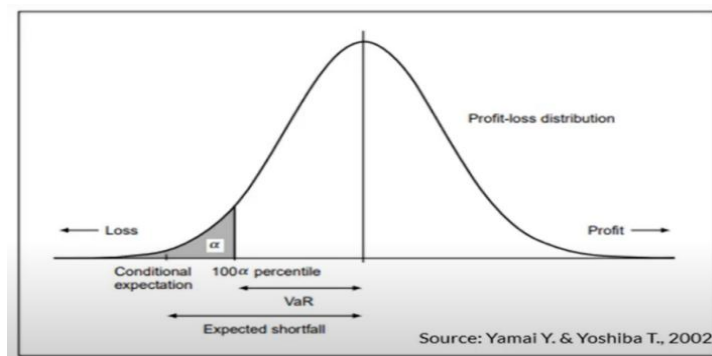


Рисунок 1. Візуалізація VaR та CVaR на графіку розподілу портфелю



До основних методів оцінювання VaR та CVaR відносяться: історичний метод, параметричний метод та метод моделювання Монте Карло, який відрізняється від попередніх більшою складністю обчислень, однак показує більш точні результати. Для полегшення обчислення рівня ризику та уникнення можливих помилок особою, що приймає рішення, було запропоновано розробити систему прийняття рішень (СППР), яка виконуватиме обрахування вартості під ризиком для певного фінансового портфелю.

СППР складається таких основних підсистем: підсистеми інтерфейсу користувача, підсистеми управління даними та основної підсистеми, яка тісно пов'язана із підсистемою функціональних модулів. Найважливішими модулями є модуль попереднього аналізу даних, що виконує завантаження даних, проводить оцінку дохідності портфелю та обраховує волатильність. Надалі ці підрахунки використовуються у модулі виконання обчислень, що виконує, власне, оцінку вартості під ризиком для різних методів та заданих глибини аналізу та рівня довіри. Рівень довіри або довірчий інтервал показує, з якою ймовірністю економічний капітал повністю покриє наслідки реалізації ризиків, які будуть меншими за розмір економічного капіталу [4]. Для проектованої СППР було реалізовано можливість встановлення рівнів довіри: згідно з рекомендаціями Базеля II, що складає 99%, а також згідно із системою “Risk Metrics” – 95% [5]. СППР надає можливість обирати для аналізу один із наступних методів: метод параметричної оцінки, найбільш поширений у формі варіаційно-коваріаційної моделі, метод історичного моделювання та метод імітаційного моделювання Монте-Карло, а також виконує порівняння результатів, отриманих під час використання вищевказаних методів. Провівши експериментальні обчислення у спроектованій СППР, можна не лише визначити розмір можливих втрат, а й зробити висновок про те, який з методів обчислення є найкращим для використання на практиці.

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для дослідження було сформовано інвестиційний портфель з цін акцій компаній Tesla, Amazon, Apple, Cisco та Microsoft Corp. У якості часового горизонту було взято період у 800 днів. Початкова інвестиція була обрана у розмірі 10 000 \$, а в якості періоду для прогнозування обрано один торговельний рік – 252 дні.

Для обчислення VaR необхідно розглянути розподіл прибутковості портфелю (Рис.2).

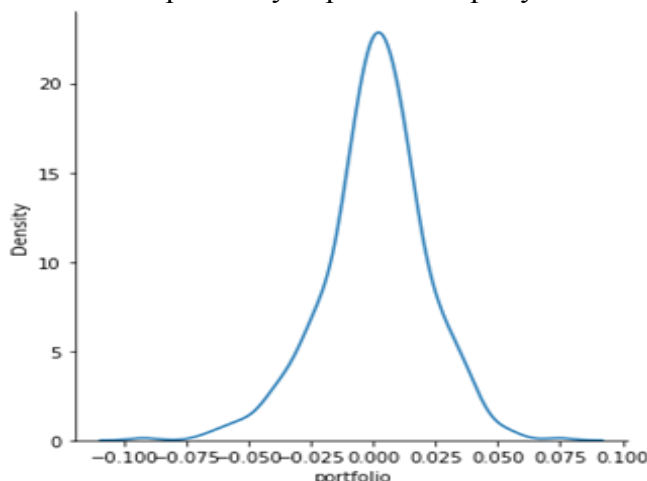


Рисунок 2. Розподіл прибутковості портфелю

Для розрахунку VaR припущення полягає в тому, що доходи розподілені нормально. Перший погляд на прибутковість показує, що щоденні доходи для портфелю розподіляються

приблизно нормально, але хвости більш товсті. Це натякає на доцільність використання поряд із нормальним розподілом розподілу Стюдента під час реалізації параметричного методу оцінки VaR. Використання розподілу Стюдента-t для оцінки VaR може значно покращити оцінку VaR [7].

Найпростішим методом розрахунку VaR є історичний метод, для якого припущення про розподіл доходів не робиться, а моделюється на основі історичних емпіричних даних. Візуалізуємо для цього методу на гістограмі розподілу доходностей портфелю (Рис. 3) підраховані значення VaR та CVaR для рівня довіри 95%.

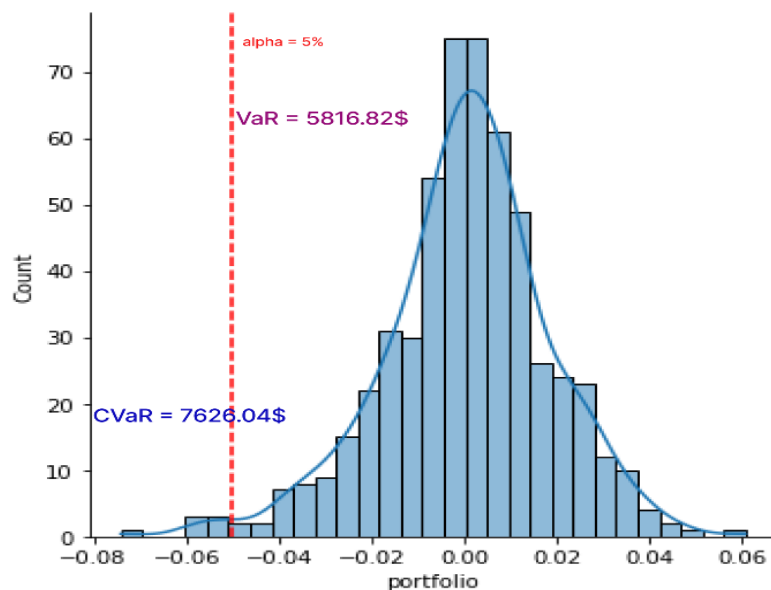


Рисунок 3. Значення VaR та CVaR для  $\alpha = 0.05$  для історичного методу

Для параметричного методу робиться припущення щодо розподілу доходів і використовуються історичні доходи портфелю та стандартні відхилення (оцінки) для визначення параметрів моделі. Для порівняння було реалізовано дві параметричні моделі VaR: нормальний розподіл та  $t$ -розподіл (жирні хвости).

Найскладнішим у реалізації виявився метод Монте-Карло, який хоч і відрізняється високою обчислювальною складністю, проте вважається найбільш точним та гнучким серед усіх методів обчислення VaR. У даному методі кожен фактор ризику моделюється за допомогою псевдовипадкових чисел. Метод Монте-Карло використовує для побудови траєкторії цін геометричний броунівський рух та дозволяє імітувати будь-який розподіл факторів ризику, включаючи важкі хвости розподілів. При цьому, збільшення кількості симуляцій підвищує точність обчислень. Застосувавши 400 симуляцій, було визначено вартість під ризиком для заданих довірчих інтервалів, а також визначено показники очікуваних втрат (CVaR) за межами інтервалів (Рис. 4).

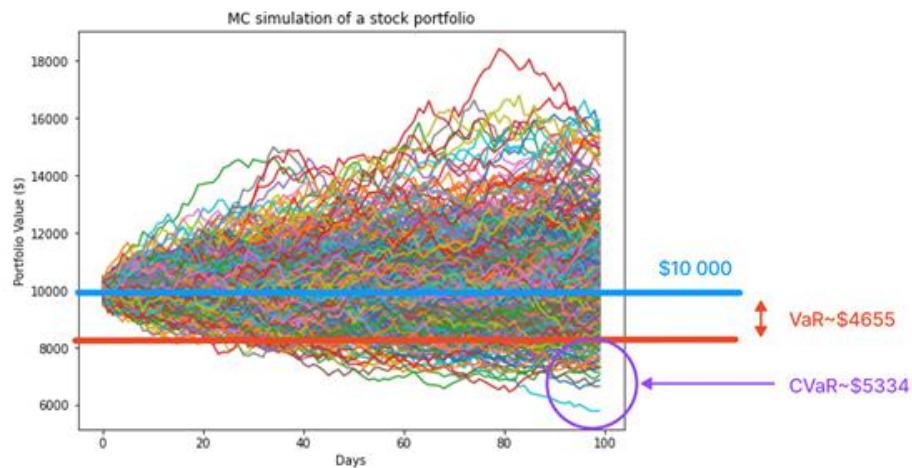


Рисунок 4. Значення VaR та CVaR для методу Монте Карло  $\alpha = 5\%$

Результати обчислень VaR та CVaR наведені у таблиці нижче (Табл. 1).

Таблиця 1. Значення VaR та CVaR для різних рівнів довіри та різних методів обчислення

Довірчий інтервал	Метод Знач. VaR/CVaR	Історичний	Параметричний $N(0,1)$	Монте Карло
		95%	VaR	5816.82
	CVaR	7626.04	4935.01	4174.38
99%	VaR	8690.92	5796.53	4754.52
	CVaR	10515.36	6903.9	5258.14

Зазначимо, що величини VaR та CVaR, оцінені за параметричним методом та методом Монте-Карло схожі, порівняно з історичним методом. Історичний метод припускає, що майбутні зміни у курсі акцій будуть подібні до минулих, у той час як метод Монте-Карло додає у передбачення випадковість шляхом застосування випадкової вибірки, що додає аргумент на користь використання саме цього методу для обчислення VaR.

Загалом, VaR надає хороше уявлення про втрати, які відбуваються менше заданого довірчого інтервалу, але не про ті, що лежать за його межею.

Наприклад, для досліджуваного портфелю значення VaR на рівні 95%, підраховане за допомогою методу Монте-Карло, становить близько 3300 доларів США. Це означає, що інвестор повинен мати достатньо капіталу, аби покрити ці втрати. Проте у випадку отримання збитків більшого розміру за межами заданого довірчого інтервалу (95%), ймовірно, інвестор не буде готовий до таких збитків, якщо розглядатиме виключно VaR.

Саме тому для кількісного визначення усіх збитків за межею VaR та врахування втрат у випадку настання найгіршого сценарію, варто застосовувати CVaR. Таким чином інвестор зможе отримати краще уявлення про потенційні найгірші втрати.

#### 4. ВИСНОВКИ

На сьогоднішній день фінансові системи стають все більш складними та невизначеними, тому виникає необхідність знаходження більш ефективних інструментів аналізу та прогнозування фінансових ризиків для якісного управління ризиками.

Одним із найбільш ефективних підходів до оцінювання ризиків на сьогоднішній день є системний підхід, який можна визначити як напрям методології досліджень, що полягає у дослідженні об'єкта як цілісної множини елементів в сукупності відношень і зв'язків між ними. Для ефективно оцінки ризиків із застосуванням системного підходу доцільним є створення інтелектуальної системи підтримки прийняття рішень.

У ході дослідження було реалізовано таку систему, яка дозволяє виконувати оцінку вартості під ризиком для певного інвестиційного портфелю. Ця система використовує методологію VaR та CVaR та реалізує низку методів обчислення значень VaR та CVaR. Результати дослідження показали, що найбільш ефективним методом виявився імітаційний метод Монте-Карло, який дозволяє найбільш точно підрахувати величину можливих втрат. Окрім того, було визначено, що для кращої оцінки ризику для найгірших сценаріїв варто застосовувати CVaR.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кузнєцова Н.В., Бідюк П.І. Теорія і практика аналізу фінансових ризиків: системний підхід, монографія. Київ 2020, 400 с.
2. Куцик П.О., Васильців Т.Г., Сороківський В.М., Стефаняк В.І., Сокорівська М.В. . Управління фінансовими ризиками навч. посібник. Львів: Растр-7, 2016. 318 с.
3. Башкіров О.В. Порівняльний аналіз VAR-методів оцінки ризику активів банку О.В. Башкіров, Проблеми і перспективи розвитку банківської системи України : зб. наук. праць ДВНЗ «УАБС НБУ». – Вип. 14. –С. 302–309.
4. Moraux F. How valuable is your VaR? Large sample confidence intervals for normal VaR. F. Moraux. Journal of risk management in financial institutions. – 2011. – № 4.2. – P. 189–200. – [Електронний ресурс]. – Режим доступу : <https://perso.univ-rennes1.fr/franck.moraux/research/JRMFI.pdf>
5. Longerstaey J. Risk Metrics TM – Technical Document . J. Longerstaey, M. Spencer. Morgan Guaranty Trust Company of New York: NewYork, 1996. [Електронний ресурс]. – Режим доступу : <http://yats.free.fr/papers/td4e.pdf>
6. Theodore Barnhill and William Maxwell. Modeling correlated market and credit risk in fixed income portfolios. Journal of Banking and Finance, 26:347–374, 2002
7. Lin, C. and Shen, S. (2006), "Can the student- t distribution provide accurate value at risk?", Journal of Risk Finance, Vol. 7 No. 3, pp. 292-300. <https://doi.org/10.1108/15265940610664960>

# **ПРОГНОЗУВАННЯ ІНДЕКСУ АКЦІЙ НА ОСНОВІ ХМАРНИХ ОБЧИСЛЮВАЛЬНИХ СЕРВІСІВ**

Тараненко О.С.<sup>1</sup>, Бідюк П.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>alloha.t.99@gmail.com

Торгівля акціями та цінними паперами набрала великої популярності в останній час. Але такий вид діяльності може нести за собою ризики втрати фінансів. Метою роботи є збір та попередня обробка вхідних даних та розробка моделі для прогнозування індексів акцій на основі хмарних обчислювальних сервісів. Результатом дослідження є навчена модель у вигляді багатошарової, згорткової нейронної мережі та мережі довгої короткочасної пам'яті із використанням хмарного сервісу AmazonWebServices. В роботі було використано емпіричні й теоретичні методи дослідження.

**Ключові слова:** індекс акції, машинне навчання, згорткові нейронні мережі, багатошарова нейронна мережа, мережа довгої короткочасної пам'яті, AmazonWebServices

## **1. ВСТУП**

Будь-які дії, пов'язані з цінними паперами, корелюють з певними ризиками, тому вони потребують точних розрахунків і точного аналізу. Хмарні сервіси забезпечують швидке та гнучке розгортання інфраструктури, забезпечують їх повсякчасну доступність. В якості архітектури доцільно використовувати нейронні мережі, бо вони не обмежуються типами і кількістю даних.

Із цього випливає необхідність у розробці моделей прогнозування індексу акцій, побудованих на основі штучних нейронних мереж, що використовують хмарні сервіси.

## **2. СХЕМА РЕАЛІЗАЦІЇ МОДЕЛІ СИСТЕМИ ПРОГНОЗУВАННЯ ІНДЕКСУ АКЦІЙ В РЕАЛЬНОМУ ЧАСІ НА БАЗІ ХМАРНИХ ОБЧИСЛЮВАЛЬНИХ СЕРВІСІВ**

Розглянемо схему реалізації моделі системи прогнозування індексу акцій на базі хмарних обчислювальних сервісів, функції її компонентів та логіку їх взаємодії (рис. 1). Система базуватиметься в Amazon Web Services, тому усі умовні позначення на схемі відповідають умовним позначенням із документації Amazon Web Services.

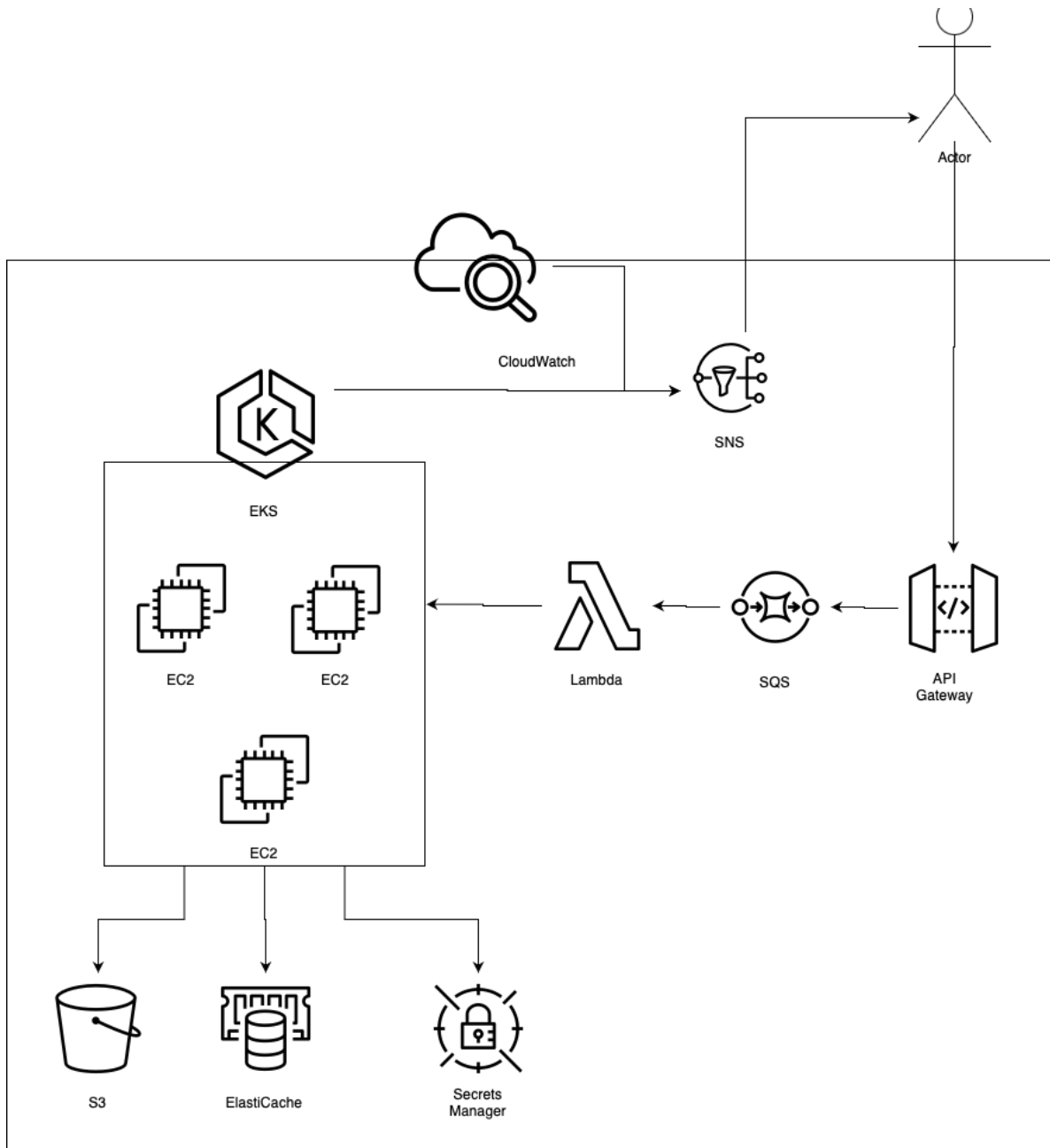


Рисунок 1. Схема реалізації моделі системи прогнозування індексу акцій на базі хмарних обчислювальних сервісів

Розглянемо взаємодію компонентів системи.

1. Користувач за допомогою командного рядка, або графічного інтерфейсу надсилає запит на зміни до системи на ендпоінт Amazon API Gateway.

2. На рівні Amazon API Gateway налаштоване правило фільтрації кількості запитів, щоб унеможливити DDoS атаку на сервіс.

3. Від Amazon API Gateway повідомлення складається у чергу Amazon SQS, яку налаштовано працювати за принципом FIFO, що забезпечує збереження порядку надісланих повідомлень.

4. Подія наявності у Amazon SQS черзі повідомлень запускає роботу Amazon Lambda, яка робить перевірку запитів з точки зору безпеки. Це може досягатись за допомогою

використання різних паролів, 2FAключів, або підтверджуючими запитами до джерела, яке має право надсилати запити.

5. Окрім того, у разі декількох одночасних запитів можливо сортувати їх за пріоритетом. До прикладу запити, які відповідають за фундаментальні зміни в системі, такі як її термінове вимкнення, будуть опрацьовуватись позачергово.

6. Потім запит направляється до Kubernetes кластера (Amazon EKS), де безпосередньо опрацьовується і вносить зміни по роботі системи.

7. Kubernetes кластер використовуватиме безпосередньо збір даних про індекс акцій, їх обробку, насичення даних додатковою інформацією, передбачення майбутньої ціни та реалізовуватиме стратегію купівлі та продажу акцій.

8. Для зберігання великих обсягів даних, які не потребуватимуть миттєвого доступу, використовуватиметься Amazon S3 (Simple Storage Service). А саме такі дані, як давньо історичної ціни на акції, застарілу додаткову інформацію про акції та про компанії, результати старих передбачень та інше.

9. Для зберігання даних, до яких потрібен миттєвий доступ буде використовуватись ElastiCache for Redis, що забезпечить миттєвий доступ до інформації, необхідної для роботи системи.

10. Для того, щоб Kubernetes кластер міг безпечно отримати доступ до ключів доступу до необхідних для роботи сервісів, використано Amazon Secrets Manager, в IAM полісі якого буде прописано, що доступ до секретних значень матиме лише необхідний Kubernetes кластер.

11. Для простого поверхневого моніторингу системи використано Amazon CloudWatch, який дозволить своєчасно виявити та проаналізувати проблеми функціонування системи.

12. В разі виникнення проблеми (сервіс Amazon CloudWatch) або в разі потреби сповістити користувача про певну подію (сервіс Kubernetes) направить повідомлення до сервісу Amazon SNS, який доставить його користувачу одним з обраних методів (Email, SMS, HTTPS, Push).

Для швидкого розгортання системи буде використовуватись IaC (Infrastructure as Code) інструмент Terraform, розроблений компанією HashiCorp. Це декларативний безкоштовний програмний продукт, який за допомогою стандартизованого синтаксису дозволяє відправити API запит до провайдера (Amazon Web Services) та за лічені хвилини створити усі інфраструктурні елементи з необхідними налаштуваннями. Це дозволить прописавши усі необхідні компоненти лише раз розгорнути дану інфраструктуру в будь-якому обліковому записі Amazon Web Services.

### **3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ**

Для реалізації компоненту системи, який відповідає за прогнозування, використано декілька поширених варіацій нейронних мереж, а саме багатошарова нейронна мережа, згортова нейронна мережа та мережа довгої короткої пам'яті. Для реалізації цих мереж використовувалась мова прогнозування Python та бібліотеки Pandas, Numpy, Sklearn, Keras та Tensorflow, а для візуалізації використовувалась бібліотеки Matplotlib та Seaborn. Метою даного дослідження є реалізувати декілька поширених архітектур нейронних мереж та обрати найбільш придатну для задачі прогнозування в реальному часі, а саме ту, яка має оптимальну точність та швидко працює. Для прогнозування використовувались по хвилинні дані індексу S&P500.

Результати роботи багатошарової нейронної мережі подано на рисунках 2-4.



Рисунок 2. Динаміка зміни похибки багатос шарової нейронної мережі

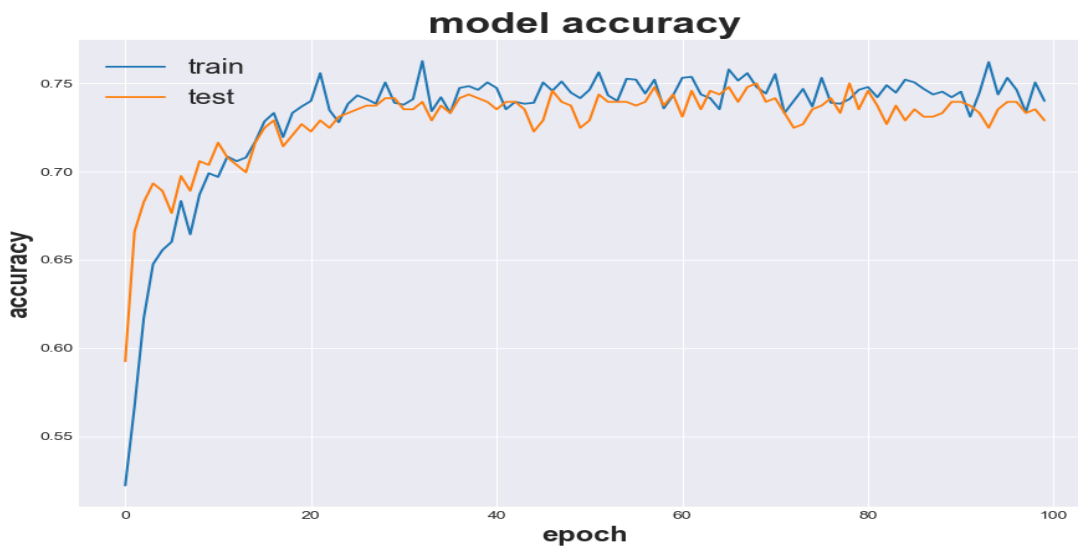


Рисунок 3. Динаміка зміни точності багатос шарової нейронної мережі

	precision	recall	f1-score	support
0	0.73	0.76	0.75	242
1	0.74	0.71	0.73	234
accuracy			0.74	476
macro avg	0.74	0.74	0.74	476
weighted avg	0.74	0.74	0.74	476
True positive =	0.76			
False positive =	0.248			
True negative =	0.714			
False negative =	0.277			

Рисунок 4. Результати роботи багатос шарової нейронної мережі



## 4. ВИСНОВКИ

В поданій роботі проведено огляд засобів аналізу та прогнозування індексу акцій та можливості хмарних обчислювальних сервісів, а також було розроблено модель системи прогнозування індексу акцій в реальному часі на базі хмарних обчислювальних сервісів та реалізовано прототип компоненту прогнозування індексу акцій. В результаті модель системи передбачає автономну роботу системи з можливістю віддаленого керування цією системою за допомогою вхідних API запитів. Базуватиметься система на хмарному сервісі Amazon Web Services.

Також розроблено і протестовано компоненту для прогнозування індексу акцій. Були побудовані багатошарова нейронна мережа, мережа довгої короткої пам'яті та згортована нейронна мережа, які аналізували по хвилинні дані індексу S&P500. Вони передбачали, як чи зросте або впаде індекс S&P500 через 5 хвилин після моменту прогнозування на основі даних за останні 30 хвилин. Звичайна багатошарова нейронна мережа показала найкращий час та трохи кращий результат, тож для даного компонента доцільно обрати її.

В майбутньому можна вдосконалити систему реалізувавши інші компоненти моделі, такі як Engicler, що дасть системі додаткові дані про компанії, акції яких вона намагається передбачити.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про цінні папери і фондовий ринок: Закон України від 23.02.2006 р. – №3480-IV. URL: <https://zakon.rada.gov.ua/laws/show/3480-15#Text>.
2. Про акціонерні товариства : Закон України № 514-VI від 17.09.2008 // Відомості Верховної Ради України. 2008. № 50–51.
3. AWS Services List - Top 10 AWS Services. URL: <https://mindmajix.com/top-aws-services>
4. Amazon API Gateway features. URL: <https://aws.amazon.com/api-gateway/features/>
5. Top programming languages. Most popular and fastest choices for developers. URL: <https://www.zdnet.com/article/top-programming-languages-most-popular-and-fastest-growing-choices-for-developers/>
6. Amazon Lambda features. URL: <https://aws.amazon.com/lambda/features/>
7. Serverless computing with AWS Lambda. URL: <https://www.infoworld.com/article/3210726/serverless-computing-with-aws-lambda.html>.
8. Amazon Leads \$150-Billion Cloud Market. URL: <https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/>
9. Google Cloud vs AWS: Difference Between AWS and GCP. URL: <https://www.guru99.com/google-cloud-vs-aws.html>.
10. Засоби прогнозування індексу акцій на основі апарату штучних нейронних мереж, 2020. URL: <https://ela.kpi.ua/handle/123456789/37035>

# ІНСТРУМЕНТИ АНАЛІЗУ ПРОЕКТНИХ РИЗИКІВ

Гогуля О.О.<sup>1</sup>, Тимошук О.Л.<sup>2</sup>

Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”, Київ, Україна

<sup>1</sup>aleksandragogulia@gmail.com,

<sup>2</sup>o.tymoshchuk@kpi.ua [0000-0003-1863-3095]

**Можливість визначати та управляти ризиками є досить актуальною задачею сьогодення. Багато процесів наразі є досить непередбачуваними, тому мати можливість передбачити та мінімізувати ризики є досить важливим процесом. Метою роботи є дослідження інструментів для аналізу ризиків та їх порівняння. Результатами роботи є застосування методів на конкретному процесі однієї із компаній.**

**Ключові слова:** проектні ризики, ризик-менеджмент, аналіз ризиків.

## 1. ВСТУП

Управління ризиками є досить актуальною задачею сьогодення. В сьогоденнішніх реаліях існує багато нестабільних та неконтрольованих процесів, на які досить важко повпливати. Тому можливість передбачати та контролювати ризики є важливою сферою, яка активно розвивається.

Невизначеність та пов'язані з цим ризики важлива частина управління проектами на сьогодні. В роботі було проведено задачу ідентифікації ризиків для конкретного проекту та аналіз вибору ефективних методів та стратегії реагування на них.

## 2. МЕТОДИ АНАЛІЗУ РИЗИКІВ

На практиці ризик-менеджмент виділяється в окремий специфічний напрям діяльності. Під цим поняття варто розуміти систему управління ризиками проекту, яка передбачає вибір та застосування інструментів та методів, спрямованих на знаходження ризиків, обрахунок імовірності їх настання, визначення фінансових величин втрат (або прибутку у випадку позитивного впливу ризику на проект) та нейтралізацію самих ризиків [1].

Головною ціллю управління ризиками є мінімізація впливу ризику на фінансові втрати. Існують наступні етапи управління ризиком [2]:

- Ідентифікація та аналіз ризику.
- Аналіз альтернативних методів управління ризиком.
- Вибір методів управління ризиком.
- Виконання обраного методу управління ризиком.
- Моніторинг результатів і вдосконалення системи управління ризиком.

Розглянемо кожен із них детальніше.

Під ідентифікацією та аналізом ризиків можна розуміти процес виявлення та опису ризиків, розбір їх специфіки, яка виражається певними притаманними рисами, за допомогою яких їх можна охарактеризувати та віднести до певної групи. [3] Також це включає дослідження розмірів збитку, потенційною зміною ризиків в часі, взаємозв'язками між ними та вивчення факторів, які на них впливають. Даний етап забезпечує інформаційну базу для

проведення подальших етапів ризик-менеджменту. Для реалізації цього етапу процесу використовуються наступні методи:

- метод Дельфі;
- мозковий штурм;
- метод номінальної групи;
- аналіз схожих проектів в тій ж, або суміжній областях.

Наступним етапом є аналіз альтернативних методів управління. Метою цього етапу є дослідження інструментів, які можна застосовувати для усунення реалізації та впливу ризику на процеси проекту та різні його вкладові. Варто зазначити, що методи управління будуть обиратись індивідуально для кожного з типів ризиків.

Етапом вибору методів управління ризиком дозволяє сформувати загальну стратегію по боротьбі з ризиком для всього життєвого циклу проекту. Вибір буде робитись на основі доцільності та ефективності застосування того чи іншого методу управління до кожного конкретного ризику з урахуванням обмежень у фінансових ресурсах та загальним впливом ризику на фінальний результат проекту.

Наступний етап передбачає пряме застосування обраних методів у випадку настання тієї чи іншої ризикової ситуації. Тому на даному етапі важливо буде максимально готовим швидко та ефективно застосовувати обрану стратегію, адже зазвичай присутня обмеженість в часових ресурсах. Також важливо визначитись з відповідальними за кожне рішення та з тими, хто буде контролювати процес реалізації стратегії управління ризиком.

Останнім етапом є моніторинг результатів та вдосконалення системи управління ризиком. Це передбачає ітеративний підхід з кожним наступним застосуванням обраної стратегії управління ризиком. Важливо розуміти, що цей етап може повпливати та внести зміни в усі попередні. Ми отримуємо та аналізуємо інформацію про ризики, наскільки ефективним був обраний метод для їх управління, який збиток був заданий проекту.

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

Було запропоновано алгоритм аналізу ризиків при плануванні процесу розширення компанії та переходу її на новий рівень. Хоч алгоритм було застосовано на даному проекті, він може бути валідним і для багатьох інших.

Розглянемо кроки, завдяки яким даний алгоритм реалізується:

- складення плану проекту;
- ідентифікація можливих ризиків на кожному з етапів;
- за допомогою методу Дельфі обраховуються ймовірності настання ризику;
- побудова матриці ризиків дозволяє визначити вплив ризику на проект в разі його настання;
- створення методів реагування на кожен із ризиків.

Було проведено аналіз ризиків в різних сферах управління компанією. Було побудовано таблицю ризиків, розглянемо один із них для прикладу.

Таблиця 1. Опис ризику, сфери його впливу та відповідальних експертів

Сфера ризику	Ризик	Номер ризику	Відповідальні
Найм спеціалістів	Найм некваліфікованого спеціаліста призведе до затримки роботи над проектом та грошових втрат на конкретному проекті.	1	- Recruiting Team - СТО

Після проведення серії опитувань серед експертів було складено таблицю з оцінкою ймовірності кожного з ризиків.

Таблиця 2. Оцінка ймовірності настання ризику кожним із експертів

Номер ризику	Відповідальний	Компетентність експерта	Оцінка ймовірність
1	Recruiting Team	0.8	0.05
	СТО	1	0.02

Після проведення всіх обчислень за формулами отримуємо наступну таблицю результатів

Таблиця 3. Проста оцінка, середньозважена оцінка настання кожного із ризиків

Номер ризику	Проста оцінка	Середньозважена оцінка
1	0.035	0.033

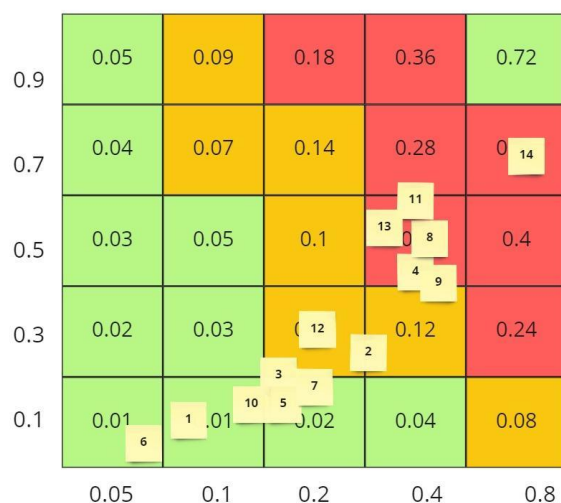


Рисунок 1. Зображення побудованої матриці ймовірностей для проекту компанії

Матриця ризиків зображає ступінь впливу ризиків на проект та показує які з ризиків мають найбільший вплив та потребують нагального реагування та чіткого плану протидії [4].

На основі попередньо проведених досліджень на складених було розроблено рекомендації для кожного з ризиків та визначено критичність їхнього впливу.

#### 4. ВИСНОВКИ

Кожного дня кількість невизначеностей в сфері управління ризиками зростає, тому є велика необхідність створення та застосування ефективних алгоритмів для цього.

У ході написання роботи було розроблено алгоритм, який дозволяє зробити аналіз проектних ризиків, визначити ймовірність їх настання та їх впливу на подальший розвиток

проекту. Результатом роботи стали дослідження, які наводять ймовірність настання того чи іншого ризику, визначені відповідальні експерти, які допоможуть з керуванням ризиків та розроблено рекомендації по впливу на ризики.

### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Фінансовий менеджмент Г.Г. Кірейцев, Житомир: ЖІТІ, 2001. – 440 с.
2. Етапи управління ризиком. URL: [https://stud.com.ua/19859/strahova\\_sprava/etapi\\_upravlinnya\\_rizikom](https://stud.com.ua/19859/strahova_sprava/etapi_upravlinnya_rizikom).
3. Бабаєв В.М. Навчальний посібник для студентів спеціальності 87 «Управління проектами». Харків: ХНАМГ, 2006. 244 с.
4. Guide to the Project Management Body of Knowledge (PMBOK® Guide)

# ЗАСТОСУВАННЯ ТЕОРІЇ ІГОР ПРИ БАНКРУТСТВІ ОРГАНІЗАЦІЇ

Тіщенко А.Є.<sup>1</sup>, Барановська Л.В.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>anna16yarmola@gmail.com,

<sup>2</sup>lesia@baranovsky.org [0000-0003-0024-8180]

**У роботі розглядається використання теорії кооперативних ігор при банкрутстві організації. Обговорюється особливість таких випадків та складності в розподілі ліквідаційної вартості організації між кредиторами, позивачами та заявниками. Представлено рішення ситуації банкрутства певної організації за допомогою побудови динамічної моделі.**

**Ключові слова:** Кооперативні ігри, банкрутство, вектор Шеплі, динамічна модель, характеристична функція.

## 1. ВСТУП

Банкрутство фірми – це довгий і дуже складний процес. Воно настає, коли компанія не може розплатитися за своїми боргами. Цьому сприяють різні причини – підприємство стало збитковим, фінансова криза призвела до того, що власники не в змозі повернути борги банкам та іншим кредиторам.

Успіх діяльності фірми породжується різними зовнішніми та внутрішніми чинниками, і, якщо вона веде цю діяльність неефективно, настає момент, коли її потрібно вивести з ринку. І тому виконується послідовність дій, однією з є оцінка ліквідаційної вартості. Після цього постає питання про розподіл цієї вартості між заявниками, кредиторами та позивачами, що призводить до великої кількості юридичних конфліктів. Ця проблема полягає в тому, що вартості фірми здебільшого недостатньо, щоб виплатити всі борги. Необхідно встановити найприйнятніші правила розподілу.

У літературі представлено ряд таких правил, які використовуються на практиці, але необхідно ці правила якимось порівнювати між собою для визначення переваг та недоліків кожного з них. Найпоширеніше правило - правило пропорційності, у якому кошти між кредиторами розподіляються пропорційно вимогам. Але найчастіше цей поділ супроводжується великою кількістю суперечок між позивачами процедури банкрутства і тоді доводиться вдаватися до перегляду справ.

З іншого боку, мета розподілу – задовольнити потреби кредиторів, позивачів та заявників. А вони, в свою чергу, хочуть максимізувати свою частину виплат. Тож для вирішення цієї проблеми логічно використовувати математичне моделювання засобами теорії кооперативних ігор. Такі ігри моделюють ситуації, в яких учасники гри, об'єднуючись, можуть отримати додатковий прибуток.

Крім того, серйозним кроком у математичному моделюванні банкрутства фірм є розробка динамічних моделей, які враховують тимчасові витрати процесу, що розглядається. Якщо дослідити ліквідацію великих підприємств, можна помітити, що виплати боржникам відбуваються поетапно. Це може бути пов'язано з наявністю дочірніх фірм, дебіторськими заборгованостями, банківськими особливостями та іншими економічними факторами. Тоді необхідно розуміти, яку частину коштів потрібно виплачувати конкретному заявнику в

кожний момент часу. Звідси надходить ідея використання теорії багатокрокових кооперативних ігор.

## 2. ТЕОРІЯ КООПЕРАТИВНИХ ІГОР ТА ЇЇ ЗАСТОСУВАННЯ

Розглянемо моделювання ситуації банкрутства підприємства інструментами теорії кооперативних ігор. Цей метод має значення у різних галузях економічних наук. Він застосовується не тільки як вирішення загальногосподарських завдань, а й у дослідження стратегічних проблем ринків, галузей, підприємств, систем управлінського обліку, і форм стимулювання ефективної діяльності. За допомогою теорії ігор менеджмент підприємства отримує можливість передбачити ходи своїх партнерів та конкурентів.

Перед побудовою моделі необхідно згадати низку базових понять.

Під грою ми розумітимемо процес, у якому беруть участь дві і більше сторін, які ведуть боротьбу реалізації своїх інтересів. Нехай умови гри допускають спільні дії та перерозподіл виграшу. Головне завдання дослідження – це оптимальний розподіл благ між членами об'єднання.

Нехай  $N = \{1, 2, \dots, n\}$  - це множина всіх гравців у рамках цієї моделі. Тоді будь-яку непорожню підмножину  $S \subset N$  ми називатимемо коаліцією.

Під характеристичною функцією  $v$  будемо розуміти функцію, яка для кожної можливої коаліції ставить у відповідність дійсне число. Для будь-яких двох непересічних коаліцій  $T \subset N$  та  $S \subset N$  виконується нерівність:

$$v(T) + v(S) \leq v(T \cup S)$$

Це означає, що коаліція  $T \cup S$  має не менше можливостей, ніж дві коаліції, що не перетинаються,  $T$  і  $S$ , що діють поодиночці.

Тоді кооперативною грою назвемо пару  $(N, v)$  та визначимо її рішення. Найчастіше використовуються принципи оптимальності, такі як  $C$ -ядро,  $NM$ -рішення, вектор Шеплі. Але ми оберемо метод, який підходить для вирішення задачі справедливого розподілу і який гарантує єдиність рішення. Цей принцип вводиться аксіоматично.

Аксіоми Шеплі.

Аксіома 1. Якщо  $S$  - будь-який носій гри  $(N, v)$ , то виконується:

$$\sum_{i \in S} \varphi_i v = v(S).$$

Аксіома 2. Для будь-якої підстановки  $\pi$  і  $\forall i \in N$  вірно:

$$\varphi_{\pi(i)} \pi v = \varphi_i v.$$

Аксіома 3. Якщо  $(N, v)$  і  $(N, u)$  - дві довільні кооперативні ігри, то:

$$\varphi_i u + v = \varphi_i u + \varphi_i v.$$

Нехай  $\varphi$  - це функція, яка ставить у відповідність згідно з аксіомами Шеплі будь-якої гри  $(N, v)$  вектор  $\varphi(v)$ . Тоді цей вектор називатимемо вектором Шеплі гри  $(N, v)$ .

Сформулюємо просте завдання повернення боргів кредиторам. Характеристичну функцію такої гри побудуємо виходячи з вже існуючих правил розподілу. Кожній коаліції поставимо у відповідність поступки кредиторів, які не входять до об'єднання, що розглядається. У разі коли поступка матиме негативне значення, поставимо нульове значення коаліції. Отже, отримуємо наступну характеристичну функцію:

$$v^{d,E} S = \max E - d_i, 0, i \in N \setminus S. \quad (1)$$

### 3. ПРИКЛАД РОЗПОДІЛУ ЛІКВІДАЦІЙНОЇ ВАРТОСТІ ОРГАНІЗАЦІЇ ЗА ДОПОМОГОЮ ПОБУДОВИ ДИНАМІЧНОЇ МОДЕЛІ

Деяка будівельна компанія подала до суду на прийняття її банкрутом. Ліквідаційна вартість її становить 38041000 гривень. Вимоги позивачів, кредиторів та заявників рівні (13 787 000, 18 655 537, 37 530 244). Відомо, що грошова сума надходить на рахунок боржника у два етапи: продаж основного майна та повернення заборгованостей з боку іншої фірми. Таким чином,  $E = (E(t_1), E(t_2)) = (26801000, 11240000)$ . Розрахуємо виплати трьом агентам за допомогою вищезгаданого кооперативного метода.

Крок 1. Розглянемо першу гру  $\Gamma^1 = (N, v^{(d^1, E^1)})$ , де вектор  $d^1 = (13\ 787\ 000, 18\ 655\ 537, 37\ 530\ 244)$ , а  $E^1 = E(t_1) + E(t_2)$ . Будуємо характеристичну функцію (табл. 1).

Таблиця 1. Характеристична функція для кроку 1

Коаліція	Значення
$v(\emptyset)$	0
$v(1)$	0
$v(2)$	0
$v(3)$	5 598 463
$v(1,2)$	510 756
$v(1,3)$	19 385 463
$v(2,3)$	24 254 000
$v(1,2,3)$	38 041 000

Розрахуємо вектор Шеплі

$$Sh_1^1 = \frac{1}{3} \cdot -24254000 + \frac{1}{6} \cdot 510756 - 5598463 + \frac{1}{6} \cdot 19385463 + \frac{1}{3} \cdot 38041000 = 6\ 978\ 626.$$

$$Sh_2^1 = \frac{1}{3} \cdot -19385463 + \frac{1}{6} \cdot 510756 - 5598463 + \frac{1}{6} \cdot 24254000 + \frac{1}{3} \cdot 38041000 = 9412894,5.$$

$$Sh_3^1 = \frac{1}{3} \cdot 510756 - 5598463 + \frac{1}{6} \cdot 19385463 + \frac{1}{6} \cdot 24254000 + \frac{1}{3} \cdot 38041000 = 21\ 649\ 479,5.$$

Вважаємо, що ми зробили виплати кожному кредитору, тобто. отримали вектор виплат  $x^1 = (x_1^1, x_2^1, x_3^1)$ . Чисельне значення вектора ми дізнаємося на наступному кроці. Але при побудові наступної гри врахуємо ці виплати та віднімемо їх із початкових вимог гравців.

Крок 2. Розглянемо гру  $\Gamma_2$ . Зменшуємо розподільну суму  $E_2 = E(t_2)$  та обчислюємо новий вектор вимог

$$d^2 = (13787000 - x_1^1, 18655537 - x_2^1, 37530244 - x_3^1).$$

Будуємо характеристичну функцію (табл. 2) та знаходимо вектор Шеплі:



Таблиця 2. Характеристична функція для кроку 2

Коаліція	Значення
$v(\emptyset)$	0
$v(1)$	0
$v(2)$	0
$v(3)$	0
$v(1,2)$	$-26\,290\,244 + x_3^1$
$v(1,3)$	$-7\,415\,537 + x_2^1$
$v(2,3)$	$-2\,547\,000 + x_1^1$
$v(1,2,3)$	11 240 000

$$Sh_1^2 x^1 = \frac{-2x_1^1 + x_2^1 + x_3^1}{6} - \frac{2043927}{2}.$$

$$Sh_2^2 x^1 = \frac{x_1^1 - 2x_2^1 + x_3^1}{6} - 1412305.$$

$$Sh_3^2 x^1 = \frac{x_1^1 + x_2^1 - 2x_3^1}{6} - \frac{21699317}{2}.$$

Складаємо різницю  $Sh^1 - Sh^2 x^1 = x^1$ , вирішуємо систему з трьох рівнянь із трьома невідомими:

$$x^1 = (7067500, 7067500, 12666000).$$

Отже, отримано виплати для кожного гравця у перший момент надходження.

Користуючись цим же алгоритмом, не складно розрахувати вектор виплат на другому кроці:

$$x^2 = (2239833\frac{1}{3}, 4500083\frac{1}{3}, 4500083\frac{1}{3})$$

Винесемо повне рішення багатокрокової гри та порівняємо його з результатами при однокроковому розподілі:

Таблиця 3. Результати однокрокової та багатокрокової гри

Період виплати	Вектор виплат
$t_1$	(7 067 500, 7 067 500, 12 666 000)
$t_2$	(2 239 833.3, 4 500 083.3, 4 500 083.3)
Сумарна виплата	(9 307 333.3, 11 567 583.3, 17 166 083.3)
Однокрокова гра	(6 978 626, 9 412 894.5, 21 649 479.5)

## ВИСНОВКИ

На підставі проведеного дослідження можна зробити такі висновки.

Проблема розподілу коштів при банкрутстві підприємства є актуальною та активно обговорюється у літературі. У роботі зроблено аналіз чинних правил розподілу фіксованої суми серед позивачів, що висунули свої вимоги. Крім їх перерахування, розглянуто аксіоматичний метод порівняння.

Крім того, були розроблені багатокрокові теоретико-ігрові моделі, що враховують складність надходження коштів на рахунок боржника. Дані моделі ґрунтуються на принципі оптимальності – векторі Шеплі, який гарантує існування та єдиність рішення. На числовому прикладі показано, що такі моделі допомагають справедливо розподілити доступну суму кожному проміжку часу.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. O'Neill A problem of rights arbitration from the Talmud // *Mathematical Social Sciences* , 2011. No 2. P. 345 – 371.
2. Gintis, H. *Game theory Evolving*. — Princenton : Princenton University Press, 2000.
3. Aumann R., Maschler M. E. Game theoretic analysis of a bankruptcy problem from the Talmud // *Journal of Economic Theory*, 1985. Vol. 36, No 1. P. 195 – 213.
4. Herrero C., Villar A. The three musketeers: four classical solutions to bankruptcy problems // *Mathematical Social Sciences*, 2001. Vol. 39, No 3. P. 307 – 328.
5. Петросян Л. А., Зенкевич Н. А., Шевкопляс Е. В. Теорія ігор. М.: БХВ-Петербург, 2014.- 423 с.
6. Thomson W. Axiomatic and Game-theoretic Analysis of Bankruptcy and Taxation Problems // *Mathematical Social Sciences*, 2003. Vol. 45, No 3. P. 249 – 280.

# МЕТОДИ І МЕХАНІЗМИ УПРАВЛІННЯ І ПІДТРИМКИ КОМПЛЕКСОМ УПРАВЛІННЯ СКЛАДОМ

Торліна Н.М.<sup>1</sup>, Мухін В.Є.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> nadiia.torlina@gmail.com, <sup>2</sup> v.mukhin@kpi.ua [0000-0002-1206-9131]

**Організація складу є важливим етапом розвитку будь-якої фірми, адже складські операції є однією з найважливіших складових в ціноутворенні товару. Тому мінімізації витрат на шляху товару від виробника до кінцевого споживача можна досягти навіть якщо не оптимально організований процес купівлі-продажу, виробництва, фінансових розрахунків, тощо. Метою роботи є аналіз процесу впровадження та підтримки комплексу управління складом для мінімізації часу та зусиль витрачених на дані процеси на основі методу з використанням механізмів управління. Результатом дослідження є програмний продукт, що автоматизує процеси під час впровадження та підтримки системи управління складом. У роботі було використано емпіричні та теоретичні методи дослідження.**

**Ключові слова:** система управління складом, WMS, механізм управління, впровадження, підтримка, комплекс управління складом.

## 1. ВСТУП

Система управління складом – це система із застосуванням комп'ютеризованих технологій, за допомогою якої оптимізується та автоматизується вся складська діяльність. Скорочена назва цієї системи – WMS (англ. Warehouse Management System – система управління складом) [1].

Економічний сенс впровадження комплексу управління складом є всюди, де може здійснюватись зберігання, переміщення та облік товарів, відправлень, сировини, тощо, в тому числі архівів. Застосування комплексу управління складом має сенс не тільки у великих логістичних центрах або підприємствах чи складах оптової торгівлі, а також у невеликих складах, які мають невеликі операційні обсяги або невелику кількість товару. Адже незалежно від кількості товару, замовлень, розмірів приміщення, автоматизація значно зменшує витрати часу, і внаслідок цього коштів [2].

Однак успіх впровадження системи WMS базується на деяких організаційних та технічних факторах. Наприклад, пошук часу на впровадження, вибір досвідченого керівника проекту, пошук та навчання співробітників, пошук консультантів та команди впровадження, тестування системи.

Крім того, перед початком впровадження комплексу управління складом потрібно переконатись, що процеси загалом підібрані правильно. Якщо ж компанія вже використовує якусь систему автоматизації, наприклад ERP, і бажає інтегрувати її в нову систему, необхідно переконатись, що дані системи сумісні і можуть використовуватись разом без перебоїв. Також важливо строго дотримуватись методів управління проектом. Незважаючи на складність, системи WMS пропонують підприємствам значні переваги. Буде не тільки

скорочено час циклу розміщення та видалення, але й покращиться точність інвентаризації. Це на додаток до збільшення обсягу зберігання, збільшення організованості зберігання матеріалів і більшої гнучкості складських операцій.

## **2. МЕТОДОЛОГІЯ ВПРОВАДЖЕННЯ І ПІДТРИМКИ КОМПЛЕКСУ УПРАВЛІННЯ СКЛАДОМ**

Складність впровадження і підтримки комплексу управління складом залежить від проекту, бізнесу та доступних людських ресурсів. Тому для успішного впровадження системи управління складом потрібно гарантувати, що ключовий персонал, необхідний для впровадження зможе виконувати поставлені задачі швидко та якісно.

Загалом проект з впровадження комплексу управління складом можна поділити на такі ключові етапи [3]:

- Ініціація;
- Планування та дизайн;
- Реалізація;
- Валідація;
- Введення в експлуатацію.

На етапі ініціації проекту відбувається схвалення проекту, обирається команда з впровадження. Зазвичай команда з впровадження складається з керівника проекту, спеціаліста WMS, дизайнера рішень WMS, спеціаліста з інтеграції, спеціаліста з автоматизації. На цьому етапі проводиться огляд процесів та операцій, які мають бути сконфігуровані в системі, попередньо оцінюється складність проекту і ризики. Складність та ризики оцінюються також наприкінці етапу ініціації. Як результат етапу ініціації проект має бути повністю визначений.

Етап планування і дизайну передбачає створення концепції та вимог щодо майбутніх процесів і операцій. На цьому етапі, а саме на початку та в кінці проводиться огляд та оновлення складності і ризиків проекту. Крім того, на даному етапі виявляються та узгоджуються функції, для яких необхідна розробка.

На етапі реалізації середовище для інтеграції системи має бути готове до налаштування, а також команда впровадження повинна мати список конфігурації, яку треба налаштувати в системі. Прогрес налаштування системи відстежується протягом всього етапу. Коли конфігурація проекту повністю виконана, вона має бути перенесена вручну з середовища для інтеграції в середовище для тестування, на якому відбувається етап валідації проекту. Також в кінці етапу реалізації проводиться демонстрація сконфігурованих процесів для команди з впровадження та операторів системи управління складом для валідації налаштованих процесів та перевірки чи залишилися певні недоліки в процесах.

Етап валідації передбачає тестування всіх процесів, що були налаштовані в системі. Також на цьому етапі спеціаліст WMS переносить вручну налаштування з середовища для тестування до робочого середовища.

На етапі введення в експлуатацію система управління складом відбувається процес останньої перевірки системи, та безпосередньо введення в експлуатацію [3].

Кожен з останніх трьох етапів: реалізації, валідації і введення в експлуатацію, може займати від двох тижнів до декількох місяців, що збільшує вартість проекту.

Після успішного запуску комплексу багато підприємств виявляють, що ресурсів, необхідних для підтримки роботи системи WMS, більше, ніж до впровадження. Насамперед це пов'язано з інтенсивним об'ємом даних програмного забезпечення та тим фактом, що ситуація на складі постійно змінюється: додаються нові елементи, розробляються нові процеси, тощо [3].

Після впровадження проект підтримується операторами складу та спеціалістами WMS. У разі виникнення проблем, спеціалісти WMS мають бути навчені вирішувати їх і якщо першопричину не вдається знайти, звертатись до служби підтримки. Зазвичай технічна підтримка складається з декількох рівнів підтримки для того щоб швидко вирішувати дрібні або прості проблеми, і визначати відповідних спеціалістів та дії для складніших проблем [3]:

- 0 лінія підтримки: самостійна підтримка локальних спеціалістів WMS, використовуючи інформацію з керівництва користувача, поширених запитань, форумів, залучення соціальних контактів, тощо;

- 1 лінія підтримки: базова служба підтримки найбільш частих проблем користувачів та виконання запитів служби підтримки, які потребують участі ІТ;

- 2 лінія підтримки: поглиблена технічна підтримка більш досвідченими спеціалістами системи управління складом проблем користувачів, які не можуть бути вирішені попередньою лінією підтримки;

- 3 лінія підтримки: експертна підтримка продуктів і послуг для вирішення проблем або створення нових функцій залежно від виявлених першопричин;

- 4 лінія підтримки: зовнішня підтримка проблем за контрактом для товарів, що надаються організацією, але не обслуговуються безпосередньо, включаючи підтримку принтерів, підтримку програмного забезпечення постачальників, технічне обслуговування та інші послуги стороннього виконавця.

Компанії можуть змінювати цей шаблон та поєднувати декілька рівнів підтримки для економії часу та людських ресурсів на вирішення проблем. Також може використовуватись часова шкала і пріоритетність для передачі проблеми на наступний рівень. Наприклад, якщо рішення проблеми з високим пріоритетом займає більше 2 годин у рівня 0, вона має бути передана на рівень 1 [4].

В даному дослідженні проаналізовано витрати на впровадження та підтримку комплексу управління складом використовуючи наведену методологію. А також використано підхід на основі методу з використанням технічних механізмів управління та представити, як підприємство може скоротити витрати на впровадження та підтримку комплексу управління складом. У дослідженні використовуються дані по загальним витратам на впровадження системи, витратам на період окремих етапів впровадження та витрати на підтримку WMS.

### **3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ**

Для дослідження ефективності програмного продукту було обрано дані, що базуються на даних логістичної компанії, яка займається впровадженням і підтримкою комплексом управління складом. Було обрано 5 проектів, які впроваджувались у період 2020-2022 року. А також представлено дані витрат на підтримку системи управління складом за 2020-2022 роки.

Аналіз існуючих підходів до оцінки логістичних витрат показує, що переважна більшість дослідників [5], [6], [7] при визначенні даного виду витрат зосереджує увагу, здебільшого, на ключових логістичних активностях або ж на конкретному переліку бізнес процесів, що здійснюються у межах підприємства.

У таблиці 1 наведено витрати на впровадження комплексу управління складом на п'ятьох проектах різних за сферою та обсягом без використання технічних механізмів управління.

Таблиця 1. Витрати на впровадження комплексу управління складом без використання технічних механізмів управління

№ проекту	Початок проекту	Кінець проекту	Загальна кількість робочих днів	Вартість послуг за робочий день (дол.)	Вартість комплексу управління складом (дол./місяць)	Загальна вартість проекту (дол.)
1	29.09.2020	01.04.2021	128	1000	2000	136533
2	16.11.2020	26.07.2021	172	1000	2000	183352
3	01.02.2021	15.07.2021	113	1000	2000	120458
4	23.02.2021	14.01.2022	223	1000	2000	237718
5	03.05.2021	05.01.2022	169	1000	2000	180154

Розглянемо кожен проект на етапі реалізації більш детально (Табл. 2).

Таблиця 2. Витрати на етапі реалізації проекту без використання технічних механізмів управління

№ проекту	Початок етапу	Кінець етапу	Загальна кількість робочих днів	Кількість годин на порівняння середовищ	Вартість послуг за робочий день (дол.)	Вартість комплексу управління складом (дол./місяць)	Грошові витрати на порівняння середовищ (дол.)
1	23.10.2020	08.03.2021	93	58	1000	2000	7728,5
2	11.03.2021	22.06.2021	70	50	1000	2000	6662,5
3	12.06.2021	28.06.2021	9	8	1000	2000	1066
4	08.04.2021	30.07.2021	77	56	1000	2000	7462
5	09.06.2021	06.12.2021	125	760	1000	2000	101270

В розробленому програмному продукті порівняння середовищ займає від 1 до 10 хвилин, в залежності від кількості даних з налаштувань системи управління складом. Витрати на етапі реалізації проекту з використанням технічного механізму управління представлені у таблиці 3.

Таблиця 3. Витрати на етапі реалізації проекту з використанням технічних механізмів управління

№ проекту	Загальна кількість днів (робочих)	Кількість годин на порівняння середовищ	Вартість послуг (дол./робочий день)	Вартість комплексу управління складом (дол./місяць)	Грошові витрати на порівняння середовищ (дол.)
1	86	0,16	1000	2000	21,32
2	64	0,16	1000	2000	21,32
3	8	0,16	1000	2000	21,32
4	70	0,16	1000	2000	21,32
5	95	0,16	1000	2000	21,32

В якості кількості годин на порівняння середовищ взято максимальне значення у 10 хвилин. В результаті в середньому грошові витрати за час відведений на порівняння середовищ можна скоротити у більш ніж 100 разів.

У таблиці 4 наведено порівняння витрат на впровадження комплексу управління складом без та з використанням технічних механізмів управління.

Таблиця 4. Порівняння витрат на впровадження комплексу управління складом

	Без використання технічних механізмів управління	З використанням технічних механізмів управління
Середні витрати робочих днів на впровадження комплексу управління складом	161	151
Середні грошові витрати на впровадження комплексу управління складом (дол.)	180421	146805

В результаті використання технічних механізмів управління на етапі реалізації витрати часу зменшились на 6,2%, а грошові витрати на 18,6%.

У таблиці 5 розглядаються витрати на підтримку комплексу управління складом за 2020-2022 роки.

Таблиця 5. Витрати на підтримку комплексу управління складом без використання технічних механізмів управління

Рік	Тип проблеми	Кількість запитів	Мінімальний час на вирішення 1 проблеми (год)	Середній час на вирішення 1 проблеми (год)	Максимальний час на вирішення 1 проблеми (год)	Вартість послуг зовнішньої підтримки (дол/год)
2020	Налаштування	35	0,8	312	2736	1000
2021	Налаштування	44	0,25	192,75	1210	1000
2022	Налаштування	38	1.23	446	3718,5	1000

Розглянуто лише запити з типом проблеми «Налаштування», що передбачає випадки, коли конфігурацію системи було змінено і процес припинив працювати коректно. В середньому за три роки середній час на вирішення 1 проблеми становить 316,9 годин. Також варто зазначити, що час на вирішення запиту залежить від кількості ескалацій проблеми на вищі рівні підтримки.

В таких випадках також можна використати розроблений програмний продукт для порівняння середовищ і виявлення конфігурації, яка була змінена і скорочення часу на пошук проблеми. Зовнішню підтримку в такому випадку можна не залучати до вирішення проблеми. Отже грошові витрати на послуги зовнішньої підтримки для типу проблеми «Налаштування» будуть дорівнювати 0. Що призведе до загального скорочення грошових витрат на зовнішню підтримку комплексу управління складом. Порівняння середовищ, як і раніше, займає від 1 до 10 хвилин. Зміна відповідної конфігурації та перевірка коректності роботи процесу займе в середньому 1 робочий день або 8 годин. Отже, витрати часу на підтримку комплексу управління складом в середньому зменшаться у 40 разів.

## 4. ВИСНОВКИ

На сьогоднішній день автоматизація складських процесів грає важливу роль в розвитку підприємства, адже вона скорочує час на виконання складських операцій, збільшує точність та забезпечує безпеку товару, що зберігається на складі. Впровадження системи управління складом має економічний сенс всюди, де здійснюється зберігання, облік і переміщення будь-яких одиниць зберігання. Організація технічної підтримки також грає важливу роль при впровадженні комплексу управління складськими процесами.

Впровадження і підтримка системи управління складом є складним завданням, що залежить від багатьох факторів, зокрема проекту, бізнесу та доступних людських ресурсів.

В даній роботі було досліджено та проаналізовано витрати на впровадження та підтримку комплексу управління складом, використано підхід на основі методу з використанням технічних механізмів управління та представлено, як підприємство може скоротити витрати на впровадження та підтримку системи управління складом за допомогою автоматизації процесів під час впровадження та підтримки системи управління складом.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Warehouse management system. URL: [https://en.wikipedia.org/wiki/Warehouse\\_management\\_system](https://en.wikipedia.org/wiki/Warehouse_management_system) (дата звернення: 16.11.2022)
2. Дудар Т. Г., Основи логістики: навч. посіб. Тернопіль: Економічна думка, 2006. 163 с.
3. 7 Steps Towards a Successful WMS Implementation. URL: <https://www.linkedin.com/pulse/7-steps-towards-successful-wms-implementation-marcil-msc-> (дата звернення: 16.11.2022)
4. IT Support Levels Clearly Explained: L1, L2, L3 & More. URL: <https://www.bmc.com/blogs/support-levels-level-1-level-2-level-3/> (дата звернення: 16.11.2022)
5. Шевців Л. Ю., Петецький І., Логістичні витрати підприємства: формування та оцінювання, монографія. Львів: НУ «Львівська політехніка», 2011. 244 с.
6. Сумець О. М., Логістичні витрати підприємства: теоретичний аспект, монографія. Харків: КП «Міськдрук», 2013. 224 с.
7. Лифар В. В., Комерційна логістика та методика розрахунку логістичних витрат. Вісник Національного Університету «Львівська політехніка». 2011. Вип. 416. С. 293 – 297.



# СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ КОРОТКОСТРОКОВОГО ПРОГНОЗУВАННЯ НЕЛІНІЙНИХ НЕСТАЦІОНАРНИХ ПРОЦЕСІВ В ЕКОНОМІЦІ ТА ФІНАНСАХ

Харченко Р.А.<sup>1</sup>, Бідюк П.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>romankharchenko99@gmail.com,

<sup>2</sup>pbidyuke@gmail.com [0000-0002-7421-3565]

**Метою дослідження є розробка гнучкої системи підтримки прийняття рішень (СППР) для короткострокового прогнозування фінансово-економічних процесів. Побудова моделей та оцінювання прогнозів в рамках СППР передбачає використання гнучкого підходу на базі декомпозиції часового ряду та використанні ряду моделей різних класів. Система надає інструменти для попереднього аналізу даних, автоматизованого підбору структурних параметрів моделей, візуалізації та дозволяє будувати високоточні короткострокові прогнози не вимагаючи від користувачів спеціальних знань в галузі прогнозування.**

**Ключові слова:** СППР, декомпозиція часового ряду, ARIMA, SVR, LSTM.

## 1. ВСТУП

Прогнозування фінансово-економічних процесів є однією з найбільш актуальних задач аналізу даних в цілому. Оцінки прогнозів динаміки розвитку процесів у економіці та фінансах є вхідними даними для вирішення задач планування, прийняття рішень, інвестування, оцінювання ризиків тощо. Універсального рішення, яке буде оптимальним для будь-яких вхідних даних та в кожному окремому контексті вирішення задачі прогнозування цих процесів не існує. Це зумовлено тим, що фінансово-економічним процесам властиві такі характеристики як нестационарність, нелінійність, наявність сезонних та циклічних патернів, залежність від великої кількості зовнішніх, як правило невимірюваних, впливів та збурень. Задачі прогнозування фінансово-економічних процесів та часових рядів у цілому присвячено досить багато літературних джерел, наприклад [1] містить докладні відомості про природу фінансово-економічних процесів у цілому та особливості їх моделювання, у роботах [2, 3] наведено докладні відомості про випадкові процеси, підходи до побудови моделей, прогнозування, критерії якості моделей, надано практичні рекомендації. Згадані вище роботи хоча і надають практичні рекомендації щодо використання методу або групи методів для вирішення задачі прогнозування, не приділяють достатньої уваги безпосередній реалізації методів, підбору гіперпараметрів моделей тощо. Спроектowana і реалізована в даній роботі СППР дозволяє використовувати множину моделей різних класів, інструменти для попереднього аналізу даних, автоматизованого підбору структурних параметрів моделей і при цьому не потребує поглиблених знань та навичок в галузі моделювання і прогнозування.

## 2. ПРОГНОЗУВАННЯ ФІНАНСОВО-ЕКОНОМІЧНИХ ЧАСОВИХ РЯДІВ НА БАЗІ ДЕКОМПОЗИЦІЇ

Ідея декомпозиції часового ряду полягає у його розкладенні на набір компонент, які, як правило, є зручнішими та простішими з точки зору моделювання та прогнозування. Виділяються наступні 3 компоненти: компонента тренду  $T$ , сезонна компонента  $S$ , залишкова компонента  $R$ .

Компонента тренду описує загальну тенденцію зміни значень ряду, сезонна компонента описує регулярні патерни, які повторюються через рівні проміжки часу, до залишкової компоненти відносять всі впливи, які не враховані іншими компонентами, наприклад невимірювані впливи та випадкові збурення тощо. Іноді також виділяється циклічна компонента, яка описує досить тривалі патерни, які не обов'язково повторюються через однакові проміжки часу, однак в даній роботі вважаємо, що компонента тренду враховує циклічні зміни.

Використання декомпозиції часового ряду також забезпечує гнучкість процесу моделювання та прогнозування. Наприклад, можна використовувати різні моделі в тому числі з різних класів моделей для прогнозування кожної з компонент. Виокремлення залишкової компоненти дозволить зменшити вплив невимірюваних збурень на якість моделей. На практиці найбільш поширеними є адитивна та мультиплікативна декомпозиції, адитивна декомпозиція полягає у представленні вихідного часового ряду у вигляді суми компонент  $T$ ,  $S$ ,  $R$ , в той час як мультиплікативна у вигляді їх добутку. Вибір того чи іншого підходу залежить від характеру досліджуваних даних, наприклад, існує емпіричне правило, яке стверджує, що у випадку, якщо амплітуда сезонних коливань практично не змінюється в часі, то варто обирати адитивну модель, якщо амплітуда залежить від тренду, то кращим вибором буде мультиплікативна модель [4]. Однак не всі методи декомпозиції передбачають мультиплікативний варіант, в такому випадку можна використовувати операцію логарифмування для представлення мультиплікативної декомпозиції у вигляді адитивної використовуючи таке співвідношення:

$$\log(y) = \log(S * T * R) = \log(S) + \log(T) + \log(R). \quad (1)$$

Надалі за замовчуванням матимемо на увазі адитивну декомпозицію, враховуючи, що за потреби можна отримати мультиплікативний варіант за виразом (1).

Формування остаточної оцінки прогнозу на базі декомпозиції також є досить гнучким і дозволяє використовувати дещо різні підходи. На практиці одним з найбільш поширених є підхід, що передбачає прогнозування компоненти сезонності та вихідного ряду з вилученою сезонною компонентою. Даний підхід може бути оптимальним у випадку якщо залишкова компонента не є цілком випадковою та може містити корисну інформацію про властивості динаміки розвитку процесу. Інший підхід полягає у відкиданні залишкової компоненти та прогнозуванні лише компоненти тренду та сезонної компоненти, в такому випадку фактично використовується фільтрація часового ряду з вилученою сезонною компонентою. Кожен з згаданих вище підходів реалізовано в рамках розробленої СППР.

Незалежно від обраного з наведених вище підходів, остаточна оцінка прогнозу для вихідного ряду формується як сума прогнозованих значень компонент. Прогнозування сезонної компоненти часто зводиться до використання найвісного прогнозу, тобто повторення відповідних значень останнього сезонного циклу. Використання найвісного прогнозу для сезонної компоненти базується на припущенні про незмінність або дуже повільні зміни сезонних впливів, що часто, хоча і не завжди, виконується на практиці, крім того, деякі методи декомпозиції, не передбачають врахування зміни сезонних впливів за побудовою.

В реалізованій системі підтримки прийняття рішень користувач має змогу обрати метод для декомпозиції часового ряду, зокрема імплементовано метод класичної декомпозиції [4] та метод STL [5].

Метод класичної декомпозиції є досить простим та інтуїтивно зрозумілим, невимогливим до обчислювальних ресурсів. Ідея методу базується на використанні ковзних середніх для оцінювання тренду та оцінюванні сезонної компоненти шляхом усереднення значень факторів сезонності, в якості факторів сезонності можуть розглядатися дні тижня, місяці, пори року тощо в залежності від досліджуваної довжини сезонного циклу та дискретизації вимірів ряду. Основним недоліком цього методу є неможливість оцінювання декілька перших та останніх вимірів, що пов'язано з використанням центрованих ковзних середніх, для отримання оцінок останніх вимірів, які є особливо важливими для прогнозування, можна використовувати не центровані ковзні середні, а усереднювати значення лише попередніх вимірів, також можна використовувати методи екстраполяції. Іншими недоліками є надмірне згладжування тренду, неможливість врахування змін сезонних впливів, низька робастність.

Метод STL також базується на процедурі згладжування, проте алгоритм роботи методу значно складніший, ніж у випадку класичної декомпозиції. Цей метод є універсальним та гнучким, при цьому не є надто вимогливим до обчислювальних ресурсів. Основними перевагами методу є можливість врахування сезонних циклів при будь-якому періоді дискретизації вимірів, врахування змін сезонних впливів з часом, невисока у порівнянні з методом класичної декомпозиції чутливість до екстремальних значень та нетипових вимірів даних. Серед недоліків методу можна виділити відсутність автоматичного врахування "календарних ефектів" та відсутність мультиплікативного варіанту декомпозиції. Однак, використовуючи перетворення, наприклад за виразом (1), або в більш загальному випадку перетворення Бокса-Кокса можна отримати результат, подібний до мультиплікативної декомпозиції або навіть деякий проміжний результат між варіантами адитивної та мультиплікативної декомпозицій [7].

В рамках реалізованої СППР користувач має змогу обирати декілька моделей різних класів для побудови оцінок прогнозів компоненти тренду або вихідного ряду з вилученою сезонністю та сезонної компоненти. Зокрема для прогнозування тренду можуть використовуватись класична регресійна модель ARIMA [6] та її різновиди, а саме моделі AR та MA, модель SVR [7], що є модифікацією методу опорних векторів для вирішення задачі регресії та нейронну мережу LSTM [8]. Для прогнозування сезонної компоненти реалізовано модель наївного прогнозу, SVR та LSTM. Відмінності в наборі моделей для прогнозування сезонної компоненти зумовлені тим, що наївний прогноз є оптимальним варіантом за умови припущення про незмінність сезонних впливів, а також може ефективно використовуватись у разі повільної зміни сезонних впливів. Модель ARIMA не була обрана для прогнозування сезонної компоненти, оскільки за умови зміни сезонних впливів з часом, сезонна компонента буде гетероскедастичною, оскільки змінюватиметься амплітуда коливань циклів, що негативно впливатиме на ефективність використання моделі.

### **3. ОПИС РЕАЛІЗОВАНОЇ СППР ТА РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНИХ ЕКСПЕРИМЕНТІВ**

СППР призначена для математичного моделювання та прогнозування нелінійних нестационарних процесів у економіці та фінансах, динаміка яких представлена у вигляді одновимірних часових рядів. Для реалізації СППР було обрано мову програмування Python, взаємодія з користувачем відбувається за допомогою віконного інтерфейсу, розробленого за допомогою фреймворку PyQt. Функціональна схема реалізованої системи наведена на Рис. 1.

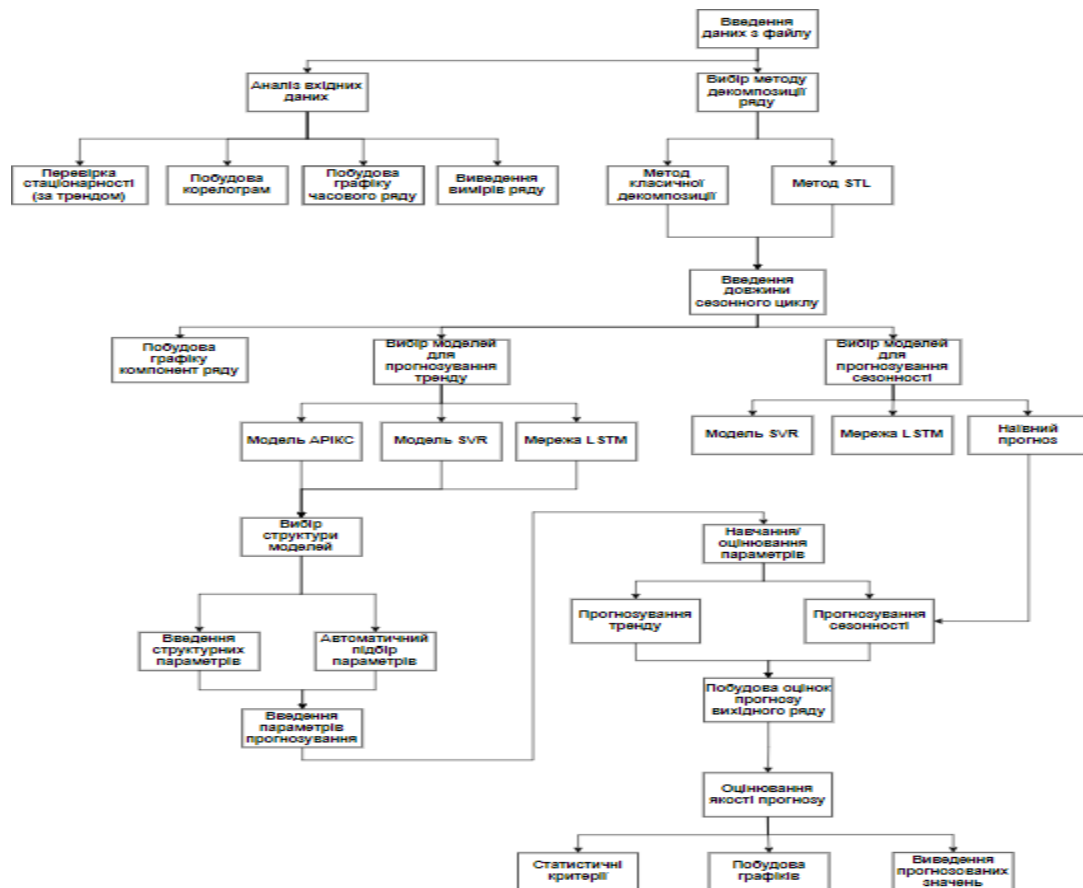


Рисунок 1. Функціональна схема СППР

Окрему увагу слід приділити структурним параметрам моделі, якість підбору яких безпосередньо впливає на якість побудови моделей та оцінок прогнозів. В рамках системи передбачено два варіанти визначення гіперпараметрів — введення користувачем вручну та автоматизований підбір. Для автоматизованого підбору параметрів моделі ARIMA система встановлює порядок тренду шляхом застосування оператора взяття різниць до тих пір, поки в результаті не утвориться стаціонарний за трендом ряд, для підбору порядку авторегресійної частини та ковзного середнього система будує ряд моделей на базі ЧАКФ, якщо при деякому значенні лагу абсолютне значення більше 0.15, то досліджуються моделі авторегресії, ковзного середнього та авторегресії і ковзного середнього, порядок параметрів яких відповідає значенню цього лагу. Для вибору кращої моделі використовується значення інформаційного критерію AIC.

Підбір параметрів мережі LSTM та моделі SVR виконується за допомогою процедури баєсівської оптимізації. В якості функції, яку необхідно мінімізувати, використовується значення метрики MAPE, для моделювання цільової функції в рамках процедури використовується гаусівський процес. В поточній реалізації системи для виконання процедури баєсівської оптимізації використовується програмний модуль для бібліотеки Python sickit-optimize. Для мережі LSTM реалізовано можливість підбору кількості нейронів та довжини вікна вхідних даних, для моделі SVR — тип ядерної функції та довжина вікна вхідних даних.

Наведемо результати роботи СППР на прикладі часового ряду, що описує кількість безробітних у відсотках від працездатних осіб в США, період дискретизації вимірів — місяць. Ряд містить дані на проміжку з 1 січня 2008 року по 1 січня 2020, усього 145 вимірів,

горизонт прогнозування – 5 місяців. Графік ряду та компонент, отриманих в результаті декомпозиції за методом STL з довжиною сезонного циклу 12 наведено на Рис. 2.

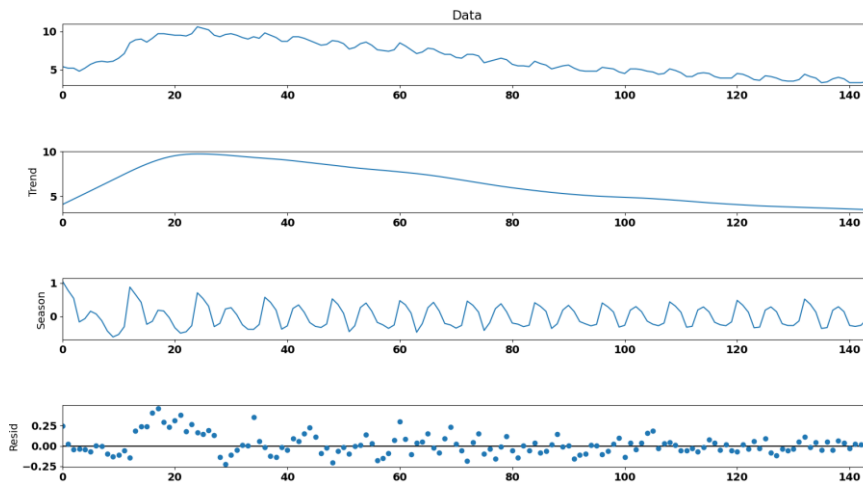


Рисунок 2. Графік досліджуваного ряду та його компонент

Для прикладу виберемо моделі та інші параметри системи, як зображено на Рис. 3.

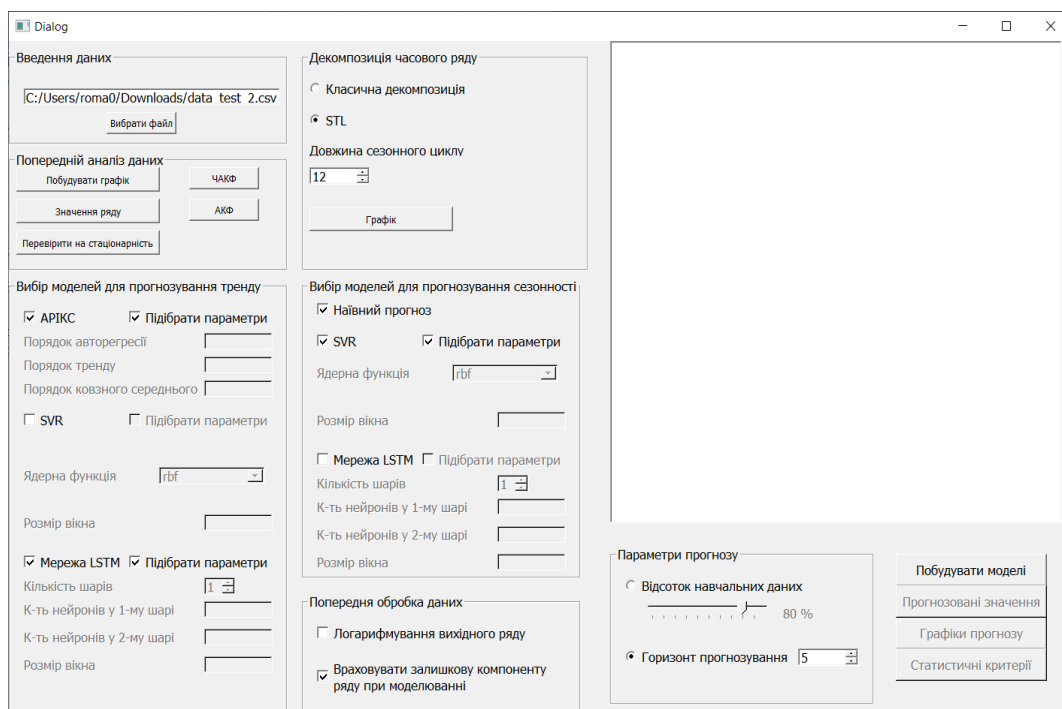


Рисунок 3. Головне вікно програми

На Рис. 4 наведено графіки значень оцінок прогнозів та тестової множини.

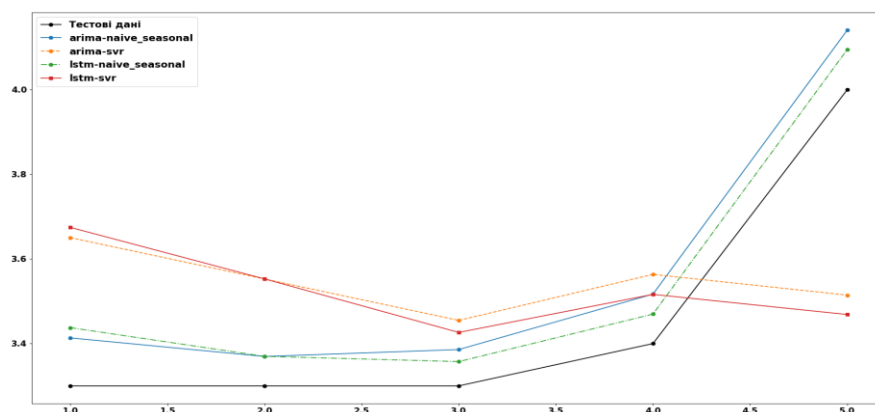


Рисунок 4. Графіки прогнозованих значень тестової множини

Статистичні критерії якості прогнозів наведено у Табл. 1.

Таблиця 1. Статистичні критерії якості прогнозів (перша модель у назві використовується для прогнозування тренду, друга – сезонності)

Метрика \ Модель	MAPE	RSME	ME	MPE	MSE	SSE
ARIMA-naive	3.020509	0.108212	-0.105291	-3.020509	0.011710	0.058549
ARIMA-SVR	7.982132	0.307560	-0.087178	-3.127477	0.094593	0.472965
LSTM-naive	2.488964	0.090508	-0.085868	-2.488964	0.008192	0.040959
LSTM-SVR	7.908984	0.321255	-0.067754	-2.595933	0.103205	0.516025

#### 4. ВИСНОВКИ

В рамках проведеного дослідження було реалізовано систему підтримки прийняття рішень для короткострокового прогнозування нелінійних нестационарних процесів в економіці та фінансах. Підхід на базі декомпозиції часового ряду, імplementований в даній системі, дозволяє отримувати високоточні оцінки прогнозів, про що свідчать статистичні критерії та графік прогнозованих значень, отримані в результаті обчислювального експерименту. Висока якість прогнозів зокрема зумовлена гнучким підходом до побудови моделей. Автоматизовані процедури підбору параметрів дозволяють покращити якість моделей та прогнозів та значно спрощують процес моделювання, надаючи змогу користувачеві використовувати систему без поглиблених знань в області моделювання та прогнозування, однак система дозволяє за бажання користувача самостійно задавати структурні параметри. Також в рамках системи наявні модулі для попереднього аналізу даних, побудови графіків.

Отже, система відповідає поставленим до неї вимогам, для подальшого вдосконалення системи доцільно додати можливість працювати з багатовимірними часовими рядами, розширити набір доступних моделей та вдосконалити процедури підбору параметрів, наприклад розширити множину гіперпараметрів, які можуть бути оптимізовані.

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cıpra T. Time Series in Economics and Finance. Springer, 2020. 353 с.
2. Бідюк П. І., Романенко В. Д., Тимошук О. Л. Аналіз часових рядів : навч. посіб. Київ, Київ обл. : Політехніка, 2010. 317 с.
3. Montgomery D. C., Jennings C. L., Kulahci M. Introduction to Time Series Analysis and Forecasting. Wiley & Sons, Incorporated, John, 2011. 472 с.

4. Kotu V., Deshpande B. Data Science: Concepts and Practice. Morgan Kaufmann, 2018. 568 с.
5. STL: A Seasonal-Trend Decomposition Procedure Based on Loess / R. B. Cleveland та ін. Journal of Official Statistics. 1990. Т. 6, № 1. С. 3–31. URL: <https://www.scb.se/contentassets/ca21efb41fee47d293bbee5bf7be7fb3/the-usa39s-bicentennial-census-new-directions-for-methodology-in-1990.pdf> (дата звернення: 13.11.2022).
6. Hyndman R. J., Athanasopoulos G. Forecasting: Principles and Practice. 2-ге вид. Melbourne : OTexts, 2018. 382 с. URL: <https://otexts.com/fpp2/index.html> (дата звернення: 13.11.2022). Time Series Forecasting Using LSTM Networks: A Symbolic Approach
7. Awad M., Khanna R. Support Vector Regression. Efficient Learning Machines. Berkeley, CA, 2015. С. 67–80. URL: [https://doi.org/10.1007/978-1-4302-5990-9\\_4](https://doi.org/10.1007/978-1-4302-5990-9_4) (дата звернення: 13.11.2022).
8. Elsworth S., Güttel S. Time Series Forecasting Using LSTM Networks: A Symbolic Approach. arxiv.org. URL: <https://arxiv.org/abs/2003.05672> (дата звернення: 13.11.2022).

# **АНАЛІЗ ВПЛИВУ ВІЙСЬКОВОГО КОНФЛІКТУ НА СОЦІАЛЬНО-ЕКОНОМІЧНІ ПРОЦЕСИ В УКРАЇНІ**

Худіков П.В.<sup>1</sup>, Бідюк П.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>pavelkhudikov77@gmail.com

**На даний момент війна в Україні впливає на економічні та соціальні процеси, тому важливо їх досліджувати і спробувати спрогнозувати розуміння подальших дій в цій ситуації. Об'єктом дослідження в роботі є соціально-економічні процеси в Україні, зокрема такі: ВВП, Інфляція, кількість біженців. Метою роботи є розробка і застосування математичних моделей для оцінювання впливу війни на соціально-економічні процеси в Україні та інших розвинутих країнах. Результатом дослідження є побудована система прогнозування, що дає можливість прогнозувати вплив військового конфлікту на соціально-економічні процеси в Україні. У роботі використано теоретичні та емпіричні методи дослідження.**

**Ключові слова: військовий конфлікт, соціально-економічні процеси, задача прогнозування, математичні моделі.**

## **1. ВСТУП**

Війна в Україні стала жорстоким випробуванням на виснаження військових, яка призвела до значної військової підтримки України з боку США та Європи. Однак це лише частина історії. Війна також перетворилася на великий політичний та економічний конфлікт між Заходом і Росією, який зрештою може мати набагато більший вплив на глобальну стабільність і політичну стратегію США та Європи, ніж самі бойові дії. Кінцевим результатом є те, що громадянська сторона війни тепер виходить далеко за межі територій, де відбувалися військові конфлікти та удари по українських містах під час перших етапів війни. Зараз війна змінила міжнародні витрати на енергоносії на глобальній основі, призвела до створення глобального дефіциту продовольства, завдала серйозної шкоди всій російській економіці та допомогла спровокувати значне зростання рівня інфляції на глобальному рівні.

## **2. МЕТОДИ ПРОГНОЗУВАННЯ СОЦІАЛЬНО-ЕКОНОМІЧНИХ ПРОЦЕСІВ**

Соціально-економічні процеси – це зміни в суспільстві та економіці, які відображаються на рівні життя учасників цих процесів, стабільності політичної та економічної ситуації в країні, умовах безпеки та захищеності її населення. Ці процеси обумовлюють соціальні зміни, у яких відображаються соціально-економічні властивості політичних процесів [1].

Для прогнозування цих процесів доречно використовувати регресійні моделі, які будуються на основі статистичних даних.

Лінійна парна регресія намагається змоделювати взаємозв'язок між двома змінними шляхом підгонки лінійного рівняння до спостережуваних даних. Одна змінна вважається пояснювальною, а інша - залежною.



Діаграма розсіювання може бути корисним інструментом у визначенні міцності зв'язку між двома змінними. Якщо, як видається, немає зв'язку між запропонованими пояснювальними та залежними змінними (тобто графік розсіювання не вказує на якісь тенденції до збільшення чи зменшення процесів), то пристосування моделі лінійної регресії до даних, ймовірно, не забезпечить корисної моделі. Цінною числовою мірою асоціації між двома змінними є коефіцієнт кореляції, який приймає значення від -1 до 1, що вказує на силу асоціації спостережуваних даних для двох вибраних змінних.

Логістична регресія – це нелінійна множинна регресія, яка аналізує функціональну залежність між двома або частіше декількома незалежними змінними (регресорами) і залежною змінною. В свою чергу бінарна логістична регресія застосовується для класифікації у тому випадку, коли вихідна змінна може приймати тільки два значення. Найчастіше у задачах кредитування застосовується саме бінарна логістична регресія [2].

Нас цікавить ймовірність появи події у залежності від значень змінних  $x = x_1, x_2, \dots, x_n$ . Виходом є використання логіт-функції. Вона приймає значення від 0 до 1.

Нейронна мережа - це множина алгоритмів, які намагаються розпізнати основні взаємозв'язки у вибірці даних за допомогою процесу, що імітує роботу людського мозку. У цьому сенсі нейронні мережі відносяться до систем нейронів, органічних або штучних за своєю природою. Нейронні мережі можуть адаптуватися до змінних входів; таким чином мережа дає найкращий можливий результат без необхідності переробляти критерії формування висновку. Концепція нейронних мереж, корінням якої є штучний інтелект, стрімко набирає популярність зокрема у розвитку торгових систем.

Нейронні мережі у світі фінансів допомагають розв'язувати такі задачі, як прогнозування часових рядів, прогнозування соціально-економічних процесів, алгоритмічна торгівля, класифікація цінних паперів, моделювання кредитного ризику та оцінювання власних показників і похідних цін.

Метод групового врахування аргументів (МГВА) базується на задаванні правил ускладнення моделі, системі опорних функцій, критерію селекції та методу регуляризації згідно зовнішнім критеріям. ЕОМ проводить генерування моделей-претендентів, селекцію згідно зовнішнім критеріям та відсів моделей, що не пройшли селекцію. В зв'язку з цим основну структуру алгоритму самоорганізації можна навести у такому вигляді:

1. попередня обробка спостережень з урахуванням системи обраних опорних функцій (скорочується кількість претендентів);
2. генерування множини моделей-претендентів;
3. обчислення критеріїв селекції, що є зовнішніми доповненнями, та пошук моделі оптимальної складності.

Головна ідея МГВА полягає в наступному: стверджується, що для задачі однократного прогнозу доцільно знизити точність визначення оцінок коефіцієнтів рівняння регресії, але за рахунок цього придати йому більшу регулярність. Тому нашою метою в цій задачі є не мінімізація похибок на вже відомих вузлах інтерполяції, а мінімізація похибок на нових точках, які ми в момент синтезу рівняння регресії ще не мали відповідних значень [2].

Random forest (англ. випадковий ліс) — ансамблевий метод машинного навчання для класифікації, регресійного аналізу та інших завдань, який працює за допомогою побудови численних дерев прийняття рішень під час тренування моделі й продукує моду для класів (класифікацій) або усереднений прогноз (регресію) на побудованих деревах. Недоліком цього методу є схильність до перенавчання [3].

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для дослідження було взято дані з сайту Світового Банку, Trading Economics, Управління Верховного комісара ООН у справах біженців. В даних було отримано таку інформацію:

- ВВП України (1991-2022 рр.);
- промисловість (включаючи будівництво), додана вартість (% ВВП) (1991-2022 рр.);
- площу земель сільськогосподарського призначення (кв. км) (1991-2022 рр.);
- об'єм виробництва, додана вартість (% ВВП) (1991-2022 рр.);
- кількість українських біженців у різних країнах Європи (станом на березень 2022 р.).

Для коректного проведення регресійного аналізу досліджено наявність можливих лінійних зв'язків між змінними, які в подальшому пояснюватимуть поведінку змінної (табл. 1), та визначити взаємозв'язок ризику з іншими змінними та змінних між собою.

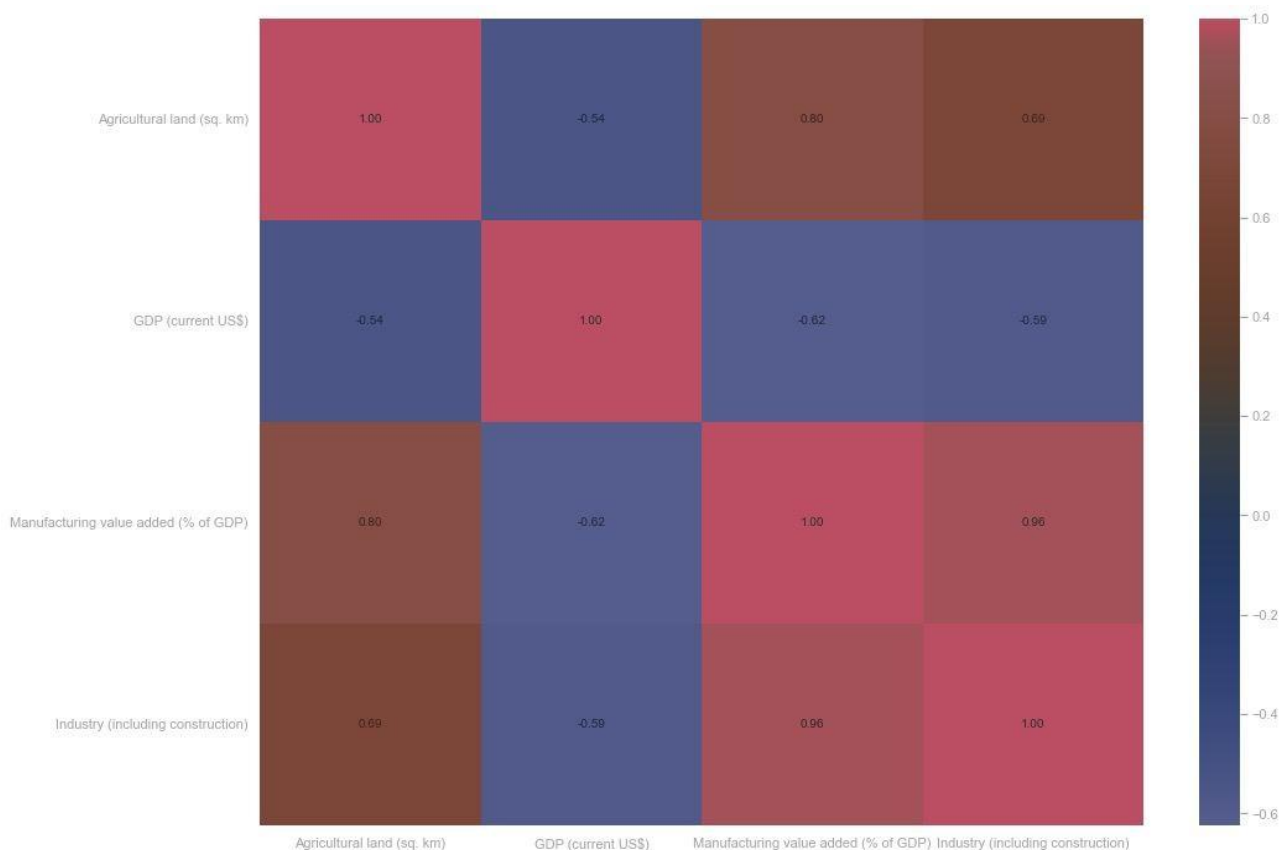


Рисунок 1. Кореляція між змінними

Визначено, що згідно з кореляційними показниками найбільше ВВП залежить від промисловості. Автокореляційний аналіз є важливим кроком у дослідницькому аналізі даних прогнозування часових рядів. Він допомагає виявити шаблони та перевірити на випадковість. Це особливо важливо, коли є намір використовувати модель авторегресії з ковзним середнім (АРКС) для прогнозування, оскільки це допомагає визначити структуру і параметри моделі. Аналіз передбачає перегляд графіків автокореляційної функції (ACF) і часткової автокореляційної функції (PACF).

При оцінюванні спектра параметрів ARMA спочатку оцінюються параметри AR, а потім параметри MA оцінюються на основі цих параметрів AR. Потім отримують спектральні оцінки моделі ARMA. Тому оцінка параметра моделі MA часто розраховується як процес асоціації із спектром параметрів моделі ARMA.

Таблиця 2. Автокореляція та часткова автокореляція

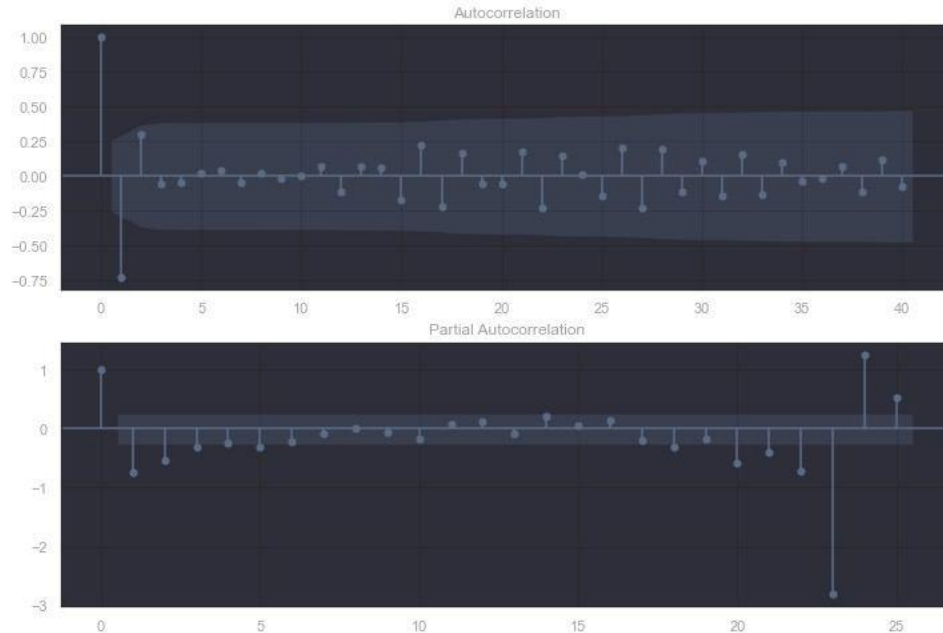


Рисунок 2. Автокореляція та часткова автокореляція

ARMA — це модель прогнозування, у якій методи аналізу авторегресії (AR) і ковзного середнього (MA) застосовуються до даних часових рядів, які добре поведуться. В ARMA припускається, що часовий ряд є стаціонарним і, коли є коливання, то це відбувається рівномірно протягом певного часу.

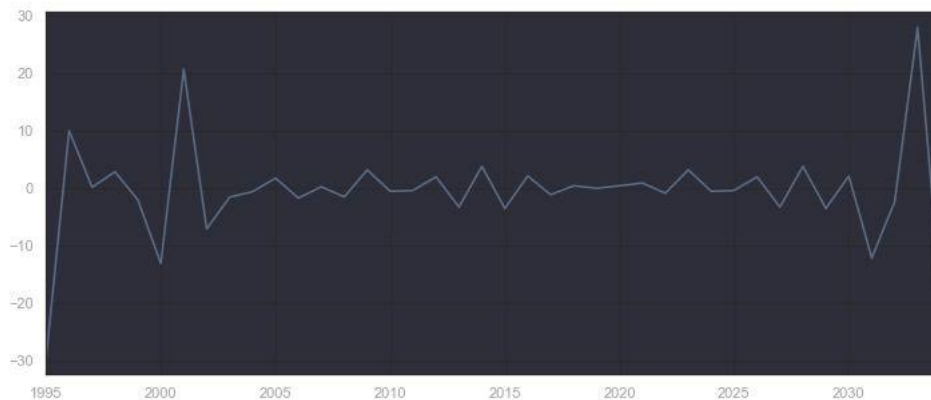


Рисунок 3. Прогноз росту ВВП(%) України

Після очищення даних і дослідницького аналізу даних було проведено побудову наступних регресійних моделей для прогнозування річної зміни відсотка ВВП країн, в які виїхали біженці з України: лінійна регресія; регресія КНН; регресія на деревах рішень; Random Forest; регресія Adaboost; Lasso та Ridge. Базова модель, зміна середньорічного відсотка ВВП, дала середньоквадратичну похибку 2,46%. Середньоквадратична похибка

вказує на те, що отримані похибки приблизно на 2,46% менші від фактичної річної процентної зміни ВВП.

Спрогнозовано річну зміну ВВП у 2022 році для країн, на які сильно вплинули українські біженці. Залишається оцінити, наскільки це точно, оскільки накопичується більше біженців і нові дані використовуються для прогнозів. Для оцінювання прогнозів було використано кількість біженців від 22 березня 2022 року, а також умовні середні або медіанні значення для відсутніх економічних характеристик.

	Country	Refugees under UNHCR's mandate	GDP Annual Change
0	Poland	2083854	3.366897
1	Romania	535461	3.410381
2	Hungary	312120	3.743343
3	Slovakia	250036	3.054826
4	Russian Federation	231764	2.449776
5	Belarus	3765	2.378090

Рисунок 4. Прогноз змін відсотка ВВП країн

Використано дані Організації Об'єднаних Націй, щоб забезпечити початкове дослідження того, як потоки біженців можуть вплинути на економіку приймаючих країн.

Після порівняння та налаштування моделей Random Forest було визнано найкращим, оскільки він дає меншу середньоквадратичну помилку 2,15% для тестових даних. Крім того, модель може пояснити 38,65% дисперсії нашої цілі, що краще, ніж базова модель.

Є впевненість у адекватності побудованої моделі, усвідомлюємо обмеження наших даних і часу. Щоб рухатися вперед із точнішою моделлю, необхідно збирати більше даних і провести глибше занурення. Також необхідно використати моделі часових рядів, щоб дослідити вплив минулих і теперішніх потоків біженців на майбутнє зростання ВВП.

## 4. ВИСНОВКИ

На сьогоднішній день маємо одну з головних проблем України і усього світу - війна. Виникає проблема, що в ході бойових дій страждає економіка, інфраструктура і, що найважливіше, людські життя. Тому виникає потреба дослідити масштаб трагедії і спрогнозувати можливі результати війни хоча б у короткостроковій перспективі.

У ході дослідження було реалізовано такі моделі, які дозволяють виконувати оцінку економічних показників України та інших країн. Для цього було використані дані Світового Банку, Управління Верховного комісара ООН у справах біженців, використані алгоритми регресійного аналізу та машинного навчання.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Прийма С.С., Приймак В.І. Теоретичні основи трактування поняття “Соціально-економічні процеси в ринковій економіці”.
2. Кузнєцова Н.В. Системний підхід до аналізу характеристик моделей оцінювання ризиків кредитування.
3. Random Forest. URL: [https://ru.wikipedia.org/wiki/Random\\_forest](https://ru.wikipedia.org/wiki/Random_forest)

# СИСТЕМНИЙ ПІДХІД ЩОДО ПРОЕКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Шахворостова В.Д.<sup>1</sup>, Тимошук О.Л.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> vdshakh@gmail.com,

<sup>2</sup> o.tymoshchuk@kpi.ua [0000-0003-1863-3095]

**Мікросервісна архітектура** – популярний підхід до розробки програмного забезпечення. Його основна риса полягає у тому, що додаток розбивається на невеликі автономні компоненти (мікросервіси) з чітко визначеними інтерфейсами. Система будується як набір незалежних і слабозв'язаних сервісів, які можна створювати, використовуючи різні мови програмування та технології для збереження даних.

**Ключові слова:** мікросервісна архітектура, мікросервіси, патерни проектування програмного забезпечення, хмарні провайдери.

## 1. ВСТУП

Архітектурний стиль мікросервісів — це підхід, при якому єдиний додаток будується як набір невеликих сервісів, кожен з яких працює у власному процесі та комунікує з іншими, використовуючи прості механізми, як правило HTTP. Ці сервіси побудовані навколо бізнес-потреб і розгортаються незалежно, з використанням повністю автоматизованого середовища. Самі по собі ці сервіси можуть бути написані на різних мовах і використовувати різні технології збереження даних.

Найкраще всього порівняти мікросервісний підхід з монолітним – додатком, що побудований як єдине ціле [1]. Монолітний сервер — досить очевидний спосіб побудови подібних систем [1]. Вся логіка по обробці запитів виконується в єдиному процесі, при цьому є можливість використовувати переваги обраної мови програмування. Монолітні додатки можуть бути успішними, однак все більше людей розчаровуються у них, особливо враховуючи те, що все більше додатків розгортаються у хмарі. Будь-які зміни, навіть найменші, вимагають передеплою та розгортання всього моноліту. З часом зберігання якісної модульної структури стає все складнішою задачею: зміна логіки одного модулю мають тенденцію впливу на код інших. Тож доводиться масштабувати увесь додаток цілком, навіть за умови, що це потрібно лише для одного модулю цього додатку.

## 2. ПІДХОДИ ДО ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Шаблони проектування програмного забезпечення — це загальні багаторазові рішення поширених проблем у проектуванні програмного забезпечення. Шаблони дизайну допомагають ділитися спільним словником та використовувати перевірене на практиці рішення замість того, щоб щоразу експериментувати та ризикувати.

Патерном проектування називається певний звичний спосіб вирішення конкретної проблеми чи виклику, які регулярно можна зустріти при проектуванні архітектури програмного забезпечення [2].

У рамках аналізу підходів до проектування програмного забезпечення, вдалось виділити наступні шаблони:

1. Агрегатор.
2. Посередник.
3. Ланцюжок.
4. Гілка.
5. Дані, що розділяються.
6. Асинхронні повідомлення.

Для подальшої роботи було прийнято рішення орієнтуватись на патерни агрегатор, ланцюжок та дані, що розділяються. Надалі походу реалізації програмного забезпечення було остаточно визначено з яким саме із наведених патернів була продовжена робота, крім того, також розглядався варіант комбінації підходів. Прийняття рішення базувалось на тому який варіант був найефективнішим – чиста реалізація конкретного патерну чи їх комбінація.

### 3. ХМАРНА ІНФРАСТРУКТУРА

Хмарні послуги – це інфраструктура, платформи або програмне забезпечення, які розміщуються сторонніми постачальниками та надаються користувачам через Інтернет [3].

На сьогоднішній день на глобальному ринку хмарних технологій домінують три основних постачальника, які ділять 64% ринку [4]:

1. Amazon Web Services або AWS.
2. Microsoft Azure.
3. Google Cloud Platform (GCP).

Згідно з Synergy Research Group ситуація на ринку нині має вигляд, що проілюстрований на рисунку 1.

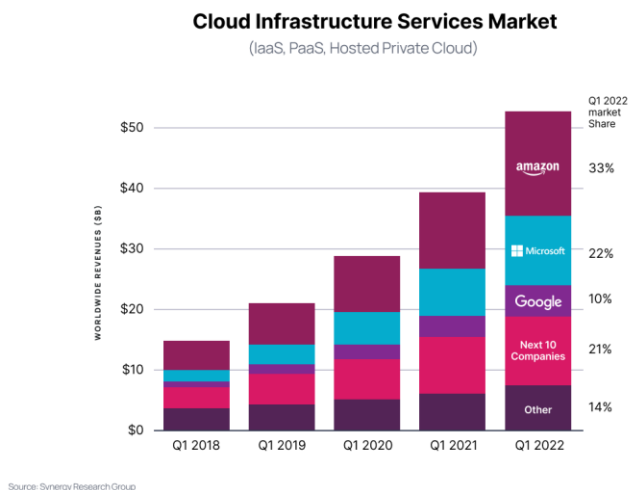


Рисунок 1. Динаміка популярності різних хмарних сервісів впродовж останніх п'ятьох років

Amazon Web Services (AWS) пропонує комп'ютерні ресурси та послуги, які можуть створювати програми за лічені хвилини з оплатою за використання [4]. Наприклад, є можливість орендувати сервер на AWS для підключення, налаштування, захисту та роботи так само, як фізичний сервер. Відмінність полягає в тому, що віртуальний сервер працює поверх мережі планетарного масштабу, керованої AWS.

Microsoft Azure — це загальнодоступна хмарна платформа, яка надає рішення інфраструктури як послуги (IaaS), платформи як послуги (PaaS) і програмного забезпечення як послуги (SaaS) для аналітики, віртуальних обчислень, зберігання, мереж та інших служб. Він може покращити або замінити локальні сервери [4].

Google Cloud, спочатку App Engine, — це набір хмарних сервісів, створений Google у 2008 році [4]. GCP пропонує підприємствам у всьому світі інфраструктуру як послугу (IaaS), платформу як послугу (PaaS) і програмне забезпечення як послугу (SaaS). . Наприклад, GCP — це в першу чергу служба для розробки та підтримки оригінальних програм, які потім можна публікувати з його гіпермасштабованих центрів обробки даних.

## 4. АНАЛІЗ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

На початку розробки було свідомо прийняте рішення щодо відмови від графічного інтерфейсу. Під час імплементації перевірка роботоспроможності написаного перевірялась у різних форматах та у кілька етапів.

Спершу коректна робота написаного коду перевірялась локально – запуск програми у Терміналі через команду «go run main.go» та співставлення очікуваного результату з фактичним: зміни у базі, зміна у поведінці на фронті відповідно до нового стану бази.

Наступний етап полягав у деплої нового функціоналу на проект у Google Cloud та у подальшій перевірці в умовах, що найближчі до оточення продакшену.

Після успішного деплою проводилось тестування функціоналу у нових умовах. У нагоді стала програма Postman, адже саме її можливості дозволяють тестувати API. Отже, вказавши URL для запиту, тіло самого запиту та пройшовши авторизацію, вдалось отримати відповідь та статус 200OK, що проілюстровано на рисунку 2. Перевіривши зміни у базі, вдалось переконатись що частина написаного функціоналу справді відпрацював коректно, тож можна рухатись далі та розробляти наступні елементи для повноцінного функціонування програмного забезпечення.

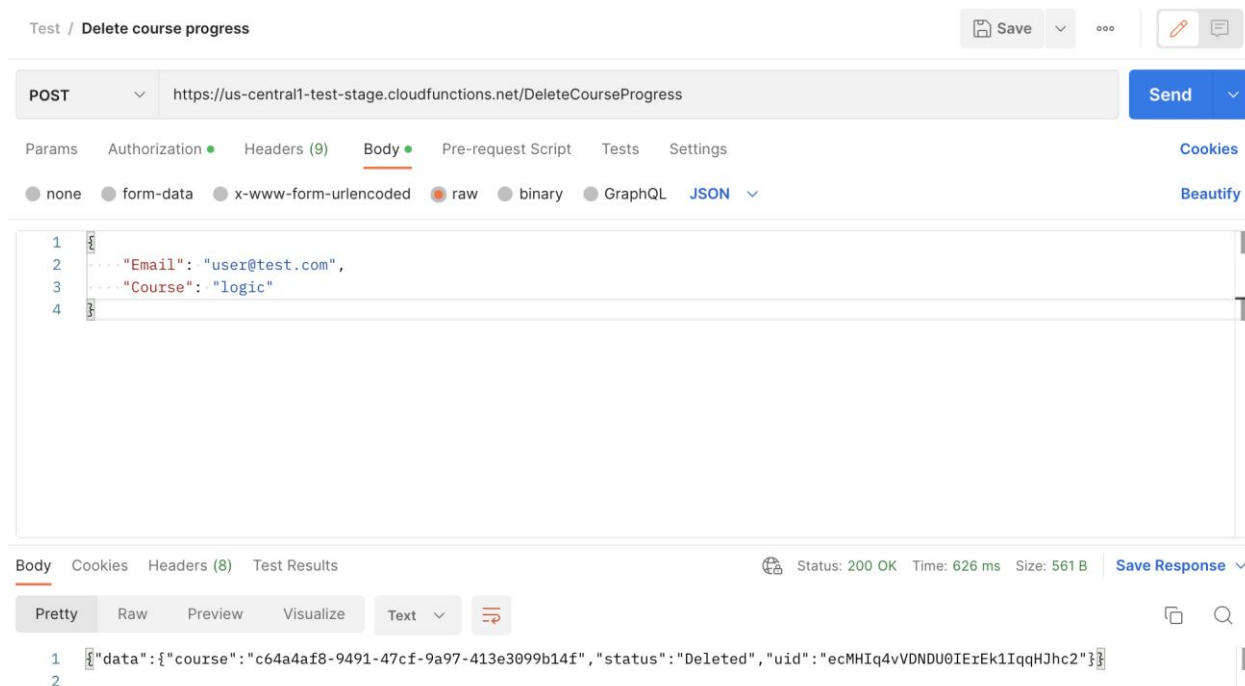


Рисунок 2. Тестування хмарної функції через Postman

Також на кожному етапі додатково перевірялись логи роботи функції у Google Cloud, щоб вкотре пересвідчитись, що все працює правильно.

Крім наведених особливостей розробки та тестування, також додатково варто додати з яким підходом реалізовувались складніші задачі, наприклад, такі як створення нової статті, що може складатись з різних інтерактивних елементів.

Введення великої кількості інформації у тіло запиту може бути проблематичним як для користувача, так і для обробки програми. Тож було прийнято рішення про заповнення файлу у Google Spreadsheet (приклад шаблону для заповнення проілюстровано на рисунку 3), прикріплення заповненого файлу до запиту у Postman та його відправка на сервер, де буде проводитись подальша обробка інформації.

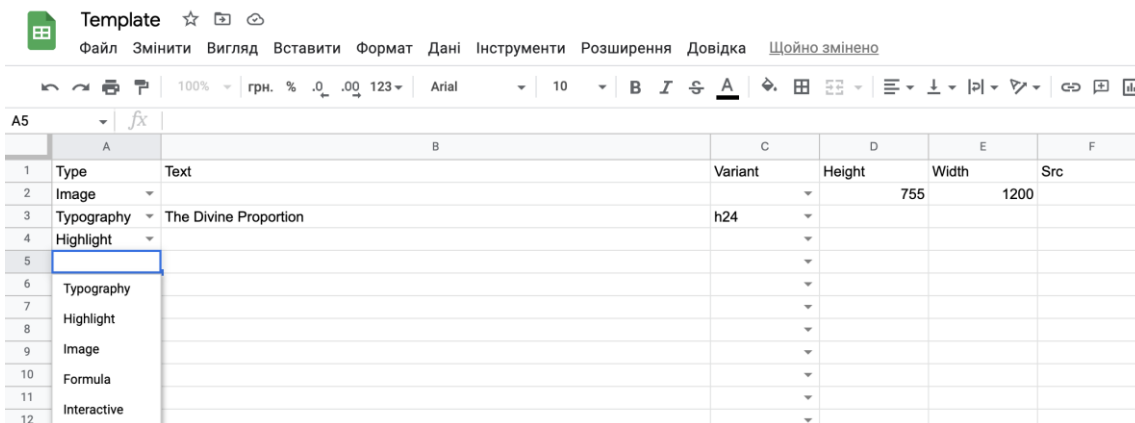


Рисунок 3. Шаблон для заповнення статті у Google Spreadsheet

У подальшому заповнений файл прикріплюється до запиту у Postman та відбувається відправка на сервер, де буде проводитись подальша обробка інформації.

## 5. ВИСНОВКИ

Популярність мікросервісів останнім часом зростає, оскільки вони можуть вирішити багато поточних ІТ-завдань, таких як збільшення швидкості, масштабованість додатків і процеси швидкого тестування. Більше того, мікросервісна архітектура впевнено та поступово стає лідером у підході до розробки систем, замінюючи монолітні рішення, які є застарілими та менш гнучкими.

У ході дослідження вдалось проаналізувати системний підхід щодо проектування архітектури програмного забезпечення: існуючі патерни та ринок хмарних провайдерів. Окрім того, на основі дослідження вдалось побудувати платформу для взаємодії з великою кількістю контенту різного роду.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Dataversity [Електронний ресурс] : A Brief History of Microservices / Keith D.Foote – Режим доступу : [dataversity.net/a-brief-history-of-microservices/](https://dataversity.net/a-brief-history-of-microservices/) (дата звернення 13.11.2022).
2. Refactoring Guru [Електронний ресурс] : What's a design pattern? – Режим доступу : <https://refactoring.guru/design-patterns/what-is-pattern> (дата звернення 13.11.2022).
3. RedHat [Електронний ресурс] : What are cloud services? – Режим доступу : <https://www.redhat.com/en/topics/cloud-computing/what-are-cloud-services> (дата звернення 13.11.2022).
4. Brocoders [Електронний ресурс] : AWS vs. Azure vs. GCP. Comparing the TOP 3 Cloud Providers – Режим доступу : <https://brocoders.com/blog/aws-vs-azure-vs-gcp/> (дата звернення 13.11.2022).



# МОДЕЛЮВАННЯ ЗАЛЕЖНОСТІ ФАКТОРІВ РИЗИКУ ЗА ДОПОМОГОЮ КОПУЛ

Шепель І.О.<sup>1</sup>, Бідюк П.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>irene.shepel@gmail.com

**В останні роки управління ризиками стало одним із найважливіших напрямів досліджень у фінансовій та економічній сферах. Одним з основних інтересів в цих сферах є моделювання залежностей факторів ризику, що дозволяють зробити апарат спеціальних функцій – копул і статистичний аналіз. Копула об'єднує одновимірні маргінальні розподіли разом, щоб сформувати багатовимірний розподіл, і у результаті виходить спільна функція розподілу стандартних рівномірних або нормальних випадкових величин. Метою роботи є аналіз методів побудови копул та моделювання функції спільного розподілу. Результатом дослідження є змодельовані багатовимірні розподіли та оцінки міри ризику.**

**Ключові слова:** фінансові ризики, копули, аналіз ризиків, залежність, VaR.

## 1. ВСТУП

Останнім часом дослідження, присвячені моделюванню внутрішньої залежності системи показників, використовують апарат спеціальних функцій – копул. Фундаментальна ідея копул полягає у поданні спільного розподілу множини показників у вигляді двох компонентів: перша відповідає за поведінку цих показників, а друга визначає характер залежності між ними.

Довгий час статистичне моделювання в сфері фінансів та економіки здебільшого ґрунтувалося на спрощених припущеннях. Нормальний розподіл домінував у дослідженнях багатовимірних розподілів: його часто припускали, але рідко заперечували [1]. Багатовимірні нормальні розподіли привабливі через їхню легку математичну обробку та керованість. У цьому випадку зв'язок між двома випадковими наслідками можна повністю описати, знаючи лише маргінальні розподіли та один додатковий параметр — коефіцієнт кореляції [2]. З роками статистики почали розуміти необхідність вивчення альтернатив нормальному розподілу. Істотною перевагою копул порівняно з іншими методами, що використовуються в моделюванні та аналізі, є його гнучкість – можливість побудови моделей різноманітних нелінійних структур залежності між досліджуваними рядами даних [3].

## 2. КОПУЛИ ТА МЕТОДИ ОЦІНЮВАННЯ ЇХ ПАРАМЕТРІВ

У теорії ймовірностей відправною точкою є поняття ймовірнісного простору  $(\Omega, \mathfrak{F}, Pr)$ , де  $\Omega$  – область визначення відповідних випадкових змінних;  $\mathfrak{F}$  – сігма-алгебра, тобто сімейство випадкових подій, які є підмножинами  $\Omega$ ;  $Pr$  – ймовірнісна міра. При цьому сімейство підмножин  $\mathfrak{F}$  в області визначення  $\Omega$  називають сігма-алгеброю, якщо воно

задовольняє таким властивостям: (1)  $\Omega \in \mathfrak{F}$ ; (2)  $A \in \mathfrak{F} \Rightarrow A \in \mathfrak{F}$ ;  $A_1, A_2, A_3, \dots \in \mathfrak{F} \Rightarrow \bigcap_{i \geq 1} A_i \in \mathfrak{F}$ .

Надалі під ризиком будемо розуміти випадкову подію, яка може призвести до фінансових втрат. Формальним означенням ризику може бути таке: ризик  $X$  – це випадкова подія, яка характеризується двома невід’ємними величинами – рівнем випадкових виплат або втрат та ймовірністю цієї події. Спільний розподіл ризиків розглянемо у вигляді:

$$H(x_1, \dots, x_n) = P(X_1 \leq x_1, \dots, X_n \leq x_n) = C(F_1(x_1), \dots, F_n(x_n)),$$

де  $F_1, \dots, F_n$  – маргінальні функції розподілу окремих ризиків;  $C$  –  $n$ -копула, що характеризує структуру залежності між ризиками.

**Теорема 1** (Скляра) [5]: Нехай  $H$  –  $n$ -вимірна функція спільного розподілу з маргінальними розподілами  $F_1, \dots, F_n$ . Тоді існує така  $n$ -копула  $C$ , що

$$H(x_1, \dots, x_n) = C(F_1(x_1), \dots, F_n(x_n)) \text{ для всіх } x \in R^n.$$

Якщо  $F_1, \dots, F_n$ , є неперервними, тоді  $C$  унікальна; в іншому випадку функції  $C$  є унікально визначеними на  $\text{Rng } F_1 \times \dots \times \text{Rng } F_n$ . І навпаки, якщо  $F_1, \dots, F_n$  є розподілами, а  $C$  є  $n$ -копулою, тоді  $H(x_1, \dots, x_n)$  –  $n$ -вимірна функція спільного розподілу з маргінальними розподілами  $F_1, \dots, F_n$ .

**Побудова сімейства копул.** Теорема Скляра гарантує існування копули та її унікальність за певних умов, але не надає методу її знаходження. Розглянемо методи побудови сімейств копул.

### 1. Метод оберненої функції

Ідея цього методу полягає в тому, що з теореми Скляра для спільної функції розподілу  $H$  та неперервних маргінальних розподілів  $F_1, \dots, F_n$  копула  $C$  визначається так:

$$C(u_1, \dots, u_n) = H(F_1^{(-1)}(u_1), \dots, F_n^{(-1)}(u_n)),$$

де  $F_i^{(-1)}(u_i)$  – обернена функція до функції маргінального розподілу  $F_i$ .

Найбільш поширеними у моделюванні фінансових випадкових величин є еліптичні розподіли і зокрема багатовимірний нормальний розподіл. При застосуванні до нього зворотного методу отримуємо багатовимірну нормальну копулу або як її ще називають – Гаусову копулу.

**Означення 1:** Нехай  $\rho$  – симетрична, додатно визначена матриця з діагоналлю 1. Гаусовою багатовимірною копулою називають функцію:

$$C(F_1, F_2, \dots, F_n, \rho) = \Phi_\rho(\Phi^{-1}(F_1), \Phi^{-1}(F_2), \dots, \Phi^{-1}(F_n)),$$

де  $\Phi$  – функція стандартного одновимірного нормального розподілу;  $\Phi_\rho$  – функція багатовимірного стандартного нормального розподілу з кореляційною матрицею  $\rho$ .

Щільністю багатовимірної Гаусової копули є:

$$c(F_1, F_2, \dots, F_n, \rho) = \frac{1}{\rho} \exp \left\{ -\frac{1}{2} \begin{pmatrix} \Phi^{-1}(F_1) \\ \vdots \\ \Phi^{-1}(F_n) \end{pmatrix} \times \rho^{-1} - I \times \begin{pmatrix} \Phi^{-1}(F_1) \\ \vdots \\ \Phi^{-1}(F_n) \end{pmatrix} \right\}.$$

Тобто Гаусова копула повністю визначається кореляційною матрицею  $\rho$ , а тому її параметри досить легко оцінити.

### 2. Архімедові копули

**Означення 2:** Нехай  $\phi$  – неперервна строго зростаюча функція з  $I$  в  $[0, \infty]$  такою що  $\phi(1) = 0$ . Псевдо-інверсією для  $\phi$  називається така функція  $\phi^{[-1]}$  з  $\text{Dom } \phi = [0, \infty]$  в  $I$ , що

$$\phi^{[-1]}(t) = \begin{cases} \phi^{-1}(t), & 0 \leq t \leq \phi(0), \\ 0, & \phi(0) \leq t \leq \infty. \end{cases}$$

Відзначимо, що  $\phi^{[-1]}$  є неперервною та неспадаючою на  $[0, \infty]$ , та строго спадаючою на  $[0, \phi(0)]$ . Більш того,  $\phi^{[-1]}(\phi(u)) = u$  на  $I$ , та

$$\phi(\phi^{[-1]}(t)) = \begin{cases} t, & 0 \leq t \leq \phi(0), \\ \phi(0), & \phi(0) \leq t \leq \infty. \end{cases}$$

Якщо  $\phi(0) = \infty$ , то  $\phi^{[-1]} = \phi^{-1}$ .

**Лема 1:** Нехай  $\phi$  є неперервною строго спадаючою функцією з  $I$  в  $[0, \infty]$  такою, що  $\phi(1) = 0$ , та нехай  $\phi^{[-1]}$  буде псевдо-інверсією для  $\phi$  і функція  $C$  з  $I^2$  в  $I$  задається

$$C(u, v) = \phi^{[-1]}(\phi(u) + \phi(v)). \quad (1)$$

Тоді  $C$  задовольняє умовам обмеження для копули.

**Доведення:**  $C(u, 0) = \phi^{[-1]}(\phi(u) + \phi(0)) = 0$  та  $C(u, 1) = \phi^{[-1]}(\phi(u) + \phi(1)) = \phi^{[-1]}(\phi(u)) = u$ . Аналогічно для  $C(0, v) = 0$  та  $C(1, v) = v$ . Якщо  $\phi(0) = \infty$  тоді  $\phi$  називають строгим генератором. В цьому випадку  $\phi^{[-1]} = \phi^{-1}$  та

$$C(u, v) = \phi^{-1}(\phi(u) + \phi(v)) \quad (2)$$

називається строгою архімедовою копулою.

Всі копули, що можуть бути представлені у вигляді (1), називають архімедовими. Функція  $\phi$  називається генератором копули. Цей клас копул є одним з найбільш вживаних, тому що до нього належить значна кількість параметричних родин копул, що дає можливість відобразити значну різноманітність структур залежності. До того ж, побудова копул цього класу досить легка.

З (2) маємо:

$$C_\theta(u, v) = \varphi^{-1}(\varphi(u) + \varphi(v)) = \exp(-[(-\ln u)^\theta + (-\ln v)^\theta]^{\frac{1}{\theta}}).$$

Сімейство копул називають копулами Гумбела у випадку двох змінних. До архімедових також належать сімейства копул Франка у випадку двох змінних

$$C(F_1, F_2) = -\frac{1}{\beta} \ln \left( 1 + \frac{(e^{-\beta F_1} - 1)(e^{-\beta F_2} - 1)}{e^{-\beta} - 1} \right), \quad \beta \in \mathbb{R} \setminus \{0\};$$

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для оцінювання можливих ринкових втрат при виконанні валютних операцій часто застосовують відому модель VaR (Value-at-Risk). Ця модель використана для аналізу обмінних курсів HRK, PLN, CZK відносно EUR. Для оцінювання параметрів моделі взяті щоденні курси з 04.2003 по 01.2008, що дало вибірку розміром 1598 спостереження. Змодельовано функції спільного розподілу (Рис. 1–4) та оцінено параметри одновимірних маргінальних розподілів для кожного з курсів та параметри копул за методом максимальної правдоподібності (Табл. 1). Результати обчислювальних експериментів отримано за допомогою спеціалізованої системи підтримки прийняття рішень для моделювання і прогнозування фінансових процесів.

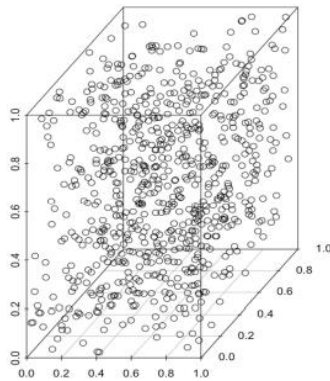


Рисунок 1. Спільний розподіл на основі копули Гумбела

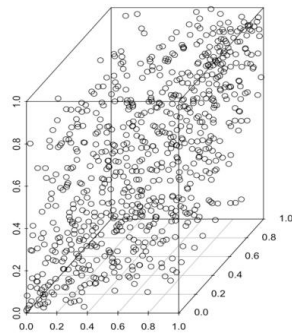


Рисунок 2. Спільний розподіл на основі нормальної копули

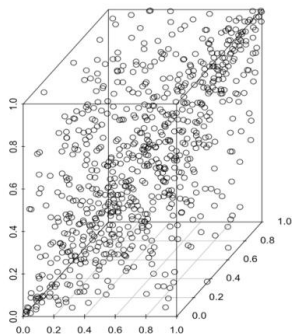


Рисунок 3. Спільний розподіл на основі копули Франка

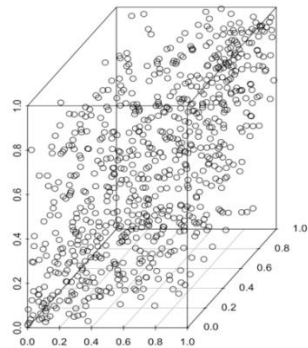


Рисунок 4. Емпіричний спільний розподіл курсів валют

Далі наведені результати оцінювання параметрів копул.

Таблиця 1. Результати оцінювання параметрів копул

Копула	Параметр	Значення	Середньоквадратична похибка
Гумбела	$\theta$	1.8736	0.0155
Нормальна	$\rho_1$	0.6617	0.0119
	$\rho_2$	0.3541	0.0138
	$\rho_3$	0.6038	0.0117
	$\rho_4$	0.8313	0.0055
	$\rho_5$	0.8672	0.0052
	$\rho_6$	0.8078	0.006
Франка	$\beta$	4.8546	0.0923

Емпірична оцінка міри ризику VaR для квантиля 0,05, тобто для 47 спостережень, що перевищують поріг, дорівнює 3,2412; для квантиля 0,01, відповідно для 15 спостережень, що перевищують поріг, оцінка міри ризику складає 3,7045. Найвних спостережень для квантиля 0,03 достатньо для отримання можливості використання цієї емпіричної оцінки на практиці, а для квантиля 0,01 вибірка є замалою.

#### 4. ВИСНОВКИ

Виконано аналіз можливості застосування класу спеціальних функцій – копул до опису багатовимірних розподілів у задачах менеджменту ризиків. Запропоновано метод побудови комбінованих маргінальних розподілів, який дозволяє враховувати важкі хвости одновимірних розподілів ризиків. Маргінальні розподіли поєднані у спільні розподіли ризиків через структуру залежності, що характеризується копулами.

На основі аналізу методів побудови сімейств копул запропоновано використовувати для моделювання ризиків кілька сімейств копул із корисними для менеджменту ризиків властивостями. В експерименті над тривимірними розподілами змін курсів валют продемонстрована можливість застосування запропонованого в роботі методу до моделювання багатовимірного розподілу через комбіновані маргінальні розподіли і структуру залежності між ними. Також наведено приклад застосування методу максимальної правдоподібності до оцінювання параметрів функції розподілу з їх подальшим використанням для оцінювання міри ризику.

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Embrechts P., McNeil A. J., Straumann D. Correlation and Dependence in Risk Management: Properties and Pitfalls. Risk Management. 2002. С. 176–223.
2. Frees E. W., Valdez E. A. Understanding Relationships Using Copulas. North American Actuarial Journal. 1998. Т. 2, № 1. С. 1–25.
3. Bouchaud J.-P. Theory of financial risk and derivative pricing: from statistical physics to risk management. 2-ге вид. Cambridge : Cambridge University Press, 2003. 379 с..
4. Cherubini U., Luciano E., Vecchiato W. Copula Methods in Finance. Wiley & Sons, Incorporated, John, 2007. 310 с.
5. Extreme Value Theory and Applications, Vol. 3 // International conference NIST, 1993, 2-7 May. – Gaithersburg, 1993. – 230 с.

# СТОХАСТИЧНА МОДЕЛЬ В БІОЛОГІЇ ТА ВІРУСОЛОГІЇ

Яблунівський О.В.<sup>1</sup>, Мальцев А.Ю.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>alexandryablunovskiy@gmail.com

**Віруси є досить великою та повсякденною загрозою, саме тому досить важливо слідкувати за ними та прогнозувати якомога точніше наслідки до яких вони можуть призвести. Саме тому для даної статті я обрав тему математики в вірусології оскільки вченні невпинно б'ються з даною проблемою та намагаються вирішити її якомога швидше. Результатом дослідження є модифікована модель SIRI, яка може описувати поведінку вірусів в наш час якомога краще.**

**Ключові слова: модель SIRI, стохастична модель в вірусології.**

## 1. ВСТУП

Наше сьогодні показує нам, що проблема епідемій та пандемій у світі є високою, як ніколи. Людство нещодавно пережило досить важкі часи та навіть зараз, ця проблема не зникла повністю. Науковці та лікарі невпинно б'ються з пандемією коронавірусу у світі, проте ми бачимо, що це не приносить повноцінного його зникнення. Тому людство повинно розуміти, що цю проблему треба вирішити як можна швидше.

Проблема широкомасштабних епідемій та пандемій походить з досить давнього часу. Людство стикається з цими проблемами кожне сторіччя і доля зводила нас з такими хворобами як чума, віспа, багато різних штамів вірусу грипу та досить багато всього іншого. Проте людство невпинно та досить швидко намагалось вирішувати ці проблеми і вдавалось це досить добре та ми робили досить вагомні внески в розвиток медицини. Проте науковці математики невпинно боролись та створювали математичні моделі розповсюдження вірусів, щоб мати змогу прогнозувати, як буде ширитись світом ця чи інша хвороба та що ми можемо зробити аби зменшити наслідки від неї для всього людства. Одна з досить вагомих та відомих стохастичних моделей є модель SIRI. Вона показує не тільки динаміку переходу населення з однієї категорії в іншу, аде і враховує зміни та фактори навколишнього середовища.

## 2. ОПИС ІНСУЮЧОЇ МОДЕЛІ

Отже, що з себе представляє модель SIRI. Вона побудована на принципі того, що з плином часу людство може переходити з однієї категорії в іншу, а саме:

1. S – це категорія сприйнятливих, тобто тих людей, які є в небезпеці та можуть бути інфікованими
2. I – це категорія інфікованих людей, тобто ті у яких уже почався перебіг хвороби та вони наражають на небезпеку людей з категорії сприйнятливих, оскільки вони можуть заражати їх.
3. R – це категорія, скажемо так, видалених, тобто це ті люди які вже не несуть загрозу іншим, тобто вони або одужали, або були ізольовані або померли від вірусу.

Дана модель, як і класична модель SIR фундаментально побудована на декількох принципово важливих припущеннях, які трохи згладжують наш світ, оскільки виразити всі аспекти в світі майже неможливо. Отже:

- Перше наше припущення, це те що ми вважаємо, що наше населення не є сталим та може змінюватись від загального коефіцієнту поповнення за рахунок народження чи імміграції, та те що він дорівнює коефіцієнту смертності;

- Друге припущення – це є той факт, що люди інфікуються з якоюсь швидкістю і вона не є сталою, а може змінюватись з часом та/або з урахуванням різних факторів, які можуть мати вплив на неї;
- Третє припущення – це те що люди переходять з категорії інфікованих в категорію видалених теж з якоюсь швидкістю, проте ця швидкість також не є сталою та може бути залежною від багатьох факторів, таких як час, навколишнє середовище та багато всього іншого;
- Четвертим є припущення, що люди з категорії видалених можуть і переходять в категорію інфікованих.

Отже перейдемо до самих рівнянь. Перше рівняння описує зміну кількості людей в категорії сприйнятливих до захворювання.

$$\frac{dS}{dt} = \mu - \mu S(t) - \frac{\beta S(t)I(t)}{1 + \alpha I(t)}$$

Де  $\mu$  - це загальний коефіцієнт поповнення за рахунок імміграції або народження він дорівнює природній смертності, він має бути додатнім та більше нуля;

$\frac{\beta I}{1 + \alpha I}$  – це швидкість передачі захворювання, вона є додатною;

$\alpha$  – це константа напівнасичення.

Друге рівняння описує зміну кількості людей, які були інфікованими та будується з декількох факторів, які є досить важливими при зміні стану, а саме зі швидкості передачі захворювання, коефіцієнта поповнення за рахунок народження та імміграції, швидкості з якою неінфекційні люди повертаються до заразного стану та коефіцієнту одужання. Отже, маємо що:

$$\frac{dI}{dt} = \frac{\beta S(t)I(t)}{1 + \alpha I(t)} - (\mu + e)I(t) + \frac{\gamma R(t)}{1 + kR(t)}$$

Де  $\frac{\beta I}{1 + \alpha I}$  – це швидкість передачі захворювання, вона є додатною;

$\mu$  - це загальний коефіцієнт поповнення за рахунок імміграції або народження він дорівнює природній смертності, він має бути додатнім та більше нуля;

$e$  – це коефіцієнт одужання;

$\frac{\gamma}{1 + kR(t)}$  – це швидкість з якою неінфіковані люди повертаються до заразного стану, та можуть наражати на небезпеку іншу частину населення;

$k$  – це константа напівнасичення.

Третє рівняння має показувати нам з якою швидкістю будемо мати поповнення даної категорії людей та швидкості з якою вони будуть вибувати з неї та переходити у стан, коли вони знову можуть наражати на небезпеку інших людей.

$$\frac{dR}{dt} = eI(t) - \mu R(t) - \frac{\gamma R(t)}{1 + kR(t)}$$

Де  $e$  – це коефіцієнт одужання;

$\mu$  - це загальний коефіцієнт поповнення за рахунок імміграції або народження він дорівнює природній смертності, він має бути додатнім та більше нуля;

$\frac{\gamma}{1 + kR(t)}$  – це швидкість з якою неінфіковані люди повертаються до заразного стану, та можуть наражати на небезпеку іншу частину населення;

Ці три рівняння в цілому створюють модель SIRI яка показує нам плинність епідемії чи пандемії з плином часу. Тепер поглибимось в деталі. Маємо що для того щоб порахувати кількість людей в населенні, то ми маємо таке рівняння, що в момент часу  $t$  населення буде складати:

$$N(t) = S(t) + R(t) + I(t)$$

Де  $S$  – це кількість сприйнятливих в момент часу  $t$ ;

$R$  – це кількість видалених в момент часу  $t$ ;

$I$  – це кількість інфікованих в момент часу  $t$ .

Також ми маємо коефіцієнт який буде показувати нам базову репродукційну кількість. Це означає, що при  $R_0 > 1$  епідемія відбудеться, якщо ж він буде менше ніж одиниця, то ми матимемо ситуацію коли епідемії не буде.

$$R_0 = \frac{\beta}{\mu + e + \frac{e\gamma}{\mu + \gamma}}$$

Де  $\beta$  - це швидкість передачі захворювання

$\mu$  - це загальний коефіцієнт поповнення за рахунок імміграції або народження він дорівнює природній смертності, він має бути додатним та більше нуля;

$e$  - це коефіцієнт одужання;

$\gamma$  - це швидкість з якою неінфіковані люди повертаються до заразного стану, та можуть наражати на небезпеку іншу частину населення;

Ми припускаємо, що коефіцієнти швидкості передачі захворювання та швидкості з якою неінфіковані люди повертаються до заразного стану, та можуть наражати на небезпеку іншу частину населення перебувають під впливом навколишнього середовища, а саме:

$$\beta \rightarrow \beta + \sigma_1 B_1(\dot{t}), \gamma \rightarrow \gamma + \sigma_2 B_2(\dot{t})$$

$\dot{B}_i(t) (i = 1, 2)$  - позначає білий шум, тобто ми маємо, що  $B_i(t)$  - це стандартний Броунівський рух, який є визначений на повному ймовірному просторі  $(\Omega, \mathcal{F}, P)$ .  $B_1(0) = 0$  та  $B_2(0) = 0$  - є стандартним відхиленням білого шуму та  $\sigma_i^2 > 0$  - представляє інтенсивність білих шумів. Отже в фіналі матимемо таку результуючу модель:

$$dS = \left[ \mu - \mu S(t) - \frac{\beta S(t)I(t)}{1 + \alpha I(t)} \right] dt - \frac{\sigma_1 S(t)I(t)}{1 + \alpha I(t)} dB_1(t)$$

$$dI = \left[ \frac{\beta S(t)I(t)}{1 + \alpha I(t)} - (\mu + e)I(t) + \frac{\gamma R(t)}{1 + kR(t)} \right] dt + \frac{\sigma_1 S(t)I(t)}{1 + \alpha I(t)} dB_1(t) + \frac{\sigma_2 R(t)}{1 + kR(t)} dB_2(t)$$

$$dR = \left[ eI(t) - \mu R(t) - \frac{\gamma R(t)}{1 + kR(t)} \right] dt - \frac{\sigma_2 R(t)}{1 + kR(t)} dB_2(t)$$

Коливання середовища є важливою частиною екосистеми в природному світі, включаючи температуру, імунологічний стан господаря, інкубаційні періоди тощо. Іноді невеликий шум може придушити вибухи в динаміці популяції. Теорія стохастичних диференціальних рівнянь (SDE) є кращим інструментом для представлення таких факторів впливу, оскільки SDE може забезпечити деякий додатковий ступінь реалізму порівняно з детермінованими системами. Багато вчених вводять ефекти стохастичних збурень у свої моделі як з біологічної, так і з математичної точки зору.

### 3. ЗАПРОПОНОВАНА ЗМІНА В МОДЕЛЬ

За основу мною була взята та сама модель SIRI, яка була описана вище. Нагадаю, що основа полягає в поділенні людей на три основні категорії між котрими вони можуть переміщуватись, отже категорії:

- S - сприйнятливі;
- I - інфіковані;
- R - видалені.

Перша зміна про яку я хочу поговорити - це умова про те що населення для котрого ми будемо дану модель можна поділити на чотири рівні категорії за віком. Ця зміна може бути зумовлені тим що, кожна категорія людей хворіє по різному та для кожної категорії населення ми матимемо свій статистичний коефіцієнт смертності. Отже, маємо:

- Перша категорія - це люди молодого віку, приблизно від нуля до двадцяти років, це молоді люди, які мають найнижчий коефіцієнт смертності та найлегший перебіг хвороби COVID-19;
- Друга категорія - це люди віком від двадцяти до сорока років, ми знаємо, що для даної категорії статистичний коефіцієнт смертності буде вищим за першу категорію та коефіцієнт швидкості перебігу хвороби буде трохи інший, оскільки люди цього віку хворіють трошки інакше;
- Третя категорія - це люди віком від сорока до шістдесяти років, вони матимуть ще вищий коефіцієнт смертності ніж друга категорія та будуть мати більший коефіцієнт швидкості перебігу хвороби, оскільки вони хворітимуть довше;



- Четверта категорія - це люди віком від шістдесяти років вони мають найвищий статистичний коефіцієнт смертності та найважчий перебіг хвороби, що являє собою найвищий коефіцієнт швидкості перебігу хвороби.

З усього вище сказаного, ми можемо сказати, що такий розподіл буде показувати стан речей стосовно хвороби найкраще, та допоможе покращити модель. Але, як відомо коефіцієнт смертності відрізняється не дуже сильно то ми будемо брати середній з усіх категорій.

Для даної зміни ми маємо додати ще декілька припущень про які, ми говорили раніше, а саме:

- Ми можемо поділити населення на чотири рівні категорії за віком;
- кількість інфікованих розподіляється на чотири категорії однаково, тобто мається на увазі, що якщо чотири інфікованих за один день, то до кожної з категорій перейде по одному інфікованому;

Перейдемо до математичних аспектів даної зміни. Оскільки ми маємо тепер чотири категорії населення, то для кожної групи населення будемо мати, що:

$$\frac{dS}{dt} = \mu - \mu S(t) - \frac{\beta}{1 + \alpha(I_1(t) + I_2(t) + I_3(t) + I_4(t))} (I_1(t) + I_2(t) + I_3(t) + I_4(t))(S_1(t) + S_2(t) + S_3(t) + S_4(t))$$

Де  $\mu$  - це загальний коефіцієнт поповнення за рахунок імміграції або народження він дорівнює природній смертності, він має бути додатнім та більше нуля;

$\frac{\beta I}{1 + \alpha I}$  - це швидкість передачі захворювання, вона є додатною;

$\alpha$  - це константа напівнасичення.

Рівняння для інфікованих 1 група:

$$\frac{dI}{dt} = 0,25 * \frac{\beta(S_1(t) + S_2(t) + S_3(t) + S_4(t))(I_1(t) + I_2(t) + I_3(t) + I_4(t))}{1 + \alpha(I_1(t) + I_2(t) + I_3(t) + I_4(t))} - (\mu + e_1)I(t) + \frac{\gamma_1(R_1(t) + R_2(t) + R_3(t) + R_4(t))}{1 + k(R_1(t) + R_2(t) + R_3(t) + R_4(t))}$$

Де  $\frac{\beta I}{1 + \alpha I}$  - це швидкість передачі захворювання, вона є додатною;

$\mu$  - це загальний коефіцієнт поповнення за рахунок імміграції або народження він дорівнює природній смертності, він має бути додатнім та більше нуля;

$e_1$  - це коефіцієнт одужання;

$\frac{\gamma}{1 + kR(t)}$  - це швидкість з якою неінфіковані люди повертаються до заразного стану, та можуть наражати на небезпеку іншу частину населення;

$k$  - це константа напівнасичення.

Для усіх інших категорій таке саме рівняння, проте з іншими коефіцієнтами одужання, швидкості з якою неінфіковані люди повертаються до заразного стану, та можуть наражати на небезпеку іншу частину населення

Рівняння для категорії видалених 1 категорія населення:

$$\frac{dR}{dt} = e_1(I_1(t) + I_2(t) + I_3(t) + I_4(t)) - \mu(R_1(t) + R_2(t) + R_3(t) + R_4(t)) - \frac{\gamma_1(R_1(t) + R_2(t) + R_3(t) + R_4(t))}{1 + k(R_1(t) + R_2(t) + R_3(t) + R_4(t))}$$

Де  $e$  - це коефіцієнт одужання;

$\mu$  - це загальний коефіцієнт поповнення за рахунок імміграції або народження він дорівнює природній смертності, він має бути додатнім та більше нуля;

$\frac{\gamma}{1 + kR(t)}$  - це швидкість з якою неінфіковані люди повертаються до заразного стану, та можуть наражати на небезпеку іншу частину населення.

## 4. ВИСНОВКИ

В даній частині дипломної роботи, мною було надане біологічне та математичне обґрунтування зміни, яку я вводив на основі базової моделі. За базову модель мною було вирішено взяти модель SIRI.

Я вважаю, що в наш час ця модель буде видавати набагато реальнішу картину світу, бо в класичній моделі немає різниці від віку. В новій моделі – це все враховано, тому я вважаю, що в наш час, потрібна саме ця модель.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Anderson, R., May, R.: Population biology of infectious diseases: Part I. *Nature* 280, 361–367 (1979)
2. Ruan, S., Wang, W.: Dynamical behavior of an epidemic model with a nonlinear incidence rate. *J. Differ. Equ.* 188, 135–163 (2003)
3. Meng, X., Chen, L., Wu, B.: A delay SIR epidemic model with pulse vaccination and incubation times. *Nonlinear Anal. Real.* 11(1), 88–98 (2010)
4. Driessche, P.V.D., Zou, X.: Modeling relapse in infectious diseases. *Math. Biosci.* 207, 89–103 (2007)
5. CAMBRIDGE STUDIES IN MATHEMATICAL BIOLOGY: 15 Editors C. CANNINGS University of Sheffield, UK F. C. HOPPENSTEADT Arizona State University, Tempe, USA L. A. SEGEL Weizmann Institute of Science, Rehovot, Israel
6. Fatini, M., Lahrouz, A., Pettersson, R., Settati, A., Taki, R.: Stochastic stability and instability of an epidemic model with relapse. *Appl. Math. Comput.* 316, 326–341 (2018)
7. Capasso, V., Serio, G.: A generalization of the Kermack–McKendrick deterministic epidemic model. *Math. Bios.* 42, 41–61 (1978)

# МЕТОДИ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ФІНАНСОВОГО СТАНУ КОРПОРАЦІЇ НА ПРИКЛАДІ КОМПАНІЇ “PHILIP MORRIS INTERNATIONAL”

Ярмола А.О.<sup>1</sup>, Зайченко О.Ю.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> anna16yarmola@gmail.com,

<sup>2</sup> zaichenko.helen@lil.kpi.ua [0000-0001-9662-3269]

Одним з найважливіших етапів в управлінні підприємством, незважаючи на розміри компанії та виду її комерційної діяльності є аналіз фінансового стану. Правильний підхід до вирішення даного завдання надає змогу прийняти правильні рішення в управлінні підприємницькою діяльністю, виявити сильні та слабкі сторони в роботі, а також прийняти оптимізаційні рішення для досягнення кращих результатів та оцінити перспективи майбутнього розвитку. За допомогою аналізу, а також і прогнозування майбутнього фінансового стану, компанія отримує наглядну картину для розуміння фінансової ситуації та прийняти обґрунтоване та документально підтвержену інформацію, щодо ефективності та стабільності її функціонування.

**Ключові слова:** аналіз фінансового стану, прогнозування, ліквідність, конкурентоспроможність, платоспроможність, Philip Morris International

## 1. ВСТУП

Одним з найпоширеніших фактів на сьогодні є те, що вся економіка у світу існує та розвивається завдяки підприємствам та корпораціям, в основі яких і лежить економічна та фінансова діяльність. Саме завдяки компаніям, які в свою чергу не тільки перетворюють процеси та ресурси на прибуток, а й задовольняють потреби суспільства. Поняття «фінансовий стан» не має одного визначення, багато науковців трактують це поняття по різному, але найбільш точним визначенням, що:

«Фінансовий стан- це складова аналітичного економічного дослідження щодо вивчення рівня, змін і динаміки фінансових показників у їх взаємозв'язку і взаємозумовленості, з метою прийняття ефективних управлінських рішень для забезпечення внутрішньої і зовнішньої діяльності господарюючих систем, сталого економічного розвитку і стійкого фінансового стану.»

При аналізі фінансового стану найчастіше розглядають такі елементи, як: ліквідність, рентабельність, платоспроможність та стабільність.

## 2. МЕТОД АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ФІНАНСОВОГО СТАНУ

Існує безліч методів аналізу фінансового стану, серед яких можна виокремити такі, як: горизонтальний (часовий) аналіз, вертикальний, трендовий аналіз, аналіз відносних показників, порівняльний та факторний аналіз. Також в першу чергу варто зазначити, що кожен з представлених методів базується на джерелах інформації про діяльність компанії: Баланс, Звіт про фінансові результати, Звіт про рух грошових коштів та Звіт про власний

капітал. Основу для якісного проведення аналізу фінансового стану є забезпеченість та повнота інформації, необхідної для дослідження.

Існують різні формули — значення, різні коефіцієнти — які можна використовувати відповідно до того, який фінансовий звіт ви аналізуєте. Щоб здійснити базовий аналіз балансу буде достатньо розрахунок лиш є таких показників:

- Коефіцієнт поточної ліквідності - вимірює вашу ліквідність, наскільки легко ваші поточні активи можна конвертувати в готівку, щоб покрити ваші короткострокові зобов'язання. Чим вищий коефіцієнт, тим більш ліквідні ваші активи.

- Коефіцієнт швидкої ліквідності - він вимірює, наскільки добре ваш бізнес може погасити свої борги

- Коефіцієнт співвідношення боргу до власного капіталу - показує, наскільки ваш бізнес залежить від власного капіталу порівняно з позиченими коштами.

Фінансова стійкість формується з ефективного формування, використання та використання фінансових ресурсів, а як результатом правильного використання впливає і платоспроможність компанії.

Коефіцієнт платоспроможності та ліквідності – є двома самостійними показниками, але пов'язані між собою, значення яких в свою чергу фінансовий стан підприємства. Поняття платоспроможності показує здатність компанії виплачувати свої зобов'язання.

По іншому можна описати платоспроможність, як стан грошовим забезпечення виробничої та інвестиційної діяльності.

А поняття ліквідності показує можливість будь-якого активу перетворитися на гроші, також є засобом забезпечити підприємство платоспроможністю.

Але також в компанії можуть виникати ситуації, коли воно є ліквідним, але неплатоспроможним: це наявність майна, якого достатньо для виплати всіх зобов'язань, але не вистачає для покриття поточних боргів, коли компанія по якійсь причині не може продати майно.

Так и навпаки, є платоспроможним, але не є ліквідним: має в наявності засоби для виплат поточних боргів, але недостатньо для всіх зобов'язань.

Для визначення ділової активності компанії проводиться на основі коефіцієнтів оборотності.

Абсолютні показники ділової активності підприємства характеризують співвідношення "витрати – прибуток", що є основою оцінювання ефективності діяльності. Витрати подані вкладеним в активи капіталом, прибуток становить різницю між виручкою від реалізації і витратами.

При підвищенні ділової активності компанії прибуток має збільшуватися більшим темпом від виручки реалізації та активи. Навіть такі випадки можуть виявлятися і успішних корпорація, що це відношення порушується. Компанія буде мати достатній рівень ділової активності, нарощуючи кількість об'єктів діяльності, реалізує весь об'єм покупцям та отримує оплату, яка вказана в умовах договору.

Важливість прогнозування фінансового стану підприємств в сучасних умовах полягає у необхідності дослідження впливу різних ризиків на їх функціонування та здійснення при потребі відповідних заходів щодо їх мінімізації з метою запобігання виникненню кризових процесів на підприємстві, погіршенню ліквідності чи фінансової стійкості

Прогнозування фінансового стану підприємства здійснюється у декілька етапів, які включають [1]:

1. Визначення об'єктів прогнозу, мети і задач.

2. Відбір об'єктів, які прогнозуються, встановлення ієрархічності, взаємозв'язків між ними.

3. Визначення часових горизонтів прогнозу.
4. Формування інформаційної бази даних для прогнозування.
5. Вибір методів прогнозування, їх обґрунтування, оцінювання точності прогнозу.
6. Складання прогнозу.
7. Відстеження результатів

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

#### 3.1 Аналіз фінансового стану

Для дослідження фінансового стану та прогнозування було обрано американську компанію «Philip Morris International». Вона є однією з найбільших виробників тютюнових виробів у світі. Свою продукцію компанії вдалося реалізувати в близько 180 країнах.

Мета компанії «Philip Morris International»: «Наша мета – замінити сигарети на бездимні продукти в інтересах повнолітніх споживачів, які інакше продовжували б курити, суспільства, компанії та своїх акціонерів.» [2]

Для аналізу було зібрану повну інформацію у вигляді фінансових звітів компанії за період 2019-2021 роки. Після перевірки повноти представлених звітів перейдемо до оцінки фінансового стану. Зі звітності було досліджено основні показники для визначення та оцінення фінансового стану на теперешній час. Після проведення дослідження можна зробити ряд висновків, які характеризують активи, капітал та прибуток компанії. Завдяки експортному потенціалу компанії, у 2021 році чистий дохід від реалізації продукції склав близько 30 мільярдів гривень, також у 2021 році РМІ демонструє стабільно високі показники фінансової діяльності. Прибуток компанії становить 3.3 мільярда гривень у 2021 році. Активи компанії на кінець 2021 року склали 11 мільярда гривень, в тому числі оборотні активи – 10.6 мільярда гривень – та необоротні – 0.4 мільярда гривень. Пасиви на кінець 2021 року переважно сформовані із поточних зобов'язань в сумі 9.3 мільярда гривень і власного капіталу – 1.5 мільярда гривень. Для кращого розуміння є необхідність продемонструвати динаміку змін фінансово-економічних показників у вигляді графіків.



Рисунок 1. Динаміка активів РМІ у 2019-2021 рр.

В результаті бачимо стабільне співвідношення суми активів, а також у компанії значно більше переважають оборотні активи над необоротними, що каже про неоптимальне їх співвідношення.

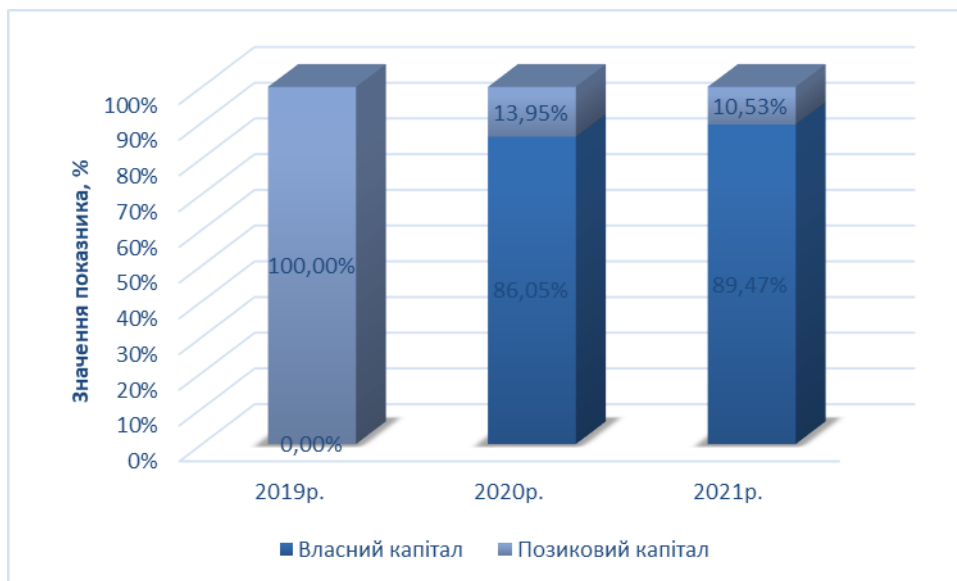


Рисунок 2. Динаміка структури капіталу РМІ за 2019-2021 рр.

Мета компанії при управлінні капіталом полягає у забезпеченні подальшої роботи як безперервно функціонуючого підприємства, щоб приносити прибуток учасникам товариства та вигоди іншим зацікавленим сторонам, а також підтримувати оптимальну структуру капіталу. Сума капіталу, управління яким компанія здійснює станом на 31 грудня 2021 року, складає 1 155 550 тис. гривень. Компанія контролює величину капіталу на основі співвідношення власних та позикових коштів.

Також продемонструємо динаміку чистого доходу від реалізації та чистого прибутку РМІ у 2019-2021 рр.

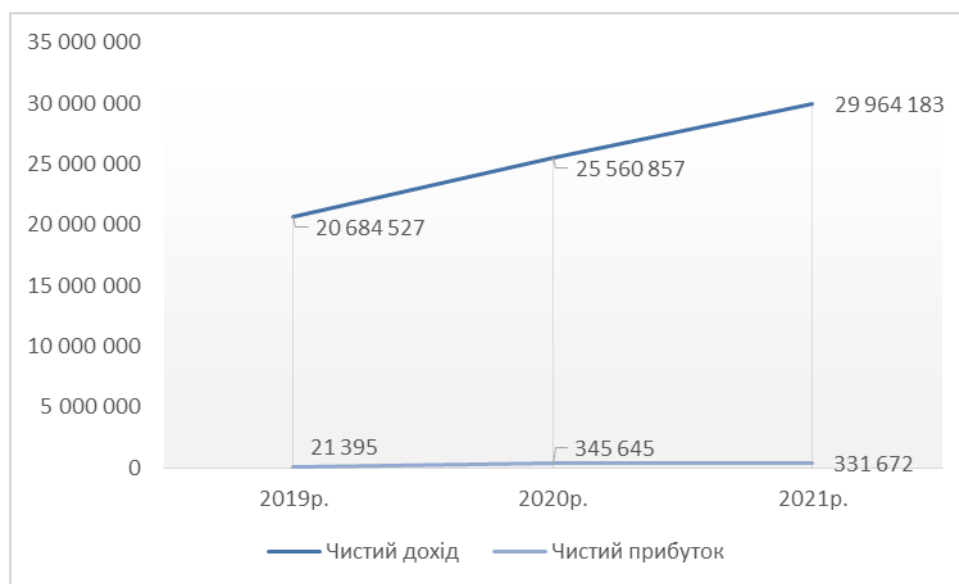


Рисунок 3. Динаміка чистого доходу від реалізації та чистого прибутку РМІ у 2019-2021 рр.

Для визначення ділової активності компанії проводиться на основі коефіцієнтів оборотності. Абсолютні показники ділової активності підприємства характеризують співвідношення "витрати – прибуток", що є основою оцінювання ефективності діяльності.



Рисунок 4. Динаміка обсягів доходів, витрат та чистого прибутку РМІ за 2019-2021 роки

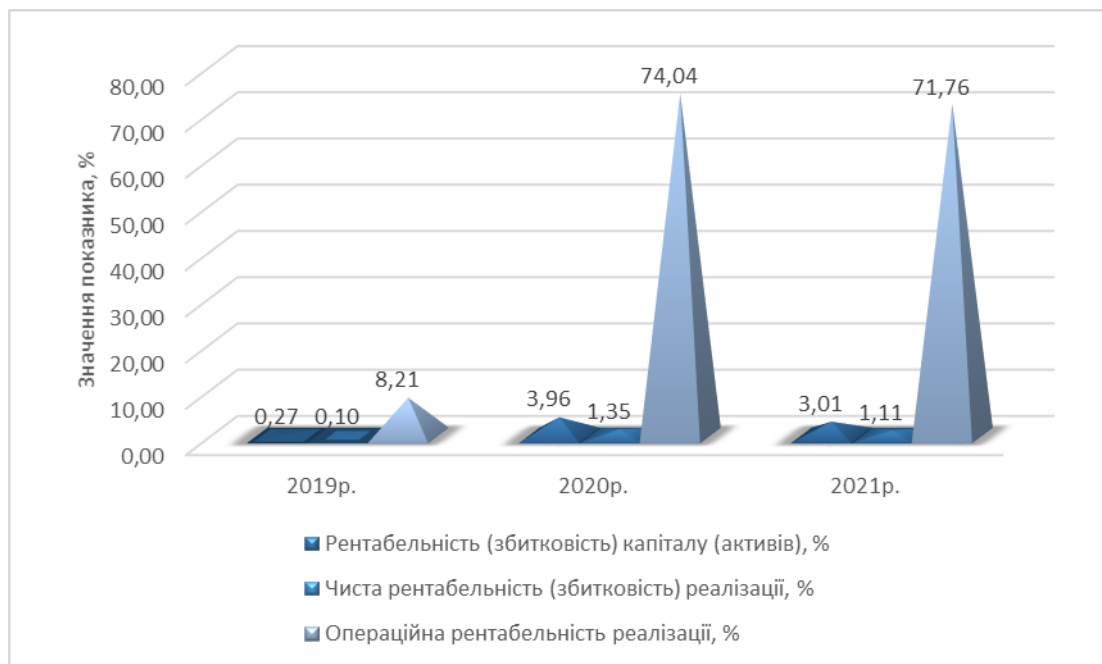


Рисунок 5. Динаміка показників рентабельності (збитків) капіталу, операційної та чистої реалізації продукції РМІ за 2019-2021 рр.

Ми змогли побачити справедливу величину прибутку компанії порівняно з сумою її витрат, капіталу та ресурсів. За результатом дослідження компанія має високий рівень платоспроможності, оскільки коефіцієнти абсолютної, швидкої та загальної ліквідності перевищують нормативні значення.

Також побудуємо динаміку коефіцієнтів термінової, мобільності активів та загальної ліквідності.

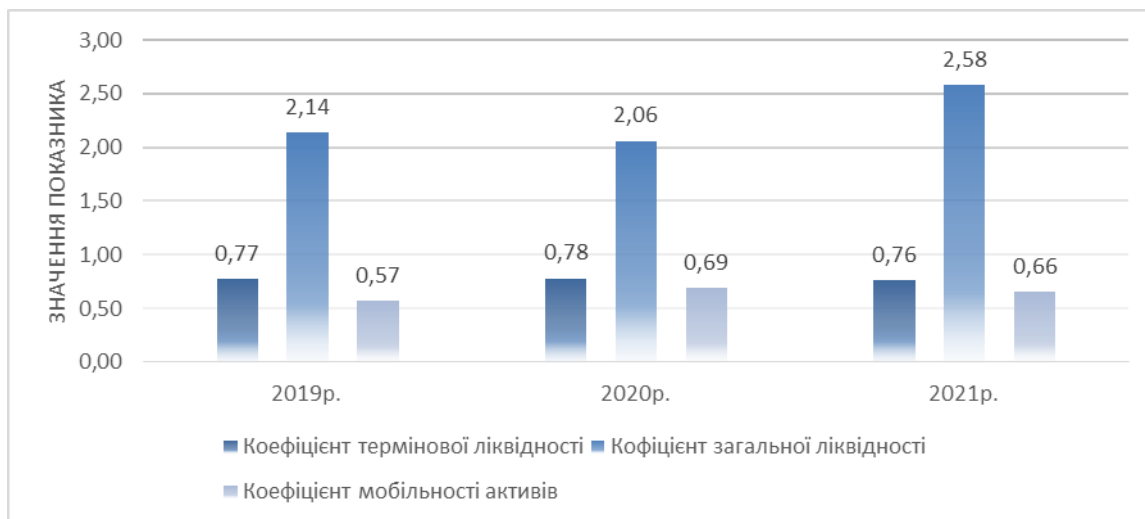


Рисунок 6. Динаміка коефіцієнтів ліквідності РМІ за 2019-2021 рр.

Отже, розглянувши деякі фінансові показники можна зробити ряд висновків, щодо фінансового стану компанії РМІ:

1. Компанія протягом року демонструє стабільний стан ліквідності. Коефіцієнт загальної ліквідності станом на кінець 2021 року становить 2.8.

2. Активи на кінець 2021 року склали 11.5 мільярда гривень, в тому числі оборотні активи – 9.9 мільярда гривень, необоротні активи – 1.6 мільярда гривень.

3. Пасиви сформовані із поточних зобов'язань в сумі 3.5 мільярдів гривень і власного капіталу – 7.9 мільярда гривень.

4. Внаслідок використання компанією фінансових інструментів компанія наражається на наступні ризики:

- Кредитний ризик

Станом на кінець 2021 року у компанії був один дебітор, розмір заборгованості якого складала приблизно 75%-85% всієї дебіторської заборгованості.

- Ринковий ризик

Кожного року компанія поновлює свою фінансову політику, яка встановлює ліміти для зберігання коштів на поточних, депозитних та кредитних рахунках компанією окремо по кожному банку, з якими співпрацює.

- Валютний ризик

Ризик розраховується лише для грошових залишків у валютах, інших ніж функціональна валюта компанії. Валютний ризик на кінець 2021 року не відображає типовий рівень ризику протягом року.

- Ризик ліквідності

Для того, щоб звести до мінімуму ризик ліквідності компанія підтримує стабільну базу фінансування, заключаючи кредитні угоди з міжнародними банками.

5. Також основним ризиком для діяльності компанії є нестабільність та непередбачуваність змін у податковому законодавстві.

### 3.2 Прогнозування фінансового стану підприємства

Для нашої задачі в прогнозуванні фінансового стану підприємства доцільно використовувати метод прогнозування екстраполяції тренду, він базується на зміні тенденцій характеристик підприємства. Та є одним з найпоширеніших методів прогнозування динаміки. Моє рішення було прийнято у виборі цього методу шляхом проведеного



дослідження та було зроблено висновок, що компанія має стабільний результат ліквідності та платоспроможності, та має прогресивний ріст її основних запасів та прибутку, що в свою чергу свідчить про постійний розвиток компанії, залучення більшого об'єму фінансування та розширення асортименту свого продукту.

Мета даного методу полягає в тому щоб показати до яких результатів можна дійти в майбутньому, якщо діяльність компанії буде розвиватися з таким же темпом та прискоренням, як в минулі періоди.

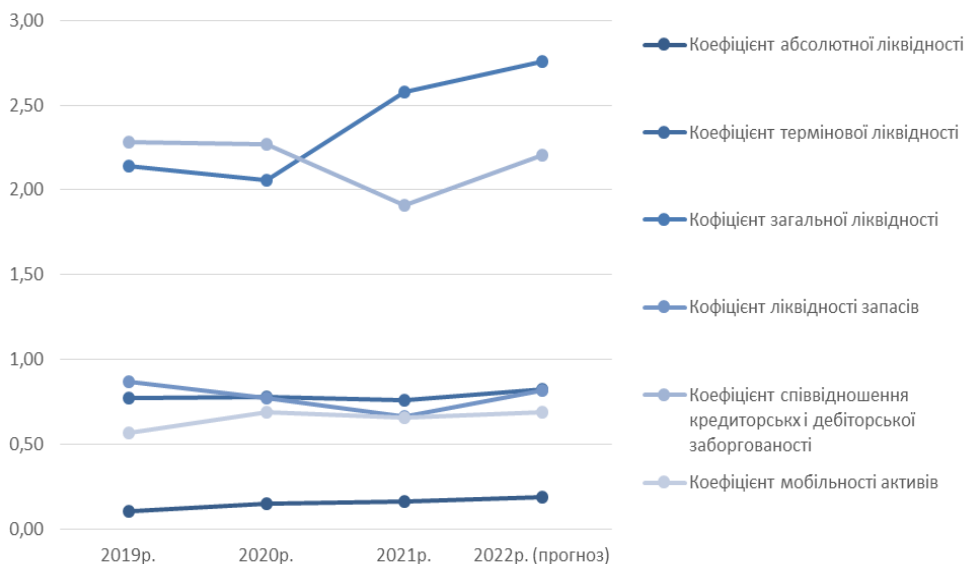


Рисунок 7. Динаміка зміни показників на прогностичний період

Отже, наведений прогноз демонструє, більшість показників мають стабільний темп росту, окрім коефіцієнта загальної ліквідності, на мою думку це є результатом росту коефіцієнту абсолютної ліквідності, таким чином в прогностичному періоді компанія буде мати нормальний фінансовий стан та в перспективі абсолютно ліквідним і платоспроможним підприємством.

## ВИСНОВКИ

Тож яка мета сутності фінансового аналізу – це є визначенням напрямків та обчислення резервів зростання ринкової вартості компанії, основується на дослідження стану і динаміці зміни показників, які охарактеризовують фінансову діяльність підприємства, а також факторів, які безпосередньо впливають на зміну цих показників.

І завдання вважається вирішеним, якщо після проведення фінансового аналізу були розроблені певні рекомендації та шляхи вирішення або покращення ситуації, задля розвитку компанії, які в свою чергу і призводять до досягнення поставленої цілі.

У ході дослідження було реалізовано ряд методів для аналізу фінансового стану та метод екстраполяції трендів для прогнозування ситуації на майбутній період, за допомогою якого ми змогли провести дослідження фінансово-економічної ситуації досить великої корпорації як “Philip Morris International”, було виявлено сильні та слабкі сторони, а також досліджено фінансовий стан. І як висновок нашого дослідження має, що компаніє має достатній рівень ліквідності, платоспроможності та конкуренто спроможності, що і дозволяє компанії конкурувати на ринку тютюнових виробів з такими ж великими корпораціями.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Приходько Н.І. Поняття фінансового стану підприємства: деякі аспекти його визначення / Н. І. Приходько // ФП ФІП РСЕ. – 2010. – Т. 8. – С. 188–190
2. <https://www.pmi.com/>
3. Ковтуненко Ю.В. Методичні основи аналізу фінансового стану промислового підприємства / Ю.В. Ковтуненко // Економіка. Фінанси. Право. – 2016. – № 8/2. – С. 40-41.
4. <https://studfile.net/preview/7292831/page:21>
5. Комариця Л.Л. Аналіз фінансового стану підприємства у кризових умовах економіки / Л. Л. Комариця, Ю.Г. Кіцак – [Електронний ресурс] Режим доступу: [http://www.rusnauka.com/9\\_DN\\_2010/Economics/61392.doc.htm](http://www.rusnauka.com/9_DN_2010/Economics/61392.doc.htm).
6. Коробов М. Я. Фінансово-економічний аналіз підприємств / М.Я. Коробов. – К. : Знання, 2000. – 378 с.

# ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ КОРОТКОСТРОКОВОГО ПРОГНОЗУВАННЯ

Ярошенко В.О.<sup>1</sup>, Бідюк П.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> valiajaroshenko@gmail.com, <sup>2</sup> pbidyuke\_00@ukr.net

**Питання прогнозування супроводжує людство майже весь час його існування. Завжди було важливим спрогнозувати стан хворої людини, знати майбутню поведінку погоди та що є на сьогодні одним з ключових питань, отримати інформацію про майбутні економічні показники на основі поточних даних. Одним із найбільш популярних підходів є прогнозування на основі моделей, побудованих за експериментальними даними, а одним із результатів застосування якого є створення Системи підтримки прийняття рішень (СППР), яка дасть змогу суттєво полегшити аналіз відповідних ризиків. Метою роботи є порівняльний аналіз методів короткострокового прогнозування для отримання можливості подальшого розуміння фінансових ризиків для обраних економічних процесів. Результатом дослідження є СППР, що виконує короткостроковий прогноз. У роботі використано теоретичні та емпіричні методи дослідження.**

**Ключові слова:** короткостроковий прогноз, моделювання, нейронні мережі, фінансові ризики, економічні процеси.

## 1. ВСТУП

З великої кількості навколишніх процесів у суспільстві, фінансові та економічні є одними з найбільш поширених. До них відносять інвестиційні фінансові та процеси глобалізації і державної інтеграції у світову економіку, курси валют, акцій і облігацій, процеси виникнення міжнародних фінансових потоків, формування та використання бюджетів на державному рівні, фінансово-економічну діяльність підприємств, біржові процеси формування цін на активи різного характеру та ще багато інших. Однією із складностей, які виникають в моделюванні таких процесів, є те, що більшість таких процесів мають нелінійний нестационарний характер при будь-якому розвитку економіки, особливо в економіці перехідного періоду. Особливістю перехідного періоду в економіці є те, що в цей період еволюції суспільства відбувається те, що стара система сходить з історичної арени й водночас народжується і утверджується нова. У зв'язку з цим розвиток перехідної економіки має особливий характер, що істотно відрізняється від звичайного, нормального економічного розвитку. В цілому, однією з основних рис перехідної економіки є альтернативність шляхів подальшого розвитку та цілей, що досягаються при цьому.

Людина, як вид, протягом всього свого існування намагалася зробити прогнози та оцінити подальші наслідки різноманітних дій. Задовго до сьогодні людство уже мало попит на отримання достовірної інформації та з плином часу важливість прогнозів у житті людини не зменшилася, але їхні цілі та напрями набули кардинально нового вигляду. Для будь-якого об'єкта, будь то велике підприємство чи, навіть, держава є дуже важливо заздалегідь знати про всі можливі ризики для будь-якого розвитку подій та попередити їх. У

випадку, коли необхідно прийняти важливе рішення, потрібно добре зважити всі можливі наслідки від прийняття того чи іншого рішення та оцінити ризики для кожної можливої ситуації. Загалом, прогнозування відіграє дуже велику роль у роботі з фінансовими процесами. Воно допомагає краще розуміти ці процеси, їх взаємодію з іншими процесами та оцінювати майбутню поведінку.

Фінансово-економічне прогнозування – це процес оцінювання фінансово-економічних прогнозів, побудований на наукових методах пізнання соціально-економічних явищ з використанням усієї сукупності методів, засобів та способів прогностики. Загалом, основною метою прогнозування можна назвати отримання необхідної інформації для розуміння майбутніх ситуацій, передбачення та, за можливості, своєчасного реагування на незадовільний прогноз. Якісне прогнозування дозволяє виявляти тенденції розвитку ринку і здійснювати свою діяльність відповідно до даних тенденцій, займати лідируючу позицію на ринку і успішно розвиватися, тому держава і великі компанії та середній бізнес витрачають великі кошти на прогнозування процесів.

## **2. ОПИС МЕТОДІВ КОРОТКОСТРОКОВОГО ПРОГНОЗУВАННЯ**

За останні десятиліття в світі було створено велику кількість математичних методів, що дозволяють спрогнозувати лінійні, нелінійні нестационарні та інші процеси. Всі методи прогнозування, як правило, розділяють на три широкі класи:

1. Прогнозування, за основу ідею якого взято суб'єктивне судження, оцінка з урахуванням досвіду й інтуїції, поглиблені розуміння предметної області та інші міркування, які мають пряме, або опосередковане відношення до досліджуваного процесу.

2. Методи, що ґрунтуються на застосуванні часового ряду однієї змінної. Тут можливі такі варіанти, як авторегресія (АР), авторегресії з ковзним середнім (АРКС), АРКС з урахуванням тренду та інші подібні моделі.

3. Векторні процеси, або ж часові ряди декількох змінних, слугують як третій клас. Тут враховується залежність ендогенної змінної від деякого числа регресорів у правій частині.

Звичайно, що ці три класи не є взаємовиключними, тому цілком доцільне використання комбінацій двох, або навіть трьох приведених методів [1 – 3]. Одним із способів прогнозування є регресійні моделі, що передбачають оцінювання значення змінної  $Y$  на основі значень змінних  $X$ . Якісно проведений регресійний аналіз дає також оцінку того, наскільки припущена форма залежності відповідає даним спостереження, але тільки в межах діапазону значень наявних незалежних змінних. Це означає, що будь-яка екстраполяція значно залежить від припущень щодо структурної форми регресійної залежності. Не слід обирати залежність лінійності за змінними і лінійність за параметрами лише з міркувань зручності розрахунків, для побудови моделі слід залучати всі доступні знання. Якщо відомо, що залежні змінні не можуть вийти за межі певного діапазону значень, це може стати корисним при виборі моделі – навіть якщо в отриманій вибірці немає значень близьких до таких меж. В цілому, математичні моделі можна будувати у формі різницевого рівняння. Таких як, авторегресія, авторегресія з ковзним середнім тощо.

Розглянемо узагальнений алгоритм (Рис. 1) для побудови математичної моделі на основі статистичних даних. Алгоритм є узагальненим для застосування до систем чи процесів будь-якого типу.

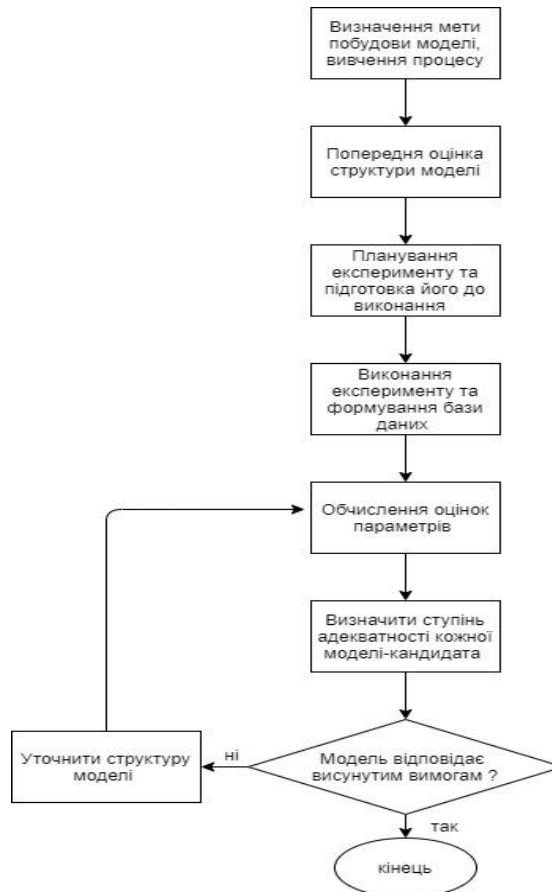


Рисунок 1. Алгоритм побудови математичної моделі на основі статистичних даних

Серед численної кількості методів моделювання, що існують в світі, можна виділити такі: аналітичне моделювання, математичне моделювання, імітаційне моделювання. Перевагою аналітичного методу моделювання є можливість отримання залежності в явному вигляді і застосування до неї методів класичного математичного аналізу. Якщо є можливість побудувати аналітичну модель системи, то завжди віддають перевагу цьому методу моделювання. Зауважимо також, що алгоритм відшукування точного розв'язку задачі може бути реалізований дослідником самостійно, за допомогою спеціального програмного забезпечення або за допомогою чисельних методів. Існують системи, опис яких не піддається опису аналітичними функціями, але процес функціонування їх може бути описаний алгоритмом імітації [4]. Під імітацією розуміють відтворення за допомогою комп'ютерної програми процесу функціонування складної системи в часі. У результаті багатократних прогонів імітаційної моделі дослідник отримує інформацію про властивості реальної системи. При виборі процесів для подальшого моделювання та прогнозування було обрано економічні процеси, оскільки більшість таких процесів є нелінійними та нестационарними. Для побудови прогнозів було обрано два методи: за допомогою нейронних мереж та системи моделювання EViews [5].

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Обраними даними для моделювання та подальшого прогнозування є даними з фінансової біржі. Файл містить дані з 1 січня 2016 року по 6 травня 21 року. Ці дані можна використовувати для прогнозування цін на акції в майбутньому.

Розглянемо графік вхідних даних – Рис. 2.

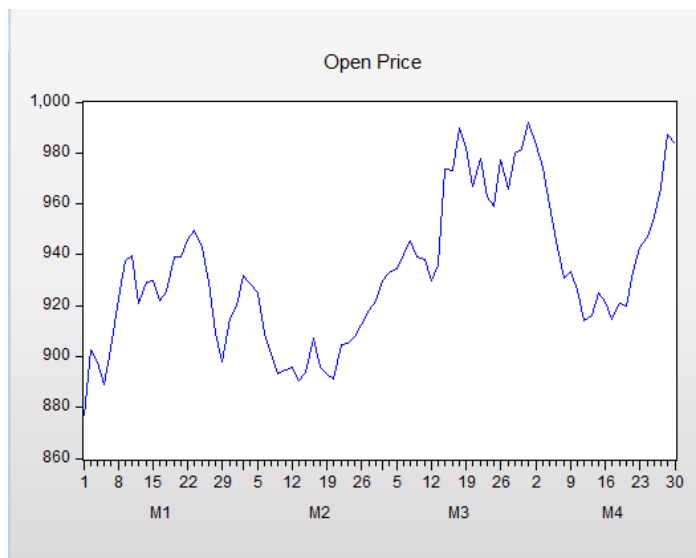


Рисунок 2. Графічне зображення даних

За допомогою системи EViews для вхідних даних були побудовані моделі другого, четвертого та сьомого порядку, але по параметрам вони не давали адекватних результатів, тому було прийняте рішення побудувати модель авторегресії. Після аналізу корелограми було прийнято рішення побудувати модель авторегресії 31-го порядку (АР (31)) та оцінивши залишки цієї моделі модель авторегресії з ковзним середнім 5-го порядку (АРКС (31, 5)). Порівняємо графік вхідних даних та вихідних даних (Рис. 3)

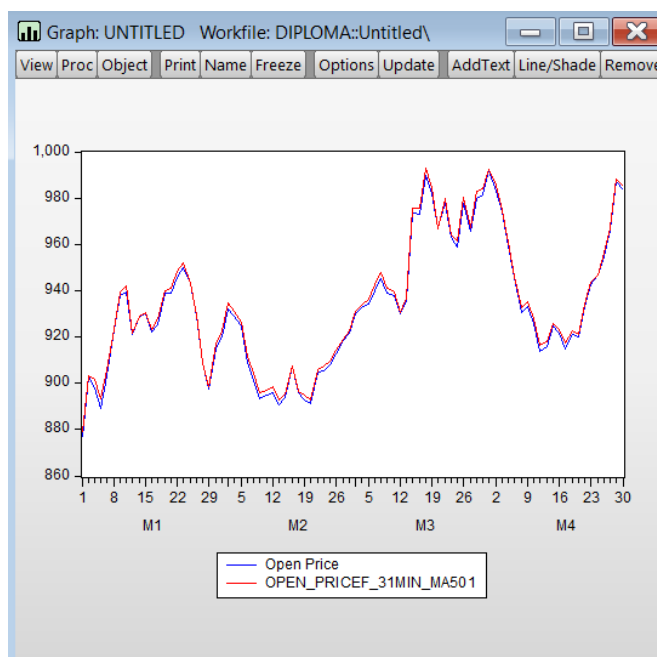


Рисунок 3. Графік вхідних даних та оцінених за моделлю АРКС(31, 5)

Подивившись ще раз на графік вхідних даних можна спробувати побудувати модель за допомогою нелінійних функцій. Оцінивши графік бачимо, що можна спробувати функцію синуса. А також Оскільки ціна відкриття в певний день на біржі на пряму залежить від ціни закриття на попередній день, то було прийняте рішення побудувати прогноз на ціну відкриття за ціною закриття попереднього дня. Порівняємо статистичні характеристики цих

моделей. Будемо порівнювати за наступними критеріями: інформаційний критерій Акаїке (це оцінювач похибки позабіркового прогнозування, і відтак відносної якості статистичних моделей, для заданої вибірки даних). Також зручними для практичного використання є критерій Дарбіна-Уотсона і коефіцієнт детермінації (Табл. 1)

Таблиця 1. Порівняння статистичних параметрів моделей

	Критерій Акаїке	Критерій Дарбіна-Уотсона	Коефіцієнт детермінації
AP (31)	9,92	1,94	0,992
АРКС (31, 5)	9,92	1,95	0,992
sin	9,92	1,93	0,992
Попередній день	9,31	1,99	0,990

У перших трьох моделях критерій Акаїке та коефіцієнт детермінації однакові, тому дивлячись саме на критерій Дарбіна-Уотсона, вибираємо кращу серед них АРКС (31,5). У моделі побудованій на даних попереднього дня найкраще значення критерію Дарбіна-Уотсона. При порівнянні прогнозованих даних по кожній моделі до реальних найближчими даними до реальних є дані прогнозовані по моделі AP (31), також досить близькими є дані прогнозовані за допомогою нелінійної моделі. Також, не дивлячись на гарні статистичні параметри моделі, побудованою за даними попереднього дня, результати прогнозування є найбільш віддаленими від реальних.

За допомогою нейронних мереж (використовуючи інструмент Google Colab), побудовані одношарові та багатошарові нейронні мережі, використовуючи різні види рекурентних нейронів, такі як Simple RNN, LSTM, GRU. Кожен з цих видів нейронів показував себе краще при збільшенні кількості нейронів на один шар. Також були використані згорткові нейронні мережі, що показали себе не досить ефективно, що можна пояснити. Адже згорткові нейронні мережі мають ефективне прикладне застосування в розпізнаванні зображень та відео, рекомендаційних системах та обробці природної мови. Порівняння статистик якості прогнозів різних нейронних мереж подано в Табл. 2.

Таблиця 2. Порівняння результатів застосування нейронних мереж

№	Вид	Шарів	Нейронів	drop	epochs	MSE		MAPE		MAE	
						train	val	train	val	train	val
1	SimpleRNN	1	128	-	20	0.003	0.035	20.249	44.227	0.042	0.121
2	LSTM	1	128	-	20	0.004	0.032	19.768	47.555	0.045	0.109
3	GRU	1	128	-	20	0.002	0.015	15.902	33.849	0.033	0.069
4	GRU	3	256	0.3	40	0.004	0.025	18.364	32.569	0.05	0.099
5	LSTM	2	256	-	30	0.003	0.023	17.514	38.291	0.038	0.09
6	LSTM	2	256	0.1	35	0.011	0.089	32.738	54.326	0.08	0.189
7	RNN + Conv1d	3 + 2	128	0.4	20	0.006	0.016	20.077	32.82	0.062	0.077

Всі розглянуті моделі мають спільний недолік: вони здатні прогнозувати лише на крок вперед. В даних умовах, зміна реального фінансового ряду не визначається лише авторегресійною складовою, а здебільшого зовнішніми чинниками, інформацію про які ми не маємо. В даному випадку: коли ми перевіряємо здатність узагальнення одноточкового прогнозу на великому періоді даних, то однією з найкращих моделей слід прийняти мережу з одним шаром GRU-комірок, що показала найкращий результат за усіма метриками на валідації, та при цьому має доволі мало параметрів. Також достатньо ефективними моделями показали себе GRU з трьома шарами нейронів та RNN + Conv1d.

#### **4. ВИСНОВКИ**

Сьогодні важко назвати область діяльності людини, де не має застосування моделювання. Моделювання виступає як процес поглиблення пізнання, повніше розкриває сутність явищ дійсності, що досліджуються. Сучасні технології моделювання не тільки полегшили і прискорили процес побудови та дослідження моделі, але й значно наблизили сприйняття інформації спеціаліста з моделювання систем і спеціаліста, що працює у галузі, яка моделюється. В якості прогнозованих даних було обрані дані з фінансової біржі. Були розроблені прогнози за допомогою системи EViews та нейронних мереж.

За допомогою системи EViews побудовано п'ять моделей для подальшого прогнозу: AP (31), APKC (31,5), модель на основі нелінійної функції ( $\sin$ ), AP (6) для логарифмованих даних та модель на основі даних попереднього дня. Було обрано робити нелінійну модель зважаючи на графік вхідних даних. Найкраще з цих моделей показала себе APKC (31,5). У моделі побудованій на даних попереднього дня найкраще значення критерію Дарбіна-Уотсона, але якщо дивитись саме на значення прогнозованих даних, то вони є найбільш віддаленими від реальних. У моделі AP (6) для логарифмованих даних найменший критерій Акайке серед усіх значень, тому на даний момент можемо вважати її найкращою.

За допомогою нейронних мереж було побудовано багато прогнозів, варіюючи кількість нейронів, епох та саме типу нейронів. Найкраще себе показали нейронні мережі GRU, як багатошарові так і одношарові, оскільки вони були одразу максимально наближеними до реальних даних. Всі нейронні мережі давали значно кращі результати зі збільшення кількості нейронів та епох.

#### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Бідюк П.І., Половцев О.В. Аналіз та моделювання економічних процесів перехідного періоду. К.: НТУ КПП, 1999. 230 с.
2. Ставицький А. В. Навчально-методичний комплекс з курсів «Прогнозування» та «Фінансове прогнозування». Київ: Центр учб. літ., 2006. 107 с.
3. Половцев О. В. Системний підхід до моделювання, прогнозування та управління фінансово-економічними процесами. Донецьк: Східний видавничий дім, 2009. 286 с.
4. Бідюк П. І., Романенко В. Д., Тимошук О. Л. Аналіз часових рядів: навч. посіб. / ННК «Інститут прикладного системного аналізу» Національний технічний університет України «Київський політехнічний інститут», 2010. 317 с.
5. Молчанов И. Н. Компьютерный практикум по начальному курсу эконометрики (реализация на Eviews): практикум / Ростовский государственный экономический университет. Ростов-н/Д. 2001. 58 с.



# ЗАСТОСУВАННЯ МІКРОСЕРВІСНОГО ТА МІКРО-ФРОНТЕНДНОГО ПІДХОДІВ ДЛЯ ЗАДАЧІ СТВОРЕННЯ ВІРТУАЛЬНОГО КАБІНЕТУ ЛІКАРЯ ТА ПАЦІЄНТА

Безносик О.Ю.<sup>1</sup>, Стефура О.Я.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>beznosyk.oleksandr@lil.kpi.ua [0000-0003-2775-6070], <sup>2</sup>stefura2010@ukr.net

Впродовж останніх років відмічається пришвидшений темп розвитку різноманітних технологій та підходів для реалізації широкого спектру веб-систем. З одного боку це стимулюється інтеграцією все нових і нових систем у сферу веб, з іншої – зростаючою складністю вже існуючих. Нові підходи та технології покликані задовільнити технологічні запити складних систем, спростити їх реалізацію. Разом із тим, однією із найбільш цікавих і водночас складних галузей є галузь охорони здоров'я. Зазвичай, такі системи вирізняються високими технологічними та етичними вимогами, складними внутрішніми процесами, потребою в постійній підтримці та оновленнях. Це створює природний інтерес до застосування сучасних підходів для спрощення та покращення функціонування таких систем. Спроба такої інтеграції представлена в цій роботі.

**Ключові слова:** мікросервісна архітектура, мікрофронтенд, складні системи, охорона здоров'я.

## 1. ВСТУП

Галузь охорони здоров'я є однією із найбільш популярних та таких, що динамічно розвиваються, з точки зору застосування різноманітних програмно-апаратних розробок. Існує дуже велика кількість різноманітних веб-систем, додатків, ресурсів в цій галузі, доступних для користування як професіоналами, так і звичайними користувачами. Не є чимось новим і онлайн системи для взаємодії лікарів, пацієнтів, представників страхових компаній, які, втім, часто є представниками саме класу складних систем, що висувають більш складні технічні вимоги до розробників. Тому природньо, що ця галузь входить до числа доменів, що особливо активно інтегруються із мікросервісним підходом. Існує досить велика кількість присвячених цій проблематиці статей, зокрема [1] або [2] тощо, де автори розглядають переваги та можливі проблеми, пов'язані з імплементацією такого підходу в галузь охорони здоров'я.

Зазвичай, подібні системи характеризуються високими вимогами до охорони даних користувача, вимагають поєднання різноманітних механізмів, але при цьому з точки зору користувача мають бути доволі простими для опанування, інтуїтивно зрозумілими, адже, згідно статистики, більшість активних користувачів подібних систем (саме через їх певну складність) все ще знаходиться в межах 18-34 років [3], хоча ця група людей напевне не є тою, якій найчастіше необхідна допомога лікарів.

На погляд авторів, означений набір проблем може бути вирішений за допомогою декомпаративного підходу. У цій статті пропонується архітектура додатку, що реалізує

віртуальний кабінет лікаря та пацієнта, яка є ефективною з точки зору вирішення вищезначених задач і проблем.

## **2. АРХІТЕКТУРА СИСТЕМИ ВІРТУАЛЬНОГО КАБІНЕТУ ЛІКАРТЯ ТА ПАЦІЄНТА**

### **2.1. Вимоги до системи**

Головним чином, дана система створюється для взаємодії між лікарем і пацієнтом в рамках створення та проведення плану лікування тих чи інших проблем. Відповідно, система має забезпечувати, в першу чергу, комфортний інтерфейс для лікаря для створення плану лікування, його кроків та обмежень, а також не менш зручний інтерфейс для пацієнта для перегляду плану та взаємодії з ним (зокрема, можливість ставити відмітки про виконання тих чи інших приписів, уточнювати план лікування, вносити показники чи результати тих чи інших процедур в рамках кроків лікування). Система, звісно, повинна передбачати й механізм спілкування між лікарем та пацієнтом для швидкого обміну повідомленнями, можливості встановлення швидкого контакту. Але це тільки кореневий функціонал, який представляє основну практичну цінність. Як було сказано раніше, системи із галузі охорони здоров'я зазвичай відрізняються поєднанням різноманітних компонентів, що і обумовлює їхню складність. Тому система, що пропонується, повинна передбачати ще деякий функціонал, зокрема:

- перегляд персональної інформації про пацієнта, зокрема його електронної медичної карти (ЕКМ);
- моніторинг поточного стану пацієнта на основі даних, що ним вносяться;
- автоматичне сповіщення про стан здоров'я на основі внесених показників;
- допоміжні системи супроводу плану лікування, що допомагають як лікарю в менеджменті плану, так і пацієнту в його проходженні.

Безумовно, система також повинна бути готова й до внесення нового функціоналу, до інтеграції із зовнішніми сервісами, і, звісно, передбачати додаткові заходи зі збереження конфіденційності даних користувачів.

В результаті, пацієнт та лікар отримують комфортні інтерфейси, насичені підказками та всім необхідним для ефективного створення та проходження плану лікування. При цьому присутні механізми запобігання когнітивному перевантаженню користувачів: всі дані відображаються лише за потреби, є можливість керування їх відображенням.

Вже із самого опису випливає, що система містить в собі різний функціонал, котрий може (і повинен) бути рознесений в різні її компоненти. Так, можна виділити такі ключові компоненти системи

- безпосередній менеджмент плану лікування;
- механізми доступу до медичної карти та історії пацієнта;
- механізми аналітики поточного стану пацієнта;
- підсистема рекомендацій (лікарю та пацієнту);
- окремий механізм забезпечення конфіденційності даних.

В той же час, в системі мають бути передбачені можливості для майбутнього розширення функціоналу системи, покращення окремих компонентів.

### **2.2. Опис архітектури**

Згідно заданих вище умов, було запропоновано архітектуру системи, що містить набір компонентів та забезпечує взаємодію між ними, як показано на рисунку 1.

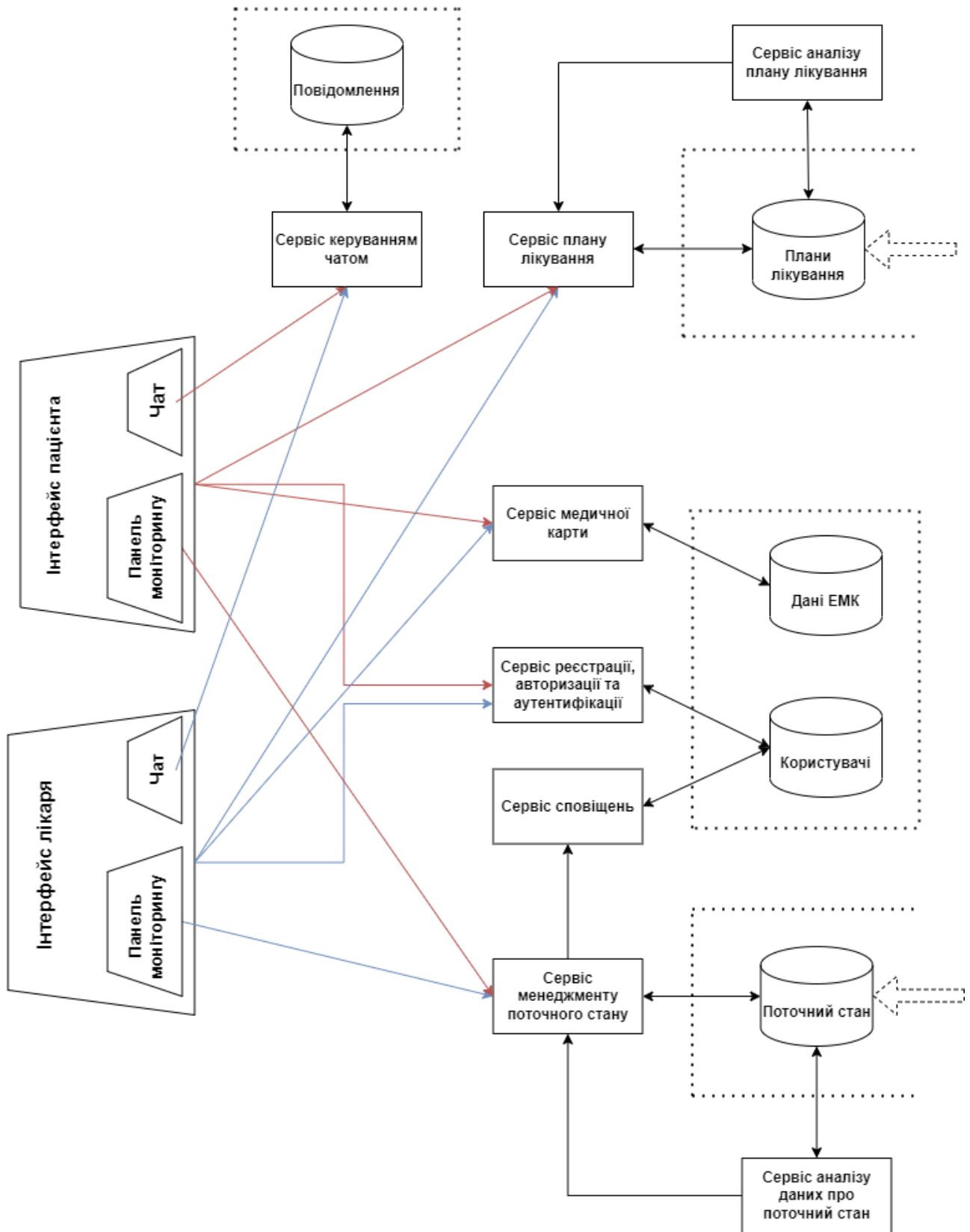


Рисунок 1. Архітектура системи віртуального кабінету лікаря та пацієнта

Компоненти системи виконують наступну роль:

Сервіс реєстрації, авторизації, автентифікації – один із фундаментальних сервісів, відповідає за правильну політику надання доступу до ресурсів, зокрема – до приватних даних.

Сервіс медичної карти – забезпечує надання персональної медичної інформації про пацієнтів.

Сервіс плану лікування – надає механізми менеджменту плану лікування, такі як створення кроків, відмітки про виконання, доступ на перегляд тощо.

Сервіс керування чатом – забезпечує механізм спілкування лікаря\пацієнта.

Сервіс сповіщень – організовує сповіщення пацієнтів\лікарів про різні події (зокрема, про кризові стани) через різні канали, як, наприклад, СМС-повідомлення або електронна пошта.

Окремий інтерес представляють такі “допоміжні” або “аналітичні” сервіси:

Сервіс аналізу даних про поточний стан – постійно аналізує вміст відповідної бази даних, та, за необхідності, видає сигнал із попередженнями або підказками сервісу менеджменту поточного стану. Той, у свою чергу, може ініціювати сповіщення чи оновлення інформаційної панелі інтерфейсу.

Сервіс аналізу плану лікування – аналізує наявні плани лікування та видає рекомендації та підказки.

Варто відмітити деякі технічні особливості системи:

- «чутлива» персональна інформація, яка дозволяє однозначно ідентифікувати користувача системи, зібрана в одному місці (на рисунку – блок з елементами «Користувачі», «Дані ЕМК»). Таким чином, ці таблиці можна винести в окремий простір, навіть в окрему базу даних, яка буде мати додаткові механізми захисту (відповідно, потребуватиме більше ресурсів). Таблиці в інших просторах не мають посилання на конкретних користувачів, відповідно, при викраденні зловмисниками, не дозволяють однозначно ідентифікувати, кому вони належать. Необхідні дані для співставлення планів лікування\повідомлень\поточного стану можуть бути отримані користувачами лише у відповідному просторі з «чутливими даними». Для наочності, приклад відповідної організації таблиць приведено на рисунку 2;

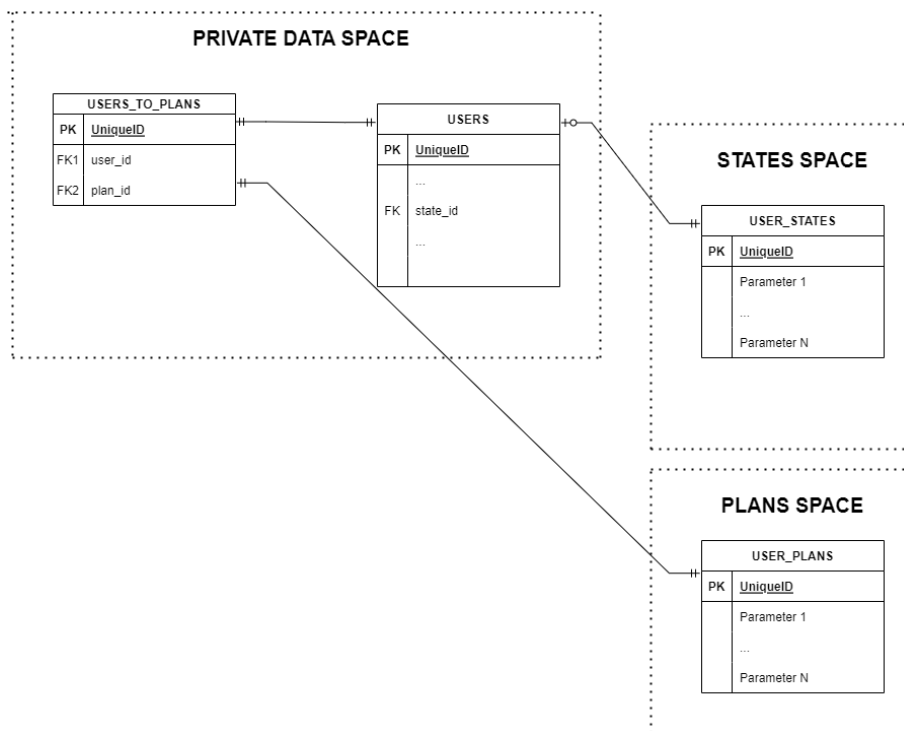


Рисунок 2. Відношення між таблицями з окремих просторів

- на сервісному рівні та\або на рівні бази даних можна обмежити можливість деяких сервісів вносити зміни в ті чи інші таблиці. Наприклад, аналітичні сервіси тільки зчитують дані, щоб далі їх аналізувати; сервіс ЕМК тільки видає дані про картки, не маючи можливість вносити зміни в них; лікар може лише переглядати дані поточного стану, не маючи змоги їх змінити;
- оскільки таблиці, котрі описують поточні стани та плани лікування є, по суті, «анонімними», є можливість відкрити до них доступ на читання стороннім сервісам для виконання іншого роду практичної аналітики або для надання інформації зовнішнім статистичним сервісам.

### 2.3. Декомпозиція інтерфейсу

Підхід із розбиттям на компоненти (мікросервіси) застосовується не тільки у внутрішній складовій системи, але й на стороні інтерфейсу користувача. Такий підхід має назву «мікрофронтендний» [4, 5]. В даній системі він використовується для інтеграції із аналітичною панеллю, чатом, в перспективі – з іншими компонентами.

Це дозволяє не тільки спростити роботу в рамках команд, що працюють над одним проектом, але й відкриває дорогу для використання зовнішніх постачальників послуг, а також для підходу «white label» [6].

Схематично, приклад сторінки плану лікування в такому випадку може виглядати так, як показано на рисунку 3. В ньому панель моніторингу (виділено синім кольором) та чат (виділено оранжевим кольором) імпортуються як окремі залежності ззовні та, за потреби, можуть бути вимкнені користувачами.

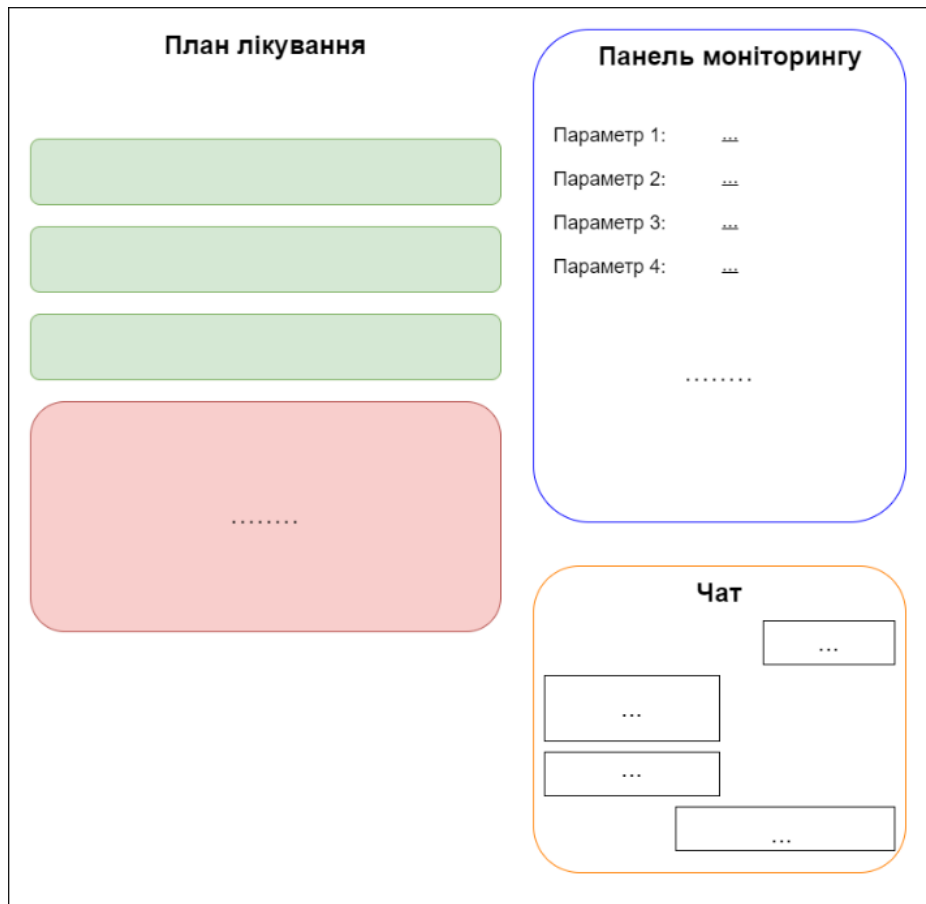


Рисунок 3. Схематичний вигляд інтерфейсу, скомпонованого із різних частин (окремих субінтерфейсів)

### 3. ВИСНОВКИ

Ця робота пропонує модель архітектури системи з акцентом на декомпозицію та слабку зв'язність компонентів не тільки на серверній стороні (що на даний момент доволі розповсюджено), але й на стороні інтерфейсу користувача (що поки не є стандартом). В даній системі така декомпозиція дозволяє, з одного боку, інкапсулювати певні критичні дані і механізми роботи над ними, відмежувавши їх від інших складників системи (з метою забезпечення), а з іншого – вивести інші, не критичні дані в своє окреме місце, що відкриває дорогу для інтеграції зі сторонніми сервісами.

Враховуючи специфіку галузі, така декомпозиція (в тому числі, на стороні інтерфейсу користувача) є досить вдалою, оскільки системи в цій галузі доволі часто є своєрідними "акумуляторами" даних з різних джерел і систем, відповідно, модульний підхід дозволяє без особливих перешкод розширювати систему, налаштовувати канали зв'язку із зовнішніми системами (як-то постачальники медичної інформації, аналітичні системи, "white label" продукти, наприклад, чати чи моніторингові системи).

Звісно, така декомпозиція має свою ціну і складнощі реалізації, однак у даному випадку є виправданою, враховуючи значний функціонал системи, вимоги до безпеки, масштабованості та інтеграції (і, разом із тим, відносно не критичні вимоги до продуктивності та швидкодії, на що така декомпозиція має безпосередній негативний вплив).

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. R. Hill, D. Shadija, M. Rezai, “Enabling Community Health Care with Microservices,” *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, Sept. 2017, pp. 1444-1450, doi: 10.1109/ISPA/IUCC.2017.00220.
2. R. Muhammad, N.F. Ahmad, R. Astari, “Microservices Architecture Design: Proposed for online HealthCare,” *International Journal of Emerging Trends in Engineering Research*, Volume 8, No. 4, Apr. 2020, pp. 1040-1047, doi: 10.30534/ijeter/2020/14842020.
3. Health app usage among adults in the United States as of November 2019, by demographics. URL: <https://www.statista.com/statistics/1193612/health-app-user-demographics-us/> (дата звернення: 17.11.2022).
4. Michael, G. Micro Frontends. Techniques, strategies and recipes for building a modern web app with multiple teams that can ship features independently. URL: <https://micro-frontends.org/> (дата звернення: 18.11.2022).
5. Sam, J. Micro Frontends. URL: <https://martinfowler.com/articles/micro-frontends.html> (дата звернення: 16.11.2022).
6. Amanda, L. White Label Software Explained: Benefits and Examples. URL: <https://growsurf.com/blog/white-label-software> (дата звернення: 18.11.2022).

# ПІДХОДИ ДО ОПТИМІЗАЦІЇ ЗАПИТІВ ДО РЕЛЯЦІЙНИХ БАЗ ДАНИХ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

Булах Б.В.<sup>1</sup>, Загородній Д.О.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>2</sup>dimkazhr@gmail.com

**Використання машинного навчання, а саме нейронних мереж для підвищення ефективності та оптимізації створення плану запиту. Метою роботи є дослідження сучасних СУБД з акцентом на механізми оптимізації запитів. Дослідити можливості використання нейронних мереж для задач оптимізації запитів. Визначити чи ефективно використовувати нейронні мережі для цієї задачі. Розробити додаток що може продемонструвати ефективність використання нейронних мереж для оптимізації запитів.**

**Ключові слова:** реляційні бази даних, нейронні мережі, оптимізація запитів, СУБД, навчання з підкріпленням.

## 1. ВСТУП

Основні сучасні інформаційні технології базуються на концепції зберігання даних, згідно якої усі дані мають бути організовані у бази даних, так щоб вони могли відображати реальний світ та його мінливість, а також бути зручними та задовольняти усім інформаційним потребам користувачів. Для створення і функціонування таких баз даних існують спеціальні програмні комплекси, які називаються системами управління базами даних (СУБД).

Усі підприємства та установи, починаючи від банків, яким необхідно зберігати інформацію про клієнтів, їх рахунки та транзакції, та закінчуючи лікарнями, для яких необхідна база пацієнтів, хвороб та іншого, використовують сучасні бази даних. Усі сучасні інформаційні системи використовують сучасні СУБД для зберігання своїх даних. Наразі дані та їх структура з кожним роком стають все складніші та складніші, кількість користувачів стрімко зростає, а питання оптимізації операцій з даними стає все більш актуальним. Через це і постає проблема оптимізації СУБД. У даній роботі будуть розглянуті способи оптимізації запитів до сучасних СУБД, а саме методи використання нейронних мереж для оптимізації запитів.

Оптимізація запитів — це 1) функція СУБД, яка виконує пошук оптимального плану виконання запиту з усіх можливих для заданого запиту, 2) процес зміни запиту та/або структури БД в цілях зменшення використання розрахункових ресурсів при виконанні запиту. Один і той же результат може бути отриманий СУБД різноманітними способами (планами виконання запиту), які можуть суттєво відрізнятися як за затратами ресурсів, так і за часом виконання. Головним завданням оптимізації запитів є системні скорочення ресурсів, необхідних для виконання запиту, що в кінцевому підсумку надає користувачеві результати за менший проміжок часу, що робить додаток комфортнішим для користувача; надає можливість програмному додатку обслуговувати більше запитів за одиницю часу, оскільки кожен запит займає менше часу, ніж неоптимізовані запити; зменшує навантаження на обладнання і дозволяє серверу працювати більш ефективно.



Наразі є не мало досліджень про можливості оптимізації баз даних з використанням машинного навчання, починаючи від лінійної регресії, яка використовується з метою більш точного розрахунку використання ресурсів кожним з планів, закінчуючи використанням нейронних мереж з дослідженням їх використання для заміни сучасних методів оптимізації плану (грубої сили, динамічне програмування, генетичний алгоритм). У порівнянні з сучасними методами оптимізації запитів, використання нейронних мереж дозволяє пришвидшити роботу пошуку плану запиту до 10 раз.

## 2. ПРОБЛЕМА СУЧАСНИХ АЛГОРИТМІВ ОПТИМІЗАЦІЇ ЗАПИТІВ

SQL – декларативна мова. Виходячи з цього маємо, що користувач вказує СУБД лише операції мають бути проведені з даними. За вибір способу виконання цих операцій відповідає вже сама СУБД.

Для прикладу простий запит:

```
SELECT user FROM users WHERE year > 1980;
```

може бути виконаний двома способами: читання усіх записів з таблиці users та перевірка кожної з них на виконання умови, або використовувати індекс по полю year. У другому випадку ми не проглядаємо зайві записи, але витрачаємо більше часу на обробку одного запису через операції з індексами.

Розглянемо складніший запит:

```
SELECT messages.content FROM messages, users WHERE messages.user_id = users.id;
```

Це з'єднання можливо виконати вже трьома різними способами:

- вкладений цикл проглядає усі можливі пари записів з двох таблиць та перевіряє для кожної пари виконання умови;
- злиття відсортує обидві таблиці по полям id та user\_id відповідно, а далі використає метод двох вказівників для пошуку всіх пар записів, які задовольняють умові. Цей метод аналогічний методу сортування злиттям;
- хешування будує хеш-таблицю по полю найменшої таблиці. Хеш-таблиця дозволяє для кожного запису з однієї таблиці знайти запис з відповідністю.

Також якщо в запиті необхідно виконати більше однієї операції з'єднання таблиць, то також можна виконувати їх у різному порядку. Отриманий результат буде деревом виконання запиту (рис.1), що і являє собою план виконання запиту.

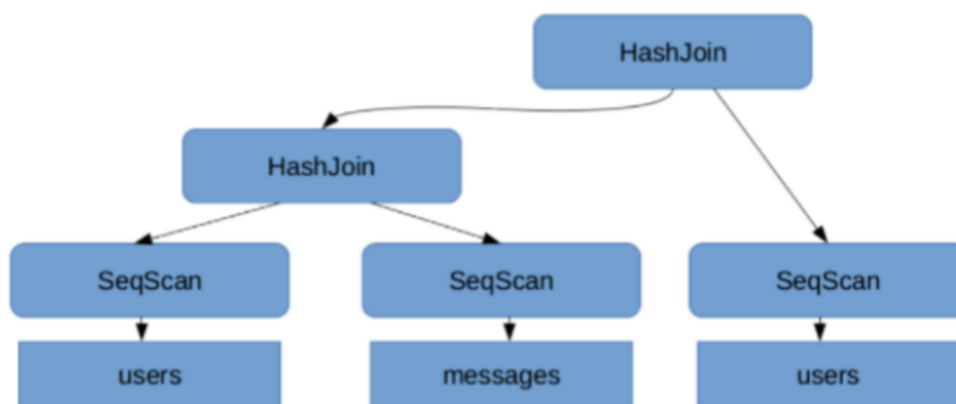


Рисунок 1. Дерево виконання запиту

Час виконання різних планів одного й того ж самого запиту може суттєво відрізнятись на багато порядків. Тому правильний вибір плану запиту сильно впливає на роботу СУБД.

На прикладі СУБД PostgreSQL розберемося як вибирається план запиту зараз. Процес пошуку оптимального плану можна розділити на дві частини:

- по-перше, необхідно вміти оцінювати вартість будь-якого плану – кількість ресурсів, що необхідні для його виконання. У випадку, коли на сервері виконуються інші задачі та процеси, оцінений час є прямо пропорційний кількості витрачених на нього ресурсів. Тому можна вважати, що вартість плану – це його час виконання в деяких умовних одиницях;
- по-друге, необхідно вибрати план з мінімальною оцінкою вартості.

Легко показати, що кількість планів зростає експоненційно зі збільшенням складності запиту, тому не можна просто перебрати усі плани, оцінити вартість кожного та обрати найбільш дешевий. Для пошуку оптимального плану використовуються більш складні алгоритми дискретної оптимізації: динамічне програмування по підмножинам для простих запитів та генетичний алгоритм для складних.

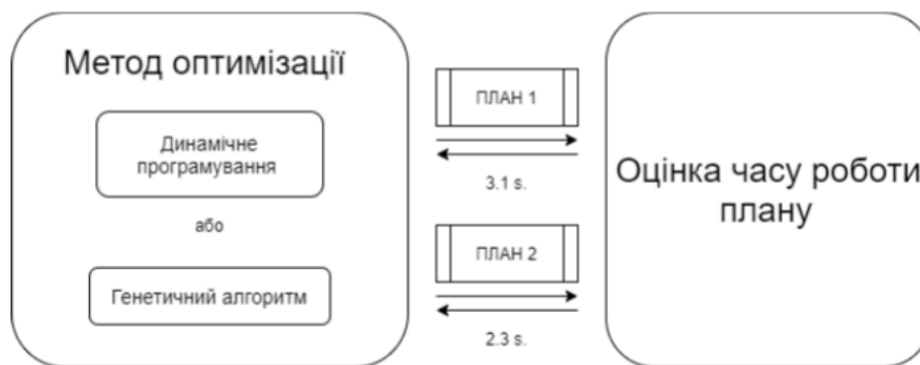


Рисунок 2. Схема процесу пошуку оптимального плану

Основною проблемою даного підходу є те що СУБД ніяк не враховує попередній досвід, через що, наприклад, деякі запити, які часто виконуються можуть мати не оптимальний план запиту, і СУБД все одно буде використовувати його. Проблема є з двома етапами побудови запиту, як кожен раз буде проводитись помилкова оцінка часу плану роботи, так і обраний метод оптимізації може виявитись не оптимальним. Саме цю проблему можливо вирішити завдяки використанню машинного навчання, в нашому випадку ми будемо розглядати оптимізацію побудови плану з використанням навчання з підкріпленням.

### 3. ВИКОРИСТАННЯ НАВЧАННЯ З ПІДКРІПЛЕННЯМ ДЛЯ ОПТИМІЗАЦІЇ ЗАПИТІВ

Навчання з підкріпленням – область машинного навчання, яка вивчає те як інтелектуальні агенти повинні приймати рішення у середовищі з ціллю максимізації отриманої нагороди.

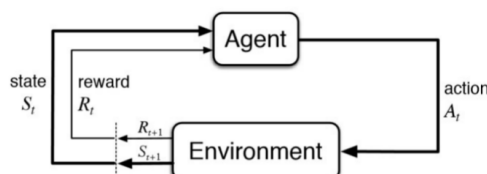


Рисунок 3. Схема навчання з підкріпленням

Постановка задачі: замість вирішення проблеми впорядкування з'єднань з використанням методів динамічного програмування або генетичних алгоритмів, ми

формуємо проблему як процес прийняття рішень Маркова (MDP) та вирішуємо її за допомогою використання навчання з підкріпленням. Тобто ми маємо визначити певне середовище, що може надавати агенту свій стан, далі в залежності від дії обраної агентом, змінювати стан, та очікувати нової дії від агента. Виконуємо цей процес ітеративно доки не досягаємо деякого кінцевого стану, коли не залишиться доступних дій – отримання плану впорядкування з'єднань, після цього агент завершує серію та отримує нагороду в залежності від прийнятих рішень. Ціль агента – максимізувати нагороду, спираючись на досвід.

Щоб використовувати навчання з підкріпленням для вирішення задачі про порядок з'єднань, ми збираємося використати наступні елементи:

- кожен стан буде представляти собою бінарне дерево з'єднань. Далі ми перетворюємо це дерево у вектор стану та додаємо додаткову інформацію, таку як предикат з'єднання та предикат вибору;

- кожна дія являє собою об'єднання двох піддерев у одне дерево;

- епізод закінчується, коли ми приєднали усі відносини;

- винагорода буде розраховуватись на основі оцінки вартості моделі з результату порядку з'єднань.

Можемо побудувати схему додатку що буде використовувати навчання з підкріпленням для пошуку оптимального порядку з'єднань (рис. 4).



Рисунок 4. Схема використання навчання з підкріпленням для задачі створення оптимального порядку з'єднань

#### 4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Поглянемо на результати навчання. Маємо набір тестових запитів з різною кількістю з'єднань. Для прикладу візьмемо наступний набір запитів з великою кількістю з'єднань: 19a.sql, 19b.sql, 19c.sql. Поставимо загальну кількість епізодів на 900. Результати будуть подані у вигляді графіків для кожного епізоду окремо, де по осі x епізоди, а по осі y вартість побудованого запиту. Поглянемо на результати навчання:

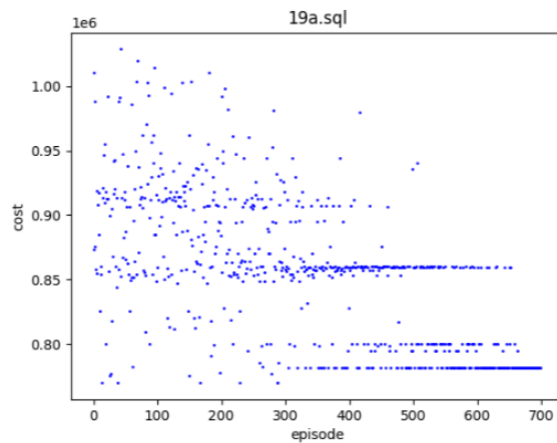


Рисунок 5. Результати навчання для запиту 19a.sql

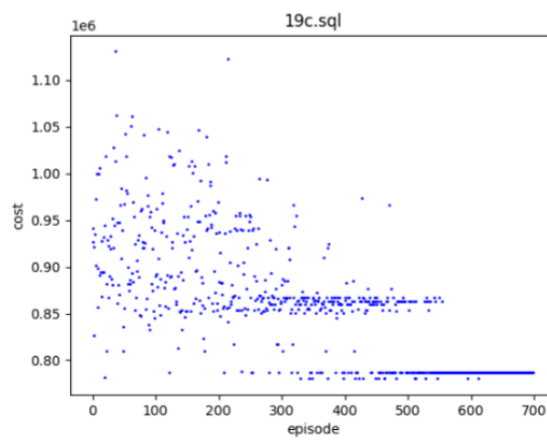


Рисунок 6. Результати навчання для запиту 19c.sql

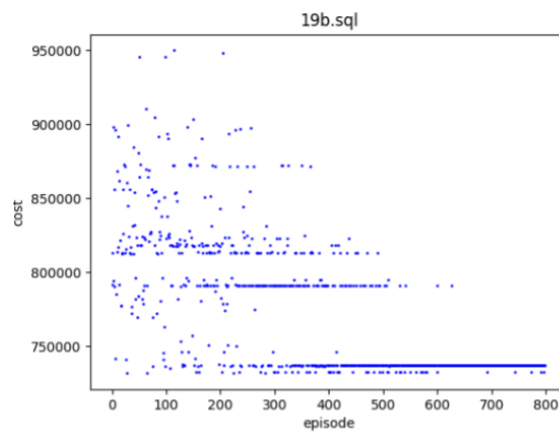


Рисунок 7. Результати навчання для запиту 19b.sql

Складемо таблицю у якій порівняємо отримані результати з вартістю запити, який складає СУБД PostgreSQL:

Таблиця 1. Порівняння результатів навчання з роботою СУБД.

Запит	Вартість за результатами навчання	Результат СУБД PostgreSQL
19a.sql	751255	767747
19b.sql	730011	731337
19c.sql	765125	766012

Як бачимо був отриманий хоч і не великий, але все ж приріст ефективності. Збільшити його можна було б якби агент вибирав не лише порядок з'єднання, але і метод приєднання. Провівши тестування для різних груп запитів приходимо до середніх результатів навчання агента:

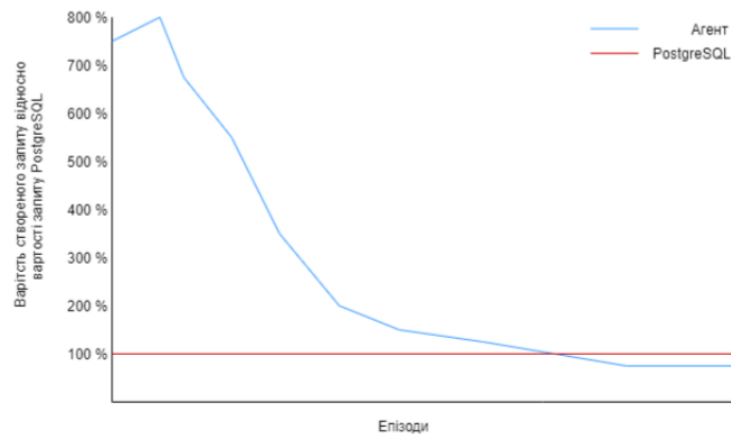


Рисунок 8. Результати тестування

Отже робимо висновок, що використання навчання з підкріпленням є ефективним для вирішення проблеми пошуку порядку з'єднань для запитів з великою кількістю з'єднань та великих БД.

## 5. ВИСНОВКИ

Отримані результати тестування показують, що використання нейромереж на запитах з великою кількістю з'єднань може бути ефективним. Було виявлено що оптимізатор запитів, що розроблений на основі навчання з підкріпленням, після навчання, ефективно вирішує проблему оптимізації порядку з'єднань у плані запиту. Для ще більшого підвищення ефективності оптимізації даним методом можна створити оптимізатор, що не лише створює порядок з'єднань у запиті, але і обирає метод з'єднань. Але є і недоліки розробленого рішення, а саме те що для кожної бази даних необхідно використовувати індивідуального агента, якого необхідно навчати окремо, а також при змінах у схемі бази даних агента необхідно перевчити. Отже робимо висновок що використання нейронних мереж для оптимізації запитів є ефективним рішенням, що дозволить пришвидшити час створення запиту та його виконання, але його доцільно використовувати для баз даних з чітко визначеною схемою. Використовуючи навчання з підкріпленням для вирішення цієї проблеми можна обійти у швидкодії сучасні динамічні алгоритми що для цього створені.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Система управління базами даних. URL:  
[https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0\\_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0)
2. План виконання запиту. URL:  
[https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%BD\\_%D0%B2%D1%8B%D0%BF%D0%BE%D0%BB%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F\\_%D0%B7%D0%B0%D0%BF%D1%80%D0%BE%D1%81%D0%B0](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%BD_%D0%B2%D1%8B%D0%BF%D0%BE%D0%BB%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B7%D0%B0%D0%BF%D1%80%D0%BE%D1%81%D0%B0)
3. Методи оптимізації виконання запитів у реляційних СУБД. URL:  
[http://citforum.ru/database/articles/art\\_26.shtml](http://citforum.ru/database/articles/art_26.shtml)
4. Query Optimizer in PostgreSQL. URL: <https://bitnine.net/blog-postgresql/query-optimizer-in-postgresql/>
5. Reinforcement learning. URL: [https://en.wikipedia.org/wiki/Reinforcement\\_learning](https://en.wikipedia.org/wiki/Reinforcement_learning)
6. Using Reinforcement Learning to Produce Better Join Ordering Strategy. URL:  
<https://towardsdatascience.com/using-reinforcement-learning-to-produce-better-join-ordering-strategy-2fd2761ebf3a>
7. SQL Query Optimization Meets Deep Reinforcement Learning. URL:  
<https://rise.cs.berkeley.edu/blog/sql-query-optimization-meets-deep-reinforcement-learning/>
8. Join-order-benchmark. URL: <https://github.com/gregrahn/join-order-benchmark>

# МОДЕЛІ НЕЧІТКОГО ПРОГНОЗУВАННЯ В ВИМІРЮВАННЯХ

Грищенко О.Ю.<sup>1</sup>, Рогоза В.С.<sup>2</sup>

КПІ ім. Ігоря Сікорського, Київ, Україна

<sup>1</sup> helengrischenko00@gmail.com [https://orcid.org/0000-0001-6888-8665],

<sup>2</sup> rosvetnik@gmail.com

## 1. ВСТУП

Проблема прогнозування стану будь-якого процесу є складною та багатогранною, як і складність вироблення єдиної універсальної технології оцінки та контролю прогнозування, оскільки, як відомо, точність оцінок є обернено пропорційною складності об'єктів, характеристики яких оцінюються [5].

Складність системи прогнозування є її внутрішньою характеристикою. Різні методи штучного інтелекту (AI) були використані для реалізації системи прогнозування [3]. Моделі прогнозування на основі ШІ можна класифікувати на чотири групи: моделі на основі нейронних мереж, нечіткої логіки, генетичного алгоритму та експертних систем.

Нечіткі обчислення є основою нового наукового напрямку, названого «м'які обчислення» або «обчислювальний інтелект», який сформувався в останні 25 років, і включає також нейрообчислення, еволюційні і генетичні алгоритми, міркування на основі свідочств, мережі довіри та інші [3].

Вимірювання є найбільш фундаментальним методом науки в отриманні знань та управлінні системами. У складних інженерних і наукових системах частина інформації може надходити від датчиків, а інша частина може надаватися експертами. Через свою природу ці джерела відрізняються надійністю та невизначеністю отриманої інформації [2]. На невизначеність також можуть впливати характеристики процедур та інструментів, що застосовуються для отримання та обробки інформації [2].

Нечіткі моделі перетворюють суб'єктивне розуміння та досвід процесів у математично доступні цифри та правила для створення систем із певним ступенем невизначеності. Використання нечіткої логіки в експериментальній розробці програмного забезпечення для моделювання різних аспектів процесу еволюції програмного забезпечення все більше отримує визнання дослідницької спільноти [1].

## 2. КОНЦЕПЦІЯ НЕЧІТКИХ ОБЧИСЛЕНЬ

Поняття нечіткі обчислення або м'які обчислення було введено основоположником нечіткої логіки Л. Заде на семінарі в 1994 році, як консорціум обчислювальних методологій, які колективно забезпечують основи для розуміння, конструювання і розвитку інтелектуальних систем, зокрема, систем інтелектуального аналізу даних. Основою відмінності м'яких обчислень від традиційних жорстких обчислень - пристосованість до роботи з неточними, невизначеними або частково істинними даними, що виражається в «допустимому ставленні до неточності, невизначеності і часткової істинності для досягнення зручності маніпулювання, низької вартості рішення і кращої згоди з реальністю».

М'які обчислення не гарантують, що знайдене рішення буде оптимальним або буде досягнутий глобальний екстремум за прийнятний час. Проте вони можуть застосовуватися для пошуку допустимого рішення задачі за «достатньо короткий час». В рамках м'яких обчислень кожна з методологій має відмінності у використанні. Зокрема, нечітка логіка

працює з неточністю, зернистою структурою (гранульованою) інформації, наближеними міркуваннями і обчисленнями із словами. Слід відмітити, що поняття нечіткості цілком узгоджується з нашими інтуїтивними уявленнями про навколишній світ. Велика частина понять, які ми використовуємо, за своєю природою нечіткі і розмиті і спроба загнати їх в рамки загальної логіки приводить до неприпустимих спотворень.

Незважаючи на зовнішню простоту і природність базових понять нечітких обчислень, знадобилося більше п'яти років, щоб побудувати і довести комплекс постулатів і теорем, які роблять логіку логікою, а алгебру - алгеброю. Паралельно з розробкою теоретичних основ нової науки опрацьовувалися різні можливості її практичного застосування. І в 1973 р. ці зусилля увінчалися успіхом - вдалося показати, що нечіткі обчислення можуть бути покладені в основу нового покоління інтелектуальних систем управління. Практично відразу після виходу в світ фундаментальних робіт по нечітким обчисленням одна невелика фірма з Данії застосувала викладені в них принципи для удосконалення системи управління складним виробничим процесом. Результат, що називається, перевершив всі очікування - через чотири роки прибутки від впровадження нової системи обчислювалися сотнями тисяч доларів.

### **3. ОСНОВНІ ЕЛЕМЕНТИ СИСТЕМИ НЕЧІТКОГО ПРОГНОЗУВАННЯ**

Основними елементами керованих даними систем нечіткого прогнозування є [1]:

1. Модуль фазифікації (fuzzification): змінює чіткі вхідні дані на нечіткі значення за допомогою функції належності. У побудові керованої даними системи нечіткого висновку цей процес фазифікації виконується нечітким розподілом чітких значень змінних набору даних.
2. Системи нечіткого висновку (FIS): виконують відображення вводу-виводу лінгвістично переданої інформації у формі правил. Різні потенційні алгоритми створення створюють ці правила, вивчаючи дані.
3. Модуль спрощення: містить невелику кількість правил і зберігає узгоджену базу правил. Хоча цей модуль є необов'язковим у процесі нечіткого висновку, структура GUAJE забезпечує покращену читабельність створених нечітких правил за допомогою цього модуля.
4. Дефазифікація (Defuzzification): транскрибує нечіткі результати назад у чіткі значення.

### **4. ФОРМУЛЮВАННЯ МАТЕМАТИЧНОЇ ЗАДАЧІ**

Загальна мета процедури вимірювання полягає в тому, щоб знайти оцінки вектора параметрів  $X$  деякої доступної моделі системи. Також відомо, що доступна модель передбачення щодо цих значень параметрів. Модель передбачає значення деяких функцій  $f(X)$  значень параметрів. Функції  $f()$  вводяться для узагальнення випадку, коли прогноз можна зробити щодо деякого зв'язку кількох параметрів. Прогноз зазвичай виражається як модель типу «функція  $f$  вектора параметрів  $X$  дорівнює  $R$ », де  $R$  є обмежувальним відношенням. Можна побачити, що це передбачення належить до класу узагальненого обмеження, визначеного Л.Заде [4]. Ця модель може бути формалізована як нечітка модель і описана набором функцій приналежності  $\mu(f(X))$ . Традиційний спосіб знаходження оцінок параметрів моделі на основі вимірювань проходив би через його визначення як задачу математичного програмування та пошук оцінок параметра  $X^*$  шляхом максимізації деякого критерію:



$$X^* = \underset{x}{\operatorname{argmax}} F(Y_1, Y_2, \dots, Y_n, X),$$

де  $F(\cdot)$  – функціонал, форма якого визначається методами оцінювання,  $Y_i$  ( $i=1, n$ ) – набір  $m$  вимірювань  $i$ -ї змінної.

Розглянемо інформацію моделі прогнозу як нечітке обмеження для вектора параметрів  $X$  і задане набором функцій приналежності  $\mu_i(f_i(X))$ ,  $i = 1, m$ , де  $m$  – кількість прогнозів, зроблених експертами. Кожна функція приналежності описує одну модель передбачення, як правило, щодо можливого значення результату вимірювання або лінійної комбінації (наприклад, середнього або середньозваженого) результатів вимірювання [2].

## **5. ПЕРЕВАГИ МОДЕЛЕЙ НЕЧІТКОГО ПРОГНОЗУВАННЯ**

Представлена теорія дозволяє виявити наступні переваги моделей нечіткого прогнозування в порівнянні з іншими:

- можливість оперувати нечіткими вхідними даними: наприклад, значеннями, що безперервно змінюються в часі (динамічні задачі), значеннями, які неможливо задати однозначно (результати статистичних опитів, рекламні компанії і так далі);
- можливість нечіткої формалізації критеріїв оцінки і порівняння: оперування критеріями «більшість», «можливо», «переважно» і т.д.;
- можливість проведення якісних оцінок як вхідних даних, так і вихідних результатів: маємо можливість оперувати не лише значеннями даних, але і їх мірою достовірності і її розподілом;
- можливість проведення швидкого моделювання складних динамічних систем і їх порівняльного аналізу із заданою мірою точності: оперуючи принципами поведінки системи, описаними нечіткими методами, ми по-перше, не витрачаєте багато часу на з'ясування точних значень змінних і складання рівнянь, що їх описують, по-друге, можна оцінити різні варіанти вихідних значень.

## **6. ЗАСТОСУВАННЯ НЕЧІТКИХ СИСТЕМ**

Що стосується вітчизняного ринку комерційних систем на основі нечіткої логіки, то його формування почалося в середині 1995 року. Найбільш популярні в замовників наступні пакети [6]:

- CubiCalc 2.0 RTC — одна з найбільш могутніх комерційних експертних систем на основі нечіткої логіки, що дозволяє створювати власні прикладні експертні системи ;
- CubiQuick — дешева <університетська> версія пакета CubiCalc ;
- RuleMaker — програма автоматичного витягу нечітких правил із вхідних даних ;
- FuziCalc — електронна таблиця з нечіткими полями, що дозволяє робити швидкі оцінки при неточно відомих даних без нагромодження похибки;
- OWL — пакет, що містить вихідні тексти усіх відомих видів нейронних мереж, нечіткої асоціативної пам'яті тощо.

Сьогодні елементи нечіткої логіки можна знайти в десятках промислових виробів — від систем керування електропоїздами і бойовими вертольотами до пилососів і пральних машин. Рекламні кампанії багатьох фірм (переважно японських) підносять успіхи у використанні нечіткої логіки як особливу конкурентну перевагу. Без застосування нечіткої логіки немислимі сучасні ситуаційні центри керівників західних країн, у яких приймаються ключові політичні рішення і моделюються всілякі кризові ситуації. Одним із вражаючих

прикладів масштабного застосування нечіткої логіки стало комплексне моделювання системи охорони здоров'я і соціального забезпечення Великої Британії (National Health Service — NHS), що вперше дозволило точно оцінити й оптимізувати витрати на соціальні нестатки.

## 7. ВИСНОВОК

Вимірювання як базова концепція наукового пізнання – це процес формулювання та модифікації моделей вимірюваних систем і середовища на основі отриманих результатів спостережень. При вимірюванні використовуються різні типи моделей. Їх взаємозв'язок і місце в ієрархії моделей, представлених Л.Заде в його теорії узагальненої визначеності [4], а також переваги, які ми можемо отримати від їх застосування, вимагають подальшого вивчення.

Моделі нечіткого прогнозування мають ряд переваг, основною з яких є можливість проведення якісних оцінок вхідних даних та отриманих результатів.

Проблема використання моделі нечіткого прогнозування для підвищення якості вимірювальних процедур і точності отриманих оцінок може бути вирішена шляхом злиття інформації з різних джерел, які характеризуються різним ступенем невизначеності.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Rinkaj Goyal, Pravin Chandra, Yogesh Singh, "Fuzzy inferencing to identify degree of interaction in the development of fault prediction models", pp. 93-102, 2017

2. Leon Reznik and Vladik Kreinovich, "Fuzzy Prediction Models in Measurement", pp. 851 – 862, 2008

3. Chu S.C. And Kim H.S, "Automatic knowledge generation from the stock market data", Proceedings of 93 Korea Japan joint conference on expert systems, pp. 193-208, 1993

4. Zadeh L. "Granular computing as a basis for a computational theory of perceptions", Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, pp. 564 – 565, 12-17 May 2002, volume 1

5. А. Ковальчук, О. Суха, Л. Фабрі, "ВИКОРИСТАННЯ МЕТОДІВ НЕЧІТКОГО ВИВЕДЕННЯ В СИСТЕМАХ ПРОГНОЗУВАННЯ СТАНІВ ПРИРОДНИЧИХ ПРОЦЕСІВ", 2009

6. Нечітка логіка. URL:  
[https://uk.wikipedia.org/wiki/%D0%9D%D0%B5%D1%87%D1%96%D1%82%D0%BA%D0%B0\\_%D0%BB%D0%BE%D0%B3%D1%96%D0%BA%D0%B0#%D0%9D%D0%B5%D1%87%D1%96%D1%82%D0%BA%D0%B0\\_%D0%BB%D0%BE%D0%B3%D1%96%D0%BA%D0%B0\\_%D0%B2\\_Matlab](https://uk.wikipedia.org/wiki/%D0%9D%D0%B5%D1%87%D1%96%D1%82%D0%BA%D0%B0_%D0%BB%D0%BE%D0%B3%D1%96%D0%BA%D0%B0#%D0%9D%D0%B5%D1%87%D1%96%D1%82%D0%BA%D0%B0_%D0%BB%D0%BE%D0%B3%D1%96%D0%BA%D0%B0_%D0%B2_Matlab)

# КОМБІНОВАНА АРХІТЕКТУРА ДЛЯ МОБІЛЬНИХ ЗАСТОСУНКІВ

Зарічний Я.С.<sup>1</sup>, Гіоргізова-Гай В. Ш.<sup>2</sup>, Яременко В. С.<sup>3</sup>

КПІ ім. Ігоря Сікорського, Київ, Україна

<sup>1</sup> yarik120700@gmail.com [https://orcid.org/0000-0002-7596-5857], <sup>2</sup> high.victoria@iit.kpi.ua [https://orcid.org/0000-0001-6224-3532], <sup>3</sup> yaremenko.v.s@gmail.com [https://orcid.org/0000-0001-8557-6938]

## 1. ВСТУП

Розробка масштабованих і стійких додатків є складною технічною задачею, в якій необхідно врахувати не лише базові вимоги до продукту, а й закласти можливості до розширення функціоналу, масштабованості тощо. Однак, коли справа доходить до мобільних програм або платформ, завжди існувала проблема з чистою архітектурою [1]. У випадку з Android це може бути тому, що Google не підтримує та навіть не рекомендує жодної конкретної архітектури. Apple, з іншого боку, запропонувала архітектуру MVC для UIKit, але ця пропозиція викликала багато суперечок, і багато експертів стверджували, що це не дуже гарне рішення. В роботі було досліджено два типу архітектури: односпрямовані, двоспрямовані та загальні принципи чистого програмування та побудови архітектур. До односпрямованих належать UDF побідні архітектури, до двоспрямованих MVX подібні та Чиста архітектура. Базуючись на перевагах та недоліків кожного підходу було запропоновано побудувати комбіновану архітектуру, яка надає можливість комбінувати переваги кожного типу архітектур та слідувати загальним принципам побудови чистої архітектури.

## 2. ОГЛЯД ІСНУЮЧИХ АРХІТЕКТУРНИХ ШАБЛОНІВ

MVC - схема поділу даних програми, інтерфейсу користувача і керуючої логіки на три окремих компоненти: модель, відображення та контролер - таким чином, що модифікація кожного компонента може здійснюватися незалежно. Перші приклади для розробників для iPhone SDK мали у собі приклади цього шаблону: клас UIView працював як відображення, AppDelegate як контролер усього застосунку, а прикладів щодо моделей не було. В цей час, перші архітектурні патерни для ОС Android були дуже схожими, відображення будувалось за допомогою XML файлів, а клас Activity був у ролі контролера. Також і в цьому випадку теж існувала проблема - відсутність прикладів класів моделей. Ця проблема мала своє відображення на багатьох мобільних застосунках, оскільки, призводить до збільшення класів контролерів [2].

MVP - шаблон проектування, похідний від шаблону MVC, який використовується в основному для побудови інтерфейсу користувача [3]. В цьому шаблоні роль Презентатор виконує роль Контролера у MVC. Але, нажаль, такий архітектурний шаблон теж має тіж самі проблеми, як і MVC: розробникам не завжди зрозуміло, яка логіка має відноситися до Презентатора або до Моделі. Також слід помітити, що доволі часто сутність Моделей теж відсутня у даному шаблоні та склеюється разом с Презентором. Треба відмітити, що порівнюючи MVC з MPV у мобільній розробці все ж таки MVP має певні поліпшення, тому що, частіше за все у MVC роль Контролеру виконувало саме Відображення. Це призводило до ще більших Контролерів така проблема має назву Massive View Controller.

MVVM – двонаправлений шаблон проектування, який походить як логічний розвиток MVC та MVP. Особливість даного шаблону це використання реактивної парадигми

програмування [4]. В даному шаблоні використовується механізм «зв'язування», який будується на парадигмі реактивного програмування, тому оновлення моделі автоматично призводить до оновлення відображення.

Односпрямований потік даних — це техніка, яка в основному зустрічається у функціональному реактивному програмуванні. Він також відомий як односторонній потік даних, що означає, що дані мають один і лише один шлях для передачі в інші частини програми. По суті, це означає, що дочірні компоненти не можуть оновлювати дані, які надходять від батьківського компонента [5].

Одностороннє зв'язування даних забезпечує ряд наступних ключових переваг.

- Полегшується відладка програми, оскільки розробнику відомо, звідки надходять дані.
- Зменшується схильність до помилок, оскільки розробник має більше контролю над даними.

Одним з головних недоліків UDF є те, що він може не знадобитися у проекті (принаймні, поки що). Отже, як дізнатися, що додаток справді потребуватиме UDF?

- Є великий стан програми, який потрібен у багатьох частинах програми.
- Стан програми часто оновлюється.
- Стан програми складно оновити.
- Програма має кодову базу середнього та великого розміру, над якою працює багато людей.
- Повинно бути видно, як стан оновлюється з часом.

Зазвичай компонент відображення у мобільному застосунку є функцією даних, які йому надає сервер або будь-яке інше джерело інформації. Математичні функції є чистими функціями, а це означає, що певний вхід завжди відображається на той самий вихід.

У MVX архітектурі через свою імперативність не можливо сказати з високим ступенем упевненості, що враховуючи однакові дані з сервера, програма буде показувати те саме відображення. Це пов'язано з тим, що потенційно деякі частини відображення можуть бути оновлені без наявності контролю над ними, такі ситуації зазвичай мають назву Сайд-ефекти.

Вирішити ці проблеми у складних інтерфейсних програмах дозволяє концепція односпрямованого потоку даних.

Поширеною проблемою у застосуванні MVX і чистої архітектури є те, що неоптимальна початкова точка (незавершені шаблони, відсутність еталонних реалізацій) призвела до різноманітних втілень із іноді суперечливими, а іноді навіть неправильними інтерпретаціями.

### **3. ЗАГАЛЬНІ ПРИНЦИПИ ПРО ПОБУДОВІ АРХІТЕКТУРИ МОБІЛЬНОГО ДОДАТКА**

#### **Чиста архітектура**

Clean Architecture пропонує використовувати чотири шари в стилі цибулі: об'єкти, варіанти використання, інтерфейсні адаптери, фреймворки та драйвери. [6] (Рис. 1)

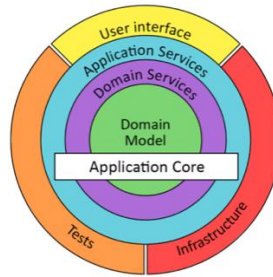


Рисунок 1. Clean Architecture [7]

Його основними цілями є незалежність від фреймворків і зовнішніх агентств (інтерфейс користувача, бази даних тощо) і можливість тестування. Ці цілі корисні для всіх видів програмного забезпечення, але вони особливо корисні для мобільних програм. Ключовою частиною досягнення цього є використання принципу інверсії залежностей для перетину меж кола. Класи внутрішнього циклу можуть спілкуватися з класами зовнішнього циклу лише через інтерфейси, реалізовані класами зовнішнього циклу (Рис. 2). [8]

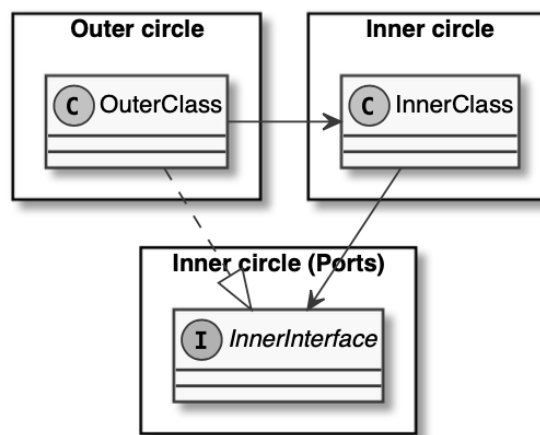


Рисунок 2. Спілкування класів через інтерфейси [8]

Однією з головних проблем чистої архітектури є відсутність офіційних еталонних реалізацій для різних платформ (наприклад, у книгах Мартіна, його блозі, його навчальних відео та обліковому записі GitHub).

Як наслідок, приклади Clean Architecture для iOS та Android, як правило, дають багато вказівок щодо конкретної платформи, часто жертвуючи незалежністю та стійкістю. Корисно повторити, що MVX — це шаблони високого рівня, які включають шаблони проектування. Чиста архітектура, також включає, принципи проектування, принципи компонентів, концепцію шарів і концепцію меж, серед інших ідей. Проект мобільного додатку з чистою глобальною архітектурою не повинен починатися з неофіційного шаблону, який уже містить низку залежних від платформи фреймворків; він повинен використовувати всі концепції, які є корисними для мобільної розробки, таким чином, щоб залишатися слабким зв'язком з платформою.

### Патерни проектування

Говорячи про шаблони високого рівня та архітектурні концепції, ми не повинні забувати про шаблони проектування, такі як шаблони GoF, які, звичайно, також можна використовувати та інтегрувати незалежно.

## Принципи SOLID

Чиста архітектура підтримує добре відомі принципи дизайну SOLID [7]. Оскільки вони сприяють досягненню загальних архітектурних цілей Clean Architecture, вони, природно, також актуальні для мобільних додатків.

### Контроль інтерфейсу сутності

Для реалізації цього принципу було запропоновано структурувати системи з трьома типами об'єктів. Спочатку вибрані типи Entity-Interface-Control пізніше були змінені на Entity-Boundary-Control, а потім на Entity-Boundary-Interactor. Об'єкти сутності містять дані, які використовує система, і всю поведінку, природно пов'язану з цими даними. Граничні об'єкти моделюють інтерфейс із системою [8]. Уся поведінка, що залишилася, поміщається в об'єкти Interactor. Martin's Clean Architecture адаптує ці типи об'єктів. Граничні об'єкти, зокрема, сприяють роз'єднанню, а отже, незалежності та тестуванню. Однак ЕВІ не дуже чітко пояснює, як отримати незалежність не тільки від інтерфейсу користувача.

### Впровадження залежностей

Впровадження залежностей допомагає підтримувати незалежність компонентів і їх тестування. Однак це не обов'язково означає, що потрібна структура впровадження залежностей. Може бути достатньо розглянути, які залежності мають бути гнучкими і організувати їх у прозорий спосіб, який легко змінювати.

## 4. ЗАГАЛЬНІ ВИМОГИ ДО ПОБУДОВИ АРХІТЕКТУРИ МОБІЛЬНОГО ЗАСТОСУНКУ

Побудова мобільних додатків не завжди простий процес, оскільки мобільний пристрій має доволі багато апаратних та прикладних обмежень. Тому при побудові треба враховувати такі моменти:

- великий обсяг UI коду;
- короткий реліз-цикл розробки програмного забезпечення;
- велика кількість різних сенсорів, датчиків у смартфоні;
- збереження даних, їх синхронізація, а також підтримка автономного режиму роботи;
- концепція програмування на життєвому циклі застосунка
- код має бути ефективний, з огляду на обмеження по пам'яті та заряд батареї смартфона;
- багатопоточність з особливістю того, що оновлення відображення зазвичай трапляється тільки в спеціальному потоці;

Наданий список може бути не вичерпаним, оскільки при розробці будь-якого мобільного застосунку треба звертати увагу на вимоги бізнесу.

Наступним кроком слід зазначити, що будь-яка гарна та підтримувана архітектура має на меті зберігати такі принципи: незалежність між своїми компонентами та фреймворками, а також можливість тестування.

*Незалежність:* незалежність між компонентами та фреймворками необхідна для побудови ефективної архітектури. Оскільки це може допомогти зменшити прогалину між апаратним забезпеченням та виконанням програми, допомагаючи модульними тестами. Маючи більш гнучку архітектуру з'являється можливість оновлювати застосунок за короткі проміжки циклу випуску з різноманітністю апаратного та програмного забезпечення та потребами налаштування.

*Автоматизоване тестування:* існує багато аспектів, які ускладнюють автоматизоване тестування для мобільних застосунків. В розрізі мобільних застосунків ми можемо поділити автоматизоване тестування на чотири види: локальні модульні тести (запускаються на приладах для розробки), модульні тести на ізольованих машинах/смартфонах, UI тести, а

також Snapshot тести. Серед трьох видів тестів слід відокремити модульні тести, бо вони надають нам швидких зворотній зв'язок, тому гарні архітектурні шаблони мають підтримувати модульні тести.

## 5. КОМБІНОВАНА АРХІТЕКТУРА

З огляду на існуючі архітектурні рішення пропонується створити комбіновану архітектуру: а саме поєднання класичних двоспрямованих архітектур MV(X) або Чистої архітектури з UDF (односпрямованою архітектурою). Тобто остаточний вибір MV(X) або різних реалізацій Чистої архітектури залежить від вимог зі сторони бізнесу, а також досвіду команди розробки.

Для цього пропонується створити DSL навколо імперативних підходів побудови компонентів відображення. В більшості випадках такий підхід виглядає більш привабливим в порівнянні з побудовою інтерфейсу у імперативному стилі або використанням XML конструкторів. Привабливість забезпечується через мінімізацію змінності даних, що покращує безпеку програми, оскільки незмінні структури даних у багатьох випадках менш схильні до помилок. Також пропонується побудова слою абстракцій, які будуть допомагати поєднувати UDF базовані архітектури з класичними MV(X). Це необхідно для того, щоб в залежності від поставлених вимог створювати той, чи інший модуль в потрібній архітектурі. Також при побудові даної архітектури треба оптимізувати роботу UDF архітектур, де можуть виникнути проблеми з продуктивністю. Для цього пропонується використовувати композицію Стану з під-станів, а також використовувати можливість порівняння одноманітності об'єктів Станів, та оновлювати відображення в разі нерівності теперішнього і нового стану.

Очевидно, з першу необхідно визначити модулі загального призначення які не прив'язані до конкретної програми і можуть бути повторно використані будь-де. Такі модулі використовуються верхнім рівнем програми, і хоча вони також можуть мати залежність один від одного, але це рідкісний випадок. Тому цей рівень називається App Independent (Рис. 3).

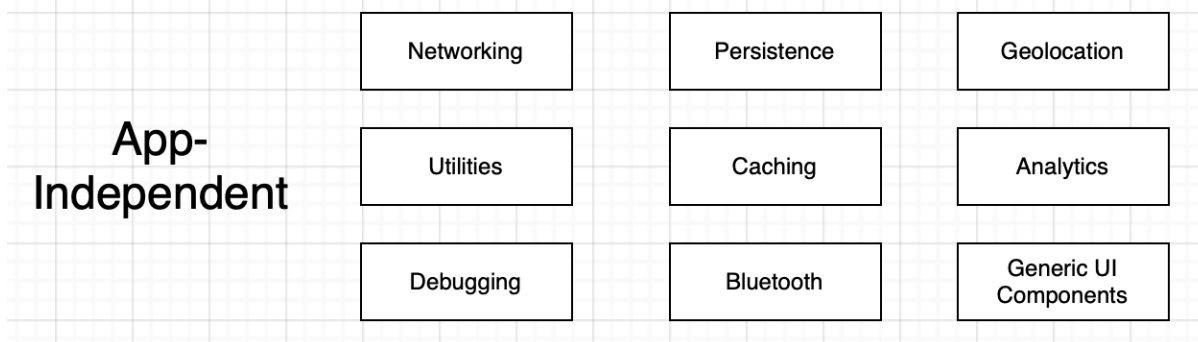


Рисунок 3. App-Independent модуль

Наступним кроком при проектуванні архітектури є створення окремих модулів функціоналу застосунку. Зазвичай можна побачити наскільки тісно окремі модулі функціоналу можуть залежати один від одного, що вказує на недоліки та слабкі сторони архітектури. Тому, основне правило, якого повинні дотримуватися розробники під час роботи над цими модулями — відсутність горизонтальної залежності на рівні функцій. Це означає, що один модуль функцій не повинен залежати від іншого. Архітектура кожного модуля може відрізнятися, тобто кожний модуль може бути реалізований ізольовано як і на UDF-подібній архітектурі, так і на MVX.

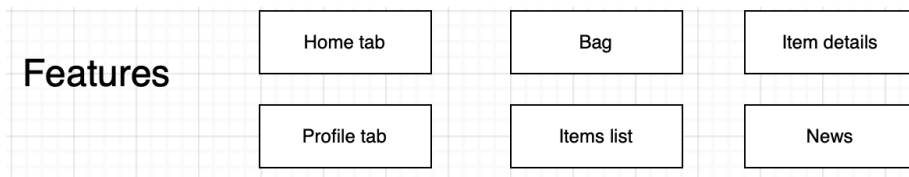


Рисунок 4. Features модуль

Наступним кроком розробки архітектури за стосунку є створення компонентів, які з одного боку, належать до рівня Features, а з іншого — не належать до жодного конкретного модуля. Таким чином, пропонується створити ще один рівень, App Specific для застосунку (Рис. 5).

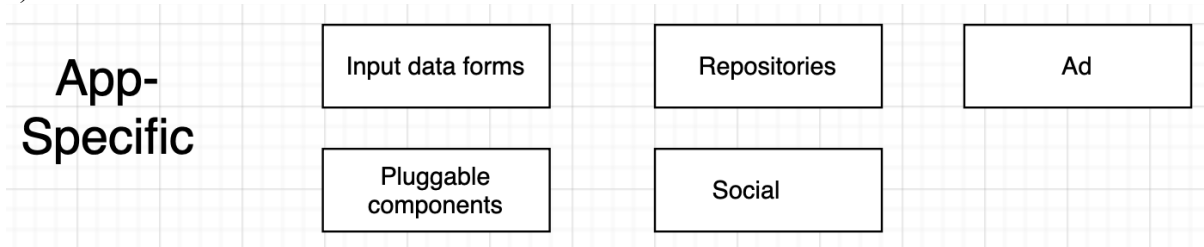


Рисунок 3. App-Specific модуль

Host App – це верхній рівень архітектури. Цей рівень визначає стан програми та тип конфігурації. Він отримує сповіщення від ОС. (Рис. 6).

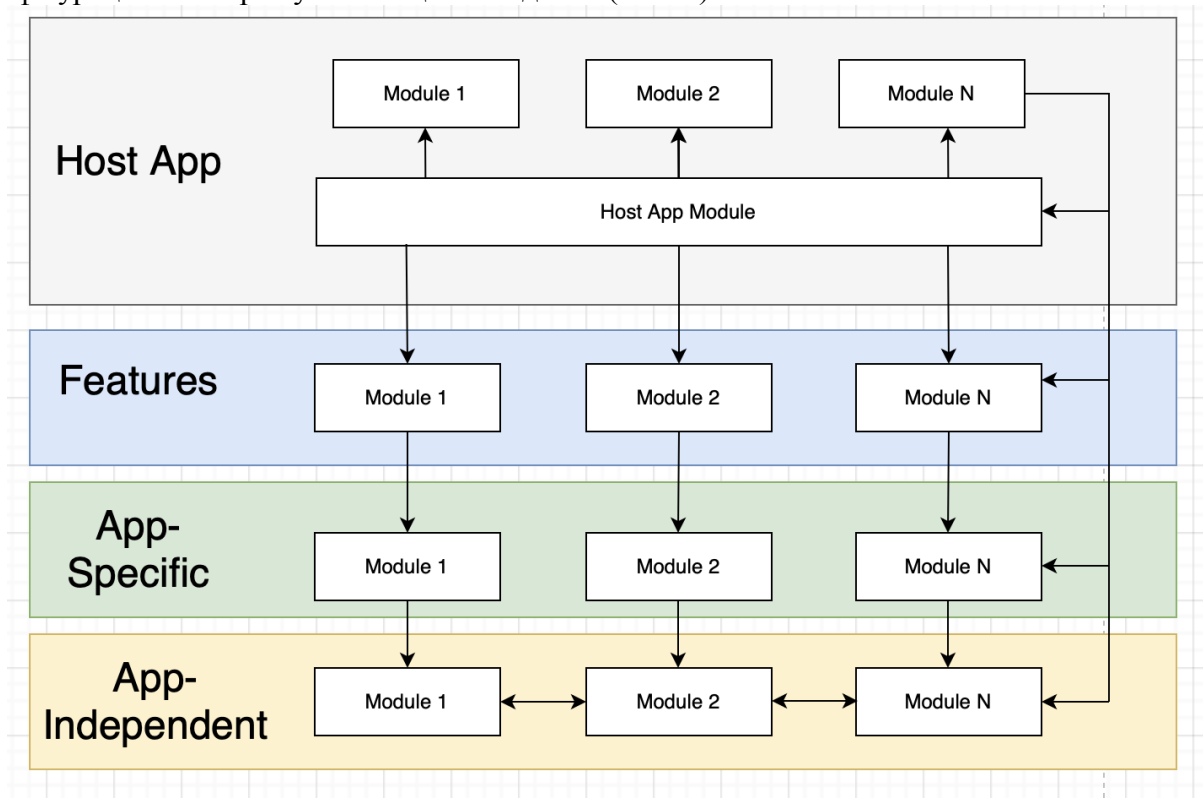


Рисунок 6. Загальний вигляд запропонованої архітектури

Таким чином, Host App є єдиним рівнем, який відповідає за навігацію між функціями, конфігурацію функцій і зв'язок між ними. Щоб забезпечити низький зв'язок модулів, ми повинні дотримуватися принципу інверсії залежностей, щоб в разі підвищити придатність до відладки та тестування. Такі залежні модулі, як репозиторії, служби та аналітика, мають



бути введені в модулі функцій за допомогою інтерфейсу з необхідними параметрами та налаштуваннями на рівні програми App Host.

Для здійснення переходів між функціями було запропоновано додати сутність Coordinator, яка належить кожному компоненту Features. Реалізація кожного Coordinator створюється в Host App модулі застосунку, який відповідає інтерфейсу навігації для певного модулю Feature. Цей підхід забезпечує детермінованість порядку показу екранів та логіку переходів. Приклад представлення збірки модуля «Список продуктів» і навігації від цієї функції до функції «Відомості про продукт» за допомогою Coordinator (Рис. 7). Пунктирними лініями відображена композиція, а суцільними відображено взаємодію модулів між собою, тобто перехід з відображення «Список продуктів» до відображення «Відомості про продукт».

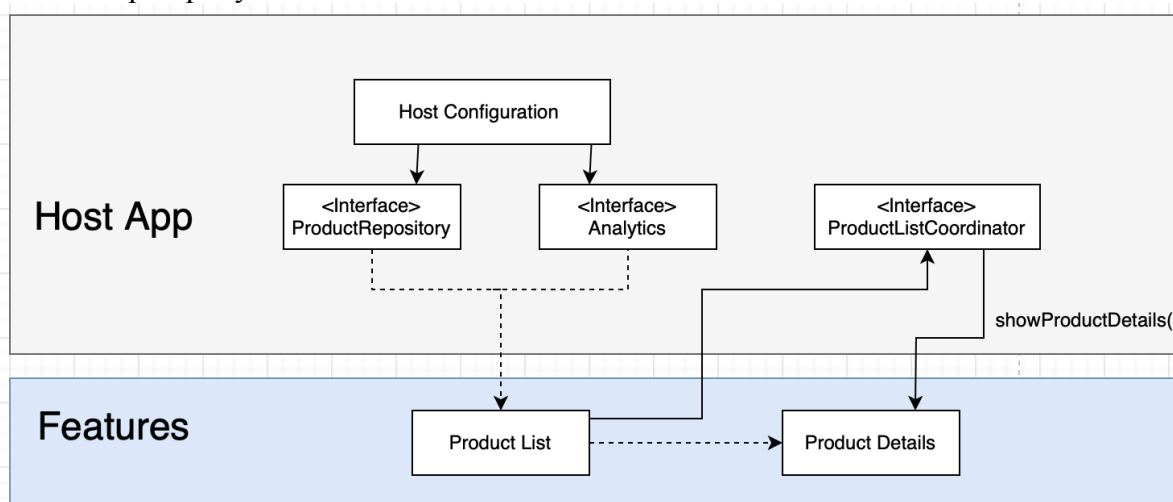


Рисунок 7. Взаємодія Coordinator з модулями функціональності застосунку

Також у кожного модуля Feature буде існувати LocalState, цей локальний Стан модифікує виключно керуючий модуль (Reducer, Presenter, ViewModel) батьківського Store, після модифікації загального AppState, LocalStore за допомогою механізму підписки отримує оновлений LocalState та зберігає його. Наступним кроком, усі підписники LocalStore отримують оновлений LocalState.

## 6. ВИСНОВКИ

З огляду на проведений аналіз можливо стверджувати, що популярні сучасні архітектурні підходи, такі як Clean, Onion необхідні для складних мобільних додатків, оскільки вони забезпечують незалежність від зовнішніх фреймворків і можливість тестування. Однак відсутність стандартних реалізацій із гарантованою якістю для iOS і Android призводить до різноманітних реалізацій і прикладів різної якості. Тому в даній роботі було запропоновано перенести їх принципи для побудови комбінованої архітектури, яка б дозволила поєднати їх переваги. На основі UDF подібних архітектур було запропоновано декомпонувати єдиний глобальний Стан на під-стани, а також створити порівняння цих станів, аби уникнути випадків оновлення відображення, коли це не потрібно.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. R. C. Martin, “Clean Architecture. A Craftsman’s Guide to Software Structure and Design”. Englewood Cliffs, NJ: Prentice Hall, 2017
2. Apple Computer, Inc., “Cocoa fundamentals guide,” 2006.
3. Архітектурний шаблон MVP. URL: <https://en.wikipedia.org/wiki/Model-view-presenter> (дата звернення: 16.11.2022)
4. Архітектура MVVM. URL: <https://jeremybytes.blogspot.com/2012/04/overview-of-mvvm-design-pattern.html> (дата звернення: 17.11.2022)
5. UDF Архітектура. URL: <https://habr.com/ru/company/inDrive/blog/571394/> (дата звернення: 18.11.2022)
6. J. Palermo, “Onion architecture”, 2008. URL: <https://jeffreypalermo.com/2008/07/the-onion-architecture-part-1/> (дата звернення: 18.11.2022)
7. R. Martin and M. Martin, Agile Principles, “Patterns, and Practices in C#”, ser. Robert C. Martin Series. Pearson Education, 2006.
8. I. Jacobson, M. Christerson, P. Jonsson, and G. Overgaard, “Object- Oriented Software Engineering”. New York, NY, USA: ACM, 1992.

# **ПРЕДСТАВЛЕННЯ ІНТЕРОПЕРАБЕЛЬНИХ СЕРВІСІВ ДЛЯ РОБОТИ З МЕДИЧНИМИ ДАНИМИ В МІКРОСЕРВІСНІЙ АРХІТЕКТУРІ**

Кандель К.В.<sup>1</sup>, Кулик В.О.<sup>2</sup>, Мельник О.Р.<sup>3</sup>, Письменний І.О.<sup>4</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> kirill.kandel@gmail.com [0000-0001-7152-5945], <sup>2</sup> slava.kulik27@gmail.com [0000-0003-3833-1529], <sup>3</sup> melnyk.oleksii@lil.kpi.ua [0000-0002-0297-6635],  
<sup>4</sup> ihor.pismennyu@gmail.com [0000-0001-7648-2593]

**Розглядається задача аналізу проблематики архітектурних підходів при проектуванні систем медичних сервісів. Досліджено наявні недоліки при побудові та впровадженні механізмів обробки та взаємодії з медичною інформацією. Завдяки порівнянню архітектурних підходів, буде спроектовано та запропоновано мікросервісну архітектуру взаємодії інтероперабельних медичних сервісів.**

**Ключові слова:** медичні сервіси, мікросервісна архітектура, інтероперабельні сервіси, медичні стандарти.

## **1. ВСТУП**

За останні десятиліття галузь охорони здоров'я досягла багатьох вагомих успіхів. Досягнення генетики та біотехнології дозволили вченим зрозуміти та лікувати хвороби на молекулярному рівні. Мабуть, найбільш вражаючим є той факт, що за останні 50 років тривалість життя зросла на 40%[1]. Незважаючи на ці успіхи, все ще є величезні можливості для вдосконалення. Найголовнішим серед цих можливостей є перехід до інформаційно-орієнтованої медицини, моделі індивідуальної медичної допомоги, за якої пацієнти отримують персоналізовані цільові рішення для лікування, специфічні для їхніх індивідуальних стадій захворювання, а також їхніх індивідуальних генетичних і метаболічних параметрів. Такий перехід, насамперед, супроводжується значними удосконаленнями в моделях взаємодії медичних сервісів - видобутку та обробці медичної інформації про пацієнта, її передача та зберігання у медичних інформаційних сервісах.

Медична сфера вважається однією з найскладніших сфер для досліджень. Найбільш поширені підходи зосереджені на забезпеченні надійних систем класифікації, які використовуються для допомоги лікарю в процесі діагностики/прогнозування. Проблеми аналізу медичних даних мають деякі унікальні особливості, які відрізняють їх від інших сфер і ускладнюють їх вирішення. Основні труднощі пов'язані зі складним характером оброблюваних даних, їх якістю і кількістю.

Хоча в лікарнях зберігається величезна кількість інформації, що відноситься до пацієнтів, які лікувалися раніше, частина цих даних не в електронному вигляді, і ідея їх цифровізації зазвичай розглядається як трудомістка. Крім того, зазвичай лікарі в різних лікарнях мають дещо різні методи дослідження. Це призводить до різних структур даних, що надходять із різних джерел, що у більшості випадків унеможливує їх поєднання. Крім того, оскільки на кону стоїть людське життя, точний діагноз має вирішальне значення. Тому

стандартизація є дуже важливим кроком, що дозволить суттєво спростити механізми передачі та взаємної обробки інформації у медичних сервісах.

Одним з поширених проблемних питань у інформатизації медицини є постійний процес виникнення та застосування нових типів медичної інформації - стрімкий розвиток та впровадження інформаційних систем у галузь охорони здоров'я призводить до більш детальних та різноманітних потоків інформації, що потребують обробки. Як результат, безліч компаній, що розробляють інформаційні системи та сервіси для роботи з медичною інформацією, зіштовхуються з проблемою надзвичайної складності та комплексності при розробці, підтримці та оновленні програмного забезпечення.

## 2. ПОСТАНОВКА ПРОБЛЕМИ

Вагомою проблемою, що постає перед розробниками інформаційних систем взаємодії медичних сервісів, є потреба у комбінуванні можливостей отримання даних про пацієнта або медичні засоби та можливостей їх обробки та транспортування. Кожна потенційна компанія-розробник оперує лише власним переліком сервісних можливостей, що можуть не включати логістичні сервіси доставки медикаментів або, наприклад, сервіси своєчасного зчитування та зберігання інформації про пацієнта. Натомість, зосереджуючи розробку на обмеженому переліку потенційних сервісів, розробляється архітектура системи, що не може забезпечити повноцінний обіг інформації про пацієнта, ліки та засоби лікування, а також не надає можливості керувати даною інформацією та перебігом подій, пов'язаних з пацієнтом чи ліками.

Іншою проблемою закритих інформаційних медичних систем, окрім неповноцінного інтерфейсу взаємодії з даними та інструментами, є обмежена можливість масштабування. Архітектура системи, що з самого початку була побудована монолітно, є надзвичайно вразливою до проблем потенційного горизонтального масштабування, якщо виникне потреба. Причиною для розширення функціоналу може слугувати будь-що: збільшення джерел надходження інформації чи кількості медичних сервісів, що надає розробник - кожен з пунктів може потребувати розробки окремого сервісу для забезпечення функціонування, тому можливість горизонтального масштабування у архітектурі системи є вкрай необхідною.

Слід також відзначити проблему незалежної розробки сервісів при обробці інформації чи впровадженні певного функціоналу. Так, наприклад, компанія-розробник може створювати певний набір медичних інформаційних сервісів, що незалежно один від одного обробляють інформацію та не мають автоматизованих засобів збереження та передачі даних між собою. Це призводить до наступних проблем:

- 1) Зменшення швидкодії повного життєвого циклу медичної інформації через відсутність єдиної автоматизованої системи
- 2) Необхідність залучення спеціалістів для керування та обміну даних між сервісами
- 3) Проблематика обробки різних форматів даних у сервісах, що не мають єдиного стандарту обміну інформацією
- 4) Проблема безпеки даних пацієнта та засобів лікування - необхідність забезпечувати конфіденційність на рівні кожного сервісу

Вирішенням даного ряду проблем може стати мікросервісна архітектура, що забезпечує повноцінну можливість оперування даними пацієнта, знаходити, обробляти та зберігати медичну інформацію. Такий підхід дозволить забезпечити гнучкість при масштабуванні, дозволивши створити необхідний перелік медичних сервісів, а також впровадити засоби їх взаємодії.

Цей матеріал має за мету запропонувати вирішення проблеми обробки медичної інформації масштабованою системою сервісів, що може бути впроваджена та вдосконалена

за потреби у разі виникнення нових технічних вимог відносно типів інформації та алгоритмів взаємодії медичних сервісів.

### 3. МЕТА ТА ЗАДАЧІ ДОСЛІДЖЕННЯ

Медичні дані пацієнта є критичними, чутливими та конфіденційними даними, оскільки є частиною медичної таємниці. Такі медичні дані, як медична картка пацієнта, діагнози, курси лікування і т.д. є важливими як для пацієнта, у разі невірної лікування, так і для страхових компаній, які повинні сплачувати страхову компенсацію. Тому ці медичні дані напрочуд важливі і їх конфіденційність та захищеність на всіх етапах використання є важливою частиною функціонування медичної установи.

Враховуючи те, що сучасні медичні системи постійно розвиваються і створюють архітектури обчислювальних систем та інфраструктури їх взаємодії, постає нагальна проблема до забезпечення безпеки даних в розподілених медичних системах, що включають в себе інтегровані медичні сервіси. Також, провайдери медичних сервісів будують власні архітектури медичних сервісів базуючись на різних підходах, у тому числі й на монолітному підході побудови обчислювальної системи. Попри те, що у монолітних систем є свої переваги і недоліки [2], варто зазначити, що сучасні архітектурні рішення будуються за допомогою мікросервісного підходу або взагалі без залучення серверів - використовуючи хмарні платформи [3]. Тобто у результаті ми отримуємо значну кількість однакових за суттю систем, які побудовані різноманітним чином.

Тобто, як було сказано вище, можна виокремити мету роботи. Метою є представлення певної уніфікованої архітектури інтегрованих сервісів для роботи з медичними даними.

Взявши до уваги варіативність існуючих обчислювальних систем та архітектур, потреб сучасної медицини та медицини наступного дня, запропонована архітектура повинна мати можливість для розширення, стабільно виконувати роботу при навантаженнях і мати повний контроль над конфіденційними даними, що є у системі без можливості їх пошкодження чи витоку. Таким чином, нам необхідно дослідити сучасні архітектурні підходи побудови таких систем і запропонувати ту, яка буде найбільш гнучкою під описані потреби.

Через значні регуляції та обмеження в сфері охорони здоров'я, багато країн мають свої сертифікаційні процедури, у тому числі й для електронних медичних систем, що накладають свої обмеження на безпеку, цілісність і конфіденційність даних в медичній системі [3]. Тобто запропонована архітектура повинна враховувати можливі рестрикції в обробці, передачі, збереженню чутливих медичних даних, що необхідно передбачити, враховуючи існуючі стандарти.

Таким чином, буде представлено структуру взаємозв'язків інтегрованих сервісів для роботи з медичними даними. Головними перевагами отриманої системи будуть наступні характеристики:

1. Безпека і конфіденційність даних при зчитуванні, передачі та взаємодії на всіх рівнях.
2. Можливість масштабування при зміні навантаження на систему.
3. Уніфікація архітектурного рішення, для спрощення роботи профільним спеціалістам та підтримки гнучкості системи.
4. Спрощена інтеграція з існуючою системою новітніх сервісів, завдяки поширеного стандарту обміну медичними даними.
5. Розширення, тестування і підтримка сервісів та їх кодової бази буде зручнішою завдяки уніфікації взаємодії та архітектурному підходу.

Такі конкурентні переваги будуть значним корисним впливом на загальний підхід до культури побудови розподілених систем, що включають в себе значні безпекові обмеження та можливості динамічної зміни самої системи на рівні її компонентів.

## 4. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

Обираючи стандарти для передачі і документування медичних даних, було розглянуто наступні:

1. Health Level 7 (HL7) - є сімейством міжнародних стандартів для передачі клінічних та адміністративних даних між програмними додатками, що можуть використовуватися різними провайдерами медичних сервісів. Є поширеним міжнародним стандартом, що постійно розвивається і використовується в різних електронних медичних сервісах. Значна кількість використань цього стандарту на науковому ресурсі говорить про його значну популярність [6].
2. Continuity of Care Record (CCR) - це стандарт медичної документації, був створений медичними працівниками на основі їхніх поглядів на дані, якими вони можуть захотіти поділитися в будь-якій ситуації. Значна, але менша за HL7, кількість використань цього стандарту на науковому ресурсі також говорить про його значну поширеність [5].
3. ISO/TC 215 - це стандарт для медичних інформаційно-комунікаційних технологій (ICT), щоб забезпечити сумісність і взаємодію між незалежними системами. Значна, але менша за HL7 та CCR, кількість використань цього стандарту на науковому ресурсі також говорить про його поширеність [7].

Розглядаючи ці стандарти, звернемо увагу на їх частоту використання у наукових роботах, також за тим, наскільки уніфікованим є стандарт і наскільки він є зручним для імплементації та розуміння. Виходячи з цих критеріїв, HL7 є розповсюдженим сімейством стандартів, які підійдуть для імплементації передачі, взаємодії (Рисунок 1) та збереження електронних карток пацієнтів між різними програмними сервісами та додатками.



Рисунок 1. Опис сімейства HL7

Розглядаючи різні тип архітектур, необхідно керуватися предметно-орієнтованим підходом для розробки такої розподіленої системи. Беручи до уваги те, що медичні системи повинні бути гнучкими і відмовостійкими, то монолітна архітектура не може конкурувати з мікросервісною архітектурою через свою природу. Приведемо порівняльну таблицю (табл. 1) цих архітектур і зможемо дізнатися, яка краще підійде для сучасної архітектури медичних інтероперабельних сервісів.

Таблиця 1. Порівняльна характеристика монолітної та мікросервісної архітектури

Монолітна архітектура	Мікросервісна архітектура
Висока зв'язаність компонентів архітектури	Низька зв'язаність компонентів архітектури
Важкість внесення змін до системи	Спрощене внесення змін
Складність масштабування	Простіше масштабування
Технологічний бар'єр та важча підтримка з часом	Кожен мікросервіс простіше перевести на нові технології без зміни інших частин системи
Простий процес розгортки додатку	Ускладнена розгортка додатку

Роблячи проміжний висновок у тому, що більшу гнучкість та кращу сумісність надає нам саме мікросервісна архітектура, оскільки можливість горизонтального масштабування є однією з головних переваг, а розподілення відповідальності між мікросервісами призводить до природної відсутності зчеплення між мікросервісами, якщо використовувати предметно-орієнтований підхід до усієї розробки такої системи, що важче уникнути в монолітних архітектурах.

Також, важливим елементом мікросервісної архітектури є уніфікованість прикладного програмного інтерфейсу кожного мікросервісу. Це робиться за допомогою специфікації OpenAPI, що дозволяє зручно описувати та документувати API кожного мікросервісу для його подальшого використання. Таким чином у поєднанні зі стандартом HL7 та предметно-орієнтованим підходом, можна провести уніфікацію API [8].

Також треба згадати про важливі елементи мікросервісної архітектури, такі як:

1. API Gateway - це інтерфейс між клієнтами та іншими сервісами, що може їх викликати для бажаного клієнту результату [9].
2. Service Discovery - це сервіс, автоматично виявляє та реєструє стан сервісів у комп'ютерній мережі [10].

Використовуючи ці сервіси, можна побудувати базову інфраструктуру для мікросервісної архітектури. На Рис. 2 зображена базова мікросервісна архітектура.

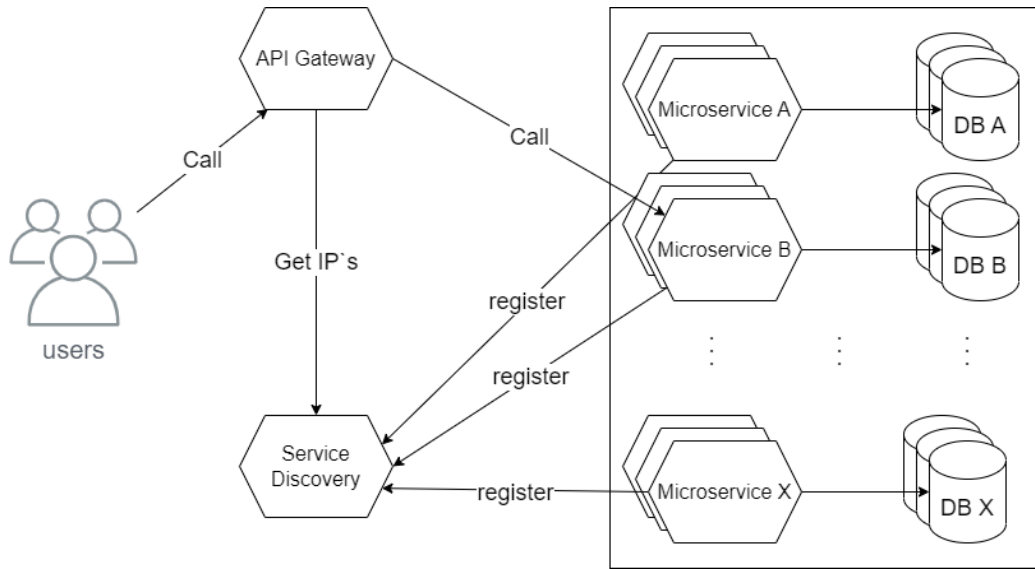


Рисунок 2. Базова мікросервісна архітектура

Використовуючи схожий підхід можна розширити його до медичної області застосування. На Рисунок 3 зображена мікросервісна архітектура для інтероперабельних медичних сервісів із обов'язковими та додатковими мікросервісами, та з прикладом виклику деякого мікросервісу.

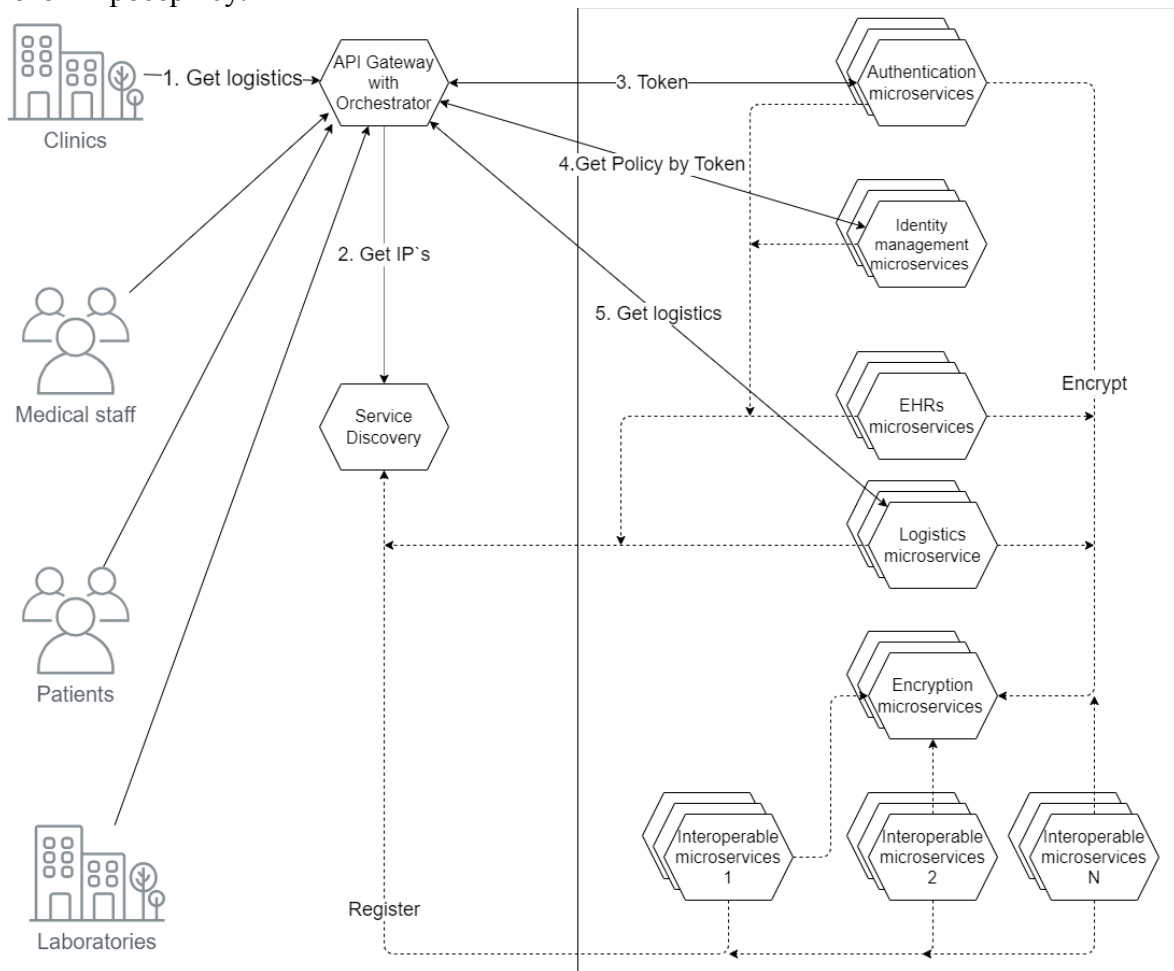


Рисунок 3. Мікросервісна архітектура медичних сервісів



## 5. ВИСНОВКИ

На сьогоднішній день розробники програмного забезпечення в галузі охорони здоров'я все більше починають стикатися з проблемами масштабування та додавання нового функціоналу в існуючі системи. Перш за все це пов'язано з тим, що компоненти системи є тісно зв'язаними. Для вирішення цих проблем пропонується застосовувати мікросервісну архітектуру.

Мікросервісна архітектура - це підхід у розробці програмного забезпечення шляхом створення незалежних компонентів - мікросервісів, які поєднуються між собою. Разом такі мікросервіси складають готовий додаток, але в будь-який момент часу він може бути розділений на незалежні блоки та інтегрований в іншу комп'ютерну систему без великих зусиль.

Використання мікросервісної архітектури в галузі охорони здоров'я надає такі переваги:

1. Портативність і сумісність. Мікросервісна архітектура використовує для взаємодії компонентів популярні стандарти JSON, HTML та ін., що дозволяє об'єднати в одному додатку мікросервіси, написані з використанням різних мов програмування. Це призводить до того, що кожен з мікросервісів є повністю незалежним та за необхідністю його можна перевикористати в іншому додатку. Це також означає, що додавання нових компонентів у існуючу систему здійснюється простіше та з меншим ризиком впливу на існуючі компоненти в процесі.
2. Ізоляція компонентів прискорює дотримання нормативних вимог. Одна з переваг, яка має особливе значення в галузі охорони здоров'я — це ізоляція між компонентами. Деякі компоненти є більш важливими за інші та потребують ретельного тестування, а також підпадають під регулятивний нагляд (наприклад, медичних стандартів). Мікросервісна архітектура дозволяє повністю ізолювати такі компоненти від інших компонентів системи, що значно спрощує їх тестування, оновлення та валідацію з існуючими стандартами.
3. Вбудована сумісність з нормами HIPAA. Мікросервіси за своїм походженням є подійно-орієнтованими та добре підходять для рішень, які вимагають відповідності нормам HIPAA, оскільки будь-яку важливу інформацію, наприклад особисту інформацію про пацієнта, можливо ізолювати від інших даних, а доступ до цих даних простіше контролювати.
4. Зменшений час розробки. Аспект «багаторазових компонентів» дозволяє не витратити час на розробку існуючого функціоналу, а зосередитись на додаванні нових компонентів до системи, що значно пришвидшує розробку та впровадження нових додатків.

Також мікросервісна архітектура має уніфікований програмний інтерфейс, створений за допомогою специфікації OpenApi, що дозволяє зручно описувати та документувати API кожного мікросервісу для його подальшого використання. Поєднавши даний інтерфейс з ключовими компонентами мікросервісної архітектури: Api Gateway та Service Discovery була розроблена модель архітектури інтероперабельних медичних сервісів.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Medical informatization: A combination of digital technology and medical care [Електронний ресурс] – Режим доступу до ресурсу: <https://www.atcinitiative.org/medical-informatization-a-combination-of-digital-technology-and-medical-care>

2. A Comparative Review of Microservices and Monolithic Architectures [Електронний ресурс] – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/abstract/document/8928192>

3. Microservices vs Serverless: A Performance Comparison on a Cloud-native Web Application [Электронный ресурс] – Режим доступа до ресурсу: [https://www.researchgate.net/publication/341483472\\_Microservices\\_vs\\_Serverless\\_A\\_Performance\\_Comparison\\_on\\_a\\_Cloud-native\\_Web\\_Application](https://www.researchgate.net/publication/341483472_Microservices_vs_Serverless_A_Performance_Comparison_on_a_Cloud-native_Web_Application)
4. Security Techniques for the Electronic Health Records [Электронный ресурс] – Режим доступа до ресурсу: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5522514/>
5. HL7 - Overview [Электронный ресурс] – Режим доступа до ресурсу: <https://www.hl7.org/>
6. Continuity of Care Record [Электронный ресурс] – Режим доступа до ресурсу: [https://en.wikipedia.org/wiki/Continuity\\_of\\_Care\\_Record](https://en.wikipedia.org/wiki/Continuity_of_Care_Record)
7. ISO/TC 215 Health informatics [Электронный ресурс] – Режим доступа до ресурсу: <https://www.iso.org/committee/54960.html>
8. OpenApi Initiative [Электронный ресурс] – Режим доступа до ресурсу: <https://www.openapis.org/>
9. What does an Api Gateway do? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.redhat.com/en/topics/api/what-does-an-api-gateway-do>
10. Service discovery [Электронный ресурс] – Режим доступа до ресурсу: [https://en.wikipedia.org/wiki/Service\\_discovery](https://en.wikipedia.org/wiki/Service_discovery)

# **МЕТОДИ РОЗПІЗНАВАННЯ ТЕКСТІВ ТА ПОШУКУ КЛЮЧОВИХ СЛІВ ДЛЯ АВТОМАТИЧНОГО РЕФЕРУВАННЯ ТЕКСТІВ**

Кузнєцов О.А.<sup>1</sup>, Кисельов Г.Д.<sup>2</sup>

НН ПСА КПІ ім. Ігоря Сікорського, Київ, Україна

<sup>1</sup>oleksiy1908@gmail.com, <sup>2</sup>g.kyselov@gmail.com

**Метою дослідження є перевірка ефективності основних підходів до автоматичного реферування тексту, а також дослідження різних методів для різних підходів реферування тексту. У доповіді представлено опис основних підходів, які на даний момент використовують для автоматичного реферування тексту, а також різні методи, які використовуються в цих підходах. На думку автора, дослідження методів автоматичного реферування текстів дуже важлива, так як вони здатні суттєво прискорити та спростити цей досить важливий алгоритм.**

**Ключові слова:** реферування текстів, добуваючий підхід, абстрактний підхід, метод тематичних слів, центрування речень, індикаторні представлення.

## **1. ОСНОВНІ ПІДХОДИ ДО АВТОМАТИЧНОГО РЕФЕРУВАННЯ ТЕКСТУ**

Існує два основних підходи до автоматичного реферування тексту :

- Extractive summarization (так називається добуваючий підхід);
- Abstractive summarization (так називається абстрактний підхід).

Добуваючий підхід – це автоматичне анотування, що засноване на виділенні з первинних документів ключових фраз, слів, чи навіть цілих абзаців, які потім додаються в вихідний документ без змін та саме в порядку їх появи в первинному тексті.

Абстрактний підхід – це автоматичне анотування, що засноване на виділенні найбільш важливої інформації, а також навіть можливого створення нових текстів на базі узагальнених первинних документів. Цей метод, на відмінну від добуваючого метода дає можливість використовувати слова, яких не було у вхідному документі. Анотації, що були отримані використовуючи саме цей підхід схожі на ті, що люди пишуть самі. Але реалізація цього методу є дуже нетривіальною задачею, адже модель має вирішувати досить складні проблеми, такі як семантичне представлення тексту та генерація природної мови. Тому, зважаючи на складність даного підходу як у реалізації так і у рамках практичного застосування, а також значні обмеження які стосуються текстів, що можна реферувати використовуючи добуваючий метод надалі будуть проведені дослідження методів реферування в рамках добуваючого підходу.

## **2. ДОБУВАЮЧИЙ ПІДХІД АВТОМАТИЧНОГО РЕФЕРУВАННЯ ТЕКСТІВ**

Зараз основа всіх добуваючих методів складається з трьох частин[1]:

– побудова проміжного представлення вхідного тексту. Існує два типи підходів до показу початкового тексту: topic representation (тематичне представлення) та indicator representation (індикаторне представлення). Тематичне представлення трансформує текст в

проміжне представлення за допомогою інтерпретацій тем, що представлені у тексті. Індикаторне представлення анує речення як перелік формальних ознак, тобто індикаторів, що відрізняються в залежності від алгоритму та використовує їх для ранжування тексту. До таких ознак відносяться довжина речення, місце в тексті, наявність ключових фраз і т.д.;

– оцінка речень на основі проміжного представлення вхідного тексту. При формуванні проміжного представлення кожному реченню присвоюється якась оцінка важливості цього речення для анотації. У різних підходах до подання теми оцінка речення відображає, наскільки добре те чи інше речення відображає найважливіші теми тексту;

– формування підсумку. Система вибирає декілька найважливіших речень для створення вихідної анотації. Деякі підходи для вибору важливих речень вибирають жадібні алгоритми, а деякі підходи можуть перетворити вибір речень у одну з задач оптимізації, де вибирається набір речень, враховуючи обмеження задачі оптимізації, на меті якого стоїть максимізувати важливість та мінімізувати надмірність деяких слів.

## 2.1. Підходи з використанням тематичного представлення

### Метод тематичних слів.

Техніка тематичних слів є загальним підходом, головною ідеєю якого є виділення слів, що описують тему вхідного тексту. Тематичні слова можуть бути визначені досить різними способами, наприклад робота[2] була однією з перших, в яких використовувався цей метод, пошук описових слів у документі відбувався за рахунок використання порогових значень частоти цих слів у тексті. У роботі[3] використовували логарифмічний алгоритм відношення правдоподібності для виявлення описових слів.

Визначення важливості речення базується на розподілі інформативних слів у реченнях. Існує два способи обчислити важливість речення: як функцію кількості тематичних слів, яку вона містить, або як частку тематичних слів у одному реченні. Обидві функції оцінки речень стосуються одного і того ж подання теми, однак вони можуть призначати різні оцінки різним реченням, через різну специфіку оцінювання. Перший метод може призначати вищі оцінки довшим реченням, оскільки вони мають більше слів, а інший більше звертає увагу на щільність тематичних слів. Найбільш поширеними методами в цій категорії є TF-IDF модель, лямбда-відношення правдоподібності та їх модифікації.

TF-IDF (term frequency — inverse document frequency, частота терму – зворотна частота документа) – статистична міра яка використовується щоб оцінити важливість терму у межах документу, який є частиною набору документів.

TF (term frequency – частота слова) – відношення кількості появи слова до загальної кількості слів у документі (1). Таким чином, оцінюється важливість терма  $t$  в рамках окремого документа  $d$ .

$$tf(t, d) = n_t / \sum_k n_k \quad (1)$$

де  $n_t$  – кількість входжень терма  $t$  в документ;

$\sum_k n_k$  – загальна кількість слів у документі.

IDF (inverse document frequency – зворотна частота документа) – інверсія частоти, з якою деяке слово зустрічається в документах колекції (2). IDF дозволяє зменшити оцінку широковживаних слів, та враховувати унікальність слів. Для кожного унікального слова в межах конкретного набору документів існує лише одне значення IDF. Більшу оцінку в TF-IDF отримують слова з високою частотою вживань в межах конкретного документа і з низькою частотою вживань в інших документах.

$$idf(t, D) = \log (|D| / |\{d_i \in D \mid t \in d_i\}|) \quad (2)$$

де  $|D|$  – кількість документів у колекції;

$|\{d_i \in D \mid t \in d_i\}|$  – кількість документів із колекції  $D$ , в яких зустрічається  $t$  (коли  $n_t \neq 0$ ).

$\lambda$ -відношення правдоподібності (log-likelihood ratio) є логарифмом відношення ймовірності спостереження слова з однаковою ймовірністю в корпусі вхідних документів і корпусі відповідних їм резюме, до ймовірності появи слова з різними ймовірностями в цих корпусах.

**Метод заснований на центруванні речень.**

Головна ідея цього підходу заснована на допущенні, що найбільш цікава для анатоції інформація знаходиться не лише в одному реченні. Він полягає в обчисленні «відстаней» між реченнями і у виборі тих з них, що в середньому знаходяться «ближче» до інших. Для визначення близькості речень використовуються алгоритми, що засновані на наборах слів (Bag-of-words). Наприклад найближчі в середньому речення можна визначити наступним чином:

- обчислити близькість між усіма парами речень за якимось параметром, наприклад перекриттям змісту одного речення іншим;
- для кожного речення визначити середню близькість до інших речень;
- впорядкувати ці значення і вибрати з мінімальну близькістю (наприклад використовуючи Латентний семантичний аналіз (LSA)).

Тематична модель, що представляється в роботі [4], це неконтрольований метод вилучення прихованої семантики тексту на того, що слова з подібним значенням частіше зустрічаються разом, чим окремо. Простіше кажучи LSA бере текстові документи та відтворює їх у декількох різних частинах, де кожна частина виражається у різний спосіб погляду на значення тексту. Якщо уявити текстові дані як ідею, існувало б декілька різних способів розгляду цієї ідеї або декілька різних способів адаптації усього тексту.

Метод, що представлений у роботі [5] полягає у виборі одного речення для кожної з тем (рисунок 1), таким чином, щоб зберегти початкову кількість тем. Ця стратегія в початковому вигляді має недолік через те, що темі може знадобитися більше одного речення для передачі своєї інформації, а ми вибираємо лише 1. Тому, були запропоновані декілька альтернативних рішень для поліпшення ефективності методів узагальнення на основі LSA. Одним із покращень було використання ваги кожної теми, що надало гнучкість у варіативності кількості речень. Інше поліпшення базується на тому, що його автори[6], зрозуміли, що речення, які обговорюють деякі важливі теми, є хорошими кандидатами для узагальнення теми, отже, для того, щоб знайти ці речення, вони визначили вагу речення за допомогою формули (3):

$$(s_i) = \sqrt{\sum_{j=1}^m d_{ij}^2} \tag{3}$$

де  $g()$  – функція «зважування»;  
 $m$  – кількість речень;  
 $d_{ij}$  – вага теми  $i$  в реченні  $j$ .

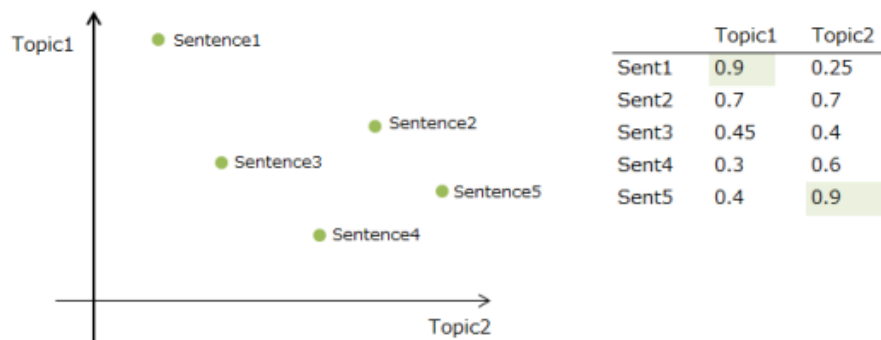


Рисунок 1. Простий вибір основних речень, що представляють теми документу (LSA)

## 2.2. Підходи з використанням індикаторного представлення

Підходи до подання показників (індикаторів) спрямовані на моделювання подання тексту у вигляді деякого набору ознак, які потім будуть використовуватися для класифікації.

До них відносяться методи на основі графів та техніки машинного навчання, що використовуються для визначення речень, які потрібно включити в кінцеве рев'ю. Основні використовуванні ознаки для цих методів:

- Речення на початку або в кінці тексту є більш інформативними;
- Занадто короткі або занадто довгі речення є малоінформативні;
- Наявність визначених сигнальних фраз або ключових;
- Слова з заголовку дають право вважати що це речення має ознаки відношення до цієї теми;
- Наявність емоційно забарвлених розділових знаків (знак питання, знак оклику, трикрапка, тощо);

Методи на основі графів, утворені під впливом алгоритму PageRank [7], тому вони представляють з себе документ, який був відображений у вигляді зв'язаного графу. Кожне речення утворює вершини графу, а вага ребр між реченнями вказують, на зв'язок подібності двох речень. Приклад подання тексту у вигляді графа можна побачити на рисунку 2. Подібність речень може бути виміряна як змістовним перекриття між реченнями, так і з використанням методу TF-IDF наприклад.

Графічна інтерпретація тексту має два основних результати. По-перше, кожен підграф – це окремий розділ, що пов'язаний однією. Другий результат – ідентифікація важливих речень у документі заснована на припущенні, що речення, які пов'язані з багатьма іншими реченнями, а також є центрами підграфів і, швидше за все, важливі і відображають тематику вхідного тексту, тому їх потрібно включити у вихідний документ.

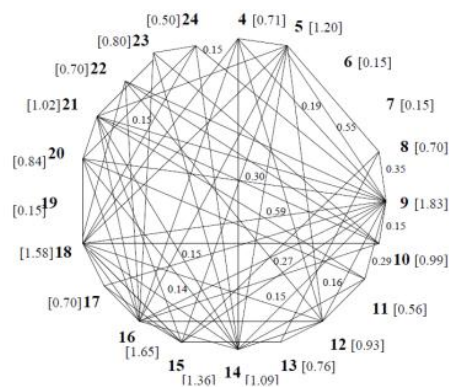


Рисунок 2. Графічне подання тексту у вигляді зваженого графу (речення представляють вершини з встановленою результуючою оцінкою)

Підходи машинного навчання представляють реферування як проблему класифікації. У роботі<sup>[8]</sup> представлена рання дослідницька спроба застосування технік машинного навчання для реферування. Автори розробили функцію класифікації, яка базувалася на наївному баєсовській класифікаторі. Ймовірності для класифікації були визначені за допомогою тренувальних даних з використанням правила Байєса (4):

$$P(s \in S | F_1, F_2, \dots, F_k) = P(F_1, F_2, \dots, F_k | s \in S) P(s \in S) / P(F_1, F_2, \dots, F_k), \quad (4)$$

де  $s$  – це речення із колекції документів;  $F_1, F_2, \dots, F_k$  – ознаки, що використовуються для класифікації;  $S$  – резюме, що має бути створено.

Отже, незважаючи на різноманітність методів та обґрунтовану дієвість кожного із них, кожен знає має досить суттєві недоліки. Наприклад, однією з основних проблем окрім складності деяких підходів є те, що для даних алгоритмів необхідна спеціальна вибірка, і не кожна випадкова вибірка документів згодиться для класифікації. Задача створення такої вибірки є більш складною, ніж створення цього всього власноруч, які містять об'єднані речення, переформульовані фрази чи навіть суто нові речення, до того ж такі навчальні дані також мають створюватися і робота алгоритму повністю залежить від якості таких навчальних даних. Якби вдалося б прибрати ці недоліки та обмеження, то можна було б скористатися даними алгоритмами для більш широкого спектру областей, але на даний момент вони накладають значні обмеження на тексти.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A. Nenkova, K. McKeown. A survey of text summarization techniques.
2. Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development* 2, 2 (1958), 159–165.
3. Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational linguistics* 19, 1 (1993), 61–74.
4. Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *JASIS* 41, 6 (1990), 391–407
5. Yihong Gong and Xin Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 19–25
6. Josef Steinberger, Massimo Poesio, Mijail A Kabadjov, and Karel Ježek. 2007. Two uses of anaphora resolution in summarization. *Information Processing & Management* 43, 6 (2007), 1663–1680.
7. Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. *Association for Computational Linguistics*.
8. Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 68–73.

# ПІДТРИМКА ПРИЙНЯТТЯ РІШЕНЬ ЗА ДОПОМОГОЮ ЗАСОБІВ DECISION INTELLIGENCE

Макаров І.В.<sup>1</sup>, Кисельов Г.Д.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> makarov@dlit.dp.ua

**Розглядається задача аналізу процесу використання засобів Decision Intelligence для підтримки прийняття рішень. Описано структуру технології та перелік переваг, які можна досягти завдяки впровадженню наявних інструментів. Розглянуто сфери застосування Decision Intelligence та описано алгоритмічну основу технології, а саме сутність глибокого навчання з підкріпленням.**

**Ключові слова: Decision Intelligence, глибоке навчання з підкріпленням, машинне навчання, Data Science.**

## 1. ВСТУП

Decision intelligence (інтелект при прийнятті рішень) — це нова сфера, яка допомагає підтримувати, доповнювати та автоматизувати бізнес-рішення, пов'язуючи дані з рішеннями та результатами. Він використовує поєднання методів (наприклад, відображення рішень і теорій прийняття рішень) і технологій (наприклад, машинне навчання та автоматизація) для покращення способів прийняття рішень у компаніях. Розвідка про прийняття рішень включає постійну оцінку результатів рішень та їх оптимізацію за допомогою системи зворотного зв'язку.

Концепція ґрунтується на ідеї, що компанії повинні ставитися до прийняття рішень як до інших сучасних бізнес-процесів і таким чином впроваджувати системи запису для моделювання, відстеження, навчання та вдосконалення конкретних управлінських рішень. Цей підхід починається з найважливіших рішень, які сприяють ефективності компанії, а потім повертається до людей, процесів і ідей, необхідних для прийняття послідовних високоякісних рішень.

## 2. РІВНІ DECISION INTELLIGENCE

Є три рівні, на яких DI може підтримувати бізнес-рішення.

Перший рівень — це підтримка прийняття рішень, на якому машини надають деякі основні інструменти для підтримки прийняття рішень людиною, такі як сповіщення, аналітика та дослідження даних. Самі рішення приймаються виключно людьми.

Другий рівень — це розширення прийняття рішень, на якому машини відіграють більшу та активнішу роль у процесі прийняття рішень. Вони аналізують дані та генерують рекомендації та прогнози для осіб, які приймають рішення, для перегляду та підтвердження. Наприклад, вони можуть дати рекомендацію на кшталт: «Вам слід купити 200 продуктів у постачальника А до 30 березня; це дозволить вам заощадити 20 000 доларів США». Люди можуть приймати рішення на основі пропозицій машини, просто прийнявши рекомендацію, або вони можуть співпрацювати з машиною, щоб змінити рекомендацію.

Третій рівень – це автоматизація прийняття рішень, яка ще більше зменшує необхідну участь людини в процесі прийняття рішень. На цьому рівні машини виконують як етап прийняття рішення, так і етап виконання автономно. На першому етапі вони приймають



автономні рішення, використовуючи комбінацію таких інструментів, як правила, оптимізація та передбачення на основі ШІ. На другому етапі вони автоматично виконують ці рішення без участі людини. Натомість люди мають огляд на високому рівні, контролюють ризики та будь-які незвичайні дії та регулярно переглядають результати для вдосконалення системи.

Це здебільшого модель зрілості з можливістю просування по рівнях. Однак не всі рішення слід автоматизувати або доповнювати. Деякі настільки чутливі, складні або рідкісні, що краще тримати їх на рівні підтримки прийняття рішень і тримати людину в курсі. Це може включати рішення високого рівня, такі як розробка нової стратегії.

Ефективна система ДІ повинна пропонувати всі три режими роботи (тобто підтримку, доповнення та автоматизацію). Це дає змогу користувачам просуватися на рівні автоматизації, коли вони розвивають довіру до технології та її можливостей.

### **3. КОМПОНЕНТИ DECISION INTELLIGENCE**

Існує цілий набір технологій і алгоритмів, які забезпечують роботу машини прийняття рішень:

Машинне навчання. Алгоритми ML працюють з певною кількістю структурованих даних і виробляють пропозиції або рішення відповідно до заданих параметрів. Найпростішим прикладом є системи боротьби з шахрайством, які використовуються банками. Наприклад, коли користувач отримує доступ до свого банківського додатку з підозрілої IP-адреси, система приймає рішення про необхідність додаткової автентифікації користувача.

Глибоке навчання. Глибоке навчання є наступним етапом еволюції машинного навчання. У цьому випадку машина прийняття рішень враховує раніше прийняті рішення та їх результати при внесенні кожної нової пропозиції.

Візуальне моделювання рішень. Прийняття рішень штучним інтелектом є надійною відправною точкою, але рішення все одно приймаються власниками бізнесу та/або їхніми працівниками. Візуальне моделювання рішень є однією з особливостей програмного забезпечення для аналізу рішень, щоб показати людям, які приймають рішення, доступні варіанти та їхні результати.

Моделювання складних систем. Однією з переваг інтелектуального прийняття рішень є швидке створення складної бізнес-логіки, керуючись наявними даними та кінцевою метою.

Прогностична аналітика. Рішення, які приймають системи ШІ, базуються на досить точних прогнозах. Найпростіший приклад – прогнозування цін і автоматизована оптимізація в роздрібній торгівлі. У цьому випадку пропозиції, зроблені системою аналізу рішень, базуються на поточних і історичних коливаннях цін, прогнозованому попиті, майбутніх тенденціях і безлічі даних про поведінку клієнтів.

### **4. СФЕРИ ЗАСТОСУВАННЯ DECISION INTELLIGENCE**

Інтелектуальні рішення трансформують організації, забезпечуючи оптимальну продуктивність у таких критичних сферах, як продажі, маркетинг, управління магазинами, управління талантами та багато іншого.

Маркетинг. Сильний бренд є результатом ретельного планування та аналізу різних точок взаємодії. Маркетологи можуть швидко виявити високоефективні ініціативи в кількох каналах від соціальних мереж до електронної пошти та оптимізувати рентабельність інвестицій для кожної кампанії, використовуючи інструменти аналізу рішень.

Продажі. Було б найкраще в режимі реального часу мати інформацію про вподобання потенційних клієнтів і те, як вони хочуть отримати інформацію про ваші продукти чи

послуги, щоб створити потенційних клієнтів для вашої організації. Ви можете отримати цінну інформацію з даних споживачів за допомогою технологій аналізу рішень і використовувати їх для адаптації повідомлень, підвищення коефіцієнтів конверсії та збільшення доходу.

Управління магазином. Вирішальне значення для забезпечення прибутковості роздрібною торгівлі має визначення точного рівня запасів і ціноутворення. Роздрібні торговці можуть використовувати інструменти аналізу рішень, щоб оптимізувати стратегії ціноутворення та рівні запасів на основі моделей попиту, гарантуючи, що вони задовольняють запити споживачів, оптимізуючи прибуток.

Управління талантами – сьогодні організації отримують пул резюме під час публікації вакансії. Залучення талановитих людей може бути одним із найскладніших завдань для вашої організації, якщо у вас обмежені ресурси. Однак ви можете використовувати програмне забезпечення для аналізу рішень, щоб перевіряти дані кандидатів і надавати пропозиції на основі їх профілю та попередніх досягнень. ДІ дозволяє рекрутерам зосередити свої зусилля на тих, хто з більшою ймовірністю досягне успіху в їхніх фірмах.

Ланцюжок поставок – Ланцюг поставок є однією з важливих сфер організації, оскільки він підтримує безперервність процесів. ДІ може творити такі чудеса, як

- Автоматизований аналіз першопричини дефектів або затримок
- Оптимізація запасів
- Прогнозування попиту
- Ефективність постачальника
- Моніторинг
- Аналітика якості

HR – ДІ може допомогти з спеціальними показниками аналізу, такими як чисельність персоналу, вивільнення, винагорода, заохочення тощо, що полегшить відділу кадрів виконання завдань.

Випадки використання Decision Intelligence в галузях

- Сектор охорони здоров'я – можна покращити медичні результати за допомогою ДІ. Ви можете отримати детальне уявлення про наступний крок лікування, краще передбачити прогноз і досягти кращих результатів, даючи лікарям суперздібність.
- Екологічний сектор – Інтелектуальні рішення можуть прогнозувати та виявляти вразливі місця на основі історичних і поточних даних. Ми можемо прогнозувати кліматичні умови з кращою точністю за допомогою ДІ. Це допомагає уникнути будь-якого лиха і озброїтися.
- Банківський і фінансовий сектор – аналіз рішень може вивчати поведінку споживачів, вимоги та больові точки та допомагає організаціям персоналізувати рішення для клієнтів. Клієнти можуть скористатися видатними методами інвестування ДІ. Потім фахівці перевіряють ці стратегії, перш ніж рекомендувати їх клієнту.
- Енергетичний сектор – ДІ допомагає користувачам краще керувати своїми енергетичними ресурсами та забезпечує автоматичне прийняття рішень щодо енергії та витрат. ДІ може передбачати сонячну енергію та відповідно регулювати ємність акумулятора.
- Страхування – ДІ може проводити андеррайтинг полісів, моделювати ризики та виявляти шахрайство, допомагаючи організаціям обробляти справжні запити та підтримувати максимальну безпеку.

## 5. ГЛИБОКЕ НАВЧАННЯ З ПІДКРІПЛЕННЯМ

Decision intelligence базується на технології Глибокого навчання з підкріпленням.

Глибоке навчання з підкріпленням — це розділ машинного навчання, який поєднує в собі навчання з підкріпленням (RL) і глибоке навчання. RL розглядає проблему обчислювального агента, який навчається приймати рішення методом проб і помилок. Deep RL включає глибоке навчання в рішення, дозволяючи агентам приймати рішення на основі неструктурованих вхідних даних без ручного проектування простору станів. Алгоритми Deep RL здатні приймати дуже великі вхідні дані (наприклад, кожен піксель, що відображається на екрані у відеогрі) і вирішувати, які дії потрібно виконати для оптимізації цілі (наприклад, максимізація результату гри). Глибоке навчання з підкріпленням використовувалося для різноманітних програм, включаючи, але не обмежуючись, робототехніку, відеоігри, обробку природної мови, комп'ютерний зір, освіту, транспорт, фінанси та охорону здоров'я.

Походження глибокого навчання з підкріпленням — це чисте навчання з підкріпленням, де проблеми зазвичай формулюються як процеси прийняття рішень за Марковим (MDP). MDP складається з набору станів  $S$  і дій  $A$ . Переходи між станами виконуються з ймовірністю переходу  $P$ , винагородою  $R$  і коефіцієнтом дисконтування  $\gamma$ . Ймовірність переходу  $P$  відображає кількість різних переходів і винагород з одного стану в інший, де послідовний стан і винагорода залежать лише від стану та дії, виконаної на попередньому кроці.

Навчання визначає середовище для виконання Агентом певних дій (відповідно до політики), щоб максимізувати винагороду. Основою оптимальної поведінки Агента є рівняння Беллмана, яке є широко використовуваним методом вирішення практичних задач оптимізації.

Коли агент існує в середовищі і переходить в інший стан, нам потрібно оцінити значення стану  $V(s)$  (позиція) — функція значення стану. Коли ми знаємо значення кожного стану, ми можемо з'ясувати, який найкращий спосіб діяти  $Q(S, A)$  — функція значення дії (просто слідкуючи за станом із найвищим значенням).

Оптимальна функція значення стану — це та, яка має вище значення порівняно з усіма іншими функціями значення (максимальна віддача), тому функцію оптимального значення також можна оцінити, взявши максимум  $Q$ :

$$v_*(s) = \max_{a \in \mathcal{A}(s)} q_*(s, a)$$

Як правило, проблеми навчання з підкріпленням вирішуються за допомогою  $Q$  — алгоритмів навчання. Тут, як було сказано вище, Агент взаємодіє з Середовищем і отримує винагороду. Мета полягає в тому, щоб розробити оптимальну політику (стратегію вибору дій), щоб максимізувати винагороду. У процесі навчання Агент оновлює таблицю  $Q(S, A)$  (припинення відбувається, коли епізод закінчився — досягнуто мети).

$Q$  — алгоритм навчання виконується за наступними кроками:

1. Ініціалізуйте таблицю  $Q(S, A)$  випадковими значеннями.
2. Виконайте дію ( $A$ ) за допомогою політики  $\epsilon$ -greedy і перейдіть до наступного стану  $S'$
3. Оновіть значення  $Q$  попереднього стану, дотримуючись рівняння оновлення:

$$Q(s, a) = \overbrace{Q(s, a)}^{\text{Old value}} + \underbrace{\alpha}_{\text{learning rate}} \underbrace{r}_{\text{reward}} + \underbrace{\gamma}_{\text{discount}} \overbrace{\max_{a'} Q(s' a')}^{\text{optimal future value}} - \overbrace{Q(s, a)}^{\text{Old value}}$$

Навчання з підкріпленням може бути застосовним до середовища, де можна керувати (ітерувати) усіма досяжними станами та зберігати їх у стандартній оперативній пам'яті комп'ютера. Однак у середовищі, де кількість станів перевищує можливості сучасних комп'ютерів (для ігор Atari є 12833600 станів), стандартний підхід Reinforcement Learning не дуже застосовний. Крім того, в реальному середовищі Агент повинен стикатися з проблемами безперервних станів (не дискретних), безперервних змінних і безперервного контролю (дій).

Беручи до уваги складність середовища, в якому агент повинен працювати (кількість станів, безперервний контроль), стандартну добре визначену таблицю Reinforcement Learning, Q — таблицю, замінено глибокою нейронною мережею (Q — мережа), яка відображає середовище станів до дій Агента. Архітектура мережі, вибір гіперпараметрів мережі та навчання виконується на етапі навчання.

## 6. ВИСНОВКИ

Decision intelligence — це поєднання існуючих технологій, зокрема штучного інтелекту та автоматизації процесів, з можливістю робити більше, ніж будь-яка з них окремо. ШІ та машинне навчання зосереджені на даних; вони можуть генерувати ідеї, але часто не пов'язані з виконанням та результатами рішень. А додатки для бізнес-процесів (зокрема роботизована автоматизація процесів, інтелектуальний аналіз процесів і виявлення процесів) орієнтовані на завдання. Вони можуть автоматизувати завдання та зробити їх ефективнішими, але можуть виконувати лише те, на що їх запрограмували, і мають обмежений вплив на ефективність рішень.

Нижче наведено основні переваги інтелектуальних рішень, на які можуть розраховувати компанії.

Рішення на основі даних. У той час як 91% компаній вважають, що прийняття рішень на основі даних може прискорити розвиток їхнього бізнесу, лише 57% з них покладаються на свої дані. Щоб отримати конкурентну перевагу, необхідно правильно проаналізувати наявні дані, зробити деякі прогнози та вибрати найкращий варіант. ШІ може краще розглянути масив даних і знайти невидимі закономірності та можливі аномалії, які можуть суттєво вплинути на результат. ДІ надає доступ до всієї необхідної інформації та інструментів, необхідних для прийняття більш обґрунтованих рішень на основі складних даних, а не на передчуттях чи інтуїційних реакціях.

Швидше прийняття рішень. Згідно з опитуванням McKinsey, лише 20% організацій задоволені своєю швидкістю прийняття рішень. Інші визнають, що витрачають надто багато часу на правильний вибір, який насправді не завжди правильний. Системи прийняття рішень ШІ максимально прискорюють процес, оскільки вони здатні майже миттєво обробляти величезні обсяги даних.

Підвищена точність – коли люди залучені, під час прийняття рішень можуть виникнути особисті упередження та неточності. ДІ пом'якшує вплив цих помилок і упереджень. Про все це піклується запрограмований алгоритм, підвищуючи точність прийняття рішень. ДІ може допомогти прийняти кращі, більш обґрунтовані рішення та уникнути конфлікту цінностей та

інтересів. Ці рішення не залежать від когнітивних упереджень і тому можуть допомогти звузати найкращі результати.

Кілька варіантів вирішення проблеми. Алгоритми прийняття рішень на основі штучного інтелекту також можуть бути досить гнучкими та виділяти кілька результатів певного рішення, коли змінюється один із параметрів. Ця функція може допомогти бізнесу зробити найкращий вибір із безлічі варіантів, враховуючи поточні цілі та стратегії розвитку.

Зменшує залежність – це позбавляє потреби залежати від аналітиків для створення звітів і інформаційних панелей і надання інформації зацікавленим сторонам. Цей метод обмежує продуктивність даних, відповідаючи лише на кілька бізнес-питань. За допомогою ДІ зацікавлена сторона сама може отримати інформацію та рекомендації, які надають безмежні можливості для прийняття високоякісних рішень.

Моніторинг і масштабованість – можливість безперервного моніторингу ДІ робить його проактивним, виявляє викиди та забезпечує персоналізований досвід для користувача. ДІ розглядає мільярди точок даних, щоб прийняти рішення. Організаціям потрібна масштабована система, оскільки дані зростають експоненціально. З цієї причини ДІ має масштабований, надійний, високошвидкісний механізм аналітики, обчислювальну структуру та засоби безпеки/контролю доступу.

## **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Is Decision Intelligence The New AI? [Електронний ресурс] - Режим доступу до ресурсу: <https://www.forbes.com/sites/forbestechcouncil/2022/05/25/is-decision-intelligence-the-new-ai> Дата доступу - 16.09.2022.

2. Why Decision Intelligence Is The Most Important Data Analytics Trend Of This Decade [Електронний ресурс] - Режим доступу до ресурсу: <https://www.atscale.com/blog/decision-intelligence-most-important-data-analytics-trend-this-decade/> Дата доступу - 16.09.2022.

3. Decision Intelligence: Diving Into One of The Top Trends of 2022 [Електронний ресурс] - Режим доступу до ресурсу: <https://statisticallyrelevant.com/decision-intelligence-diving-into-one-of-the-top-trends-of-2022/> Дата доступу - 16.09.2022.

4. Introduction to Decision Intelligence [Електронний ресурс] - Режим доступу до ресурсу: <https://www.infopulse.com/blog/introduction-decision-intelligence> Дата доступу - 16.09.2021.

5. Decision Intelligence Use Cases to Prioritize in 2022 [Електронний ресурс] - Режим доступу до ресурсу: <https://diwo.ai/blog/4-decision-intelligence-use-cases-to-prioritize-in-2022/> Дата доступу - 24.09.2022.

# **ОЦІНЮВАННЯ І ПРОГНОЗУВАННЯ СТАНУ ПАЦІЄНТІВ ТА ПОЛІПШЕННЯ ЕФЕКТИВНОСТІ ВИМІРНИХ ДАНИХ ТА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ АНАЛІЗІ НЕТОЧНИХ ТА НЕПОВНИХ ВИМІРЯНИХ ДАНИХ**

Медвідь В.А.<sup>1</sup>, Харченко К.В.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> medved.vlad01@gmail.com, <sup>2</sup> konst1970@gmail.com

**Метою дослідження є розгляд різноманітних методів, які можна використовувати для оцінювання і прогнозування стану пацієнтів, а також методів та алгоритмів прийняття рішень при неповних чи неточних даних. Буде розглянуто особливості даних методів, а також засоби мови програмування Python, котрі підходять для їх програмної реалізації. На думку авторів, розглянуті методи та програмні засоби здатні покращити та пришвидшити процес постановки діагнозу та оцінювання стану пацієнтів**

**Ключові слова: методи прогнозування, методи прийняття рішень, мова програмування Python.**

## **1. ВСТУП**

Медичні дані є одними з найцінніших і водночас найскладніших даних для аналізу. Як постачальники медичних послуг можуть використовувати сучасні інструменти та технології аналізу даних для аналізу та створення цінності складних даних? Аналітика даних, яка обіцяє ефективно виявляти цінні закономірності шляхом аналізу великої кількості неструктурованих, неоднорідних, нестандартних і неповних даних охорони здоров'я. Він не лише прогнозує, але й допомагає у прийнятті рішень і все частіше помічається як прорив у постійному прогресі з метою покращення якості обслуговування пацієнтів і зниження вартості медичної допомоги.

Через зростання витрат на охорону здоров'я рання профілактика захворювань ніколи не була такою важливою, як сьогодні. Це, зокрема, пов'язано зі збільшенням загроз нових варіантів захворювань, біотероризму, а також нещодавніх удосконалень збору даних і обчислювальних технологій. Збільшення кількості даних про охорону здоров'я збільшує попит на розробку ефективного, чутливого та економічно ефективного рішення для профілактики захворювань. Традиційні профілактичні заходи в основному зосереджені на просуванні переваг охорони здоров'я та не мають методів обробки величезної кількості даних. Використання ІТ для підвищення якості охорони здоров'я може сприяти зміцненню здоров'я та профілактиці захворювань. Це справжній міждисциплінарний виклик, який вимагає низки типів досвіду в різних областях досліджень і справді великих даних.

## **2. МЕТОДИ ПРОГНОЗУВАННЯ СТАНУ ПАЦІЄНТІВ**

Існує три основні типи методів прогнозування: якісні методи, аналіз часових рядів і прогнозування, а також моделі причинно-наслідкових зв'язків.

Перший використовує якісні дані (наприклад, експертну думку) та інформацію про особливі події, згадані вище, і може брати або не брати до уваги минуле.

Другий, з іншого боку, повністю зосереджується на шаблонах і змінах шаблонів і, таким чином, повністю покладається на історичні дані.

Третій використовує високоточну та специфічну інформацію про взаємозв'язки між елементами системи та є достатньо потужним, щоб формально враховувати спеціальні події. Як і в аналізі часових рядів і методах прогнозування, минуле є важливим для причинно-наслідкових моделей.

### **2.1. Якісні методи**

В основному вони використовуються, коли даних мало, наприклад, коли продукт вперше виходить на ринок. Вони використовують людське судження та рейтингові схеми, щоб перетворити якісну інформацію на кількісні оцінки.

Мета тут полягає в тому, щоб логічним, неупередженим і систематичним чином об'єднати всю інформацію та судження, які стосуються факторів, що оцінюються. Такі методи часто використовуються в сферах нових технологій, де розробка ідеї продукту може вимагати кількох «винаходів», так що потреби в дослідженнях і розробках важко оцінити, і де рівень прийняття на ринку та проникнення вкрай невизначений.

### **2.2. Аналіз часових рядів**

Це статистичні методи, які використовуються, коли доступні дані за кілька років для продукту чи лінійки продуктів і коли зв'язки та тенденції чіткі та відносно стабільні.

Одним із основних принципів статистичного прогнозування — фактично будь-якого прогнозування, коли доступні історичні дані — є те, що прогнозіст повинен використовувати дані про минулі показники, щоб отримати «показник спідометра» поточного темпу і як швидко ця швидкість зростає або зменшується. Основою прогнозу є поточний темп і його зміни — «прискорення» і «уповільнення». Коли вони відомі, різні математичні методи можуть розробити проєкції з них.

Зазвичай складно робити прогнози на основі необроблених даних, оскільки показники та тенденції не відразу очевидні; вони змішуються, наприклад, із сезонними коливаннями і, можливо, спотворюються різними факторами. Необроблені дані необхідно обробити, перш ніж їх можна буде використовувати, і це часто робиться за допомогою аналізу часових рядів.

Зараз часовий ряд — це набір упорядкованих у хронологічному порядку точок необроблених даних — наприклад, продажі певного продукту підрозділом за місяцями за кілька років. Аналіз часових рядів допомагає визначити та пояснити:

- 1) Будь-яка регулярність або систематична зміна в ряді даних, яка зумовлена сезонністю — «сезонність».
- 2) Циклічні моделі, які повторюються через будь-які два-три роки або більше.
- 3) Тенденції в даних.
- 4) Темпи зростання цих тенденцій.

Після завершення аналізу можна починати роботу з прогнозування. Слід зазначити, більшість методів статистичного прогнозування фактично поєднують обидві функції в одній операції.

### **2.3. Моделі причинно-наслідкових зв'язків**

Коли доступні історичні дані та проведено достатній аналіз, щоб чітко визначити зв'язки між фактором, який потрібно прогнозувати, та іншими факторами, прогнозіст часто буде причинно-наслідкову модель.

Причинно-наслідкова модель є найдосконалішим інструментом прогнозування. Вона математично виражає відповідні причинно-наслідкові зв'язки та може включати міркування щодо конвеєра (тобто запаси) та інформацію з огляду ринку. Вона також може безпосередньо включати результати аналізу часових рядів.

Причинно-наслідкова модель бере до уваги все, що відомо про динаміку потокової системи, і використовує передбачення пов'язаних подій. Якщо дані доступні, модель зазвичай включає фактори для і з'єднує їх рівняннями для опису загального потоку продукту.

Якщо певних типів даних бракує, спочатку може знадобитися зробити припущення щодо деяких зв'язків, а потім відстежити, що відбувається, щоб визначити, чи припущення відповідають дійсності. Як правило, причинно-наслідкова модель постійно переглядається, оскільки стає доступним більше знань про систему.

### 3. МЕТОДИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ НЕПОВНИХ ТА НЕТОЧНИХ ДАНИХ

Коли ми маємо неповні чи неточні дані, тоді використовуються наступні метод та алгоритми прийняття рішень.

#### 3.1. Критерій Лапласа

Критерій Байеса-Лапласа — один з критеріїв прийняття рішень в умовах невизначеності. Умовами невизначеності вважається ситуація, коли наслідки прийнятих рішень невідомі, і можна лише приблизно їх оцінити. За цим критерієм множина оптимальних альтернатив знаходиться так: критерій передбачає існування імовірнісних мір  $p(x,s)$  на  $X \times S$ ,  $u(x,s)$  де — імовірнісна міра на декартовому добутку  $X \times S$ , де  $X$  — множина альтернатив,  $S$  — множина станів, які до того ж є стабільними протягом тривалого періоду часу.

Для того, щоб це було це було так ЗПР повинна бути добре досліджена статистично(на основі тривалих або частих спостережень).

Отже спочатку обчислюється:

$$E(X_k) = \sum_{j=1}^N u_{kj} p_{kj}, k = \overline{1, N}$$

де  $u(x,s)$  — функція рішень, визначена на  $X \times S$ , де  $X$  — множина альтернатив,  $S$  — множина станів, а  $p(x, s)$  — ймовірнісна міра ситуації  $fx, sg$

Для скінченновимірною випадку набуває:

$$E(X_k) = \sum_{j=1}^N u_{kj} p_{kj}, k = \overline{1, N}$$

де  $p_{kj}$  імовірність ситуації  $\{x_k, s_j\}$ , де  $u_{kj}$  матриця рішень.

Далі вже множина оптимальних альтернатив визначається так:



$$X_{BL} = \arg \max_{x \in X} E(x)$$

$$X_{BL} = \arg \max_{k=1, N} E(x_k)$$

### 3.2 Критерій Вальда

Критерій Вальда є критерієм крайнього песимізму, оскільки статистик вважає, що "природа" діє проти нього найгіршим чином. Це критерій гарантованого результату. Нехай гру задано матрицею виграшів гравця А. Тоді на думку статистика - гравця А, дії гравця "природа", якій діє проти нього найгіршим чином, відображуються в реалізації гравцем "природа" таких своїх стані П<sub>і</sub>, при яких величина виграшу гравця А (статистика) приймає найменше значення min a<sub>ij</sub>. Виходячи з цього статистик обирає таку чисту стратегію А<sub>і</sub>, при якій найменший виграш min a<sub>ij</sub> буде максимальним.

Величина α<sub>В</sub>, що розраховується за вищенаведеною формулою, називається нижньою ціною гри - це максимальний виграш, що є гарантованим в грі з певним противником шляхом вибору однієї зі своїх стратегій при мінімальних результатах. Нехай гру задано матрицею програшів гравця А, тоді найгірші дії гравця "природа", будуть реалізовуватися в таких станах П<sub>і</sub>, при яких величина програшу гравця А (статистика) приймає найбільше значення max a<sub>ij</sub>. Виходячи з цього статистику необхідно обрати таку чисту стратегію А<sub>і</sub>, при якій найбільший програш max a<sub>ij</sub> буде мінімальним.

### 3.3 Критерій песимізму

Якщо суб'єкт управління орієнтується на вкрай несприятливий розвиток подій, для обґрунтування рішення застосовується критерій песимізму. Цей критерій передбачає визначення варіанту рішення, що дозволяє мінімізувати мінімальні виграші для кожного варіанту економічного середовища, й за вихідними даними матриці виграшів у формалізованому виразі має вигляд:

$$\alpha_s = \min_i \min_j a_{ij}$$

За умови застосування даних матриці ризиків визначення критерію песимізму передбачає знаходження варіанту рішення, який максимізує максимальні програші для кожного варіанту економічного середовища:

$$\beta_s = \max_i \max_j H_{ij}$$

### 3.4 Критерій Севіджа

*Мінімакський критерій Севіджа* використовується у випадках, коли потрібно за будь-яких умов уникнути великого ризику. Відповідно до цього критерію перевагу слід надати варіанту рішення, для якого максимальні втрати за різних варіантів стану економічного середовища виявляються мінімальними. Його формалізований вираз має вигляд:

$$\alpha_c = \min_i \max_j H_{ij}$$

### 3.5 Критерій Гурвіца

Критерій узагальненого максиміна (песимізму-оптимізму) Гурвіца використовується, якщо потрібно знайти певну комбінацію оптимістичної та песимістичної позицій щодо прийняття рішення. Відповідно до цього критерію перевагу надають варіанту рішення, для якого виконується умова:

$$H = \max_i \left( k \min_j X_{ij} + (1-k) \max_j X_{ij} \right),$$

де  $k$  – коефіцієнт, що розглядається як показник песимізму  $0 \leq k \leq 1$ ,  $a_{ij}$  – виграш, що відповідає  $i$ -му рішенню при  $j$ -му варіанті стану економічного середовища.

Значення коефіцієнту песимізму встановлюються суб'єктивно, залежно від конкретних обставин та схильності до ризику особи, що приймає рішення. При  $k = 0$  має місце оптимістична позиція та орієнтація на граничний ризик, а величина критерію Гурвіца співпадає з величиною критерію максимакса. При  $k = 1$  суб'єкт управління налаштований песимістично та прагне уникати ризику, а величина критерію Гурвіца співпадає з величиною максимінного критерію Вальда. Значення  $k$  між 0 і 1 є проміжними між ризиком і обережністю. Чим більша можлива небезпека, тим більш наближуватиметься до одиниці значення коефіцієнта песимізму. При зміні значення коефіцієнта песимізму змінюватиметься й варіант рішення, якому надається перевага. Отже, схильність до ризику суб'єкта управління значною мірою впливає на вибір рішення.

## 4. ПРОГРАМНІ ЗАСОБИ ДЛЯ РЕАЛІЗАЦІЇ МЕТОДІВ ТА АЛГОРИТМІВ

Для програмної реалізації вищерозглянутих методів якнайкраще підходить мова програмування Python, з її величезним різноманіттям модулів. Саме велика кількість різних модулів та бібліотек дозволяє цій мові бути гнучкою для вирішення різноманітних завдань, наприклад обробка великих масивів даних, машинне навчання і так далі. З огляду на специфіку розглянутої теми та методів, можна зробити висновок, що мова програмування Python ідеально підходить, так як має засоби для обробки та аналізу чималих об'ємів даних з використанням розглянутих методів. Розглянемо далі які модулі якнайкраще підходять для програмної реалізації вище зазначених методів та алгоритмів.

### 4.1 NumPy

NumPy це open-source модуль для python, який надає загальні математичні та числові операції у вигляді пре-компільованих, швидких функцій. Вони поєднуються у високорівневі пакети. Вони забезпечують функціонал, який можна порівняти із функціоналом MatLab. NumPy (Numeric Python) надає базові методи для маніпуляції з великими масивами та матрицями. SciPy (Scientific Python) розширює функціонал numpy величезною колекцією корисних алгоритмів, таких як мінімізація, перетворення Фур'є, регресія та інші прикладні математичні техніки.

Навіщо використовувати NumPy?

У Python у нас є списки, які служать для цілей масивів, але вони повільно обробляються. NumPy має на меті надати об'єкт масиву, який у 50 разів швидший за традиційні списки Python. Об'єкт масиву в NumPy називається ndarray, він надає багато допоміжних функцій, які роблять роботу з ndarray дуже легкою. Масиви дуже часто використовуються в науці про дані, де швидкість і ресурси дуже важливі.

Чому NumPy швидший за списки?

Масиви NumPy зберігаються в одному безперервному місці в пам'яті, на відміну від списків, тому процеси можуть отримувати до них доступ і маніпулювати ними дуже ефективно.

Така поведінка в інформатиці називається локальністю посилання.

Це головна причина, чому NumPy швидше, ніж списки. Крім того, він оптимізований для роботи з останніми архітектурами ЦП.

## 4.2 SciPy

SciPy в Python — це бібліотека з відкритим кодом, яка використовується для вирішення математичних, наукових, інженерних і технічних задач. Це дозволяє користувачам маніпулювати даними та візуалізувати дані за допомогою широкого спектру високорівневих команд Python. SciPy побудовано на розширенні Python NumPy.

Навіщо використовувати SciPy?

Якщо SciPy використовує NumPy нижче, чому ми не можемо просто використовувати NumPy? SciPy оптимізував і додав функції, які часто використовуються в NumPy і Data Science.

## 4.3 Matplotlib

Що означає Matplotlib?

Matplotlib — це бібліотека для побудови графіків, доступна для мови програмування Python як компонент NumPy, ресурсу чисельної обробки великих даних. Matplotlib використовує об'єктно-орієнтований API для вбудовування графіків у програми Python.

Оскільки Python широко використовується в машинному навчанні, такі ресурси, як NumPy і matplotlib, часто корисні для моделювання технологій машинного навчання. Ідея полягає в тому, щоб програмісти отримували доступ до цих бібліотек для виконання ключових завдань у ширшому середовищі Python та інтегрували результати з усіма іншими елементами та функціями програми машинного навчання, нейронної мережі чи іншої вдосконаленої машини. Утиліта NumPy і matplotlib пов'язана з числами — утиліта matplotlib конкретно пов'язана з інструментами візуального малювання. Тож у певному сенсі ці ресурси більше аналітичні, ніж генеративні. Однак уся ця інфраструктура працює разом, щоб дозволити програмам машинного навчання давати результати, корисні для людей, які працюють із ними.

## 5. ВИСНОВКИ

Постійний розвиток технологій дозволяє покращувати різноманітні сфери людського існування, якою і являється медицина. На основі розглянутих методів прогнозування та прийняття рішень, а також з використанням можливостей мови програмування Python, створюється можливість створювати програми та системи які можуть ефективно оцінювати та прогнозувати стан різних пацієнтів, приймати рішення щодо подальшого лікування чи стану пацієнту, а також покращувати дані, навіть за умови, коли вони неточні та неповні, так як не завжди є можливість отримати повний набір даних, який би ідеально підходив, як вхідні дані.

У даній роботі розглянуто як методи так і програмні засоби, які здатні покращити якість обслуговування у сфері медицині, зекономити час та кошти для запобігання, лікування а також профілактики різноманітних захворювань.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. «How to Choose the Right Forecasting Technique» by John C. Chambers, Satinder K. Mullick, and Donald D. Smith, URL: <https://hbr.org/1971/07/how-to-choose-the-right-forecasting-technique>.
2. Принятие решений при неполной информации, URL: [https://studwood.net/1682726/matematika\\_himiya\\_fizika/prinyatie\\_resheniy\\_nepolnoy\\_informatsii](https://studwood.net/1682726/matematika_himiya_fizika/prinyatie_resheniy_nepolnoy_informatsii)
3. Критерії прийняття рішень в умовах невизначеності, URL: [https://stud.com.ua/34760/finansi/kriteriy\\_laplasa](https://stud.com.ua/34760/finansi/kriteriy_laplasa)
4. NumPy Introduction, URL: [https://www.w3schools.com/python/numpy/numpy\\_intro.asp](https://www.w3schools.com/python/numpy/numpy_intro.asp)
5. SciPy in Python, URL: <https://www.guru99.com/scipy-tutorial.html>
6. What is Matplotlib in Python?, URL: <https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/>

# ПЕРЕВІРКА ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ У АВТОМАТИЗАЦІЇ РЕГРЕСИВНОГО ТЕСТУВАННЯ

Михайловин Р. Г.<sup>1</sup> Булах Б. В.<sup>2</sup>

КПІ ім. Ігоря Сікорського, Київ, Україна

<sup>1</sup>mihailovinroman@gmail.com , <sup>2</sup>bogdan.bulakh@gmail.com

**Метою дослідження є перевірка ефективності використання машинного навчання на прикладі нейронних мереж у автоматизації регресійного тестування. У доповіді представлено опис причин та аргументи на користь даного підходу, та результати тестової реалізації. На думку автора, залучення подібних методів до автоматизації тестування здатне прискорити та спростити ітерації регресивного тестування.**

**Ключові слова: нейронні мережі, машинне навчання, регресійне тестування, автоматизація.**

## 1. ВСТУП

Розробка програмного забезпечення - складний і трудомісткий процес, в ході якого неодмінно виникають помилки. Частина з них так і не будуть виявлені, поки програмний продукт не потрапить до кінцевого користувача. Для того щоб скоротити число прихованих проблем, застосовуються різні методи тестування ПЗ.

Важливою частиною тестування програмного продукту є - регресивне тестування.

Регресивне тестування— збірна назва для різних способів тестування ПЗ, що мають на меті виявлення помилок у вже протестованих ділянках коду. Помилки, що виникають при внесенні змін до програми називають регресивними.

В умовах, коли нові версії програмного забезпечення з'являються одна швидше іншої, вимагаючи скорочення часу виходу на ринок і підвищення якості, автоматизація регресійного тестування є логічним і ефективним кроком.

Одним з варіантів автоматизації є використання нейронних мереж.

Добре навчена нейронна мережа здатна передбачати результат по набору вхідних значень. Разом з цим додатково вона вирішує завдання класифікації, якщо заздалегідь задані класи у вхідній множині, або завдання кластеризації, тобто виявлення класів в множині.

## 2. РЕГРЕСИВНЕ ТЕСТУВАННЯ

### 2.1. Завдання регресивного тестування

Основне завдання регресивного тестування - перевірка того, що виправлення помилки не торкнулося існуючої функціональності. Через часте виконання одних і тих же наборів сценаріїв, рекомендується використовувати автоматизовані регресивні тести, що дозволить скоротити терміни тестування.

- перевірка та затвердження виправлення помилки;
- тестування наслідків виправлень, так як внесення виправлення можуть принести помилку в код який виконувався правильно;
- гарантувати функціональну спадкоємність і сумісність нової версії (релізу, патча) з попередніми;

- зменшення вартості і скорочення часу виконання тестів.

## **2.2. Використання регресивного тестування**

Для регресивного тестування використовуються тест-кейси, написані на попередніх стадіях розробки і тестування. Це дає гарантію того, що зміни в новій версії програми не пошкодили вже існуючу функціональність. Рекомендується проводити автоматизацію регресійних тестів, для прискорення подальшого процесу тестування і виявлення дефектів на ранніх стадіях розробки програмного забезпечення.

Проводити регресивне тестування, слід після будь-якої зміни функціоналу, для того, щоб переконатися у відсутності нових і / або усунення попередніх помилок. Включення блочного регресивного тестування в процес розробки дозволяє захиститися від помилок. Баги будуть виявлені відразу після виникнення і не зможуть стати причинами поширення помилок в додатку. Перевірка цілісності проекту після внесення змін призначена для того, щоб протестувати загальний функціонал оточення, в якому були зроблені зміни.

Повторна поява одних і тих же помилок при розробці програмного забезпечення досить часто явище. Це відбувається через помилки при роботі з системою контролю версій або через людські помилки. Але настільки ж часто вирішення проблеми буває «недовготривалим»: рішення може працювати до наступної зміни в програмі. І нарешті, при зміні частини коду часто можуть з'являтися ті ж помилки, що були в попередній реалізації [1].

Спочатку потрібно обрати спосіб тестування програми. Після того як обраний спосіб тестування програми, його потрібно оптимізувати, інакше ефективність дій буде мінімальною.

Для цього існують такі поширені і дієві методи:

1. Із застосуванням дворівневого підходу.
2. Із застосуванням сортування тест-кейсів за пріоритетом.
3. За допомогою автоматизованого підходу.

## **2.3. Методи оптимізації**

Метод оптимізації із застосуванням дворівневого підходу полягає в тому, що регресійне тестування розбивається на два етапи. На першому етапі кожен тестувальник зосереджується на тих ділянках коду, які були недавно змінені або створені. В цей час розглядаються всі внесені зміни і їх вплив на загальну функціональність проекту (іmpact analysis). На наступному етапі проводиться повне тестування ПЗ на всіх ділянках коду. Зазвичай цю процедуру виконують перед випуском нової версії, щоб бути повністю впевненими в працездатності програми. На перший погляд може здатися, що цей етап досить простий. Однак тут є свої нюанси, які обов'язково необхідно враховувати. Безперервна комунікація з розробниками програмного продукту дозволить своєчасно виявити тонкі місця в системі. Завдяки цьому забезпечується можливість проводити ефективні тести і економити час на необов'язкових перевірках.

Другий метод оптимізації із застосуванням сортування тест-кейсів за пріоритетом базується на виборі ділянок коду для тестування, в яких були зроблені останні зміни. При цьому кожен тест-кейс сортується за пріоритетністю. Вибрані тест-кейси можуть мати такі пріоритети:

- високий;
- середній;
- низький.

Тестувальники включають у високий пріоритет тести, які перевіряють критичні ділянки програми, що відповідають за її головні функції. Тестами із середнім пріоритетом перевіряються області коду, в яких раніше були виявлені недоліки. Тести з низьким пріоритетом в основному використовують перед великим випуском нової версії програми.

Метод оптимізації за допомогою автоматизованого підходу полягає в правильній розстановці етапів тестування в процесі розробки проекту. На ранній стадії розробки важливо проводити функціональне тестування тільки після того, як програмний продукт досяг певної зрілості і може виконувати покладені на нього функції. Також варто враховувати, що, незважаючи на високий рівень продуктивності автоматизованих тестів і економію часу, вони не завжди залишаються актуальними. Тому варто після випуску нових версій програмного продукту переглядати автоматизовані тести і в разі необхідності замінювати застарілі на нові пакети. Завдяки такому підходу можна забезпечити високу якість програмної продукції [2].

### **3. МАШИННЕ НАВЧАННЯ І НЕЙРОННІ МЕРЕЖІ**

#### **3.1 Машинне навчання**

У випадку автоматизації процедур тестування програмного забезпечення, стандартне машинне навчання і, більш конкретно, методи глибокого навчання, такі як когнітивні нейронні мережі, можуть бути навчені даними, що генеруються діями користувача, у поєднанні з відповідним результатом тестованої програми. Для цього дії тестера, який проводить регресійне тестування, є гілками в нейронній мережі, тоді як елементи сторінки - це вузли. Коли вимірний результат згодом кваліфікується в контрольованих навчальних моделях, він може бути застосований для прогнозування результатів для майбутніх циклів регресійного тестування, а також для автономного генерування нових тестових випадків. Значення підготовки моделей глибокого навчання для прогнозування введення користувачем і вихідних даних системи зростає і стає все більш точним, оскільки дані накопичуються в кожному тестовому циклі [3].

#### **3.2 Нейронні мережі**

Нейронна мережа являє собою систему  $N$  розподілених елементів, пов'язаних між собою змінними зв'язками. Ці елементи називаються формальними нейронами і характеризуються рівнем збудження. Нейрон - це атомарна одиниця нейронної мережі. Отримавши дані на вхід, він обробляє їх і передає вихід, який є входом для наступного рівня нейронів. Зв'язки між нейронами визначені коефіцієнтами передач.

Штучна нейронна мережа має три складові:

- Вхідний шар;
- Приховані (обчислювальні) шари;
- Вихідний шар.

Іншими словами, нейронна мережа - це математична модель, заснована на когнітивних процесах мозку. Так само як і мозок, нейронна мережа складається з вузлів (нейронів) і змінних зв'язків між ними (ваги нейронних зв'язків). Нейрони поділяються на 3 групи за своїми властивостями: вхідні нейрони, вихідні нейрони і нейрони прихованого шару (обчислювальні вузли). Фактично нейронна мережа - це особливий спосіб завдання функції[4].

Мережі бувають:

1. Одношарові. У даній структурі сигнали зі вхідного шару відразу спрямовуються на вихідний шар, який, перетворюючи сигнал, видає відповідь. Таким чином, 1-й вхідний шар тільки приймає і розподіляє сигнали, а другий шар проводить обчислення та

видає результат. Вхідні нейрони об'єднані з основним шаром за допомогою зв'язків, які називають синапсами з різними вагами, що забезпечує якість.

2. Багатошарова нейронна мережа. Має, між вхідним та вихідним ще декілька шарів. Такі шари називаються прихованими. Такі мережі мають більший функціонал ніж одношарові, але також навчаються значно повільніше [5].

Також нейронні мережі поділяють за такими критеріями:

За типом нейронів:

- Однорідні
- Гібридні

За методом навчання:

- Навчання з учителем
- Навчання без учителя
- Навчання з підкріпленням

За типом вхідних даних:

- Аналогові
- Двійкові
- Образні

За налаштуванням синапсів:

- З фіксованими зв'язками
- З динамічними зв'язками

Для навчання нейронній мережі дають приклади – навчальні вибірки. Представницькі дані підбираються користувачем, після чого запускається алгоритм навчання. Для навчання нейронної мережі потрібен набір евристичних знань про відбір і підготовку даних, вибір потрібної архітектури мережі та аналізу результатів, хоча, для успішного застосування нейронних мереж, потрібен менший об'єм знань, ніж, наприклад, при використанні традиційних методів статистики.

Привабливість нейронних мереж зумовлена їх інтуїтивністю, адже вони засновані на примітивній біологічній моделі нервових систем [6].

Процес навчання достатньо простий. Підготовується навчальна вибірка - множина пар вхідних та вихідних векторів, із завчасно відомими та правильними значеннями. Елементи цієї множини задовольняють наступним властостям - незалежність, послідовність, вибірку можна розширити, а її елементи можуть бути переставлені у будь-якому порядку. Навчальну вибірку пропускають через нейронну мережу і в результаті цього процесу ваги в нейронних зв'язках змінюються таким чином, щоб задати функцію, що задовольняє навчальну вибірку. Це призводить до того, що якість роботи нейромереж залежить від якості її навчання.

Далі готують тестову вибірку аналогічну навчальній, з іншими значеннями. Якщо результат тестування невідповідний, варто краще навчити мережу або змінити її структуру.

Нейромережеві технології спільно з широко використовуваними методами автоматичного тестування цілком здатні поліпшити результати автоматичного тестування в задачах тестування складних програмних систем [7].

## 4. ОПИС ПРАКТИЧНОЇ ЧАСТИНИ

Для дослідження було обрано багатошарову нейронну мережу навчену за методом – навчання з учителем.



Навчання з учителем (supervised learning) потребує наявності повного набору розмічених даних для тренування моделі на всіх етапах її навчання.

Це означає, що для кожного прикладу з навчального набору існує відповідь, що йому відповідає, її алгоритм і повинен отримати. Таким чином, мережу навчають. Потім, коли мережа отримує, наприклад, новий набір змін, вона порівнює його з прикладами з навчального датасета, щоб передбачити відповідь.

Зазвичай навчання з учителем використовується для задач регресії і класифікації.

Таким чином, навчання з учителем найбільше ефективно, коли є значний набір достовірних даних для навчання алгоритму. На жаль такі випадки досить рідкісні. Брак даних – одна з найбільших проблем у машинному навчанні [8].

Для дослідження було проведено експеримент у ході якого було розроблено програму з використанням нейронних мереж. Завдання програми було при отриманні набору внесених змін, обрати найбільш релевантні для проходження тест-кейси. Отже вхідними даними для програми є – тип внесених змін, а результатом її роботи – набір тест-кейсів обов'язкових до виконання. Схему роботи програми можна побачити на Рис. 1.

Програма представляє з себе набір з нейронних мереж типу багатошаровий перцептрон. Результати її тестування зображені на Рис. 2.

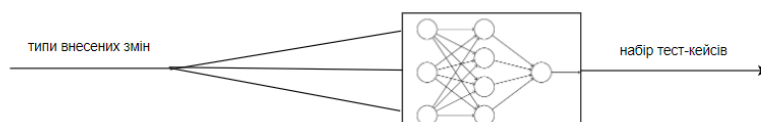


Рисунок 1. Схеми роботи програми

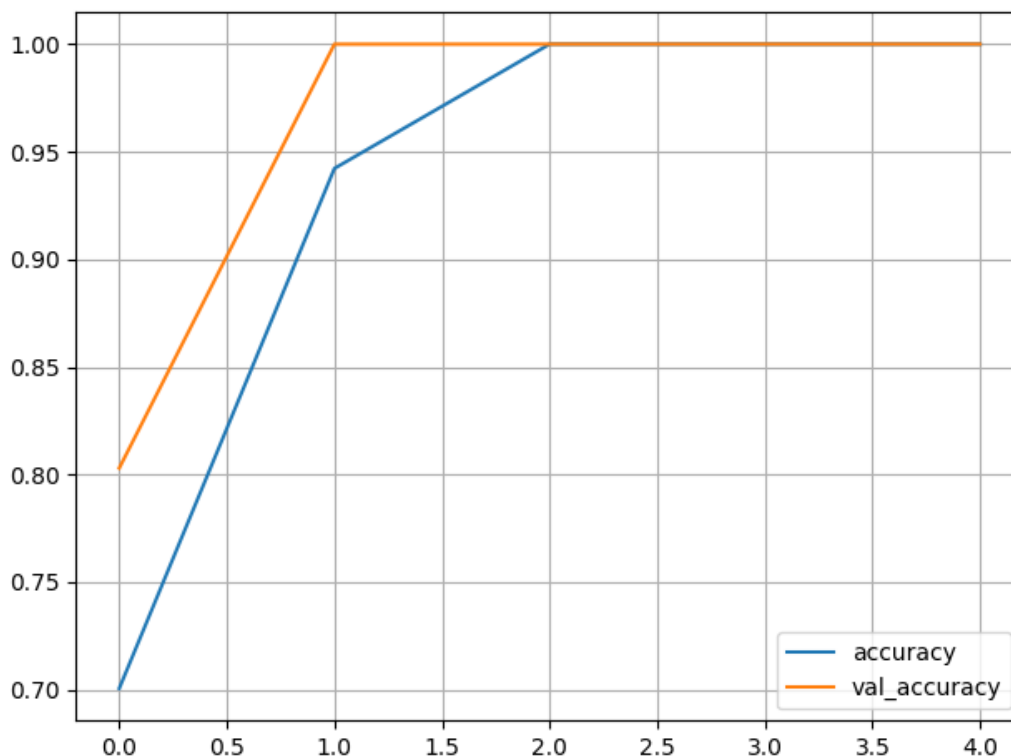


Рисунок 2. Точність програми на тестових даних.

Як бачимо, правильно навчені нейронні мережі показують високу точність. До переваг даної системи відносимо:

- Можливість паралельного навчання мереж
- Висока точність
- Можливість додати мережу без необхідності перенавчати інші

Недоліками подібної системи є:

- Необхідність у розмічених навчальних даних
- Необхідність перенавчання системи при зміні набору вхідних даних

## 5. ВИСНОВКИ

Як бачимо на прикладі даної програми, використання машинного навчання може бути ефективним у автоматизації регресивного тестування, доведення чого і було ціллю дослідження. Даний підхід може бути експериментально порівняний з іншими методами автоматизації для обрання найбільш ефективного підходу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Лучшие практики автоматизации тестирования URL: <http://getbug.ru/luchshie-praktiki-avtomatizatsii-testirovaniya/>
2. Особенности регрессионного тестирования программ А. Коттов URL: <https://dnipro.deveducation.com/blog/osobennosti-regressionnogo-testirovaniya-programm/>
3. Automated GUI Regression Testing Using AI Planning URL: <https://www.functionize.com/blog/how-ai-impacts-regression-testing/>
4. Применение нейронных сетей для построения адаптивных систем управления технологическими процессами А. Юдашкин URL: <http://tekhnosfera.com/primenenie-neuronnyh-setey-dlya-postroeniya-adaptivnyh-sistem-upravleniya-tehnologicheskimi-protsessami#ixzz6q0IpkIxO>
5. Типы нейронных сетей. Принцип их работы и сфера применения А. Павленко URL: <https://otus.ru/nest/post/1263/>
6. Нейронные сети URL: <http://statsoft.ru/home/textbook/modules/stneunet.html>
7. Использование нейронных сетей в тестировании сложных программных систем Бабин Д. В. URL: [https://www.okbsapr.ru/library/publications/babin\\_tezisy2013/](https://www.okbsapr.ru/library/publications/babin_tezisy2013/)
8. Обучение нейросети с учителем, без учителя, с подкреплением – в чём отличие? Какой алгоритм лучше? К. Беликова URL: <https://neurohive.io/ru/osnovy-data-science/obuchenie-s-uchitelem-bez-uchitelja-s-podkrepleniem/>

# СИСТЕМА МОНІТОРИНГУ В ЕЛЕКТРОННІЙ МЕДИЧНІЙ СИСТЕМІ

Музика О.А.<sup>1</sup>, Булах Б.В.<sup>2</sup>

ІІСА, КПІ ім. Ігоря Сікорського, Київ, Україна

<sup>1</sup>oleksandr.muzyka.a@gmail.com, <sup>2</sup>bogdan.bulakh@gmail.com

**Предметом дослідження є проблематика побудови системи дистанційного медичного моніторингу. Актуальність теми визначається потребою пошуку рішення для організації збору, зберігання та візуалізації медичних даних в моніторинговій системі, яка буде інтегрована в загальну інформаційну систему охорони здоров'я. В даній роботі розглянуто основні пристрої та підходи для проведення моніторингу, які використовуються як в клінічній практиці, так і для повсякденних потреб. Наведено огляд існуючих підходів до організації локального моніторингу в медичних закладах та підходи до організації дистанційного моніторингу. Враховуючи переваги та недоліки оглянутих архітектур, запропоновано концепцію моніторингової системи, яка може бути інтегрована в комплексну медичну систему.**

**Ключові слова:** медичний моніторинг, інтернет речей, телемедицина, системи моніторингу.

## 1. ВСТУП

Останні здобутки інформаційних та комп'ютерних технологій в області інтернету речей, бездротових мереж, портативних девайсів та мініатюрних сенсорів дають простір для розробки методів дистанційного моніторингу пацієнтів, аналізу великої кількості медичних метрик для клінічних досліджень та підбору персоналізованого лікування, ефективного амбулаторного та постклінічного супроводу пацієнта тощо. Даний вектор розвитку є актуальним, адже попри розширення технологічних можливостей біомедичної науки та покращення розуміння патологічних процесів та методів їх лікування, проблеми обмеженості ресурсів сфери охорони здоров'я, які виливаються в економічні та логістичні бар'єри в доступі до медичних послуг, як в країнах, що розвиваються [1], так і в розвинених країнах [2], залишаються невирішеними. Часткова автоматизація процесів діагностики, супроводу пацієнтів, створення доступних широкому загалу носимих девайсів та розвиток систем моніторингу для контролю здоров'я може зарадити цій проблемі шляхом цифровізації процесів, які вимагають дорогого часу медичних працівників та знизить необхідність особистої присутності в медичних закладах для діагностики та обстеження. Також впровадження сенсорів в повсякденне життя може стати джерелом великої кількості даних, які матимуть змогу використовувати аналітики та фахівці з машинного навчання для побудови прогностичних моделей та дати додаткові інсайти щодо здоров'я населення в статистичному переїзді та особливостей клініки окремих захворювань. [3]

## 2. МОНІТОРИНГОВІ ПРИСТРОЇ ТА ДАНІ

В контексті розробки ПЗ для моніторингу за станом здоров'я пацієнтів можна виділити дві категорії приладів та систем, які надають інформацію про ті чи інші показники: клінічні моніторингові системи та портативні девайси.

До першої категорії відносяться спеціалізовані прилади для вимірювання окремих показників та комплексні системи, які вимірюють ряд показників та мають базове ПЗ для аналізу даних: сигналізація про критичний стан, співставлення декількох показників для виявлення конкретної проблеми. Як правило, такі системи та прилади постійно використовуються в інтенсивній терапії та під час оперативного втручання чи інших медичних процедур. До них відносять інвазивні (монітор внутрішньочерепного тиску чи монітор для катетеру легенової артерії, втрачає популярність в медичній практиці) та неінвазивні кардіологічні монітори (пульсометри, ЕКГ, монітор Доплера), аналізатори крові (лейкоцитарної формули, рівню вмісту окремих речовин), респіраторні монітори та інші. [4]

До другої категорії відносять спеціальні прилади для зчитування метрик з подальшою передачею їх в хмарне сховище та універсальні пристрої загального призначення, що мають функціонал для проведення моніторингу. Окремі спеціалізовані прилади можуть використовуватись при заняттях спортом для контролю пульсу, на регулярній основі в побуті для контролю здоров'я (наприклад, прилади для вимірювання тиску, пульсоксиметри, термометри, глюкометри та ін.) а також можуть тимчасово видаватись пацієнтам для діагностики (наприклад, монітор Холтера дає зафіксувати серцевий ритм протягом доби в різних ситуаціях, як-то в стані спокою, під час сну, під час фізичної активності та ін.). [5] Також варто відзначити універсальні пристрої, наприклад смартфон, які, за наявності ПЗ та технічного функціоналу, дають змогу проводити моніторинг: вимірювання пульсу за допомогою відеофіксації, логування рутинних звичок (контроль раціону, контроль прийому медикаментів), трекінг фізичної активності (зокрема ходьби, за допомогою GPS) тощо.

Таке різномайття ПЗ та приладів для моніторингу породжує велику кількість даних, аналіз яких значною мірою ускладнює їх великі обсяги та варіативність структури. Природу та характер даних, які фігурують в таких приладах та системах, варто розглянути більш детально.

По-перше, в залежності від цільової метрики відрізняється тип та структура моніторингових даних. Частина даних за своєю природою є цілочисельним типом. До таких даних відносяться показники пульсу, кількості подихів за хвилину, для більшості приладів артеріальний тиск та ін. Також є показники, які описуються дійсним числом, як-то показники в лейкоцитарній формулі, рівень глюкози в крові, оксигенація, тиск. Інші дані можуть бути булеву природу, категоріальну (де показник описується як одне якісне значення з обмеженої кількості можливих), можуть бути описані зображенням чи відеорядом, мати складну багаторівневу структуру (наприклад, зведений звіт декількох показників), або структура може бути відсутня зовсім (наприклад, суб'єктивне відчуття пацієнта, яке він зазначає кожен день в мобільному додатку).

По-друге, такі дані відрізняються періодичністю замірів. В інтенсивній терапії ключові вітальні показники вимірюються безперервно з певним проміжком часу (одна чи декілька секунд); при супроводі пацієнта, який приймає медикаменти з серйозними побічними ефектами, може братись аналіз крові щодня чи раз на декілька днів; для пацієнтів з онкологічним захворюванням в анамнезі може раз на декілька тижні проводитись тест на рівень онкомеркерів та/або раз на рік чи пів року проводитись КТ/МРТ.

По-третє, моніторингові дані відрізняються періодом своєї актуальності. Приміром, показники в інтенсивній терапії є актуальними лише в околі критичного стану, аби лікар мав змогу його коректно визначити; виміри тиску при прийомі препаратів для визначення їх ефективності на пореби корегування актуальні в контексті курсу прийому препаратів; результати КТ/МРТ при веденні пацієнта з онкологією в ремісії актуальні протягом декількох років, аби відслідковувати прогрес захворювання.

Отже в контексті розробки моніторингових систем варто враховувати природу даних для розуміння можливих опцій представлення даних на моніторингових панелях, періодичності замірів для визначення коректних способів отримання даних з джерела та методів роботи з великою кількістю даних, та “термін життя” даних, аби правильно організувати збереження даних.

Наприклад, представлення зміни даних протягом часу на графіку актуальне тільки для чисельних даних. Для булевих та категоріальних можливо робити зведену статистику за певний період часу (кругові чи стовпчикові діаграми). Для даних зі складною багаторівневою структурою потрібно визначати алгоритми аналізу (наприклад, обрахунок якоїсь цільової метрики з масиву різних даних, чи виділення тільки одного чи декількох показників простої структури). Для неструктурованих даних доцільним буде тільки логуювання чи представлення у форматі таблиць.

### **3. АРХІТЕКТУРА МОНІТОРИНГОВИХ СИСТЕМ В МЕДИЧНІЙ СФЕРІ ТА ДЛЯ ЗАГАЛЬНОВЖИВАНИХ ДЕВАЙСІВ**

Загалом в медичній сфері використовуються локальні моніторингові системи (ПЗ, яке працює в межах локальної мережі медичного закладу) та системи для дистанційного моніторингу.

Попри значні досягнення в комп'ютерних науках, організація моніторингової інфраструктури в медичних закладах досі є складною задачею. Як зазначається в [6], головна проблема полягає у відсутності чітких стандартів інтерфейсів та формату даних, ліцензійні обмеження, несумісність приладів та відсутність можливості інтеграції деяких приладів в цифрову систему з технологічної точки зору. Через це єдина система, яка агрегує всі дані відсутня. З деякими даними доводиться працювати мануально (наприклад, показники відображаються на індикаторі приладу), деякі дані складно структурувати для ретроспективного аналізу, а для деяких категорій приладів треба використовувати різне ПЗ та орієнтуватися в великій кількості моніторів одночасно. В архітектурному плані якогось єдиного стандарту організації мережі приладів немає, однак над цим ведеться активна робота та створюються прототипи єдиних систем. Зокрема, “The Integrated Medical Environment”, описана в статі. В архітектурному плані ПЗ — це єдиний інтерфейс для отримання даних з різних сховищ, як то БД лабораторії, БД з медичними візуалізаціями та ін.

Прилади для дистанційного моніторингу в загальному випадку стикаються з такими ж проблемами. Однак, тут варто розрізняти дві категорії таких приладів: прилади, які розробляються для повсякденного вжитку (фітнес-браслети, пульсоксиметри, електронні ваги тощо) та спеціалізовані прилади для медичного використання.

Для перших, як правило, розробляється якісне програмне забезпечення, яке дозволяє на смартфонах та у веб-додатках візуалізувати дані, робити аналітику. В межах одного бренду часто можна збирати дані в одному додатку і формувати більш комплексну статистику. До того ж розробники спеціально створюють API для того, щоб вони могли з легкістю інтегруватись в інші додатки, як, наприклад, Google Fitness. Загалом архітектуру таких приладів можна описати як клієнт-серверний додаток, де клієнт — мобільний додаток, а сервер — система та сховище в хмарі. Обчислення можуть відбуватись як локально, так і в хмарі. Окремі девайси під'єднані до телефону, через який передають інформацію в систему.

При використанні спеціалізованих приладів для дистанційного моніторингу підхід трохи інший, адже дані збираються з приладів, під'єднаних до пацієнта, а аналізує дані медичний працівник. В [7] дані із сенсрів за допомогою брокерів повідомлень передаються до сховища даних, де вони зберігаються для подальшого аналізу. Окремий додаток

звертається в БД, аби проаналізувати дані та надати звіт. В [8] наведена архітектура більш схожа на моніторингові девайси та додатки, які використовуються в широкому загалі. Група сенсорів під'єднана до смартфона чи планшета, де в мікроконтрольному інтерфейсі попередньо обробляється та пріоритезується. Після чого в зашифрованому вигляді надсилається до mobile gateway через WiFi або Bluetooth (в залежності від відстані) Після чого дані потрапляють до хмарного середовища, де відбуваються відбopідні обчислення (аналіз даних за допомогою нейронних мереж та побудова прогнозів), до результатів яких медичний працівник матиме доступ через клієнський застосунок.

#### **4. ПРОБЛЕМАТИКА АКТУАЛЬНИХ МОДЕЛЕЙ ТА КОНЦЕПЦІЯ ІНТЕГРАЦІЇ СИСТЕМИ МОНІТОРИНГУ В ЕЛЕКТРОННУ МЕДИЧНУ СИСТЕМУ**

В попередньому розділі розглянуто підходи, які використовуються для створення моніторингових систем при використанні девайсів-сенсорів для широкого загалу, організації медичних моніторингових систем на локальному рівні та побудови систем дистанційного моніторингу для медичних закладів. Проблема моніторингових систем для особистого користування в тому, що вони оптимізовані під використання однією людиною в особистих цілях, а хмарне сховище даних скоріше використовується як резервна копія даних. Огляд проблематики побудови системи моніторингу на локальному рівні висвічує проблеми, з якими потенційно можна стикнутися при інтеграції моніторингової системи в екосистему мікросервісів медичної системи, адже однією з потенційних задач є робота з моніторинговими пристроями.

Головний недолік запропонованих архітектур, які використовуються для побудови дистанційного моніторингу, полягає в тому, що всі дані проходять через централізоване хмарне сховище. Така концепція є прийнятною для розробки моніторингової системи, яка використовується для візуалізації даних в ретроспективі, однак для трансляції даних в режимі реального часу призведе до значних затримок та просідань з точки зору швидкодії системи.

Однак, повністю позбутися централізованого хмарного сховища, яке є джерелом моніторингових даних, не варто. Ба більше, в системі моніторингу воно може покрити значну частину потреб в наданні інформації на відображення на моніторинговій панелі, позбавивши необхідності адаптовувати сервіс під API кожного окремого девайсу чи групи девайсів.

Оптимальним рішенням проблеми буде використання патерну Адаптер для роботи з різними джерелами даних. Для роботи з центральним сховищем функція адаптеру буде скоріше номінальною, адже дані, які надаються по API загалом відповідатимуть формату даних, який необхідний для представлення на панелі. А для інших джерел даних чи груп джерел даних можна створити окремі адаптери, які зводитимуть дані до стандартного формату, готового для представлення на панелях.

Для кожного такого адаптера має конфігуруватись чіткий перелік індикаторів, які можна створити за допомогою даних з джерела. Очевидно, наприклад, що при прямому підключенні до сервіса, який надає лише останній замір пульсу, без зберігання даних в центральному датасховищі побудувати графік зміни пульсу за останні 2 години буде неможливим.

Також центральне сховище можна оптимізувати, зробивши його тимчасовим хабом даних. Дані, які актуальні лише пару родин, немає сенсу зберігати в БД декілька місяців. Можна розбити дані на декілька категорій за їх "терміном життя" та періодично очищати БД від застарілих даних. В т.ч. для аналітики можна створити спеціальний API, за допомогою якого можна отримати деякі дані та зберегти в постійному сховищі для подальшого

статистичного аналізу чи використання як навчальних вибірок для побудови моделей машинного навчання. З таким підходом база не буде швидко розростатись, що через деякий час призведе до зниження ефективності.

## 5. ВИСНОВКИ

В даній роботі було розглянуто пристрої, які використовуються для медичного моніторингу, на локальному рівні в медичних закладах та для дистанційного моніторингу. Для таких пристроїв наведено огляд природи даних, з якими потрібно взаємодіяти при організації моніторингу стану здоров'я. Також було розглянуто наявні архітектурні рішення для організації сенсорних пристроїв, хмарних застосунків так лієнтів в єдину архітектуру. На основі наявних рішень запропоновано концепцію системи моніторингу в екосистемі мікросервісів загальної системи віртуального кабінету лікаря, де централізоване хмарне сховище поєднується з можливістю роботи з сервісами напряму за допомогою адаптерів та періодична очистка застарілих даних з централізованого хабу (періодичність визначається “терміном актуальності” даних), який використовується як основне джерело.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Poverty and Access to Health Care in Developing Countries / D. H. Peters та ін. *Annals of the New York Academy of Sciences*. 2008. Т. 1136, № 1. С. 161–171. URL: <https://doi.org/10.1196/annals.1425.011> (дата звернення: 14.11.2022).
2. Cylus J., Papanicolas I. An analysis of perceived access to health care in Europe: How universal is universal coverage?. *Health Policy*. 2015. Т. 119, № 9. С. 1133–1144. URL: <https://doi.org/10.1016/j.healthpol.2015.07.004> (дата звернення: 14.11.2022).
3. Machine Learning for Healthcare Wearable Devices: The Big Picture / F. Sabry та ін. *Journal of Healthcare Engineering*. 2022. Т. 2022. С. 1–25. URL: <https://doi.org/10.1155/2022/4653923> (дата звернення: 17.11.2022).
4. Berry C. Monitoring and Testing the Critical Care Patient - Critical Care Medicine - MSD Manual Professional Edition. MSD Manual Professional Edition. URL: <https://www.msmanuals.com/professional/critical-care-medicine/approach-to-the-critically-ill-patient/monitoring-and-testing-the-critical-care-patient> (дата звернення: 18.11.2022).
5. A Review of Wearable Internet-of-Things Device for Healthcare / N. Surantha та ін. *Procedia Computer Science*. 2021. Т. 179. С. 936–943. URL: <https://doi.org/10.1016/j.procs.2021.01.083> (дата звернення: 18.11.2022).
6. Information Technology in Critical Care: Review of Monitoring and Data Acquisition Systems for Patient Care and Research / M. A. De Georgia та ін. *The Scientific World Journal*. 2015. Т. 2015. С. 1–9. URL: <https://doi.org/10.1155/2015/727694> (дата звернення: 14.11.2022).
7. Vitabile, S. et al. (2019). Medical Data Processing and Analysis for Remote Health and Activities Monitoring. In: Kołodziej, J., González-Vélez, H. (eds) *High-Performance Modelling and Simulation for Big Data Applications*. Lecture Notes in Computer Science, vol 11400. Springer, Cham. [https://doi.org/10.1007/978-3-030-16272-6\\_7](https://doi.org/10.1007/978-3-030-16272-6_7) (дата звернення: 17.11.2022).
8. Iranpak S., Shahbahrami A., Shakeri H. Remote patient monitoring and classifying using the internet of things platform combined with cloud computing. *Journal of Big Data*. 2021. Т. 8, № 1. URL: <https://doi.org/10.1186/s40537-021-00507-w> (дата звернення: 18.11.2022).

# ОРГАНІЗАЦІЯ ЗБОРУ ДІАГНОСТИЧНИХ ПОКАЗНИКІВ В СИСТЕМАХ КРАЙОВИХ ОБЧИСЛЕННЯХ

Охота Д.Л.<sup>1</sup>, Булах Б.В.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>qwerty165343@gmail.com [0000-0001-8364-1900], <sup>2</sup>bogdan.bulakh@gmail.com  
[0000-0001-5880-6101]

**Розглядається задача аналізу процесу організації систем крайових та туманних обчислень медичних показників в трьохрівневій архітектурі. Завдяки впровадженню наявних інструментів, буде реалізовано швидке впровадження різного роду рішень моніторингу стану суб'єктів охорони здоров'я. Буде оцінено вплив такого підходу на ефективність збору та опрацювання показань датчиків .**

**Ключові слова: крайові обчислення, інфраструктура, крайовий пристрій, Інтернет речей.**

## 1. ВСТУП

Граничні обчислення являють собою технологічну пропозицію, яка намагається вирішити обмеження мережевої затримки, безпекової складової, використовуючи зростаючу конвергенцію між хмарними і мережевими інфраструктурами та забезпечуючи основу для еволюції до конвергентного прозорого середовища обробки інформації, здатного підтримувати розподілені, мобільні додатки, що вирішують набагато ширший спектр завдань, і стають ключовим інструментом для втілення обіцянок щодо мереж наступного покоління (5G і далі) в реальність.

Стрімкий розвиток Інтернету речей (IoT) трансформував бізнес та обслуговування клієнтів у багатьох аспектах сучасного життя завдяки повсюдному використанню сенсорних і обчислювальних можливостей мобільних пристроїв, таких як смартфони, фітнес-браслети, ноутбуки та планшети.

Велика тенденція використання “розумних” речей спостерігається у сфері охорони здоров'я, де відслідковування усіх життєвих показників стану людини, грає значну роль у визначенні аномалій роботи організму, хвороб. Що за собою призводить до великою утилізації інтернет каналів провайдерів мережеских послуг. Тому у даній роботі розглядається специфіка організації роботи систем граничних обчислень з використанням медичних вимірювальних пристроїв.

## 2. КРАЙОВІ ТА ТУМАННІ ОБЧИСЛЕННЯ

### 2.1. Визначення

Крайові обчислення - це парадигма розподілених обчислень, що здійснюються в межах досяжності кінцевих пристроїв. Цей тип обчислень використовується для скорочення часу мережного відгуку, а також ефективнішого використання пропускної спроможності мережі.

Туманні обчислення — це модель, яка забезпечує обчислення та зберігання даних між кінцевими пристроями та традиційними центрами хмарних обчислень. Туманні обчислення — концепція, за якою частина даних оброблюється в локальних мережах, а не виключно в дата-центрі. Будь-який пристрій, що має обчислювальні здібності, сховище та мережу



підключення, може бути вузлом туману. Приклади пристроїв включають промислові контролери, комутатори, маршрутизатори, вбудовані сервери та камери відеоспостереження [1].

Сервіси додатків, які використовують крайові обчислення, скорочують обсяги даних, які мають бути передані, наступний трафік та відстань, яку мають пройти дані. Ця архітектура обчислень забезпечує меншу затримку відгуку всередині мережі та знижує витрати на обмін даними. Розвантаження обчислень у додатках, які виконуються у реальному часі (алгоритми розпізнавання осіб) з допомогою цієї технології, показала значну ефективність у сфері поліпшення часу відгуку мережі, що було продемонстровано у ранніх досліджах. Подальші дослідження показали, що використання хмаринок (cloudlets) у мережі користувачів мобільних додатків, що пропонують послуги, зазвичай що у хмарі, забезпечує скорочення часу виконання операцій, коли деякі завдання вивантажуються на граничний вузол. З іншого боку, розвантаження кожної оброблюваної задачі може призвести до уповільнення обміну даними через збільшення часу передачі між кінцевими пристроєм і вузлами.

Інші можливі застосування технології включають управління «розумними» системами автомобілів, створення «розумних» міст з розвиненою мережевою інфраструктурою, проекти Індустрії 4.0, а також використання граничних обчислень в системах охорони здоров'я.

## 2.2. Загальна схема

Туманні обчислення - це обчислювальний шар між хмарою і периферією. Там, де периферійні обчислення можуть відправляти величезні потоки даних безпосередньо в хмару, туманні обчислення можуть отримувати дані з периферійного шару до того, як вони потраплять в хмару, а потім вирішувати, які з них є важливими, а які ні. Релевантні дані зберігаються в хмарі, в той час як нерелевантні дані можуть бути видалені або проаналізовані на рівні туману для віддаленого доступу або для інформування локалізованих моделей навчання.

Прикладом такої роботи туманних обчислень може слугувати медичний додаток пацієнтів, які знаходяться віддаленому лікуванні, де датчик температури, підключений до периферійного сервера, вимірює температуру щосекунди. Потім ці дані передаються в хмарний додаток для моніторингу температурних стрибків. Уявіть собі, що всі вимірювання температури, кожна секунду циклу вимірювань 24/7, відправляються в хмару.

У випадку з туманним шаром, крайовий пристрій спочатку надсилає дані до туманного шару через локальну мережу. Туманний сервер отримував би ці дані і, відповідно до певних параметрів, вирішував би, чи варто їх відправляти в хмару. Кінцевий результат - зменшення трафіку та часу відповіді. Для простих вимірювань показників здоров'я така економія даних може здатися незначною. Але уявіть, що ви постійно передаєте складну інформацію або великі файли, наприклад, зображення чи відео. Вплив на пропускну здатність і затримку може бути величезним в залежності від програми та пристрою.

У випадку розгортання системи моніторингу стану пацієнта, важливими є характеристики: реакції на подію, що відбулась на краю, відповідність нормативним вимогам, захищеність від несанкціонованого доступу. Також при реалізації такої моделі є можлива відсутність зв'язку із центральною хмарою, у разі чого система буде працювати у штатному режимі [2].

Така архітектура (Рис. 1) гарантує доступність крайових пристроїв до інструментів зв'язку з вузлами підтримки системи, операційною складовою та моніторинговими та контролюючими складовими. Рішення даного типу передбачає його використання в доволі малій географічній зоні, що має достатній рівень мережевого забезпечення.

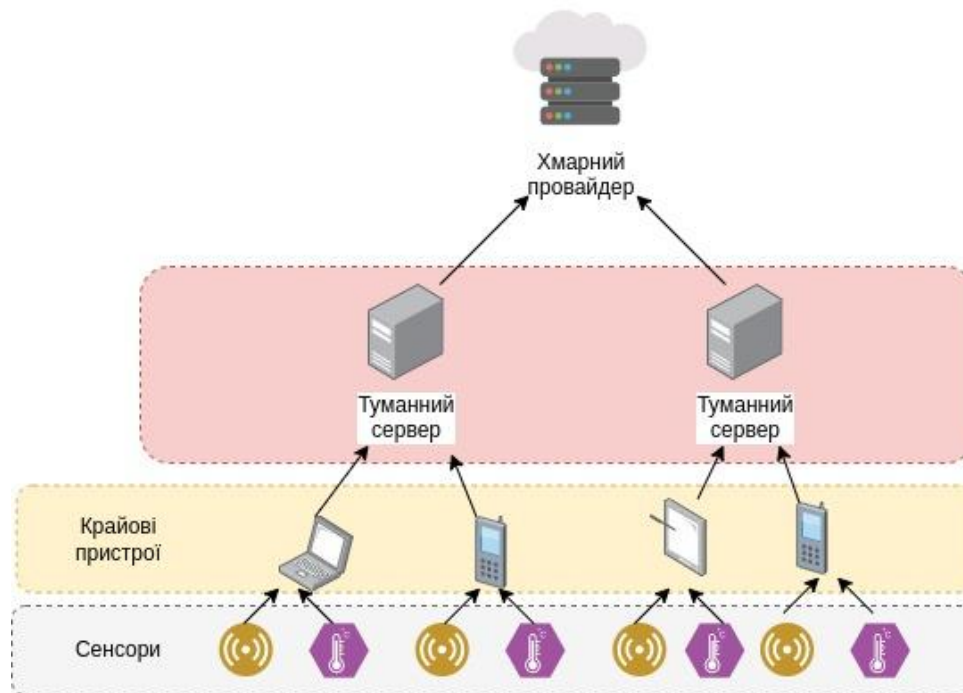


Рисунок 1. Організаційна схема крайових обчислень

### 2.3. Поведінкова схема

Поведінкова схема (Рис. 2) описує увесь алгоритм зв'язку процесів що відбувається на кожному із рівнів системи.

На нижньому рівні (рівні крайових пристроїв) розташовуються кінцеві пристрої (датчики), а також крайові пристрої та шлюзи. Цей шар також включає програми, які можуть бути встановлені на кінцеві пристрої для розширення їхньої функціональності. Елементи цього шару використовують наступний шар, мережу, для зв'язку між собою, а також між ними та хмарою. Рівень туманних пристроїв містить обчислювальні вузли, які підтримують управління ресурсами та обробку IoT завдань, що далі надходять у хмару. Поверх туманного рівня розташовується програмне забезпечення для управління ресурсами, яке керує всією інфраструктурою та забезпечує якість обслуговування додатків туманного обчислення [3].

Трирівнева архітектура для інфраструктури інтелектуальної охорони здоров'я, що складається з рольової моделі, багаторівневої хмарної архітектури та прошарку туманних обчислень, забезпечує ефективність для додатків охорони здоров'я та догляду за літніми людьми. Рівень туманних обчислень удосконалює архітектуру, забезпечуючи низьку затримку, підтримку мобільності, та безпеку. Відображення інформації відбувається на пристрої з допомогою сервіс-орієнтованого підходу [4].

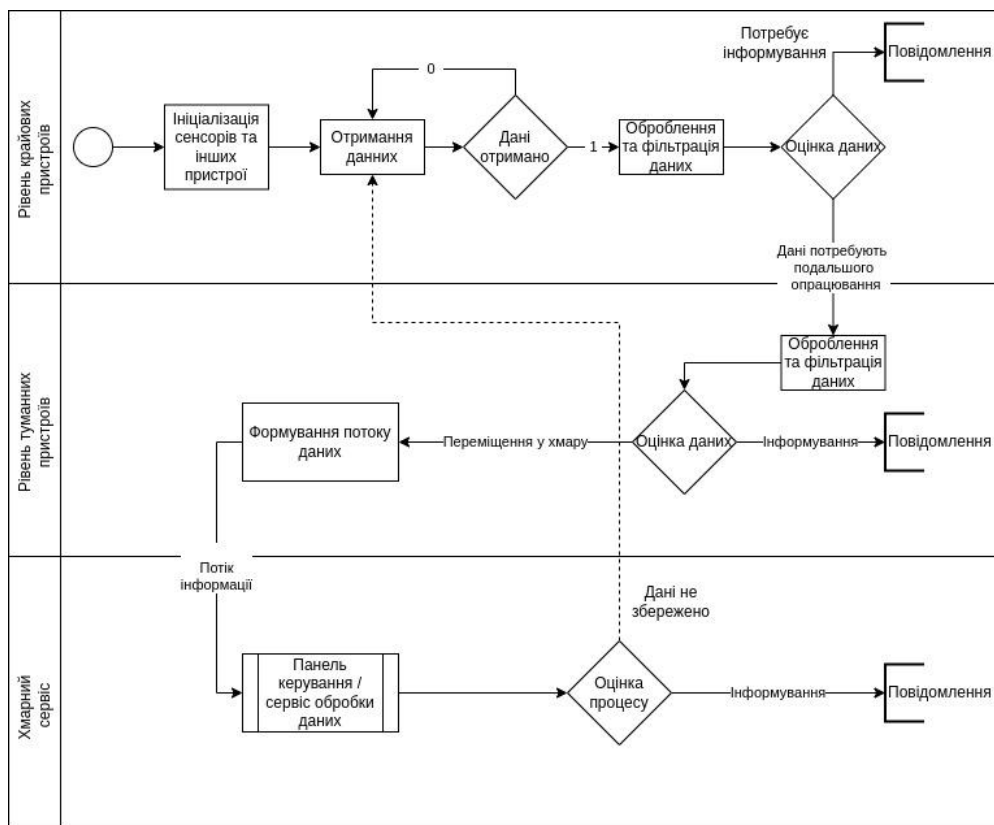


Рисунок 2. Поведінкова схема

### 3. МЕТОДИ ТА ІНСТРУМЕНТАРІЙ

#### 3.1. Моделі розгортання систем

Розробка систем крайових обчислень передбачає реалізацію різного роду методологій та принципів розгортання обчислювальних вузлів. Таким чином розглянуті нижче засоби виокремлюють та удосконалюють роботу усіх одиниць середовища.

Наприклад **Cloudlet** технологія представляє собою довірених, багатий на ресурси комп'ютер або кластер комп'ютерів, добре підключений до Інтернету і доступний для прилеглих мобільних пристроїв. Він удосконалює початкову дворівневу архітектуру "мобільний пристрій-хмара" мобільних хмарних обчислень до трирівневої архітектури "мобільний пристрій-cloudlet -хмара". При цьому Cloudlet може обслуговувати користувачів як незалежну хмару, перетворюючись на "маленьку хмару" або "центр обробки даних у коробці". Подібно до точок доступу WiFi, Cloudlet може бути розгорнутий у зручному місці (наприклад, у ресторані, кафе або бібліотеці). Декілька Cloudlet можуть утворювати розподілену обчислювальну платформу, яка може ще більше розширити доступні ресурси для мобільних пристроїв. Оскільки Cloudlet знаходиться лише в одному стрибку від мобільних пристроїв користувачів, він покращує QoS за рахунок низької затримки зв'язку та високої пропускнуної спроможності [5].

**CloudPath** – це система крайових обчислень, запропонована Університетом Торонто. У такій системі різні ресурси такі як обчислення та зберігання, надаються на шляху від пристрою до хмарного центру обробки даних. Вона підтримує розподіл на вимогу розподіл та динамічне розгортання багаторівневої архітектури. Основна ідея CloudPath полягає в реалізації так званих "обчислень по дорозі", що дозволяє скоротити час відгуку та покращити

використання пропускну́ї спроможності. час і поліпшити використання пропускну́ї спроможності порівняно зі звичайними хмарними обчисленнями.

**SpanEdge** – потокова обробка є одним із важливих типів додатків у крайових обчисленнях, де дані генеруються різними джерелами даних джерелами даних у різних географічних точках і безперервно передаються у вигляді потоків. Традиційно всі необроблені дані передаються глобальною мережею на сервер центру обробки даних, а системи обробки потоків, такі як Apache Spar. Системи обробки потоків, такі як Apache Spark та Flink, також розроблені та оптимізовані для одного централізованого центру обробки даних. Однак такий підхід не може ефективно обробляти величезні дані, що генеруються безліччю пристроїв на краю мережі, і ситуація стає ще гіршою, коли програми вимагають низька затримка та передбачуваність [6].

Перелічені вище технології володіють рівнозначними характеристики реалізації у трьохрівневих архітектурах крайових обчислень. Але зі спробами зменшення потоку інформації на центральні вузли хмар, з'являється проблема перевантаження крайових пристроїв, що ймовірно може спричинити відмову деяких із них. Тому було розроблено ряд програмних рішень для розвантаження та одночасно ефективного використання обчислювальних вузлів, які знаходяться між крайовими пристроями та хмарою.

### 3.2. Програмні рішення

Системи моніторингу стану пацієнтів, передбачають використання широкого спектру програмних рішень, які вирішують задачі: розпізнавання аномалій роботи організму, зберігання даних, оброблення та відображення. Але такий перелік програмного забезпечення із збереження повного його функціоналу неможливо розгорнути на малопотужних крайових чи туманних пристроях. Для рішення цієї проблеми було спроектовано безліч рішень у сфері глибокого навчання, моніторингу та т.п.

Для підтримки обробки даних за допомогою моделей глибокого навчання на периферії було випущено кілька периферійних фреймворків та інструментів глибокого навчання. Такими є: TensorFlow Lite, Caffe2, PyTorch, MXNet, CoreML та TensorRT.

TensorFlow Lite — це набір інструментів, який забезпечує машинне навчання на пристрої, допомагаючи розробникам запускати свої моделі на мобільних, вбудованих та периферійних пристроях [7].

Caffe2 — це фреймворк для глибокого навчання, який забезпечує простий і зрозумілий спосіб експериментувати з глибоким навчанням і використовувати внесок спільноти в нові моделі та алгоритми. Надає можливість масштабувати архітектуру, використовуючи потужність графічних процесорів у хмарі або в масах на мобільних пристроях за допомогою крос-платформних бібліотек Caffe2.

PyTorch — це фреймворк машинного навчання, заснований на бібліотеці Torch, що використовується для таких додатків, як комп'ютерний зір і обробка природної мови, спочатку розроблений компанією Meta AI, а тепер входить до складу Linux Foundation. Це безкоштовне програмне забезпечення з відкритим вихідним кодом, випущене під модифікованою ліцензією BSD. Хоча інтерфейс Python є більш відшліфованим і основним напрямком розвитку, PyTorch також має інтерфейс C++.

MXNet — це фреймворк глибокого навчання з відкритим вихідним кодом, який дозволяє визначати, навчати та розгортати глибокі нейронні мережі на широкому спектрі пристроїв, від хмарної інфраструктури до мобільних пристроїв. Вона має високу масштабованість, що дозволяє швидко навчати моделі, а також підтримує гнучку модель програмування і кілька мов.

TensorRT — це SDK для високопродуктивного глибокого навчання, включає в себе оптимізатор глибокого навчання і середовище виконання, яке забезпечує низьку затримку і високу пропускну здатність додатків для виводу.

Таблиця 1. Порівняння бібліотек машинного навчання

Назва	Підтримка мов	Типи завдань	Характеристика	Розподіленість	Багатопроекторність	Складність у розгортанні	Операційні системи	Цільовий пристрій
TensorFlow Lite	Java, Swift, Objective-C, C++, Python	Прийняття рішень	Латентність	Наявна	Середня	Легко	Android та iOS, вбудовані Linux та мікроконтролери	Мобільні та вбудовані пристрої
Caffe2	C++, Python	Прийняття рішень, тренування	Легкість, модульність, масштабованість	Наявна	Мінімальна	Складна	MacOS; Ubuntu; CentOS; Windows; Android; Raspberry Pi	Багато платформні
PyTorch	C++ Python	Прийняття рішень, тренування	Легкість у дослідженні	Наявна	Середня	Легко	Linux MacOS Windows	Багато платформні
MXNet	C++, JavaScript, Python, R, Matlab, Julia, Scala, Clojure, Perl	Прийняття рішень, тренування	Великий вибір модулів	Наявна	Повна	Середня	Linux MacOS Windows мікроконтролери	Багато платформні
CoreML	-	Прийняття рішень	Використання пам'яті та потужність	Невідомо	Повна	Легка	iOS	Apple
TensorRT	C++ Python	Прийняття рішень	Латентність та пропускну здатність	Відсутня	Повна	Середня	Linux Windows	NVIDIA GPU

### 3.3. Рекомендації щодо процесу розгортання

При розгортанні подібних систем варто виконати наступні кроки:

- Об'єм інтелекту IoT-пристроїв.

Чим більше аналітичних дій виконується на пристрої, тим менше кількість операцій виконується на хмарних серверах. Адже дані вже фільтруються в джерелі — пристрої IoT. Інтелектуальні, стандартизовані пристрої IoT забезпечать менші обсяги даних в більш легко керованих форматах.

- Групування IoT пристроїв

Об'єднування в групи виконується за декількома критеріями. Поділ на області можна виконати, в залежності, які властивості були надані пристроям у першому кроці. Або ж у

відповідності до географічного положення, це значно зменшить затримки та навантаження на мережу.

- Визначення результируючих даних

Розробка хорошого механізму правил та оновлення його за необхідності, щоб відобразити, як змінюються пріоритети і потреби у виконанні тих чи інших задач. Пріоритетом також є інвентаризація середовища в режимі реального часу, щоб гарантувати, що туманні сервери працюють в поточному середовищі.

- Використання hub-spoke підходу

Для управління потоком необхідних даних необхідно мати крайову інфраструктуру, яка складається з різних серверів, розміщених в мережі з ієрархічним способом роботи з даними між ними. Оптимальним способом роботи з такою складною системою є використання найдешевших, найменш інтелектуальних крайових серверів - це умовне використання термінології, ці системи можуть бути досить інтелектуальними і дорогими самі по собі - якомога ближче до пристроїв IoT.

- Використовувати розширену аналітику даних та звітність.

Необхідно уникати помилкових спрацьовувань і негативних сигналів, а можливі шляхи виправлення ситуації повинні бути показані будь-якій людині, яка бере участь в цьому процесі. Тому не варто економити на аналітичних інструментах, що використовуються, і переконайтесь, що звітність ведеться чітко і змістовно.

#### **4. ВИСНОВКИ**

Екосистема крайових обчислень дуже динамічна, у ній з'являється безліч нових ініціатив від різних організацій та компаній. Тільки в області відкритих кодів реалізується не менше 20 ініціатив. Поки що немає узгодженого галузевого стандарту, що охоплює всі аспекти крайових обчислень, навіть після багаторічних зусиль деяких органів зі стандартизації.

Навіть на стадії розвитку крайових обчислень, виникає велика кількість програмних та архітектурних рішень, що в повній мірі використовують усі переваги середовища, у якому їх розгорнуто. Так чудовим прикладом слугують системи "розумного" міста, моніторингу роботи сонячних панелей, система управління робочими вузлами на виробництвах, модернізується сфера охорони здоров'я.

У роботі розглядається архітектурні та програмні рішення, які є ефективним у медицині. Так на основі роботи кінцевих пристроїв, якими являються: медичне обладнання, різного роду сенсори та датчики; було реалізовано трьохрівнену архітектуру з опрацюванням показників життєдіяльності та їх подальшу передачу у центральні вузли як туманного рівня так і хмари. Така методологія, організовує швидку реакцію системи на події, про які сигналізують кінцеві пристрої, навіть в умовах недостатнього доступу до інтернет-мережі.

#### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Edge Computing. URL: [https://en.wikipedia.org/wiki/Edge\\_computing](https://en.wikipedia.org/wiki/Edge_computing) (дата звернення: 11.11.2022)
2. OpenFog Reference Architecture for Fog Computing. URL: [https://www.iiconsortium.org/pdf/OpenFog\\_Reference\\_Architecture\\_2\\_09\\_17.pdf](https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf) (дата звернення: 11.11.2022)
3. Smart Items, Fog and Cloud Computing as Enablers of Servitization in Healthcare URL: [https://www.researchgate.net/publication/284430697\\_Smart\\_Items\\_Fog\\_and\\_Cloud\\_Computing\\_as\\_Enablers\\_of\\_Servitization\\_in\\_Healthcare](https://www.researchgate.net/publication/284430697_Smart_Items_Fog_and_Cloud_Computing_as_Enablers_of_Servitization_in_Healthcare) (дата звернення: 12.11.2022)

4. A. Krolczyk, V. Stantchev, C. Senf, Service-oriented approaches for e-government, in Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services, 2009, pp. 441–446.
5. Cloudlet. URL: <https://en.wikipedia.org/wiki/Cloudlet> (дата звернення: 12.11.2022)
6. SpanEdge: Towards Unifying Stream Processing over Central and Near-the-Edge Data Centers. URL: <https://ieeexplore.ieee.org/document/7774704> (дата звернення: 12.11.2022)
7. TensorFlow Lite. URL: <https://www.tensorflow.org/lite/guide> (дата звернення: 12.11.2022)

# ВИРІШЕННЯ ЗАДАЧІ ГРАФЕМНО-ФОНЕМНОГО ПЕРЕТВОРЕННЯ ТЕКСТІВ

Самвелян А.Р.<sup>1</sup>, Кисельов Г.Д.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>yourcamelcase@gmail.com [0000-0001-8364-1900], <sup>2</sup>g.kyselov@gmail.com

**Розглядається методи вирішення задачі графемно-фонемного перетворення. Буде визначена математична модель задачі, наведені методи вирішення даної задачі та їх опис.**

**Ключові слова: графема, фонема, нейромережева модель, синтез мовлення.**

## 1. ВСТУП

Процес графемно-фонемного (G2P) перетворення генерує фонетичну транскрипцію з письмової форми слів. Написання слова називається послідовністю графем, фонетична форма - послідовністю фонем (або фонем). Розвиток фонемної лексики має важливе значення в системах перетворення тексту в мову (TTS) та автоматичного розпізнавання мови (ASR). Для цього використовуються методи G2P, і від точності G2P-перетворення залежить отримання сучасних характеристик цих систем. Так, наприклад, в акустичних моделях АСР критично важливими компонентами є вимовні лексикони та мовні моделі. Акустичні та мовні моделі будуються автоматично з великих корпусів. Лексикони вимови є середнім шаром між акустичною та мовною моделями. Для нової задачі розпізнавання мови продуктивність всієї системи залежить від якості вимовної складової. Іншими словами, продуктивність системи залежить від точності G2P. Наприклад, G2P перетворення слова "диктор" - це "S P IY K ER". У системах TTS якісна модель G2P також є невід'ємною частиною і має великий вплив на загальну якість. Неточне перетворення G2P призводить до неприродної вимови або навіть до незрозумілої синтетичної мови.

## 2. МАТЕМАТИЧНА МОДЕЛЬ ЗАДАЧІ ГРАФЕМНО-ФОНЕМНОГО ПЕРЕТВОРЕННЯ ТЕКСТІВ

Нехай  $G = \{a, b, v, \dots, y\}$  - множина графем української мови,  $P = \{A0, \dots, Y1\}$  - множина фонем української мови,  $W$  - множина слів української мови. Також:

$$\forall w \in W \exists! \underline{g} = (g_1, g_2, \dots, g_k), g_{1..k} \in G$$

Задача графемно-фонемного відображення полягає в тому щоб знайти таке відображення  $f: G_n \rightarrow P_m$  щоб:

$$\forall g \in G, p \in P, f((g_1, g_2, \dots, g_n)) = (p_1, p_2, \dots, p_m)$$

## 3. ТИПИ МОДЕЛЕЙ ДЛЯ ВИРІШЕННЯ ЗАДАЧ

**На основі правил:**

Даний підхід базується на написанні правил (регулярних виразів) для перетворення слів у набір фонем, в якому мається на увазі розміщення букв та відповідний.



### Моделі спільних послідовностей (JSM):

Фундаментальна ідея JSM полягає в тому, що послідовності графем і фонем можуть бути згенеровані спільно за допомогою послідовності спільних одиниць (графонів), які несуть як символи графем, так і символи фонем. Завданням СПМ є знаходження послідовності  $Y$  фонем, що задана послідовністю  $X$  графем. Цю задачу можна сформулювати як визначення оптимальної послідовності фонем  $Q$ , що максимізує їх умовну ймовірність  $Q$ , при заданій послідовності графем  $G$ :

$$\hat{Q} = \arg \max_Q P(Q|G).$$

Обчислення для всіх можливих послідовностей  $Q$  безпосередньо з  $P(Q|G)$  є складним і ми можемо виразити його за допомогою правила Байєса наступним чином:

$$\hat{Q} = \arg \max_Q P(Q|G) = \arg \max_Q \{P(G|Q) \cdot P(Q)/P(G)\}$$

Тут  $P(G)$  є спільною для всіх послідовностей  $Q$ . Наведене вище рівняння можна спростити наступним чином:

$$\hat{Q} = \arg \max_Q P(G|Q) \cdot P(Q)$$

### Використання нейромереж

Методи seq2seq засновані на генерації з непрямо обумовленою мовною моделлю, останнім часом показали багатообіцяючі результати в ряді завдань. У машинному перекладі моделі, обумовлені вихідними словами, використовувалися для створення тексту цільовою мовою, а в підписі до зображень - моделі, обумовлені зображенням, для генерації тексту підписів. Минулі роботи з цим підходом були зосереджені на завданнях з великими словниками та вимірюваною якістю в термінах BLEU.

Так як графемно-фонемне перетворення по своїй суті є перетворення однієї послідовності в іншу, то для вирішення даної задачі можна використати seq2seq нейромережеві моделі (Рис. 1).

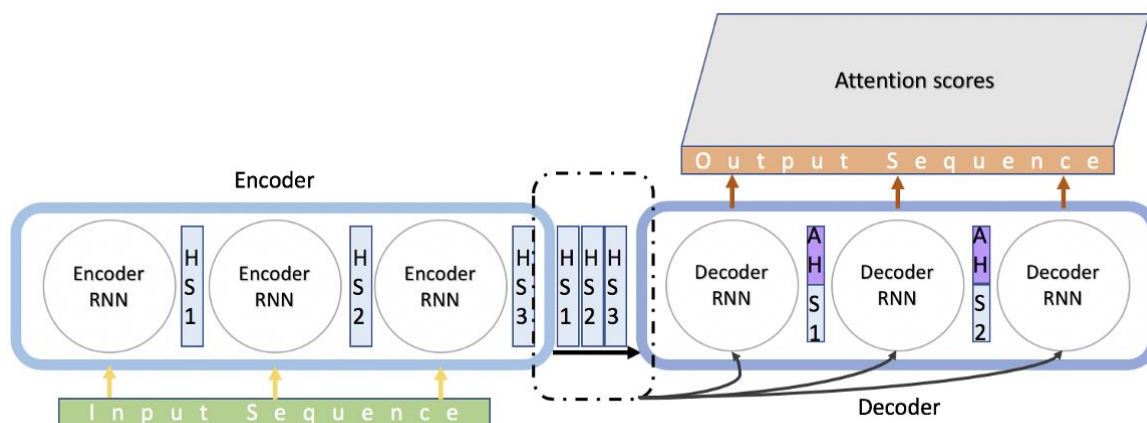


Рисунок 1. Використання seq2seq нейромережевої моделі

## 4. ВИСНОВКИ

Моделі перетворення графем у фонем (G2P) є важливими для обробки природної мови (NLP), автоматичного розпізнавання мови (ASR) та перетворення тексту в мову (TTS).

Хоча багато підходів машинного навчання застосовні для перетворення G2P, більшість з них є підходами керованого навчання, і в якості передумови ми повинні підготувати чисті анотовані навчальні дані, а це дорого коштує. На практиці нам потрібно здійснювати бутстрапінг або активне навчання за допомогою невеликого анотованого вручну словника G2P для ефективної розробки G2P-перетворювачів.

### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Comparison of Grapheme-to-Phoneme Conversion Methods on a Myanmar Pronunciation Dictionary URL: <https://aclanthology.org/W16-3702.pdf>
2. Sequence-to-Sequence Neural Net Models for Grapheme-to-Phoneme Conversion URL: <https://arxiv.org/pdf/1506.00196.pdf>
3. A Finite State and Data-Oriented Method for Grapheme to Phoneme Conversion URL: <https://aclanthology.org/A00-2040.pdf>

# НЕЙРОННА МЕРЕЖА ДЛЯ ДІАГНОСТУВАННЯ ЗАХВОРЮВАНЬ СИСТЕМИ ДИХАННЯ НА БАЗІ СИГНАЛІВ ЕКГ

Скріпченко М.О.<sup>1</sup>, Кислий Р.В.<sup>2</sup>

Інститут прикладного системного аналізу, КПІ ім. Ігоря Сікорського,  
Київ, Україна

<sup>1</sup>kvrware@gmail.com, <sup>2</sup>codegeas65@gmail.com

Незважаючи на досить розвинену індустрію медицини в сучасному світі, задача діагностування захворювань системи дихання є дуже важливою. Величезна кількість людей страждає від респіраторних захворювань кожного року, таких як, наприклад, апное, і, частина людей, на жаль, помирає від несвоєчасного діагностування, навіть якщо воно відбувається досвідченим лікарем. Одним з рішень проблеми неправильного діагностування є аналіз результатів з допомогою машинного навчання. Нейронні мережі є одним з таких рішень, де, через процес навчання на попередньо зібраному наборі результатів, що називають датасетами, система може надавати діагнози на основі певних закономірностей [1, 2]. Власне огляд та покращення точності вже існуючих нейронних мереж даного характеру, і є основною задачею роботи.

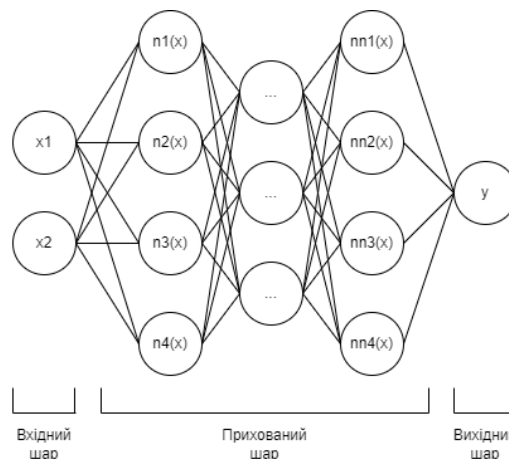


Рисунок 1. Узагальнена структура нейронних мереж

Нейронна мережа - це обчислювальна система, що базується на принципах роботи мозку людини. Тобто нейронні системи мають структуру, показану на рис.1, складаючись з нейронів та взаємозв'язків між ними, за допомогою яких відбувається навчання. З вхідного шару інформація передається на прихований шар, в якому нейрони роблять оцінку за допомогою функцій активації, і на вихідному шарі ми отримуємо результат, який представляє собою ймовірність того чи іншого діагнозу в залежності від задачі. Далі в роботу включається метод зворотного поширення (англ. backpropagation), що відповідає за коригування помилок шляхом поширення сигналів помилок в зворотному шляху, від виходу до входу, під час якого вага нейронів в прихованому шарі, що допомагає в збільшенні точності роботи мережі на наступному проході.

Задача, яка розглядається в роботі, відноситься до категорії аналізу часових рядів (англ. time series analysis), який представляє собою аналіз набору даних, зібраних на певному проміжку часу, що дозволяє виявлення значимих статистичних даних, а також подальшого

прогнозування результатів. Далі буде наведено варіанти нейронних мереж, що можуть бути використані для вирішення задач даного характеру.

Згорткові нейронні мережі (CNN) є класом глибоких мереж, що використовує спеціальні згорткові шари для більш ефективної та швидкої обробки вхідних даних, шляхом “згортки”, тобто зменшення початкових вхідних даних до певного формату меншого розміру, а також спеціальну функцію передобробки ConvNet. ConvNet складається з двох основних шарів, кожен з яких відповідає за згортку даних до більш оптимального розміру, але роблять це по-різному. Перший шар називається шар згортки (англ. convolutional layer), і він проводить згортання шляхом трансформації вхідних даних через поелементне множення на ядро шару, який являє собою сітку/матрицю ваг, які будуть тренуватися, що дозволяє визначати більш важливі елементи на наборі даних. Після чого, для більшої оптимізації розміру даних, використовується шар стягнення(англ. pooling layer), в якого є два варіанти: максимальне стягнення(вибір лише найбільш важливого елемента серед діапазону), або середнє стягнення(середнє значення елементів серед діапазону). Дані шари допомагають зменшити об’єм вхідних даних, при цьому маючи мінімальні втрати в точності прогнозування інформації.

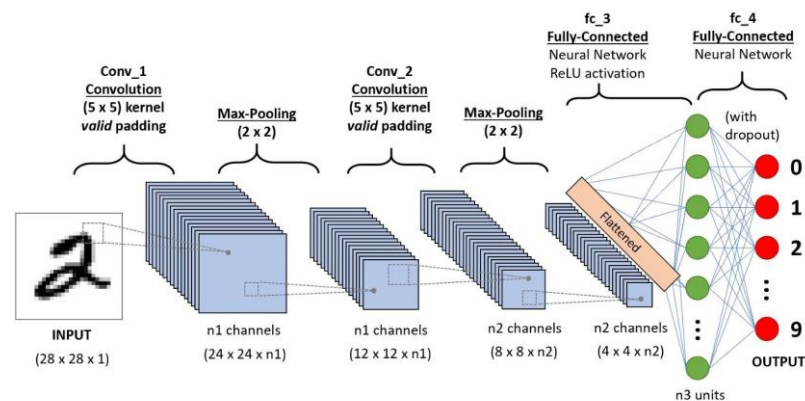


Рисунок 2. Приклад згорткової нейронної мережі

Рекурентні нейронні мережі також є гарною альтернативою при аналізі часових рядів. Основна ідея даної архітектури мереж базується на навчанні шляхом передачі інформації з попереднього стану мережі до наступного, що відкриває можливості більш точного прогнозування кожного наступного стану. Звичайні рекурентні нейронні мережі мають певні проблеми коли розрив між релевантно пов’язаними даними є досить великим, тому одним з найбільших рішень є архітектура довгої короткочасної пам’яті (англ. long short-term memory, LSTM). Основним компонентом даних мереж є стан клітини, яка на рис.3 є лінією, що проходить через верхню частину клітини. Вона зберігає інформацію про те, що мережа вважає важливим результатом попередніх результатів, значення яких можуть знаходитись в діапазоні  $[0,1]$ , де 0 означає “видалити все”, а 1 - “залишити все”, що означає, що інформація може видалятися, якщо вона перестає бути релевантною, і цей процес контролюється завдяки фільтрам, які дозволяють вирішувати зміни в стані клітини. Спочатку інформація проходить через шар фільтра забуття(англ. forget gate layer), на якому ми змінюємо кожне значення стану клітини відносно результату роботи попередньої клітини та нових вхідних даних, що дозволяє нам відсіяти менш релевантні дані. Наступним кроком є вирішення, яка нова інформація буде зберігатись в стані клітини. За це відповідає шар вхідного фільтра(англ. input gate layer), що визначає які значення потрібно оновити, для яких шар функції активації будує вектор значень, які можна занести в новий стан клітини. Після цього, ми вносимо результат роботи фільтрів на стан клітини шляхом векторного множення значень

шару фільтра забуття на стан попередньої клітини, а потім додавання нових значень станів, які ми отримали з шару вхідного фільтра та шару функції активації. В кінці кінців, ми використовуємо шар вихідного фільтра(англ. output gate layer), в спільній роботі з шаром функції активації, які відсортовують, яку інформацію ми повинні вивести. Дана архітектура мережі дозволяє нам більш контрольований метод зберігання важливих закономірностей, та відкидання випадкових повторень.

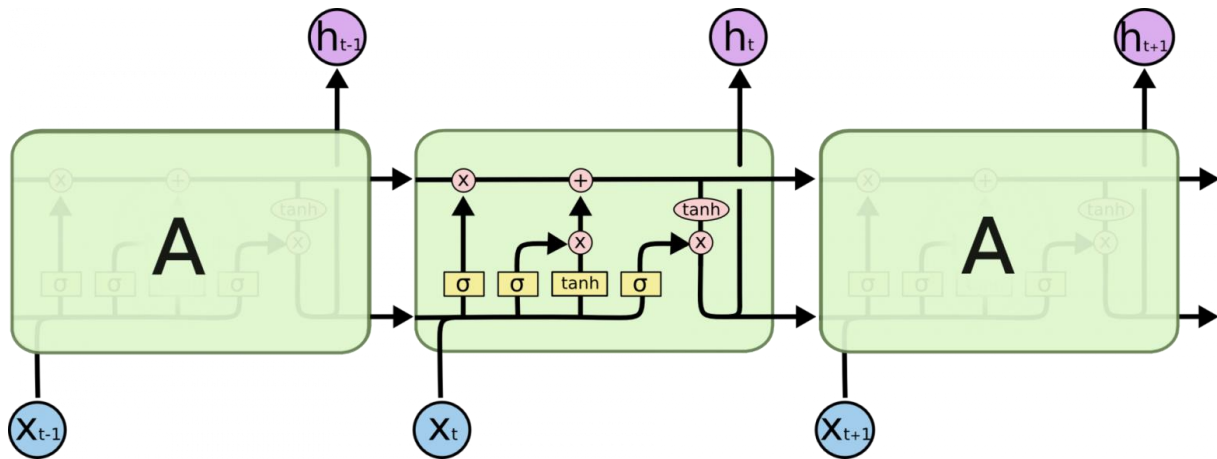


Рисунок 3. Базова структура LSTM-мережі

Також для вирішення таких задач можна використовувати статистичні методи навчання, такі, як, наприклад, алгоритм lightgbm. Даний метод базується на алгоритмі градієнтного бустингу, що відноситься до ансамблевих алгоритмів машинного навчання. Його ідея полягає в побудові ансамблей на основі моделей дерева рішень, які додаються в ансамбль по одному, і навчаються для виправлення помилок прогнозування, що були допущені попередніми моделями. Моделі власне навчаються з використанням будь-якої випадково обраної диференційованої функції витрат та алгоритма оптимізації градієнтного спуску. Lightgbm в свою чергу додає дві ключові ідеї: градієнтна одностороння вибірка(англ. Gradient One Side Sampling, GOSS), що використовується для виключення екземплярів з невеликим градієнтом, щоб віддати пріоритет екземплярам з більшим градієнтом, які матимуть більше значення на кінцеву точність; а також поєднання взаємовиключних ознак(англ. Exclusive Feature Bundling, EFB), яке поєднує розрідженні взаємовиключні ознаки, для зменшення загальної кількості ознак. Введення цих двох концепцій дозволяє отримати приріст в часі навчання алгоритму в 20 разів.

Тепер, маючи узагальнене розуміння нейронних мереж, а також різних можливостей їх реалізації, можна виділити наступні проблеми, поставленні в моїй дипломній роботі:

- дослідження впливу структури нейронної мережі, а тобто кількості проміжних шарів, типу зв'язків між ними, створеної на кафедрі, на точність і використання ЕКГ в діагностуванні;
- покращення функціонування діагностичної системи за рахунок застосування гібридних і нечітких компонентів нейронних мереж (хоч би теоретично);
- використання публічно доступних датасетів ЕКГ хворих і здорових пацієнтів, що містять тисячі записів з десятками атрибутів та призначені для визначення захворювання людини [3, 4].

Тобто запропонованим вирішенням поставленої задачі буде знаходження більш ефективних варіантів імплементації нейронної мережі для отримання точніших результатів.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Automated Prediction of Apnea and Hypopnea, Using a LAMSTAR Artificial Neural Network - Waxman, Jonathan A.; Graupe, Daniel; Carley, David W. Посилання на онлайн ресурс - <https://www.atsjournals.org/doi/10.1164/rccm.200907-1146OC>
2. Blind adaptive filtering for non-invasive extraction of the fetal electrocardiogram and its non-stationarities - D Graupe, M H Graupe. Посилання на онлайн ресурс - <https://journals.sagepub.com/doi/10.1243/09544119JEIM417>
3. MIT-BIH Arrhythmia Database - Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. Посилання на онлайн ресурс - <https://www.physionet.org/content/mitdb/1.0.0/>
4. PTB Diagnostic ECG Database - Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. Посилання на онлайн ресурс - <https://www.physionet.org/content/ptbdb/1.0.0/>

# АЛГОРИТМ РОЗРАХУНКУ ТОЧОК ДЛЯ ЗАХВАТУ ОБ'ЄКТА МАНІПУЛЯТОРОМ НА ОСНОВІ АПРОКСИМАЦІЇ 3D МОДЕЛІ ОБ'ЄКТА ПРОСТОРОВИМИ ПРИМІТИВАМИ

Харченко К.В.<sup>1</sup>, Вовк І.С.<sup>2</sup>

Інститут прикладного системного аналізу, КПІ ім. Ігоря Сікорського,  
Київ, Україна

<sup>1</sup>konst1970@gmail.com, <sup>2</sup>i.s.v.vovk@gmail.com

Задача захвату об'єкта маніпулятором є невід'ємною частиною процесу виконання роботом різноманітних завдань. Наприклад, широко використовується для автоматизації у виробничому середовищі. Зазвичай, це попередньо запрограмовані дії для одного визначеного типу маніпулятора та суворо обмеженого переліку об'єктів або їх типів. Проте, з часом сфера застосування робототехніки стрімко зростає і з'являється потреба у виконанні більш загальних, комплексних задач. Однією з яких є взаємодія зі заздалегідь невизначеною множиною об'єктів. У даній роботі розглядається взаємодія з твердими тілами.

Задача захвату невідомого об'єкта довільної форми, розміру та матеріалу є об'єктом досліджень. Одним з найбільш розповсюджених підходів є використання глибокого навчання. Наприклад у науковій роботі [1] для розрахунку захвату предмета маніпулятором з п'ятьма стрижнями пропонується використовувати глибокі згорткові нейронні мережі. Розрахунки проводяться на датасеті, отриманому за допомогою однієї RGB камери та сенсору глибини. У роботі [2] використовується метод глибокого навчання з підкріпленням. Вхідні дані отримуються з двох камер: закріпленої на основній частині робота та маніпуляторі. Захват виконується маніпулятором з двома паралельними стрижнями. Основним недоліком підходів, які потребують навчання є потреба у зборі великого об'єму даних для навчання та валідації. Такі датасети можуть бути синтезовані або зібрані дослідниками вручну. Наприклад, у роботі [3] дані для навчання моделі були отримані шляхом виконання дослідниками ряду мануальних дій з маніпулятором. Для збору датасету був сконструйований додатковий переносний сетап з аналогічним типом маніпулятора і можливістю виконувати захват предмета по натиску на кнопку. У будь-якому випадку алгоритми, основою яких є глибоке навчання, будуть ефективними лише на класах предметів, які були включені в навчальну вибірку, та є специфічними для типу маніпулятора для якого виконувалось навчання.

Іншим підходом до проблеми захвату об'єкта маніпулятором є використання технік комп'ютерного зору. У роботі [4] використовується часткове просторове зображення об'єкта, створене за допомогою сенсору глибини. Алгоритм захоплення предмета базується на математичній моделі маніпулятора (в роботі використовувався С-подібний маніпулятор). Такий підхід є більш універсальним, оскільки не виділяє якийсь окремий клас предметів. Проте, якщо модель маніпулятора складніша, ніж представлена у роботі, об'єм розрахунків збільшується, що матиме вплив на час виконання захвату.

У даній роботі запропоновано алгоритм розрахунку оптимальних точок для захвату об'єкта маніпулятором, що базується на апроксимації 3D моделі об'єкта просторовими примітивами. Модель об'єкта реконструюється з часткових просторових зображень об'єкта у процесі взаємодії з ним людини. Для отримання яких використовується RGB камера та сенсор глибини, що закріплені стаціонарно.

Основні кроки алгоритму зображені на рис. 1.

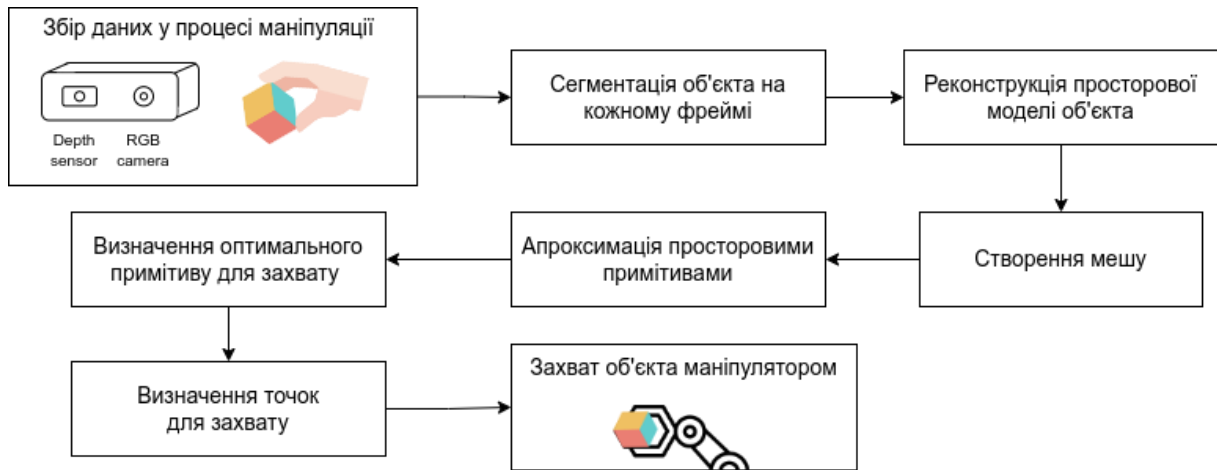


Рисунок 1. Блок-схема алгоритму визначення геометрії об'єкта шляхом взаємодії з ним, для задачі захвату маніпулятором

Алгоритм включає в себе наступні кроки:

- Збір даних у процесі маніпуляції людиною об'єктом у руках.
- Сегментація фону, рук та предмета. На цьому етапі розраховуються часткові просторові зображення предмета у форматі хмари точок.
- Реконструкція просторової моделі об'єкта з даних, отриманих на попередньому етапі. Теселяція хмари точок для отримання полігональної моделі(мешу).
- Апроксимація моделі просторовими примітивами, такими як паралелепіпед, сфера, конус, циліндр. Ці примітиви є простими для захвату будь-яким типом маніпулятора.
- Визначення оптимального для захвату примітиву та визначення точок захвату.

Запропонований у даній роботі підхід є універсальними для роботи з будь-яким класом об'єктів для захвату. Може бути використаним для будь-якого типу маніпуляторів, для яких способи захвату просторових примітивів (паралелепіпед, конус, сфера, циліндр) є визначеними. Запропонований алгоритм є детермінованим, а отже не потребує збору великої кількості даних для навчання.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Philipp Schmidt, Nikolaus Vahrenkamp, Mirko Wachter, Tamim Asfour “Grasping of Unknown Objects using Deep Convolutional Neural Networks based on Depth Images”, 2018
2. Shirin Joshi, Sulabh Kumra, Ferat Sahin “Robotic Grasping using Deep Reinforcement Learning” , 2020
3. Shuran Song, Andy Zeng, Johnny Lee, Thomas Funkhouser “Grasping in the Wild: Learning 6DoF Closed-Loop Grasping from Low-Cost Demonstrations”, 2020
4. Qujiang Leia, Guangming Chen, Jonathan Meijer, Martijn Wisse “A novel algorithm for fast grasping of unknown objects using C-shape configuration”, 2017



# РОЗРОБКА ДОДАТКУ ДЛЯ МОДЕЛЮВАННЯ ТА АНАЛІЗУ ЕЛЕКТРИЧНИХ КІЛ

Харченко К.В.<sup>1</sup>, Тараненко А.В.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>2</sup>tatanenko.and@gmail.com

**Метою дослідження є вивчення методів представлення елементів електричного кола та побудови кола на ЕОМ та методів аналізу електричного кола. Порівняння методів побудови моделі електричного кола. У доповіді описана актуальність теми та представлено опис причин для розробки додатку та результати тестової реалізації. Результатом роботи є програма, що приймає на вхід елементи електричного кола та їх параметри і на повертає характеристики утвореної схеми.**

**Ключові слова: представлення електричного кола на ЕОМ, задача моделювання, аналіз електричного кола, додаток.**

## 1. ВСТУП

Метою моделювання електронної схеми є отримання її математичної моделі і подальшого аналізу працездатності електронного пристрою на ЕОМ. Задачі моделювання електричних кіл розв'язуються в 2 етапи. Перший етап полягає у складанні рівнянь рівноваги для електричного кола. Другий етап полягає у вирішенні цих рівнянь за допомогою чисельних методів.

В області математичного моделювання динамічних систем можна виділити два основних напрямки. Одне з них орієнтовано на застосування спеціалізованих професійних прикладних програм з автоматизованого моделювання. Інший напрямок пов'язано з так званим «ручним» моделюванням, освоювати яке необхідно на стадії навчання студентів основам спеціальності і воно ефективне, так як математичне моделювання саме по собі передбачає більш глибоке вивчення фізичних процесів, що лежать в основі функціонування фізичних систем і математичного апарату для їх опису, що в сукупності сприяє ефективному навчанню в технічному університеті.

## 2. МЕТОДИ ПРЕДСТАВЛЕННЯ ЛІНІЙНИХ ЕЛЕМЕНТІВ ЕЛЕКТРИЧНОГО КОЛА ТА МОДЕЛІ СХЕМИ

Математичною моделлю реальної схеми називають сукупність співвідношень (формул, рівнянь, логічних виразів), що визначають характеристики стану системи залежно від її параметрів, початкових умов, вхідних сигналів (включаючи випадкові фактори) і часу.

У моделях розрізняють дані трьох типів:

- дані про елементи об'єкта моделювання;
- дані про властивості об'єкта й елементів;
- дані про співвідношення між елементами й об'єктом.

Абстрагування об'єкта проводять як за глибиною структурування (ієрархічна система, система елементів, нероздільний об'єкт), так і за ступенем абстрагування елементів і властивостями об'єкта (структурний, логічний та кількісний рівні).

На структурному рівні моделюється структура об'єкта на найнижчому рівні його структурування, коли математичну модель подано у вигляді множини, властивості та параметри елементів якої описані через функціональні зв'язки, співвідношення порядку, суміжності, сполучення. При цьому використовують апарат теорії множин і графів, теорії масового обслуговування тощо.

На логічному рівні моделювання кожній множині, булевій матриці бінарних співвідношень або структурному графові відповідають набори логічних співвідношень і змінних, які відображають причинно-наслідкові зв'язки. У проектуванні використовується апарат математичної логіки.

На кількісному рівні кожному елементові множини, булевій матриці або логічній змінній ставиться у відповідність алгебрична чи інша кількісна змінна, а логічні співвідношення переходять у кількісні співвідношення: рівняння, нерівності тощо.

Моделювання на кількісному рівні відображає функціональні, енергетичні, речовинні та просторові зв'язки. Ці зв'язки зазвичай описуються просторово-часовими співвідношеннями і визначаються через звичайні диференціальні рівняння або рівняння у частинних похідних.

Реактивними називають компоненти, до рівнянь яких входять похідні за часом від схемних змінних. Реактивними, зокрема, є звичайні лінійні ємності та індуктивності:

$$i_C = C \frac{du_C}{dt} \quad (1)$$

$$u_L = C \frac{di_L}{dt} \quad (2)$$

Враховуючи специфіку чисельних методів, що застосовуються для аналізу кола, для похідної за часом можна застосувати апроксимацію:

$$\frac{dy_n}{dt} = \frac{\alpha_0}{h} y_n - \frac{1}{h} \sum_{i=1}^K \alpha_i y_{n-i} \quad (3)$$

де  $K$  - порядок методу інтегрування;  $\alpha_0, \dots, \alpha_{K-1}$  - коефіцієнти апроксимаційного поліному;  $h$  - крок інтегрування,  $n$  - індекс кроків за часом.

Використовуючи наведене співвідношення, компонентне рівняння для ємності можна записати у вигляді:

$$i_{Cn} = \frac{\alpha_0}{h} C u_{Cn} - \frac{C}{h} \sum_{i=1}^{K-1} \alpha_i u_{C(n-i)} \quad (4)$$

або

$$i_{Cn} = G_{Cn} u_{Cn} + J_{Cn} \quad (5)$$

де

$$G_{Cn} = \frac{\alpha_0 C}{h}; \quad J_{Cn} = -\frac{C}{h} \sum_{i=1}^{K-1} \alpha_i u_{C(n-i)} \quad (6)$$

Таким чином, в межах даного кроку за часом можна формально представити ємність як паралельне з'єднання лінійної провідності  $G_{Cn}$  і джерела струму  $J_{Cn}$ , величина якого залежить від стану елемента в попередні моменти часу (рис. 1).

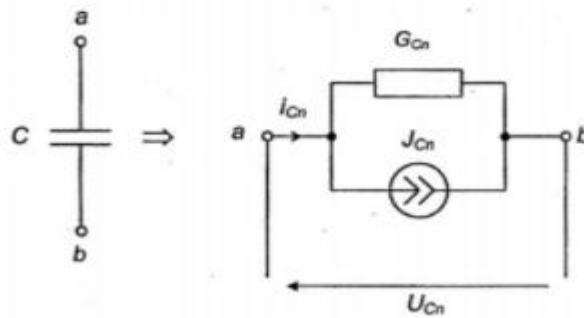


Рисунок 1. Алгебраїзація ємності

Аналогічно, компонентне рівняння індуктивності можна представити у вигляді:

$$u_{Ln} = \frac{\alpha_0}{h} L i_{Ln} - \frac{L}{h} \sum_{i=1}^{K-1} \alpha_i i_{L(n-i)} \quad (7)$$

або

$$u_{Ln} = R_{Ln} i_{Ln} + E_{Ln} \quad (8)$$

де

$$R_{Ln} = \frac{\alpha_0 L}{h}; \quad E_{Ln} = -\frac{L}{h} \sum_{i=1}^{K-1} \alpha_i i_{L(n-i)} \quad (9)$$

Таким чином, в межах даного кроку за часом можна формально представити індуктивність як послідовне з'єднання лінійного опору  $R_{Ln}$  і джерела напруги  $E_{Ln}$ , величина якого залежить від стану елемента в попередні моменти часу (рис. 2).

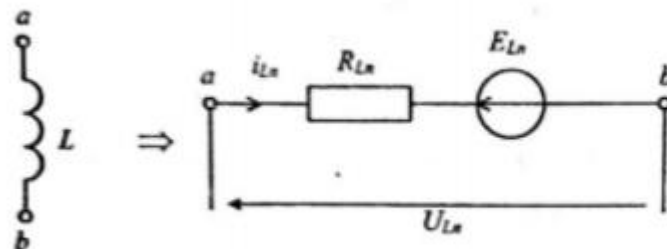


Рисунок 2. Алгебраїзація індуктивності

В практиці моделювання досить часто доводиться розглядати компоненти, властивості яких описуються нелінійними рівняннями виду  $y = f(x)$ , де  $y$  — струм або напруга на компоненті,  $x$  — вектор схемних змінних,  $f(x)$  — деяка нелінійна залежність, що з огляду на застосовувані чисельні методи аналізу вважається неперервною та диференційованою по всіх змінних. Враховуючи, що на кожній ітерації чисельного аналізу схемні змінні зазнають малих прирощень, нелінійну залежність  $f(x)$  можна лінеаризувати в околі поточних значень вектора невідомих:

$$y^{m+1} = f(x^m) + \sum_{i=1}^N \frac{\partial f}{\partial x_i} (x_i^{m+1} - x_i^m) \quad (10)$$

де  $m$  — індекс ітерацій. В загальному випадку ми вважаємо, що величина  $y$  залежить від усіх схемних змінних, однак на практиці зазвичай вона явно залежить лише від невеликої підмножини їх. Похідна  $\partial f / \partial x_i$  визначається в точці  $x^m$  і є константою в межах однієї

ітерації. На наступній ітерації її значення переобчислюється відповідно до поточних значень схемних змінних.

Останнє співвідношення можна переписати у вигляді:

$$y^{m+1} = \sum_{i=1}^N \frac{\partial f}{\partial x_i}(x_i^{m+1}) + (f x^m - \sum_{i=1}^N \frac{\partial f}{\partial x_i}(x_i^m)) \quad (11)$$

Отже, нелінійний компонент в межах даної ітерації можна представити як комбінацію лінійного компонента типу провідності або опору і джерела струму або напруги, величина змінної якого на даній ітерації є відомою.

В результаті застосування операцій алгебраїзації та лінеаризації вдається формально звести початкову схему до резистивного кола із залежними та незалежними джерелами.

### 3. РЕЗУЛЬТАТИ РОЗРОБКИ ДОДАТКУ

Завданням даної роботи є створення додатку для моделювання та аналізу електричних схем. Для цього було обрано мову програмування Python, тому що це скриптова об'єктно-орієнтована мова, що має величезну кількість додаткових модулів (бібліотек) для спрощення процесу написання коду даної програми. Також будуть обрані такі бібліотеки Python: NumPy та Matplotlib.

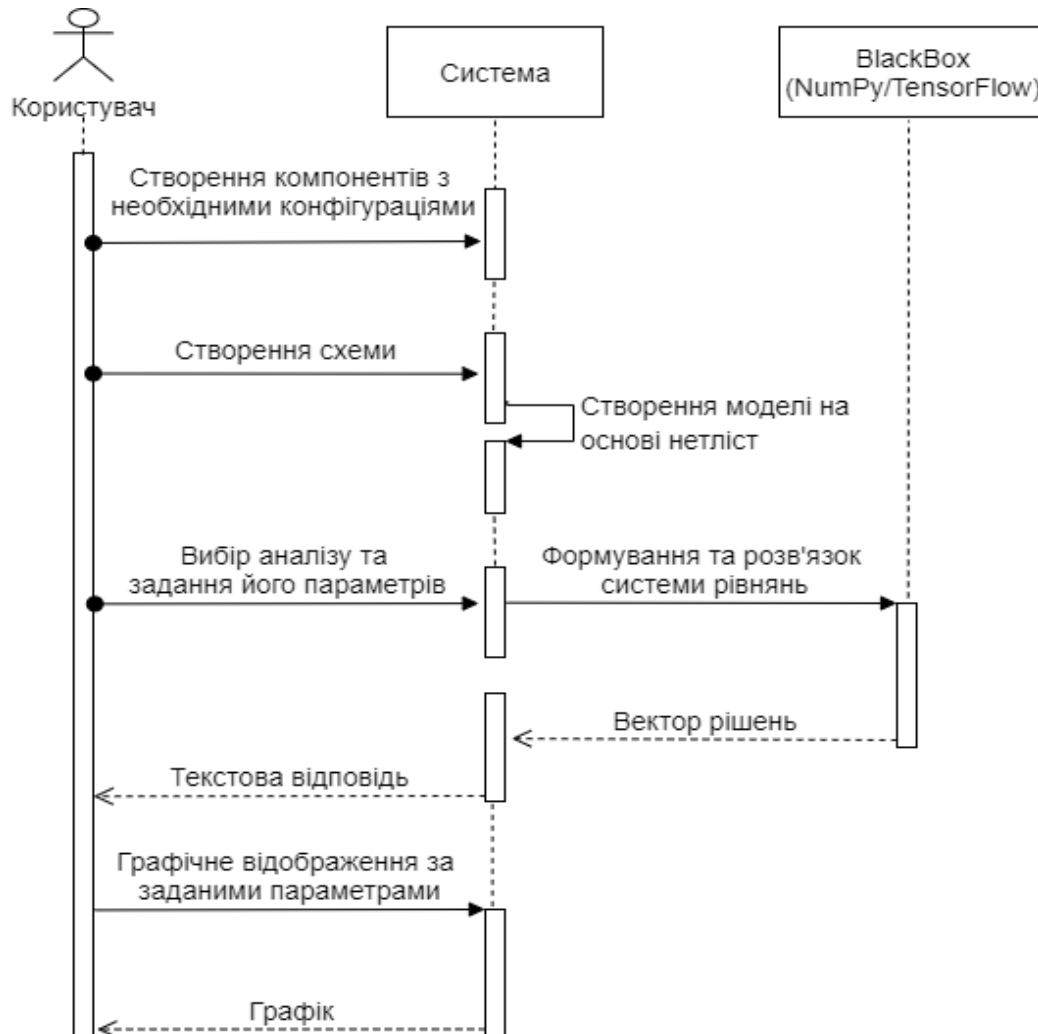


Рисунок 3. Діаграма послідовності роботи програми

Зобразимо детальний опис (Рис. 3) логіки можливого сеансу використання даної програми. Користувач створює компоненти необхідні йому для моделювання. Потім необхідно створити об'єкт схеми Circuit та передати йому список вузлів для формування математичної моделі. При виклику методу для аналізу схеми користувачу буде надано рішення рівнянь рівноваги для даної моделі у вигляді текстової та графічної відповіді.

У додатку присутній базовий набір лінійних елементів та нелінійні елементи, можливість вибору методу побудови рівнянь рівноваги для представлення електричного кола та декілька видів аналізу електричної схеми.

#### **4. ВИСНОВКИ**

На сьогоднішній день існують потужні спеціалізовані програми для моделювання динамічних систем, такі як ALLTED та PSPICE. Даний додаток не має на меті конкурувати з ними, але може стати корисним інструментом для навчання математичного моделювання та більш детальному вивченню фізичних процесів пов'язаних з ним, з можливістю переглядати кожен крок від побудови систем рівнянь за обраною моделлю до аналізу побудованої схеми.

#### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Л.О. Чуа, Пен-Мин Лин Машинний аналіз електронних схем
2. Петренко Анатолій Іванович. Конспект лекцій до курсу „МОДЕЛЮВАННЯ СИСТЕМ” // Київ-2020. Ст.3-5, ст.27-28.
3. Петренко, Чкалов, Кірюша. «МОДЕЛЮВАННЯ СКЛАДНИХ СИСТЕМ» // Київ-2015. ст. 6-18.
4. [https://kpi.ua/web\\_allted](https://kpi.ua/web_allted)
5. <https://www.pspice.com/>

# СИСТЕМА ВІДСТЕЖЕННЯ ПОГЛЯДУ В РЕАЛЬНОМУ ЧАСІ ДЛЯ ВЗАЄМОДІЇ З КОМП'ЮТЕРОМ НА ОСНОВІ МЕТОДІВ ГЛИБОКОГО НАВЧАННЯ

Гончарук О.П.<sup>1</sup>, Недашківська Н.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>honcharuk.oleksandr@lil.kpi.ua,

<sup>2</sup>nedashkovskaya.nadezhda@lil.kpi.ua [0000-0002-8277-3095]

**Різноманіття кількості способів вводу даних та покращення їх точності збільшує продуктивність та зручність взаємодії людини та будь-якої системи. У випадках коли доступ до звичайних пристроїв вводу відсутній, обмежений або неможливий, то їх доповненням або заміником можуть стати системи, що дозволяють керувати пристроєм за допомогою погляду. Метою роботи є вдосконалення методів відстеження погляду користувача, використовуючи методи глибокого навчання. Реалізовано систему відстеження погляду, використовуючи мікросервісну архітектуру.**

**Ключові слова:** відстеження погляду, глибоке навчання, згорткові нейронні мережі, механізм уваги, обробка зображень, комп'ютерний зір.

## 1. ВСТУП

Із розвитком технологій та їх активним впровадженням в повсякденне життя людини, все частіше виникає питання про ефективність взаємодії із цифровими пристроями. Такі параметри, як швидкість набору символів на клавіатурі або швидкість переміщення миші, мають конкретні межі. Також існують люди, які через фізіологічні причини не можуть повноцінно використовувати певні пристрої вводу. Враховуючи це, можливість управління пристроями за допомогою погляду є одним із найперспективніших напрямків досліджень та розробки.

Причин, щоб користуватися таким способом може бути доволі багато та вони можуть бути індивідуальними для конкретного користувача. Починаючи з особливостей будови пристрою (Наприклад, сервери в дата центрах), які не мають жодної периферії, закінчуючи виходом з ладу пристрою вводу, але гострою необхідністю використання цього пристрою.

Більшість технологій, що були створені для зчитування погляду використовують дорогі датчики. Вони надають достатньо точні виміри, але є надмірно дорогими для впровадження на глобальному ринку.

Однією із задач, що була поставлена при розробці даної системи, є надання можливості керувати пристроєм в реальному часі із присутньою камерою.

## 2. ОПИС НАБОРУ ДАНИХ

В межах цього дослідження було використано набір даних Gaze Capture [1], який містить 2.5 мільйони зображень від 1450-ти різних людей. Вказаний датасет містить інформацію, яка дозволяє ідентифікувати де саме на моніторі сконцентрований погляд, використовуючи лише зображення обличчя людини.

Мінливість даних дає загальну оцінку репрезентативності даних. Для цього в роботі було використано метод Dataset Characteristics [1], що оцінює позицію голови (h) та напрям погляду (g) для кожного зображення. Порівняння кінцевих розподілів величин h та g для обраного набору даних GazeCapture та інших наборів MPIIGaze [2] та TabletGaze [3] (Рис. 1) підтверджує варіативність положення голови та погляду у різних наборах, що безпосередньо впливає на якість навчання згорткових нейронних мереж.

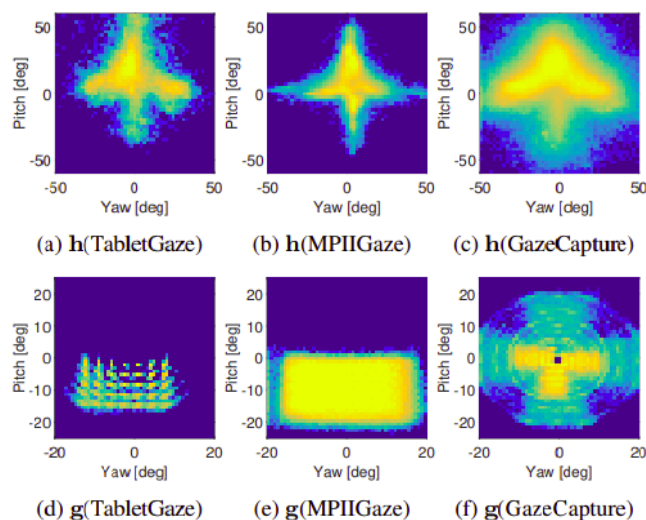


Рисунок 1. Розподіл положення голови і напрямку погляду в порівнянні з існуючими наборами даних [1]

### 3. АРХІТЕКТУРА НЕЙРОННОЇ МЕРЕЖІ ТА ОСОБЛИВОСТІ НАВЧАННЯ

У цій роботі розглядається підхід, що використовує інформацію представлену у вигляді зображення для передбачення координати погляду на моніторі персонального комп'ютера. Для отримання такого прогнозу пропонується модифікація відомої архітектури нейронної мережі [1] на основі згорткових нейронних мереж шляхом додавання механізму уваги [4].

На вхід моделі передається: зображення лица із відповідним його розміщенням на зображенні, іменується як маска лица; і додатково зображення лівого та правого ока відносно розміщення лица (голови). Комбінуючи цю інформацію, модель здатна визначати координати погляду користувача. Загальна архітектура нейронної мережі із відповідними розмірностями зображена на Рис. 2.

Варто звернути увагу, що зображення лівого і правого ока передаються на вхід мережі як окремі входи, хоча саме лице вже містить дану інформацію. Даний підхід було використано з метою збільшення роздільної здатності зображення, що спрощує процес ідентифікації напрямку погляду нейронною мережею.

Оскільки спостереження включають використання різних цифрових пристроїв: телефонів, планшетів та персональних комп'ютерів, необхідно було створити універсальну цільову зміну. В даному випадку - це двовимірний вектор з координатами x, y відносно камери, які приймають значення в сантиметрах.

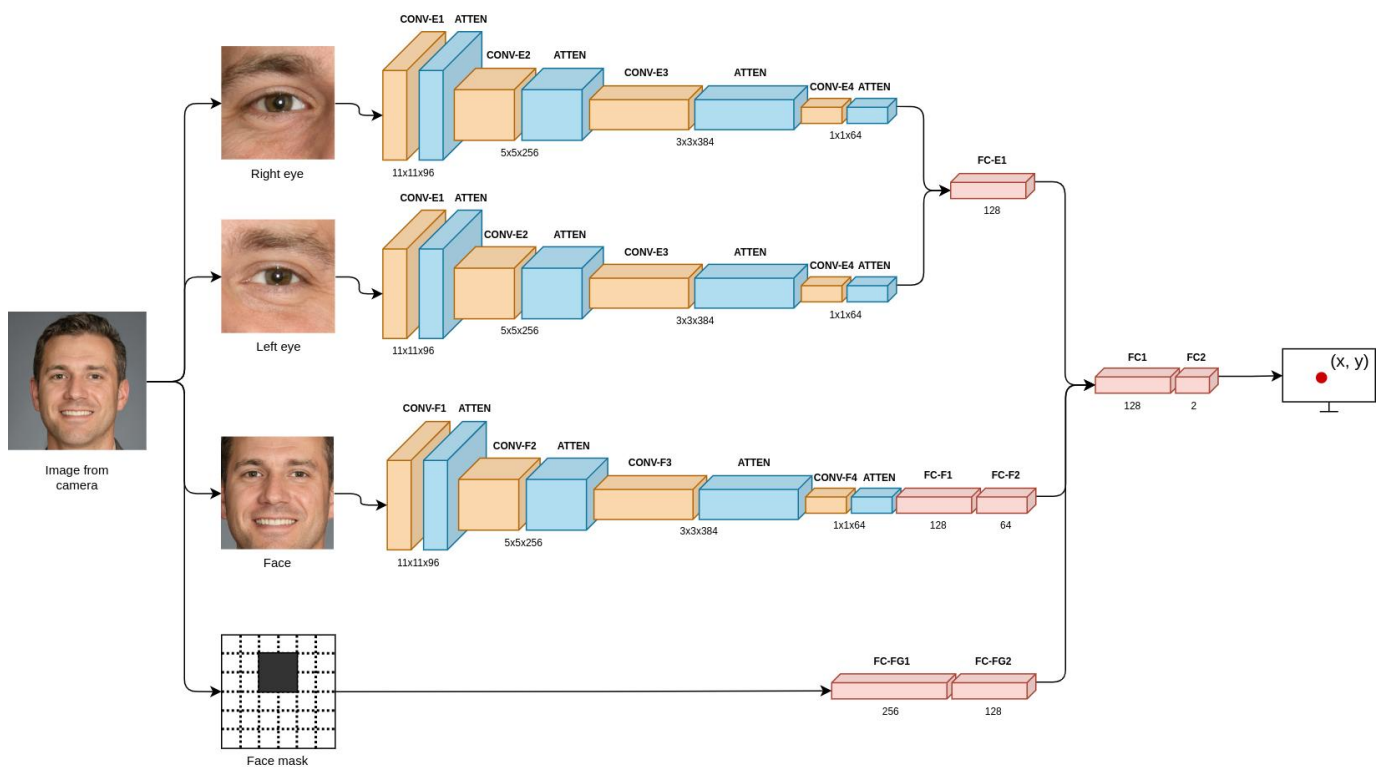


Рисунок 2. Високорівнева архітектура нейронної мережі

Датасет GazeCapture містить 2 445 504 зображень, серед яких лише 1 490 059 використовуються для навчання, валідації та тестування. А саме, спостереження отримані від 1471 людини використовувались для навчання нейронної моделі, зображення отримані від 50 людей – для валідації, і зображення від 150 людей – для перевірки кінцевої якості моделі.

Враховуючи, що розміри кожного зображення відрізняються, було проведено їх трансформацію до єдиного розміру 224 на 224 пікселів. Архітектуру нейронної мережі було реалізовано за допомогою фреймворку Torch для глибокого навчання [5].

Пропонована на Рис. 2 архітектура моделі розбита на функціональні блоки (Табл. 1 і 2).

У Табл. 1 у позначенні Conv2d(a/b/c/d) використано:  $a$  - кількість фільтрів,  $b$  - розмір ядра згортки,  $c$  - крок зміщення,  $d$  - padding; після кожного згорткового шару Conv2d використовується активаційна функція ReLU [6]; у позначенні об'єднуючого шару MaxPool2d (a/b):  $a$  - розмір вікна,  $b$  - крок зміщення; у BatchNorm2d (a) під  $a$  розуміється кількість фільтрів; у шарі MultiHead (a/b):  $a$  - кількість фільтрів,  $b$  - кількість шарів уваги.

У позначенні Linear(a/b):  $a$  - кількість вхідних нейронів,  $b$  - кількість вихідних нейронів. У кожному повнозв'язному шарі застосовується активаційна функція ReLU.

Таким чином, на виході нейронної мережі отримуємо два значення, які відповідають за координату погляду  $x$  та  $y$  відповідно.

Оскільки дана задача відстеження погляду формулюється як регресійна задача, то в якості функції втрат, що оптимізується, було обрано функцію похибки Губера [7]:

$$l_n = \begin{cases} 0.5 x_n - y_n^2, & \text{if } x_n - y_n < \text{delta} \\ \text{delta} * x_n - y_n - 0.5 * \text{delta}, & \text{otherwise} \end{cases}$$

Зазначена функція об'єднує переваги Манхетенської метрики (L1), яка менш чутлива до викидів, а також Евклідової метрики (L2), яка забезпечує згладжування в околі нуля.



Таблиця 1. Модуль обробки зображень

Номер шару	Складові модуля
1	Conv2d(96/11/4/0) MaxPool2d(3/2) BatchNorm2d(96) MultiHead(96/3)
2	Conv2d(256/5/1/2) MaxPool2d(3/2) BatchNorm2d(256) MultiHead(96/3)
3	Conv2d(384/3/1/1) Conv2d(64/1/1/0)

Таблиця 2. Повнозв'язні шари для агрегування даних

Номер шару	Eyes Module	Face	Grid	Aggregate Module
1	Linear(18432, 128)	Linear(9216, 128)	Linear(784, 256)	Linear(320, 128)
2		Linear(9216, 128)	Linear(256, 128)	Linear(128, 2)

В якості алгоритму навчання (optimizer) було вибрано Adam [8] із коефіцієнтом навчання learning rate = 0.0003, параметрами  $\beta_1 = 0.9$  та  $\beta_2 = 0.999$ . Додатково використовувався параметр weight\_decay = 0.0001, як аналог L2 регуляції із адаптацією коефіцієнта навчання.

Процес навчання зображено на Рис. 4. Батч містив 128 зображень, навчання проводилося протягом 100 епох. З Рис. 4 бачимо, що функція втрат, в даній задачі це функція похибки Губера (Huber Loss), стрімко зменшує своє значення на перших епохах навчання, і після сорокової епохи виходить на плато як на навчальній (Train), так і на валідаційній (Valid) підмножинах.

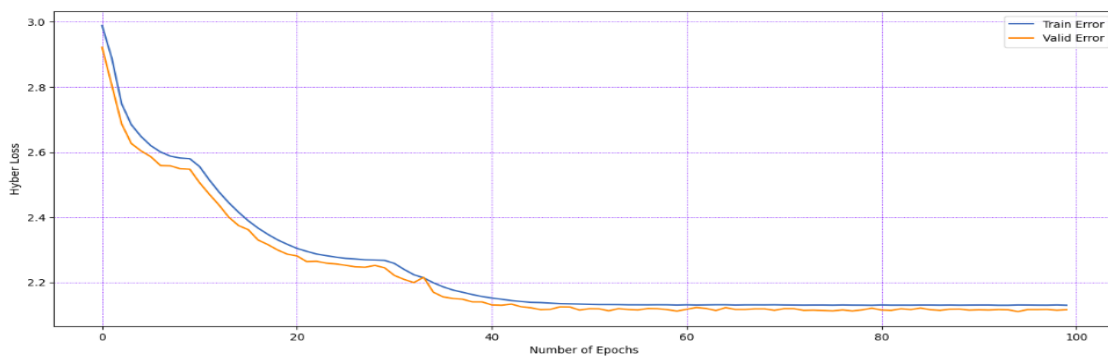


Рисунок 4. Прогрес навчання нейронної мережі

## 4. РЕЗУЛЬТАТИ

В Табл. 3 наведено значення показника точності MAE (mean absolute error) на валідаційній підмножині окремо для кожної координати прогнозованого цільового вектора. Для порівняння, в табл. 3 наведено значення для запропонованої моделі і для іншої відомої моделі iTracker [1] лише для планшетів без калібрування.

Таблиця 3. Порівняння отриманих результатів

Модель	Значення MAE, см	
	координата x	координата y
iTracker [1]	1.04	2.31
Пропонована модель	0.98	2.17

Як видно з Табл. 3, запропонована модель показала кращі результати порівняно з відомою iTracker: значення MAE є меншими на 5.769% та 6.061% для координати x та y, відповідно.

## 5. ВИСНОВКИ

В ході роботи було досліджено і встановлено інструменти прогнозування погляду користувача на основі зображення обличчя.

Запропоновано модифіковану модель глибокої нейронної мережі для відстеження погляду і взаємодії з комп'ютером в реальному часі на основі згорткових шарів із додаванням механізму уваги. Розроблено систему відстеження погляду у реальному часі з використанням фреймворку Torch Python.

В результаті порівняльного аналізу встановлено, що розроблена система за показником MAE (значення 0.98 см і 2.17 см для координат x і y), не поступається перед існуючою системою iTracker, у якої MAE сягають для координат x та y значення 1.04 см та 2.31 см, відповідно.

Перспективи подальших досліджень полягають у пришвидшенні роботи системи. Також одним із вдосконалень є додавання можливості розгортання частини системи у хмарі, що дозволить перенести ресурсоємні обчислення з клієнтської частини системи.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Eye Tracking for Everyone [Електронний ресурс] / [К. Kraffka, A. Khosla, P. Kellnhofer та ін.]. – 2016. – Режим доступу до ресурсу: <https://gazecapture.csail.mit.edu/>.
2. Appearance-based gaze estimation in the wild. [Електронний ресурс] / X.Zhang, Y. Sugano, M. Fritz, A. Bulling. – 2015. – Режим доступу до ресурсу: [https://openaccess.thecvf.com/content\\_cvpr\\_2015/papers/Zhang\\_Appearance-Based\\_Gaze\\_Estimation\\_2015\\_CVPR\\_paper.pdf](https://openaccess.thecvf.com/content_cvpr_2015/papers/Zhang_Appearance-Based_Gaze_Estimation_2015_CVPR_paper.pdf).
3. Huang Q. TabletGaze: Unconstrained Appearance-based Gaze Estimation in Mobile Tablets [Електронний ресурс] / Q. Huang, A. Veeraraghavan, A. Sabharwal. – 2015. – Режим доступу до ресурсу: <https://arxiv.org/abs/1508.01244>.
4. Islam K. Face Pyramid Vision Transformer [Електронний ресурс] / K. Islam, M. Zaigham, A. Mahmood. – 2022. – Режим доступу до ресурсу: <https://arxiv.org/abs/2210.11974>.
5. PyTorch Foundation [Електронний ресурс] – Режим доступу до ресурсу: <https://pytorch.org/foundation>.
6. Fred A. Deep Learning using Rectified Linear Units (ReLU) [Електронний ресурс] / Abien Fred. – 2018. – Режим доступу до ресурсу: <https://arxiv.org/abs/1803.08375>.
7. Gokcesu K. Generalized Huber Loss for Robust Learning and its Efficient Minimization for a Robust Statistics [Електронний ресурс] / K. Gokcesu, G. Nakan. – 2021. – Режим доступу до ресурсу: <https://arxiv.org/abs/2108.12627>.
8. Kingma D. Adam: A Method for Stochastic Optimization [Електронний ресурс] / D. Kingma, B. Jimmy. – 2014. – Режим доступу до ресурсу: <https://arxiv.org/abs/1412.6980>.

# ВИЗНАЧЕННЯ ЕМОЦІЙНОГО ЗАБАРВЛЕННЯ ГОЛОСУ ЗА ДОПОМОГОЮ ТЕХНІК DEEP LEARNING У РЕАЛЬНОМУ ЧАСІ

Загарницький Д.В.<sup>1</sup>, Шаповал Н.В.<sup>2</sup>

КПІ ім. Ігоря Сікорського, Київ, Україна

<sup>1</sup> dmitriy.zaharnitskiy@gmail.com, <sup>2</sup> shovgun@gmail.com

Метою дослідження є аналіз можливостей сучасних глибоких нейронних мереж для вирішення задач аналізу емоційного забарвлення голосу. За основу була взята глибока нейронна мережа AlexNet і за допомогою transfer learning вона була донавчена на розпізнавання емоцій на спектрограмах аудіо файлів, які були перетворені у RGB зображення. За дані було взято Berlin Emotional Speech (ЕМО-DB) датасет з якого були створенні відрізки даних довжиною до 1.04 секунди. У результаті ми отримали середню точність класифікації емоцій у 80.5%. Основне практичне значення роботи, це можливість удосконалення сервісів типу «людина-машина» для покращення їх взаємодії у відповідність до емоційного стану людини.

**Ключові слова:** емоційне забарвлення голосу, мережа AlexNet, transfer learning, RGB, голосові запити клієнтів.

## 1. ВСТУП

Розпізнавання емоцій голосу, або speech emotion recognition (SER) — це завдання розпізнавання емоційних аспектів мовлення незалежно від семантичного змісту. Хоча люди можуть ефективно виконувати це завдання як природну частину мовного спілкування, здатність виконувати це автоматично за допомогою програмованих пристроїв все ще є предметом досліджень.

Дослідження систем автоматичного розпізнавання емоцій спрямовані на створення ефективних методів виявлення емоцій у режимі реального часу користувачів мобільних телефонів, операторів колл-центру та клієнтів, водіїв автомобілів, пілотів та багатьох інших користувачів спілкування «людина-машина». Додавання емоцій до машин було визнано критичним фактором, що змушує машини виглядати та діяти схожими на людину [1].

Традиційно машинне навчання (ML) передбачає обчислення ознак (features) із необроблених даних (наприклад, мова, зображення, відео, ЕКГ, ЕЕГ). Ознаки використовуються для навчання моделі, яка вчиться класифікувати їх. Поширеною проблемою, з якою стикається цей підхід, є вибір цих ознак. Загалом невідомо, які ознаки можуть призвести до найбільш ефективної класифікації даних у різні категорії (або класи). Деяке розуміння можна отримати, протестувавши велику кількість різних ознак, об'єднавши різні ознаки в загальний вектор ознак або застосувавши різні методи вибору ознак. Якість створених вручну елементів може мати значний вплив на продуктивність класифікації.

Завдяки появі класифікаторів на базі глибоких нейронних мереж (DNN) було запропоновано елегантне рішення, яке обходить проблему оптимального вибору ознак. Ідея полягає у використанні мережі, яка приймає необроблені дані як вхідні дані та генерує мітку класу як вихідні дані. Немає потреби ні обирати вручну ознаки, ні визначати, які параметри є

оптимальними з точки зору класифікації. Ціною цього дуже зручного рішення є набагато більші вимоги до мічених зразків даних порівняно зі звичайними методами класифікації.

У багатьох випадках, і це включає SER, лише мінімальні дані доступні для цілей навчання. Як показано в цьому дослідженні, проблему обмежених навчальних даних значною мірою можна подолати підходом, відомим як трансферне навчання. Воно використовує існуючу мережу, попередньо навчену на великих даних, для вирішення загальної проблеми класифікації. Потім ця мережа додатково навчається з використанням невеликої кількості доступних даних для вирішення більш конкретного завдання.

Враховуючи, що на даний момент найпотужніші попередньо навчені нейронні мережі навчені для класифікації зображень, щоб застосувати ці мережі до проблеми SER, сигнал голосу потрібно перетворити у формат зображення [2]. Це дослідження описує кроки, пов'язані з переходом від голосу до зображення; у ньому пояснюються процедури навчання та тестування, а також умови, які необхідно виконати, щоб досягти розпізнавання емоцій у режимі реального часу з безперервної потокової мови.

## 2. МЕТОДИ ОБРОБКИ ТА ПІДГОТОВКИ ДАНИХ

Враховуючи, що доступні обчислювальні ресурси були обмежені, і була доступна лише невелика база даних емоційно позначених зразків мовлення, метою було визначити обчислювально ефективний підхід, який міг би працювати з невеликим набором навчальних даних. Для кожного блоку був розрахований масив спектрограм, який перетворювався у формат зображення RGB і передавався як вхідні дані до попередньо навченого CNN. Після відносно короткого навчання навчена CNN була готова класифікувати емоції з нерозміченої (потокової) мови, використовуючи той самий процес перетворення мови в зображення. У представлених тут експериментах продуктивність SER перевірялася з використанням двох різних частот дискретизації (16 і 8 кГц) і процедури компандування  $\mu$ -low. Система SER була реалізована за допомогою мови програмування Python та робочої станції із процесором AMD Ryzen 5 2600X, 3,6 ГГц, 32 ГБ оперативної пам'яті та NVIDIA GeForce 1060 3GB.

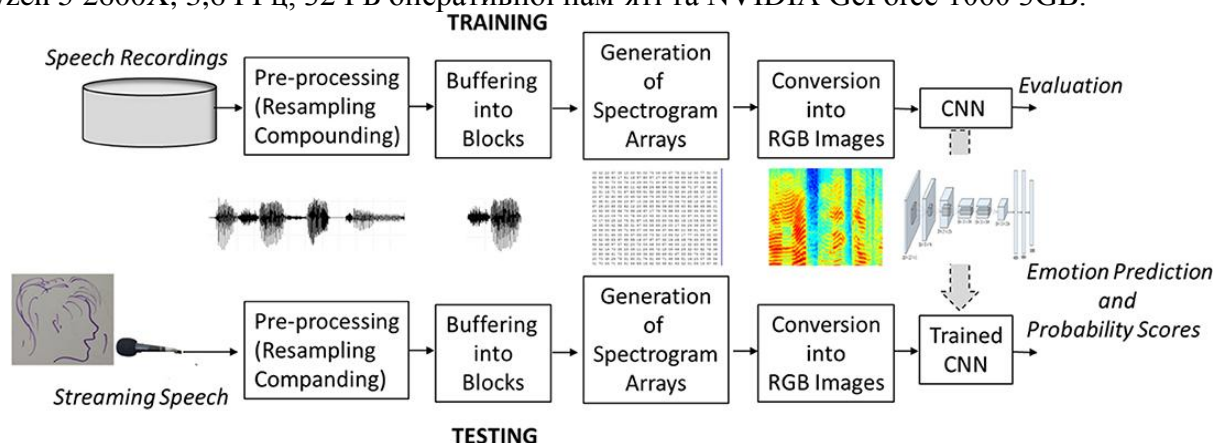


Рисунок 1. Огляд методу SER у реальному часі

У традиційних вузькосмугових системах передачі даних смуга пропускання мовного сигналу була обмежена, щоб збільшити швидкість передачі. У телефонії, наприклад, діапазон частот мови раніше обмежувався діапазоном від 300 Гц до 3,4 кГц. Цього було достатньо, щоб забезпечити базовий рівень розбірливості мови, але ціною високої якості голосу. Цілком ймовірно, що таке серйозне скорочення пропускну здатності призвело до значного зменшення емоційної інформації, що передається мовцями.

Щоб перевірити цю можливість, систему SER було навчено з двома різними частотами дискретизації: початкова 16 кГц відповідає широкій смузі пропускання 8 кГц, а зменшена частота дискретизації 8 кГц відповідає вузькій смузі частот 4 кГц.

Враховуючи оригінальні зразки мовлення  $x$ , компресія відрізків голосу  $F(x)$  була розрахована як

$$F(x) = \frac{\ln(1+\mu|x|)}{\log(1+\mu)^{\text{sgn}(x)}}$$

Тоді як реконструйовані мовні зразки  $\tilde{x}$  були розраховані як

$$\tilde{x} = F^{-1}(F(x)) = \text{sgn}(F(x)) \frac{((1+\mu)^{|F(x)|} - 1)}{\mu}$$

Значення параметра стиснення  $\mu$  було встановлено на 255 (стандарт у США та Японії) [3].

Потокове або записане мовлення було буферизовано в блоки по 1 с для проведення поблочної обробки. Між наступними блоками застосовувався короткий крок тривалістю 10 мс. Рівні амплітуди були нормалізовані в діапазоні від -1 до 1.

Процедура, яка використовується для генерації масивів спектрограм, проілюстрована на рисунку 2. Короткочасне перетворення Фур'є було виконано для кожного 1-секундного блоку мовних сигналів з використанням кадрів 16 мс, створених із застосуванням функції вікна Хеммінга зі зсувом у часі.

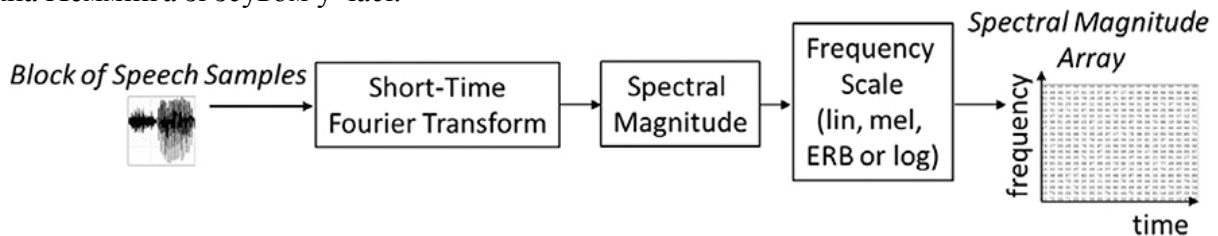


Рисунок 2. Генерація масивів спектрограм

Масиви спектральних величин  $257 \times 259$  дійсних чисел були перетворені у формат кольорового зображення RGB, представлений трьома масивами кольорових компонентів (рис. 3).

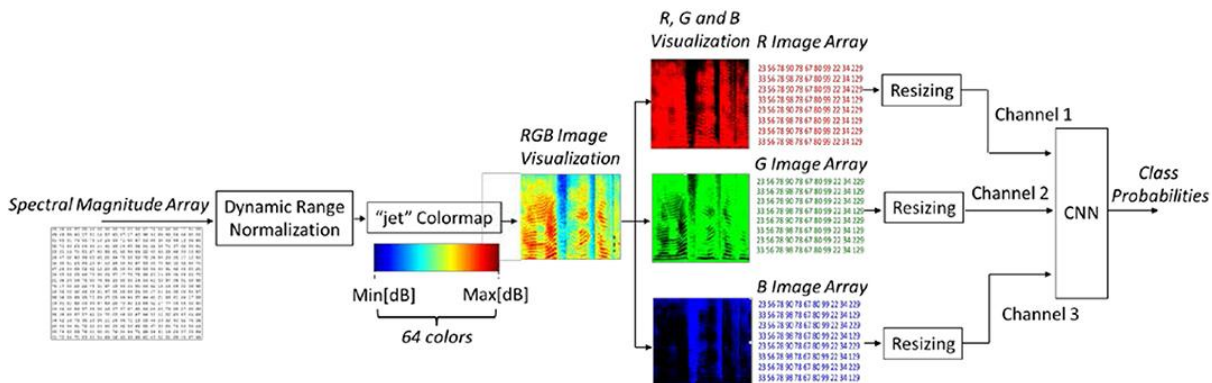


Рисунок 3. Перетворення масивів спектральних амплітуд у масиви зображень R, G та B.

### 3. МЕРЕЖА КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ ALEXNET

AlexNet — це згорточна нейронна мережа (CNN), представлена Крижевським та ін. (2012). Її було попередньо навчено на понад 1,2 мільйонах зображень із набору даних ImageNet Стенфордського університету, щоб розрізняти 1000 категорій об'єктів. Він складається з 3-канального вхідного шару, що дозволяє вводити три 2-вимірні масиви, кожен розміром  $256 \times 256$  пікселів. Після вхідного рівня йдуть п'ять згорткових шарів (Conv1–Conv5), кожен з яких має шар максимального об'єднання та нормалізації (рис. 4). Двовимірні вихідні об'єкти з останнього згорткового шару Conv5 перетворюються в одновимірні вектори та подаються в три повністю з'єднані шари (fc6–fc8). У той час як згорткові шари витягують характерні ознаки з вхідних даних, повністю пов'язані шари вивчають параметри моделі класифікації даних. Експоненціальна функція SoftMax відображає вихідні значення fc8 у нормалізований вектор дійсних значень, які потрапляють у діапазон  $[0,1]$  і в сумі дають 1. Ці значення надаються на вихідному рівні та представляють ймовірності кожного класу. Остаточна класифікаційна позначка присвоюється класу, який отримав найвищий бал вірогідності.

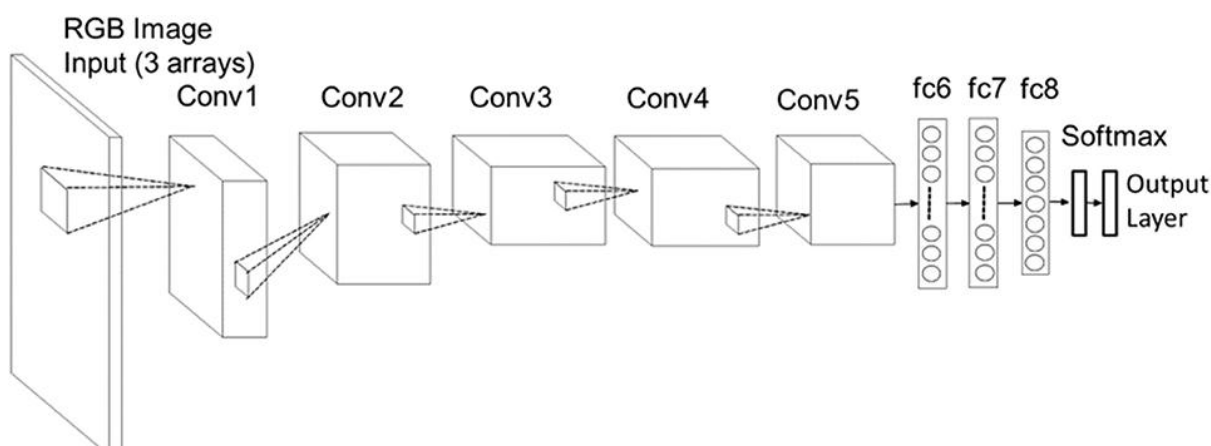


Рисунок 4. Структура AlexNet

Після адаптації до класифікації семи емоцій AlexNet було навчено (fine-tuned) на позначених даних. Оскільки мережа вже була попередньо навчена, процес навчання був набагато швидшим і “можливим” порівняно з тим, що потрібно було б під час навчання тієї самої структури мережі з нуля. Однак можливо, що за наявності необхідних ресурсів навчання з нуля могло б привести до кращих результатів. Хоча в останні роки AlexNet конкурували зі значно складнішими мережевими структурами [4].

У transfer learning процес навчання має на меті досягнути найвищого впливу навчання на кінцевих, повністю пов'язаних рівнях мережі, залишаючи попередні рівні майже недоторканими. Щоб досягти швидшого навчання в нових модифікованих шарах і повільнішого в старих шарах, початкова швидкість навчання (learning rate) була встановлена на невелике значення, а значення weight learning rate та bias learning rate були збільшені лише для кінцевих шарів.

### 4. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

Усі експерименти адаптували техніку 5-кратної cross-validation з 80% розподілу даних для навчання AlexNet і 20% для тестування. Зразки даних тестування ніколи не використовувалися під час процедури навчання мережі. Експерименти не залежали від мовця та статі.

Таблиця 1. Інформація про набір даних

Емоція	Кількість зразків	Загальна тривалість (сек.)	Кількість згенерованих зображень
Злість	129	335	27220
Нудьга	79	220	18125
Огида	38	127	11010
Страх	55	123	5463
Задоволення	58	152	12400
Нейтральна	78	184	14590
Сум	53	210	18455
Загально	390	1207	111425

Метриками для оцінки виступили Accuracy, Precision, Recall та F-score.

Застосування SER в режимі реального часу було досягнуто шляхом поблочної обробки. Для кожного блоку була створена класифікаційна мітка, яка вказує на одну з семи категорій емоційного класу. Після навчання моделі були проведені наступні експерименти:

- Експеримент 1: частота дискретизації 16 кГц (смуга пропускання = 8 кГц);
- Експеримент 2: частота дискретизації 8 кГц (смуга пропускання = 4 кГц);
- Експеримент 3: частота дискретизації 16 кГц,  $\mu$ -low компандування;
- Експеримент 4: частота дискретизації 8 кГц,  $\mu$ -low компандування.

Таблиця 2. Результати експериментів

Експеримент	Weighted precision (%)	Weighted recall (%)	Weighted F-score (%)	Weighted accuracy (%)
1	80.3	80.8	79.6	80.5
2	76.8	78.0	76.1	76.8
3	75.8	77.1	75.3	75.8
4	73.7	76.1	73.1	73.7

Таблиця 3. Середній час обчислення в мілісекундах (мс)

Експеримент	Час інференсу	Створення ознак			Загальний час
		Створення масивів спектрограм	Перетворення у RGB зображення	Загальний час створення ознак	
1	18.7	8.0	3.6	11.6	30.3
2	18.6	4.7	3.6	8.3	26.9
3	18.4	7.5	3.6	11.1	29.5
4	18.5	4.6	3.6	8.2	26.7

#### 4. ВИСНОВОК

Підсумовуючи, ми виявили, що використання CNN для класифікації емоцій голосу дає точність у 80.5% і швидкість інференсу достатня для аналізу емоцій у реальному часі. Також було показано, що обидва фактори, зменшення смуги пропускання мовлення та впровадження процедури компандування мовлення  $\mu$ -low, мають шкідливий вплив на результати SER. За рахунок зменшення частоти дискретизації з 16 до 8 кГц (тобто, зменшення смуги пропускання з 8 до 4 кГц) спостерігалось невелике зниження середньої точності SER на (близько 3,3%). Процедура компандування зменшила результат на таку саму величину (приблизно на 3,8%), а сукупний вплив обох факторів призвело до зниження приблизно на 7% порівняно з базовими результатами. У всіх експериментальних випадках SER виконувався в режимі реального часу з емоційними мітками, що генерувалися кожні  $\sim 1,04$  с.

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Андре, Е., Рем, М., Мінкер, В., і Бюлер, Д. «Наділення систем розмовного мовного діалогу емоційним інтелектом», у навчальному та дослідницькому семінарі з афективних діалогових систем, ADS 2004, ред. Е. Андре, Л. Дибк'єр, П. Хайстеркамп і В. Мінкер (Німеччина: Kloster Irsee), 2004. 178–187.
2. Столар, М. Н., Лех, М., Боля, Р. Б., і Скіннер, М. «Розпізнавання емоцій голосу у реальному часі за допомогою класифікації зображень RGB і transfer learning», 2017. 1–6.
3. Методи кодування сигналу. URL: <https://www.cisco.com/c/en/us/support/docs/voice/h323/8123-waveform-coding.html> (дата звернення: 15 листопада 2022 р.).
4. Жегеді, С., Луї, В., Джіа, І., Серманет, П., Рід, С., Ангелов, Д. та ін. «Поглиблюючись із CNN», у матеріалах Конференції IEEE з комп'ютерного бачення та розпізнавання образів, 2015. 1–9.
5. EmoDB Dataset. URL: <https://www.kaggle.com/datasets/piyushagni5/berlin-database-of-emotional-speech-emodb> (дата звернення: 10 листопада 2022 р.).



# ПРОГНОЗ ТРАЄКТОРІЇ УЧАСНИКІВ ДОРОЖНЬОГО РУХУ З ВИКОРИСТАННЯМ КІЛЬКОХ КАМЕР

Загній Є.В.<sup>1</sup>, Данилов В.Я.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> egor.zagny@gmail.com,

<sup>2</sup> danilov1950@ukr.net [0000-0003-3389-3661]

**Для роботи автопілота машини або допоміжної системи водія непогано знати траєкторії руху об'єктів навколо автомобіля. Для прогнозування траєкторій об'єктів навколо авто потрібно на зображеннях з камер розпізнати об'єкти, оцінити відстані до них щоб визначити їх місцезнаходження і відслідковувати переміщення об'єктів у часі. Для цього потрібно застосувати цілий комплекс нейронних мереж, що і буде досліджено в даній роботі.**

**Ключові слова:** машинне навчання, коп'ютерний зір, розпізнавання об'єктів, оцінка відстаней, відслідковування об'єктів, прогноз траєкторій.

## 1. ВСТУП

За останні 10 років людство досягло значного успіху у галузі штучного інтелекту. Це стало можливим завдяки значному зростанню обчислюваних потужностей комп'ютерів, завдяки чому з'явилася можливість тренувати великі нейронні мережі. Штучний інтелект вже є частиною звичного життя людей. Одним з напрямком штучного інтелекту є створення автопілотів для машин. Створення автопілоту неможливо без розв'язання задачі розпізнавання машин і пішоходів навколо машини, знаходження відстаней до них, відслідковування переміщення цих об'єктів, побудови карти з висоти пташиного польоту з розміщенням на ній цих об'єктів, та прогнозуванням траєкторій сусідніх машин.

Метою роботи є дослідження й порівняння різних методів, моделей, структур нейронних мереж для вирішення кожної з підзадач: розпізнавання об'єктів навколо машини, знаходження відстаней до них, побудови виду з висоти пташиного польоту навколо машини, відслідковування переміщення кожного об'єкта та прогнозування траєкторій об'єктів.

## 2. ЕТАПИ ЗАДАЧІ ПРОГНОЗУВАННЯ ТРАЄКТОРІЙ

Для того щоб працювати з траєкторіями об'єктів, спочатку треба розпізнати об'єкти на зображеннях що видає камера, визначити їх місцезнаходження, та відслідкувати переміщення цих об'єктів відносно попередніх кадрів з камери. [1-3]

Нейронні мережі складаються з тіла та голови. Існують багато видів тіл і голів. Доволі часто тіло і голова мережі вибираються окремо. Тіло нейронної мережі вибирається в залежності від того яка швидкість і точність роботи нейронної мережі потрібна. Чим більше тіло тим менша швидкість роботи, але тим кращу точність зможе видати нейронна мережа. Існують такі види тіл як MobileNet, VGG, ResNet. В задачах комп'ютерного зору на вхід нейронних мереж в більшості випадків подається просто зображення. Проте вихід нейронної мережі залежить від задачі і може дуже сильно відрізнятися, тому в першу чергу нас буде цікавити голова нейронної мережі.

Задача розпізнавання об'єктів полягає у тому щоб знайти на зображенні об'єкти певного класу. Зазвичай кожен об'єкт буде позначатися прямокутником з координатами його розміщення на зображенні, а також ідентифікатор класу об'єкта.

Для задачі розпізнавання об'єктів існують такі голови: CenterNet, SSD, Yolo. Розглянемо центернетівську архітектуру.

Суть центернетівської архітектури полягає в тому, що нейронна мережа повертає для кожного класу об'єкта матрицю в якій вказані ймовірності того що в цій частині зображення є даний об'єкт. Цю матрицю називають тепловою картою. Всі локальні максимуми які перевищують певне значення й будуть центрами цих об'єктів. В додаткових матрицях буде ширина й висота об'єктів.

Отже на вході зображенням  $I \in \mathbb{R}^{W \times H \times 3}$  ширини  $W$  і висоти  $H$ . На виході для кожного класу розпізнаваних об'єктів тепла карта  $Y \in [0,1]^{(W/R \times H/R \times C)}$ , ширина та висота виходу нейронної мережі менше вхідного зображення у 2 або 4 рази. Хочемо, щоб у точці, що є центром об'єкта  $Y(x,y,c) = 1$ , тоді як  $Y(x,y,c) = 0$  означає, що тут нічого немає. Тому під час тренування мережі при генеруванні ground truth даних значення на теплових картах навколо центрів об'єктів генеруємо за допомогою Гаусівського ядра середнє квадратичне якого залежить від розміру об'єкта

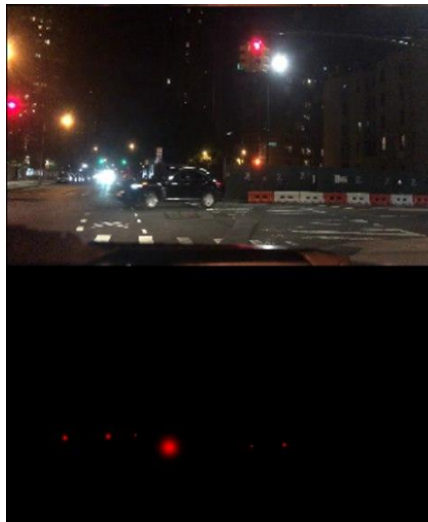


Рисунок 1. Генерація ground truth теплової карти для зображення

На цьому рисунку можна побачити приклад генерації ground truth гаусівськими ядрами, які розміщені в центрі машин та людей.

Окрім центрів об'єктів нам потрібно знайти їх ширину та висоту. Для цього до виходу моделі додаємо ще 4 шари, для визначення відстаней до правого, лівого, верхнього і нижнього країв прямокутника, що позначає об'єкт на зображенні. Тому після того як ми визначили центри об'єктів, визначаємо довжину і ширину зі значень по координатах центрів об'єктів з цих чотирьох шарів. Тому вихід CenterNet буде мати кількість каналів, що дорівнює кількості класів плюс чотири.

Функцією втрат під час тренувань буде сума функцій втрат з теплових карт та з шарів для визначення розмірів об'єкта, при цьому функція втрат для розмірів об'єкта працює тільки з точками що є центрами об'єктів, усі інші точки вона ігнорує.

Одним з головних плюсів CenterNet є те, що так само як визначається розміри об'єкту можна визначити будь які характеристики об'єкта просто додаючи додаткові шари до виходу, наприклад таким чином можна дізнатися орієнтацію машини і відстані до неї, що знадобляться до нашої задачі.

Щоб знайти відстань до об'єктів можна спробувати декілька варіантів. Наприклад знаючи середню висоту машини, можна знаходити відстань за висотою машин, а бо якщо ми знаємо розміщення камери можна пробувати знаходити відстань за розміщенням об'єкта у кадрі. Проте такі методи можуть видавати значні похибки, тому якщо є датасет в якому розмічені відстані до об'єктів можна просто додати їх до виходу CenterNet ще один канал з відстанями до об'єктів, що буде працювати аналогічно визначенню розміру об'єкта.

Задача відслідковування об'єктів полягає в тому щоб у відеопотоці зрозуміти що об'єкт, який присутній на декількох кадрах є одним і тим самим об'єктом, відслідкувати переміщення об'єкта у відео [4-6].

Найпростіший спосіб відслідкувати об'єкти це використовувати жадібний алгоритм, тобто найближчий об'єкт цього класу на попередньому кадрі і буде цим об'єктом. Або використовувати фільтри Калмана. Якщо частота кадрів велика то в принципі проблем не буде, проте існують і більш точні методи. Один з них це модифікація CenterNet CenterTrack. Головною його особливістю є те що замість одного зображення на вході ми беремо два: нинішній і попередній кадр а також теплову карту на якій розміщуємо центри об'єктів по аналогії з тим як це робилось у CenterNet. Тобто тепер вхід моделі буде не трьох каналним, а семи-каналним. На виході CenterTrack додатково додаємо два шари для визначення вектора зміщення відносно попереднього кадру.



Рисунок 2. Вхід і вихід CenterTrack

Потрібно зазначити що під час тренування на вхідній тепловій карті додатково будемо генерувати шуми, симулюючи помилки, бо якщо цього не робити, моделі буде складніше помічати нові об'єкти. Для співставлення об'єктів після врахування вектору зміщення використовуємо жадібний алгоритм.

Отже після того як ми розпізнали об'єкти знаємо на якій глибині в зображенні вони розташовані і можемо створити мапу об'єктів навколо машини. Далі можна застосувати рекурентну нейронну мережу що приймає координати об'єктів на кожному кадрі і намагається видати координати об'єкту на майбутньому кадрі.

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Розділили задачу на декілька етапів. Спочатку розпізнавання об'єктів, їх глибини та відслідковування їх переміщень по кадрах відеопотоку. Потім розміщення розпізнаних об'єктів на карті навколо нашої машини, у якій знаходиться камери, які і знімають відео. Розпізнавання об'єктів, відстаней до них можна робити паралельно. Це заслуга CenterNet, що дозволяє легко прикріплювати до голови будь які характеристики об'єктів, при цьому майже не збільшуючи нейронну мережу. Завдяки тому що на вході використовується не одне зображення, а також попереднє та результати роботи мережі на попередньому кадрі це також трішки дозволяє поліпшити розпізнавання об'єктів проте не набагато. Тому будемо розв'язувати однією і тією ж самою нейронною мережею декілька задач, що зекономить час роботи. До одного і того тіла нейронної мережі будемо додавати декілька голів. Функцією втрат в такому випадку буде сума функцій втрат з кожної голови помножених на коефіцієнти. Щодо розпізнавання об'єктів в основному будемо експериментувати з

центернетівською головою оскільки вона легше в реалізації, її легше змінювати ніж Yolo, чи SSD. При цьому будемо використовувати не оригінальний CenterNet, а його модифікацію TTF. Де замість того щоб видавати ширину, висоту і зміщення центра об'єкта, буде відстань лівого, правого, нижнього і верхнього країв об'єкту. На точність це не вплине, а якщо і вплине то в кращу сторону проте в межах похибки, але при цьому це інтуїтивно більш зрозуміла голова. Для відслідковування об'єктів або як це ще називають трекінгом об'єктів можна використовувати CenterTrack, що дозволить нам об'єднати задачі розпізнавання об'єктів, знаходження відстаней до них та трекінг в одну мережу з трьома головами. Розпізнавати будемо 3 класи об'єктів автомобілі, мотоцикли/велосипеди, люди. Отже голова з розпізнавання об'єктів буде складатися з  $3+4=7$  каналів, відстані 1 канал та трекінг 2 канали, тобто всього 10 каналів.



Рисунок 3. Візуалізація роботи розпізнавання об'єктів разом з відстеженням

Отримана точність з розпізнавання об'єктів. Евалюація проводилася на датасеті BDD100K [7]: AP-integral - 0.190; AP-0.5: 0.407.

#### 4. ВИСНОВКИ

Задачу з прогнозування траєкторії учасників дорожнього руху було розкладено на декілька підзадач: розпізнавання об'єктів, знаходження відстаней до них, відслідковування об'єктів у відеопотоці, нанесення об'єктів на карту навколо машини, прогнозування траєкторій об'єктів. Перші етапи задачі вдалося поєднати в один, тобто вирішувати однією нейронною мережею, що значно прискорює швидкість роботи. Також було враховано те що робота йде з відеопотоком, що дозволяє моделям використовувати більше інформації та спиратися на результати роботи на попередніх кадрах відеопотоку. Було натреновано нейронну мережу для вирішення кожної з підзадач та продемонстровано результати її роботи

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Objects as Points. URL: <https://arxiv.org/abs/1904.07850>
2. SSD: Single Shot MultiBox Detector. URL: <https://arxiv.org/abs/1512.02325>
3. Tracking Objects as Points. URL: <https://arxiv.org/pdf/2004.01177.pdf>
4. Orthographic Feature Transform for Monocular 3D Object Detection. URL: <https://arxiv.org/pdf/1811.08188.pdf>
5. Vehicle Trajectory Prediction Using Hierarchical Graph Neural Network for Considering Interaction among Multimodal Maneuvers. URL: <https://www.mdpi.com/1424-8220/21/16/5354>
6. Vehicle Trajectory Prediction with Lane Stream Attention-Based LSTMs and Road Geometry Linearization/ URL: <https://www.mdpi.com/1424-8220/21/23/8152>
7. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. URL: <https://arxiv.org/pdf/1805.04687.pdf>

# МОДЕЛЬ ДЛЯ РОЗПІЗНАВАННЯ ВІЙСЬКОВОЇ ТЕХНІКИ ТА ВІЙСЬКОВОСЛУЖБОВЦІВ ІЗ КАМЕРИ БПЛА В РЕАЛЬНОМУ ЧАСІ ДЛЯ СИСТЕМ З ОБМЕЖЕНИМИ ОБЧИСЛЮВАЛЬНИМИ РЕСУРСАМИ

Іванов С.І.<sup>1</sup>, Шубенкова І.А.<sup>2</sup>

Національний Технічний Університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> sergiy.ivanov0210@gmail.com,

<sup>2</sup> shubenkova1959@gmail.com [0000-0002-7433-2070]

Безпілотні літальні апарати широко використовуються в сучасній військовій сфері, одним із суттєвих факторів підвищення ефективності їх застосування є використання системи розпізнавання об'єктів із камери дрону для допомоги оператору. Метою роботи є збір та попередня обробка вхідних даних та розробка моделі для розпізнавання військової техніки та військовослужбовців із зображення з камери БПЛА в реальному часі. Результатом дослідження є навчена модель у вигляді згорткової нейронної мережі, адаптована під практичне використання із системами з обмеженими обчислювальними ресурсами. В роботі було використано емпіричні й теоретичні методи дослідження.

**Ключові слова:** комп'ютерний зір, машинне навчання, згорткові нейронні мережі, розпізнавання об'єктів, глибоке навчання, БПЛА

Системи розпізнавання образів та об'єктів із зображень широко використовуються у багатьох сферах сучасного життя, й одним із перспективних напрямів їх застосування є системи розпізнавання об'єктів з камер дронів, що набули стрімкої популярності як в цивільній, так і у військовій сфері [1, 2].

У порівнянні із цивільними застосуваннями, розробка й практичне використання систем розпізнавання образів у військовій сфері пов'язані із додатковими складнощами. Серед них обмежений доступ до даних, складні умови використання таких систем, що вимагають точності, надійності й відповідності нормам безпеки [3]. Із цього випливає необхідність у розробці моделей машинного навчання для розпізнавання об'єктів, що можуть бути застосовані на інтегрованих системах та робочих станціях операторів за умов низької обчислювальної потужності та адаптовані до реальних умов.

## 1. РОЗПІЗНАВАННЯ ОБ'ЄКТІВ У ВІЙСЬКОВІЙ СФЕРІ

Системи розпізнавання образів використовуються у військовій сфері досить широко, однак й досі зацікавленість в їх використанні не падає. Серед застосувань слід виділити розпізнавання стратегічних об'єктів на супутникових знімках, розпізнавання техніки та особового складу на фото та відео із соціальних мереж та інших відкритих джерел [4], застосування із системами камер наземної техніки та безпілотних літальних апаратів для маневрування, наведення, автономного пересування, виявлення цілей та допомоги оператору.

У військовій сфері БПЛА можуть виконувати різноманітні задачі, серед яких розвідка, патрулювання, знищення цілей та перевезення легкого вантажу. Оснащення БПЛА системами комп'ютерного зору може значно спростити роботу таких дронів, наприклад, пошвидшивши виявлення та класифікацію цілей, покращивши наведення на ціль, спростивши виявлення потенційних загроз.

Значним стримуючим фактором розвитку досліджень у сфері військового застосування систем розпізнавання об'єктів з БПЛА є наявність даних та інформації. Дуже часто моделі доводиться навчати на публічно доступних еталонних наборах даних, таких як ImageNet, COCO, публічних наборах даних для дронів для цивільних задач як VisDrone, MOR-UAV[1] та інші, після чого проводити донавчання на обмеженій кількості власних даних із використанням transfer learning [5]. Доступні у відкритому доступі набори даних часто є придатними лише до окремих способів використання та вкрай обмеженими, недієвими для навчання моделей у реальних умовах [6]. Ускладнюючими розпізнавання із камер БПЛА у військовій сфері факторами є низька якість вихідного зображення, часті спотворення форм, кутів огляду, розмиття, той факт, що під час виконання бойових задач дрони значну частину часу проводять у русі, та мають водночас слідкувати за рухомими цілями. Окрім цього, об'єкти часто можуть не повністю потрапляти в кадр, знаходитись на великій відстані що, враховуючи низьку роздільну здатність значної частини монтованих камер, значно ускладнює як виявлення так і класифікацію об'єктів на зображенні.

Додаткові складності для використання систем машинного навчання для задач розпізнавання створює те, що більшість робочих станцій операторів дронів мають досить обмежені обчислювальні ресурси. Переважна більшість використовуваних у військових цілях дронів в останні роки – це легкі дрони, які мають лише компактну камеру та іноді також певний вантаж або бойовий заряд [7]. Це виключає можливість проведення обчислювальних операцій використовуючи обладнання самого дрону. Так само неможливе й застосування хмарних обчислювальних систем [8] через необхідність постійного зв'язку із віддаленим обчислювальним пристроєм та проблеми щодо захисту даних та їх потенційної стратегічної значимості. Натомість, обчислення оптимально проводити безпосередньо на робочому пристрої оператора дрону, які мають відповідати вимогам мобільності, особливо в умовах використання безпосередньо в районах ведення бойових дій, та часто не мають достатніх обчислювальних ресурсів для використання на них більшості сучасних моделей глибокого навчання.

Таким чином існує необхідність в масштабованому рішенні, тобто створенні моделі, що була б здатна відповідати вимогам за точністю розпізнавання об'єктів, швидкості роботи для використання в режимі реального часу та могла би використовуватись на різних робочих станціях від командних пунктів до мобільних пристроїв для максимальної мобільності оператора.

## **2. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ**

Для дослідження було використано набір даних, зібраний студентами та викладачами Факультету прикладних наук Українського Католицького Університету [9]. Набір даних включає в себе більше 100 відеозаписів, зібраних із відкритих джерел за перші пів року повномасштабної війни Росії проти України. Відеозаписи являють собою реальні кадри із дронів різних класів та напрямів застосування, зібрані безпосередньо під час виконання бойових завдань. Об'єкти інтересу в кадрах із набору даних поділено на 3 класи: військові транспортні засоби, військовослужбовці та невизначені рухомі об'єкти.

Однією з головних переваг набору даних є те, що відеозаписи добре відображають типові фактори, що ускладнює розпізнавання об'єктів із камери дронів за реальних бойових



умов. Кадри часто мають низьку роздільну здатність, розмиті, зашумлені, мають певні викривлення форм об'єктів, включають кадри із складними природними умовами, як туман та сніг, ускладнюючі техногенні фактори як задимлення, горіння. Об'єкти інтересу також використовують природне й рукотворне маскування, що ускладнює розпізнавання та складне оточення, рух як об'єктів так і дрону, перекриття об'єктів іншими об'єктами інтересу або оточенням (Рис.1).



Рисунок 1. Кадри із набору даних, що демонструють деякі типові викривлення та ускладнюючі розпізнавання об'єктів фактори

Для розпізнавання об'єктів на основі вибраного набору навчальних даних було обрано модель згорткової нейронної мережі EfficientDet компактної модифікації Lite0.

Мережі EfficientDet відрізняються ефективним поєднанням високої точності розпізнавання, протестованої на таких еталонних наборах даних як COCO [10], та подекуди в рази, вищою швидкістю та компактністю моделей за аналогічні детектори [11]. Ще одна важлива особливість моделей це їх повна масштабованість, лінійка моделей включає як надточні та важкі, так і легкі, швидкі моделі із достатньою для практичного застосування точністю.

Архітектура мережі складається з 3 частин: «хребта» - попередньо навченої на наборі даних ImageNet мережі EfficientNet, що відповідає за виділення основних ознак на вихідному зображенні, двосторонньої пірамідальної мережі ознак (Bi-FPN), що дозволяє мережі краще обробляти й знаходити ознаки у об'єктах різних розмірів шляхом масштабування ознак в обох напрямках, та мережі прогнозування класів і меж об'єкта за допомогою softmax-класифікаторів та регресорів (Рис. 2) [11].

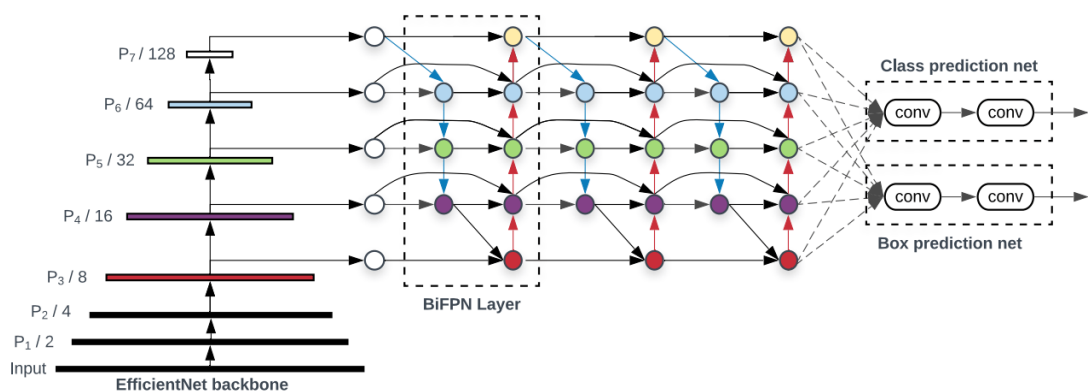


Рисунок 2. Структурні елементи архітектури мереж iEfficientDet

Модифікація мережі Lite0 відповідає найлегшому типу мережі, із найнижчим фактором масштабування, що має всього 3.9 мільйона параметрів й важить всього 4 мегабайти. Відповідно до фактору масштабування, «хребтом» мережі виступає EfficientNet-Lite0, мережа має 3 блоки BiFPN із 64 фільтрами та 3 Box/Classблоки відповідно. Головною перевагою цієї мережі є її швидкість та легкість, що дозволяє використовувати її на пристроях із низькими обчислювальними можливостями. На Google Pixel мережа показала середню швидкість обробки кадру в 36 мс, що відповідає 28 кадрам за секунду, тобто дозволяє використовувати модель в реальному часі навіть на смартфоні.

За результатами навчання побудованої моделі на наборі реальних даних із БПЛА із відкритих джерел було отримано хороші показники середньої точності як поміж класами, так і для ключового класу об'єктів Військовий транспортний засіб, що є домінантним у вихідному наборі даних (Табл. 1).

Таблиця 1. Показники точності мережі на тестовому наборі даних із 1168 зображень

Показник	mAP	AP50	AP транспортний засіб	AP військовослужбовець
Значення	22.4	43.6	41.8	10.6

Модель також показала дуже високу швидкість, витрачаючи на обробку кадру всього 23.125 мілісекунд, що відповідає роботі в реальному часі на 43 кадрах за секунду. Експерименти проводились на графічному процесорі nVidiaTesla із максимальним обсягом пам'яті графічного процесору у 32Гб, однак в процесі навчання для досягнення наведеної швидкості було достатньо лише незначної частини цього обсягу (Табл. 2).

Таблиця 2. Показники швидкості мережі на тестовому наборі даних із 1168 зображень

Показник	Затримка, мс	Кадри на секунду (FPS)	Використаний об'єм пам'яті графічного процесору, Гб
Значення	23.1	43.24	3.56

Кінцевим результатом роботи мережі є спрогнозовані околиці об'єктів інтересу на вихідному зображенні та класифікація цих об'єктів (Рис. 3).



Як ми можемо бачити, модель добре розпізнає та класифікує об'єкти навіть за вкрай складних реальних умов, включаючи перекриття об'єктів, погодні умови, викривлення й низьку якість зображення, маскування, великі відстані до об'єктів.

Розроблена модель може використовуватись на мобільних пристроях на системах Android та iOS, можливе використання на компактних мікрокомп'ютерах RaspberryPi, де достатня швидкість для розпізнавання в реальному часі досягається з версії 4 для 4 Гб вбудованої пам'яті. Модель також портується для використання на компактних процесорах EdgeTPU та може легко працювати на високій швидкості на більшості сучасних ноутбуків середньої потужності та багатьох планшетах. Можливість застосування на компактних процесорах та в інтегрованих системах також дозволяє застосовувати модель для проведення обчислень безпосередньо на обчислювальних ресурсах деяких дронів. Це значно спрощує її використання в реальних умовах на максимально мобільних робочих станціях.



Рисунок 3. Результати роботи мережі на кадрах із камер дронів

Розроблена модель може використовуватись для допомоги операторам дронів будь-якого класу, оснащених камерою, для виконання різноманітних бойових та небойових задач. Низькі вимоги до обчислювальних ресурсів, висока швидкість роботи та достатня точність розпізнавання роблять модель надійним інструментом допомоги оператору БПЛА, що готова до використання в реальних умовах.

### 3. ВИСНОВКИ

Розпізнавання об'єктів із камер дронів у військовій сфері має значні складності, пов'язані із доступністю даних, обчислювальними потужностями пристроїв, вимогами до безпеки, точності та швидкості роботи системи в реальному часі. Існує потреба у

використанні легких моделей, адаптованих під системи із обмеженими обчислювальними ресурсами, на мобільних станціях операторів.

В результаті дослідження було реалізовано модель у вигляді згорткової нейронної мережі для розпізнавання військової техніки, військовослужбовців та інших рухомих об'єктів на кадрах з камери дрону. Модель була навчена на реальних даних, зібраних із відкритих джерел, що враховують основні спотворення зображення та складності для розпізнавання що відповідають використанню системи в реальних умовах. Розроблена модель показала хороші точність та швидкість, здатність працювати в реальному часі, а також є легкою і може бути застосована на інтегрованих системах, мобільних пристроях, планшетах, ноутбуках та інших робочих станціях операторів БПЛА, котрі мають обмежені обчислювальні ресурси. Модель може використовуватись для допомоги операторам дронів у проведенні розвідки, патрулюванні території, а також для бойових задач, як система визначення та класифікації цілей.

Враховуючи масштабованість моделі, як напрям подальшого дослідження можлива розробка моделей із різними показниками точності та вимогами до обчислювальних ресурсів. Можливе розширення набору даних для покращення результатів та додаткової диверсифікації даних. Також можливе поєднання системи розпізнавання об'єктів із системами керування дронами на основі комп'ютерного зору для створення автономних систем, наприклад для задач патрулювання.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wu X., Li W., Hong D., Tao R., Du Q. Deep learning for unmanned aerial vehicle-based object detection and tracking: a survey. *IEEE Geoscience and Remote Sensing Magazine*. 2018. Vol. 10, No. 1. P.91–124.

2. Nex F. et. al. UAV in the advent of the twenties: Where we stand and what is next. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2022. Vol. 184. P. 215–242.

3. Svenmarck P., Luotsinen L., Nilsson M., Schubert J. Possibilities and challenges for artificial intelligence in military applications. In proceedings of the NATO Big Data and Artificial Intelligence for Military Decision Making Specialists' Meeting. Bordeaux, France, 30 May –1 June, 2018. P. 1–16.

4. Hiipala T. Recognizing military vehicles in social media images using deep learning. *ISI 2017: Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatic*, Beijing, China, July 22–24, 2017.

5. Yang Z., Yu W., Liang P., Guo H., Xia L., Zhang F., Ma J. Deep transfer learning for military object recognition under small training set condition. *Neural Computing and Applications*. 2019. Vol. 31, No. 10. P.6469–6478.

6. Schumacher R., Rosenbach Kh. ATR of Battlefield Targets by SAR – Classification Results Using the Public MSTAR Dataset Compared with a Dataset by QinetiQ, UK. *RTO SET: In proceedings of Symposium on “Target Identification and Recognition Using RF Systems”*. Oslo, Norway, October 11–13, 2004. P.31–42.

7. Pong B. The Art of Drone Warfare. *Journal of War & Culture Studies*. 2022. Vol. 15. P. 377–387.

8. Jangwon L. et. al. Real-time object detection for unmanned aerial vehicles based on cloud-based convolutional neural networks. *Journal Concurrency and Computation: Practice and Experience*. 2017. Vol.29.

9. Факультет прикладних наук УКУ. На передовій безпілотники відіграють надзвичайно важливу роль, тому дуже важливо вміти швидко і безперешкодно аналізувати отримані дані.

URL: <https://www.facebook.com/100063512295462/posts/pfbid02uEWNEKcAVT4 набаSgR1DJCfXnrKX1jcCMzQHnoivoNUPbD5iGv93ZTTYREFeRbBl/> (дата звернення: 02.11.2022).

10. LinTsun-Yiet. al. Microsoft COCO: Common objects in context. *ECCV 2014: Proceeding of the 13th European Conference on Computer Vision*. Zurich, Switzerland, September 6-12, 2014. P. 740–755.

11. Mingxing T., Ruoming P., Quoc V. Le. EfficientDet: Scalable and Efficient Object Detection. *IEEE/CVF: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, Washington, USA, June 13–19, 2020. P. 10781–10790.

# РЕКОМЕНДАЦІЙНА СИСТЕМА НА ОСНОВІ ОГЛЯДІВ КОРИСТУВАЧІВ І ПРОДУКТІВ

Кравченко О.В.<sup>1</sup>, Недашківська Н.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>kravchenko.olexandra@lil.kpi.ua,

<sup>2</sup>nedashkovskaya.nadezhda@lil.kpi.ua [0000-0002-8277-3095]

Рекомендаційні системи займають важливе місце в процесі життєдіяльності багатьох великих компаній. Рекомендаційні системи підвищують лояльність користувачів до різних інтернет-сервісів. Рекомендаційні системи допомагають користувачам знайти фільми, музику, інше за їх вподобаннями, рекомендують нові проекти. Одним з актуальних напрямків розвитку рекомендаційних систем є використання оглядів для рекомендацій. Метою роботи є аналіз і порівняння існуючих алгоритмів надання рекомендацій на основі оглядів, а також пропонування модифікації методу для покращення рекомендацій.

**Ключові слова:** рекомендаційна система, огляди для рекомендацій, DeepCoNN, eCommerce, нейромережа

## 1. ВСТУП

Рекомендаційні системи займають важливе місце в процесі життєдіяльності майже будь-яких інтернет-сервісів. Вони допомагають збільшити дохід бізнесу, покупцям – обрати найбільш підходящий для них продукт, а також зменшити навантаження на живих продавцях-консультантах. Ці системи пройшли шлях від набору заздалегідь сформованих рекомендацій для всіх покупців одразу до персоналізованих, що враховують не тільки загальні інтереси користувача, але і настрої і контекст сеансу.

Останніми роками деякі дослідники припустили, що текст огляду може допомогти більше, ніж характеристики товару і вподобання користувача. Таким чином з'явився відносно новий напрямок дослідження систем рекомендацій – засновані на оглядах.

## 2. МЕТОДИ НАДАННЯ РЕКОМЕНДАЦІЙ НА ОСНОВІ ОГЛЯДІВ

Рекомендаційні системи все ще знаходяться на етапі стрімкого розвитку, в наукових журналах постійно з'являються статті, які пропонують нові алгоритми надання рекомендацій і способи покращення вже існуючих.

На сьогоднішній день існує кілька популярних напрямків досліджень рекомендаційних систем для покращення точності наданих рекомендацій. Серед них можна виділити наступні:

1. Використання оглядів для рекомендацій. Загалом у цьому напрямку стверджується, що оскільки огляди «пояснюють» думки користувачів, вони повинні бути корисними для визначення базових параметрів, які передбачають оцінки чи покупки. Деякі дослідники [1] навпаки стверджують, що такі моделі показують себе мало не гіршими за еталонні моделі, якщо помістити їх в повністю однакові умови.

2. Передбачення рейтингу переходів. За допомогою рейтингу переходів (click-through rate, CTR) ми можемо в певній формі виміряти скільки кліків отримали рекомендації. Основне припущення полягає в тому, що більша кількість кліків на рекомендованих елементах означає, що рекомендації були більш релевантними для користувачів.

3. Контекстно-залежна система рекомендацій (CARS) застосовує визначення й аналіз контексту користувача для надання персоналізованих послуг. Контекстну інформацію можна отримати від датчиків, щоб підвищити точність рекомендацій.

4. Використання автокодерів для рекомендацій. В той час як традиційні моделі працюють лише з одним джерелом даних (оцінка чи текст), моделі на основі автоматичного кодувальника можуть опрацьовувати неоднорідні джерела даних (рейтинг, аудіо, відео).

5. Послідовні рекомендації. Системи послідовних рекомендацій намагаються зрозуміти поведінку користувача з часом і моделюють таким чином послідовності дій користувача.

В статті розглянемо лише перший напрямок дослідження: використання оглядів для рекомендацій. Серед них можна виділити такі моделі.

DeepCoNN [2] – це глибока модель спільного вивчення властивостей предмета та поведінки користувача на основі тексту огляду. Модель складається з двох паралельних нейронних мереж, з'єднаних на останніх рівнях. Одна з мереж зосереджена на вивченні поведінки користувачів, використовуючи відгуки, написані користувачем, а інша вивчає властивості предмета з відгуків, написаних для товару. Спільний рівень об'єднує ці дві мережі. Спільний рівень дозволяє латентним факторам, отриманим користувачами та елементами, взаємодіяти один з одним у спосіб, подібний до методів машинної факторизації.

NARRE [3] аналізує важливість кожного огляду перед використанням. Автори стверджують, що менш корисні відгуки шкодять продуктивності моделі, а також менш значущі для користувача. Для дослідження важливості оглядів модель використовує механізм уваги.

HFT [4] – особлива модель у цьому списку, використовує огляди як регуляризатор. По суті, така модель не просто використовує текст огляду для визначення прихованих вподобань, але й визначає, наскільки об'єктивною відносно тексту огляду є відповідна оцінка.

TransNets [5] – розширення моделі DeepCoNN, що вводить додатковий латентний рівень, що представляє пару користувач-елемент. Під час навчання цей рівень впорядковується, щоб бути схожим на інше приховане представлення огляду цільового користувача цільового елемента.

MPCN [6] – модель з багатоієрархічною парадигмою, що ґрунтується на припущенні, що не всі огляди однакові, тобто важливі лише кілька обраних. Важливість, однак, має динамічно виводитися залежно від поточної цілі. З цією метою пропонується схема навчання на основі вказівника рецензії за рецензією, яка виділяє важливі рецензії, а потім зіставляє їх слово за словом. Це дає змогу не лише використовувати найбільш інформативні огляди для прогнозування, але й глибше взаємодіяти на рівні слів.

NRPA [7] припускає, що важливість відгуків різна для різних користувачів і елементів, а також одне і те ж слово або схожі відгуки можуть мати різну інформативність для різних користувачів і товарів. Тому модель використовує персоналізовану увагу, щоб врахувати це.

DAML [8] – модель, що використовує локальну та взаємну увагу згорткової нейронної мережі, щоб спільно вивчати особливості оглядів для покращення інтерпретації. Потім рейтингові функції та функції перегляду інтегруються в уніфіковану модель нейронної мережі, а нелінійна взаємодія ознак вищого порядку реалізується машинами нейронної факторизації для завершення прогнозу остаточного рейтингу.

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для дослідження було обрано чотири датасети з набору Amazon product data: Apps for Android, Video Games, Cell Phones and Accessories, Office Products (розташовані в порядку спадання розміру датасету). Для спрощення процесу навчання був обраний 5-core варіант даних. Для дослідження набори даних були попередньо оброблені на розділені по схемі 80-10-10 на сети для навчання, тестування і перевірки.

Для подальшого дослідження обрано дві моделі, описані вище – DeepCoNN і NARRE. Також розглянуто NeuMF – проста і розповсюджена модель нейронної колаборативної фільтрації, що використовує машину факторизації для заповнення пропущених зв'язків. Алгоритм NeuMF не використовує текст огляду і обраний для об'єктивності дослідження.

Також для дослідження запропоновано модифікацію моделі DeepCoNN – DeepAtnCoNN. Припущення полягає в тому, що за умови використання Textual Preprocessing (64-вимірні word2vec вбудовування, навчені за допомогою Gensim), а також додавання додаткового Attention шару результат може бути набагато кращим за початковий. Припущення базується на кількох альтернативних статтях в напрямку рекомендацій за допомогою оглядів [7, 8]. Також замість машини факторизації на останньому етапі спробуємо дослідити різного роду зміщення: глобальні, користувачів і елементів, і додати їх до виходу нейронної мережі.

Показниками точності обрано MSE, nDCG@5 і HR@1 [1]. Перші дві метрики використовувалися і під час навчання, і під час тестування моделей, а остання – тільки для тесту моделей після закінчення навчання для загальної оцінки їх ефективності.

Проведено два дослідження: значень втрат протягом 10 епох навчання, і дослідження точності залежно від значення дропауту (dropout). Значення MSE для відомих моделей DeepCoNN, NARRE і NeuMF та запропонованої DeepAtnCoNN на епохах 1, 2, ..., 10 наведені на Рис.1 і 2.

Можна побачити, що запропонована модель DeepAtnCoNN працює краще за інші розглянуті моделі за умови великого набору даних. Крім того, бачимо, що моделі починають перенавчатися в середньому після п'ятої епохи, тому у подальших дослідженнях будемо переривати навчання на цьому етапі. DeepCoNN має повільно спадаючу тенденцію по метриці MSE протягом всього навчання. Запропонована модель має кращу точність ніж розглянуті три інші на наборах даних Android Apps, Cells Phones і Video Games, а на останньому наборі Office Products дає кращі результати ніж DeepCoNN і NARRE, однак ці результати майже не відрізняються від NeuMF, яка не використовує огляди для рекомендацій.

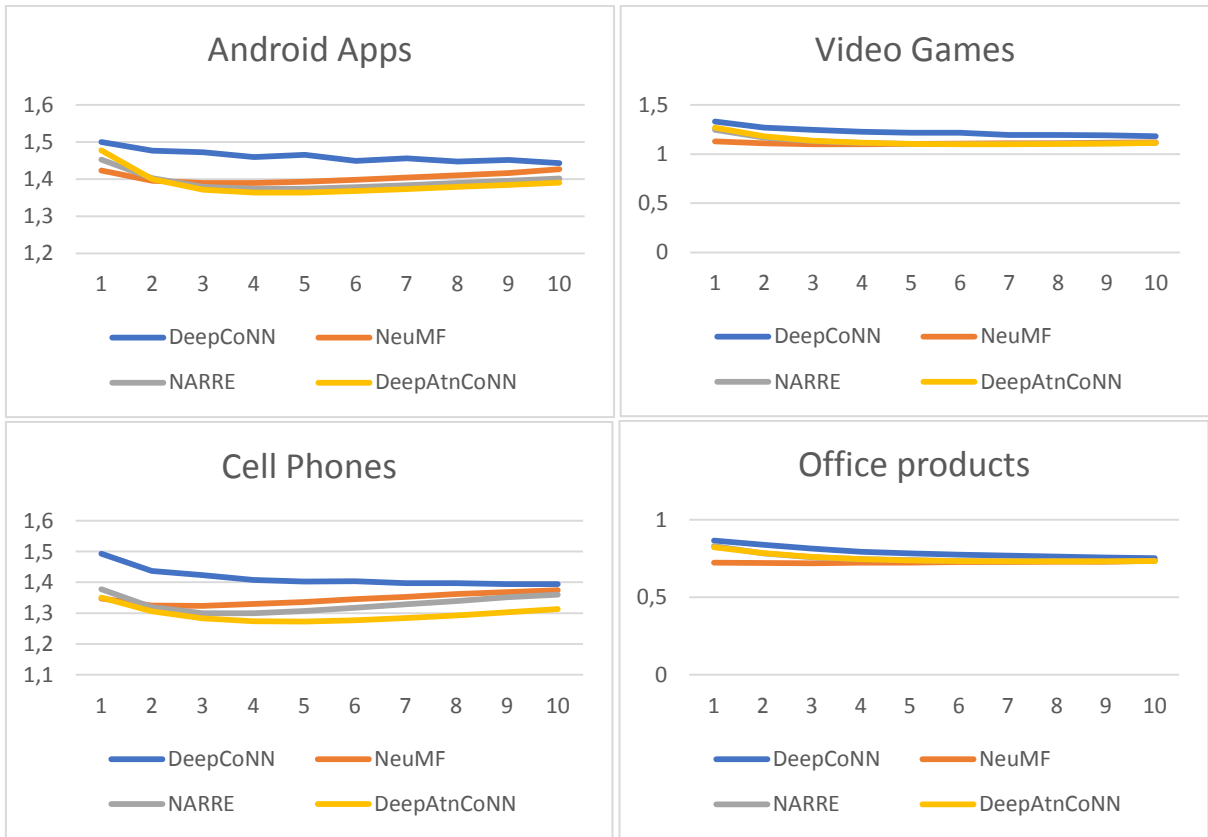


Рисунок 1. Залежність MSE від кількості епох

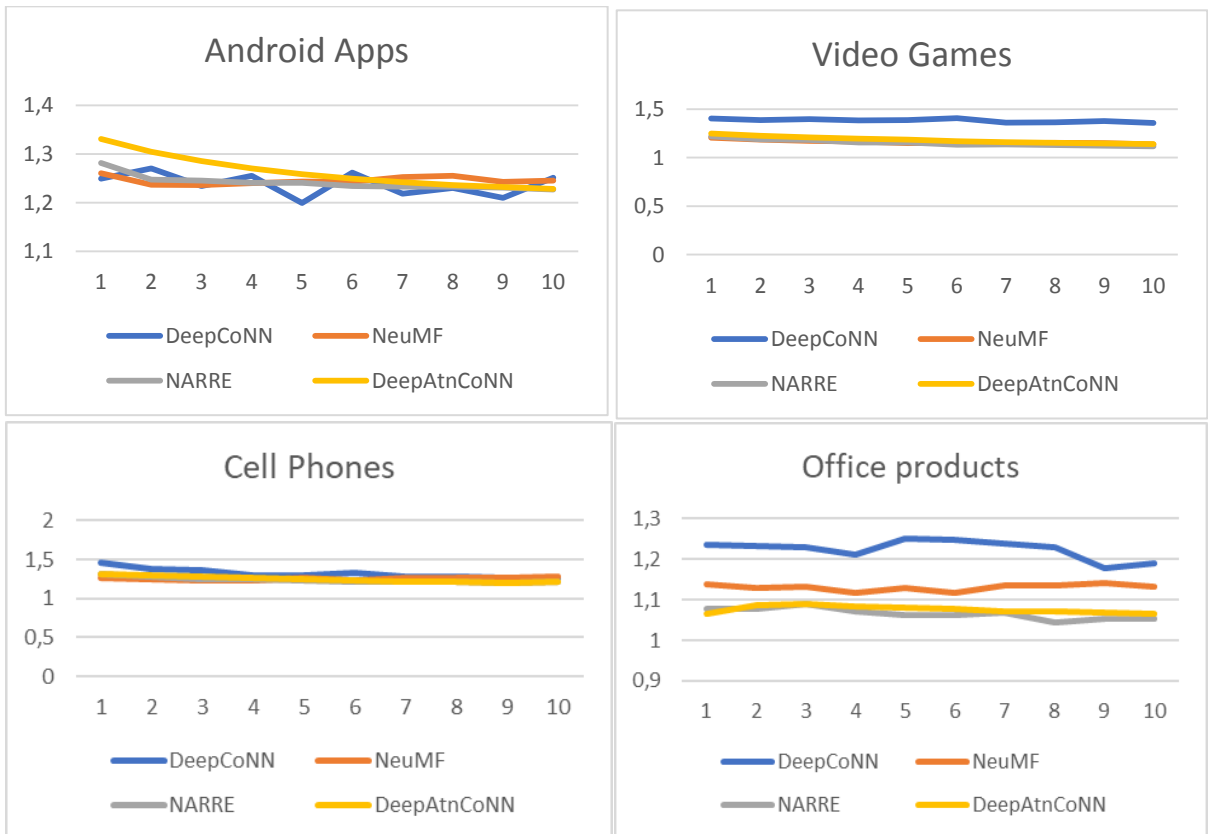


Рисунок 2. Залежність NDCG від кількості епох

Таким чином можна зробити додатковий проміжний висновок, що для маленьких розріджених наборів даних немає необхідності використовувати огляди.

В Таблиці 1 наведено фінальні результати тестування найкращих моделей на чотирьох наборах даних.

#### 4. ВИСНОВКИ

В сучасному світі постійно з'являються нові шляхи покращення механізму рекомендацій. Одним з них є рекомендації на основі оглядів. Припускається, що огляди можуть містити додаткову інформацію про приховані вподобання користувача та специфічні характеристики продукту.

У ході дослідження проаналізовано і порівняно дві класичні моделі рекомендацій на основі оглядів: DeepCoNN і NARRE, а також ту, що не використовує текст огляду – NeuMF. Також запропоновано модифікацію методу DeepCoNN – DeepAtnCoNN з механізмом уваги. Обрано підходящий набір даних Amazon Products Data і проведено його попередню обробку. Проведено дослідження кількох класичних алгоритмів обраного напрямку і порівняно їх точність за показниками MSE і nDCG@5. На базі цього запропоновано модифікацію обраного методу і проведено його дослідження і порівняння з початковими.

Таблиця 1. Значення показників точності MSE, nDCG@5 і HR@1 при фінальному тестуванні моделей DeepCoNN, NeuMF, NARRE та DeepAtnCoNN

	DeepCoNN				NeuMF			
	Apps	Game	Phone	Office	Apps	Game	Phone	Office
MSE	1.4475	1.1856	1.3621	0.7313	1.3789	1.1019	1.3364	0.7129
nDCG@5	1.2393	1.3913	1.2697	1.2025	1.2218	1.1584	1.2171	1.1294
HR@1	31.340	39.090	36.360	44.440	33.330	44.170	70.00	27.270

Продовження таблиці 1

	NARRE				DeepAtnCoNN			
	Apps	Game	Phone	Office	Apps	Game	Phone	Office
MSE	1.3599	1.1013	1.3134	0.7398	1.3708	1.1026	1.3166	0.7042
nDCG@5	1.2323	1.1847	1.1987	1.1146	1.2574	1.2053	1.2474	1.1752
HR@1	37.620	31.970	27.270	50.00	37.310	40.910	45.450	33.330

Пропонована модифікація виявилася точнішою на 5-7% в основному для всіх обраних наборів даних порівняно з іншими розглянутими алгоритмами.

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Noveen Sachdeva and Julian McAuley. How Useful are Reviews for Recommendation? A Critical Review and Potential Improvements. SIGIR '20: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval July 2020 Pages 1845–1848 <https://doi.org/10.1145/3397271.3401281>

2. Lei Zheng, Vahid Noroozi, Philip S. Yu Joint Deep Modeling of Users and Items Using Reviews for Recommendation WSDM '17: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining February 2017 Pages 425–434 <https://doi.org/10.1145/3018661.3018665>



3. Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-level Explanations. In *The Web Conference 2018*, April 23–27, 2018, Lyons, France. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3178876.3186070>
4. Julian McAuley, Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. *RecSys '13: Proceedings of the 7th ACM conference on Recommender systems* October 2013 Pages 165–172 <https://doi.org/10.1145/2507157.2507163>
5. Rose Catherine, William Cohen. TransNets: Learning to Transform for Recommendation. *RecSys '17: Proceedings of the Eleventh ACM Conference on Recommender Systems* August 2017 Pages 288–296 <https://doi.org/10.1145/3109859.3109878>
6. Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Multi-Pointer Co-Attention Networks for Recommendation. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, August 19–23, 2018, London, United Kingdom. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3219819.3220086>
7. Hongtao Liu, Fangzhao Wu, Wenjun Wang, Xianchen Wang, Pengfei Jiao, Chuhan Wu, and Xing Xie. 2019. NRPA: Neural Recommendation with Personalized Attention. In *42nd Int'l ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR'19)*, July 21–25, 2019, Paris, France. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3331184.3331371>
8. Donghua Liu, Jing Li, Bo Du, Jun Chang, Rong Gao. DAML: Dual Attention Mutual Learning between Ratings and Reviews for Item Recommendation. *KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* July 2019 Pages 344–352 <https://doi.org/10.1145/3292500.3330906>

# ДОСЛІДЖЕННЯ ГІБРИДНИХ МЕРЕЖ ГЛИБОКОГО НАВЧАННЯ В ЗАДАЧАХ ПРОГНОЗУВАННЯ У ФІНАНСОВІЙ СФЕРІ

Кузьменко О.В.<sup>1</sup>, Зайченко Ю.П.<sup>2</sup>

КПІ ім. Ігоря Сікорського, Київ, Україна

<sup>1</sup>oleksii.kuzmenko@ukr.net [0000-0003-1581-6224],

<sup>2</sup>zaychenko Yuri@ukr.net [0000-0001-9662-3269]

**У роботі розглядаються проблеми короткострокового та середньострокового прогнозування у фінансовій сфері. Для їх розв'язання запропоновано інтелектуальні методи прогнозування: LSTM та гібридні мережі глибокого навчання на основі МГУА. Знайдено оптимальні параметри LSTM та гібридних мереж. Побудовано оптимальні структури гібридних мереж. Проведено експериментальні дослідження LSTM та гібридних мереж, а також порівняно їх точність.**

**Ключові слова: короткострокове прогнозування, середньострокове прогнозування, LSTM, гібридна мережа глибокого навчання, оптимізація.**

## 1. ВСТУП

Проблемам прогнозування курсів акцій і ринкових індексів на фондових біржах приділяється велика увага інвесторів і різних грошових фондів. Для їх вирішення були розроблені та досліджені потужні інтелектуальні методи та технології, серед яких нейронні мережі та системи нечіткої логіки.

Ефективним інструментом моделювання для короткострокового та середньострокового прогнозування нестационарних часових рядів є мережі LSTM. Вони давно розроблені та успішно застосовуються для прогнозування на біржах [1-5].

В якості альтернативного підходу для прогнозування у фінансовій сфері застосовують різні типи нейронних мереж: MLP [6], нечітких нейронних мереж [7,8], нео-нечітких мереж [9] та мереж глибокого навчання (DL) [10]. Новим трендом в області мереж DL є новий клас мереж – гібридні мережі DL на основі методу МГУА [11]. Застосування самоорганізації в цих мережах дозволяє навчати не тільки ваги нейронів, але й будувати оптимальну структуру мережі. Завдяки методу навчання в цих мережах ваги коригуються не одночасно, а шар за шаром. Це запобігає феномену вибуху градієнта, що дуже важливо для мереж з багатьма шарами.

У перших роботах в цій області у якості вузлів гібридної мережі використовувалися нейрони Ванга-Менделя з двома входами [11]. Але недоліком таких нейронів є вимога навчати не лише нейронні ваги, але й параметри нечітких функцій приналежності (ФП) в антецедентах правил. Це потребує значних розрахункових витрат і тривалого часу на навчання. Тому пізніше були розроблені нео-фазі-мережі DL, в яких у якості вузлів використовуються нео-фазі-нейрони Ямакави [12,13]. Основна властивість таких нейронів полягає в тому, що необхідно навчати лише ваги нейронів, а не нечіткі ФП. Це вимагає менше обчислень порівняно з нейронами Ванга-Менделя і значно скорочує час навчання в цілому. У роботі [13] було проведено дослідження обох класів гібридних мереж DL та порівняно їх ефективність при прогнозуванні у фінансовій сфері.

Тому виникає проблема порівняння ефективності гібридних мереж DL та LSTM в задачі прогнозування у фінансовій сфері [14].

Метою даної роботи є дослідження точності гібридних мереж DL і LSTM у задачі прогнозування ринкових індексів на біржі, порівняння їх ефективності на різних інтервалах та визначення класів задач прогнозування, для яких застосування відповідних обчислювальних інтелектуальних технологій є найбільш перспективними.

## 2. ОПИС ГІБРИДНОЇ МЕРЕЖІ МГУА-НЕО-ФАЗИ

Архітектура гібридної еволюційної DL-мережі представлена на рис. 1. На вхідний рівень системи подається  $(n \times 1)$ -вимірний вектор вхідних сигналів. Після цього сигнал передається на перший прихований шар. Цей шар містить нейрони, кожен з яких має лише два входи.

На виходах першого прихованого шару формуються вихідні сигнали. Потім ці сигнали подаються на блок селекції першого прихованого шару.

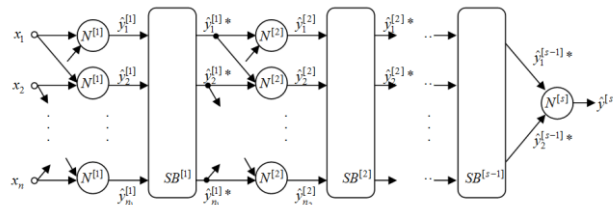


Рисунок 1. Еволюційна гібридна мережа МГУА-нео-фазі

Серед вихідних сигналів обираються найточніші  $y_l^1 \cdot n_1^*$  ( $n_1^* = F$  – так звана свобода вибору) за певним критерієм (здебільшого за середнім квадратом помилки  $\sigma_{y_l^1}^2$ ). Серед цих  $n_1^*$  найкращих результатів першого прихованого шару утворюються попарні комбінації  $y_l^{1*}, y_p^{1*}$ . Вони подаються на другий прихований шар, який утворений нейронами  $N^{[2]}$ . Після навчання цих нейронів вихідні сигнали рівня  $y_l^2$  передаються до блоку селекції  $SB^{[2]}$ , який вибирає  $F$  найкращих нейронів за точністю (наприклад, за значенням  $\sigma_{y_l^2}^2$ ), якщо найкращий сигнал другого шару кращий за найкращий сигнал першого прихованого шару  $y_l^{1*}$ . Інші приховані шари формують сигнали аналогічно. Процес еволюції системи триває до тих пір, поки найкращий сигнал блоку селекції  $SB^{s+1}$  не виявиться гіршим, ніж найкращий сигнал попереднього рівня. Потім ми повертаємося до попереднього рівня та обираємо його найкращий вузловий нейрон  $N^{[s]}$  з вихідним сигналом  $y^{[s]}$ . І рухаючись від цього нейрона (вузла) по його зв'язках назад, послідовно проходячи всі попередні шари, ми врешті отримуємо структуру МГУА-нео-фазі мережі.

Слід зазначити, що таким чином можна побудувати не тільки оптимальну структуру мережі, але й добре навчену мережу завдяки алгоритму МГУА. Крім того, оскільки навчання виконується послідовно шар за шаром, то вдається уникнути проблем високої розмірності та вибухового градієнта.

## 3. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

### 3.1. Набір даних

В якості вихідних даних було взято корпоративний індекс прибутковості Emerging Markets Bond total Return Index (EMBRI) фондової біржі NASDAQ за період з січня по

серпень 2022 року. Вибірка складалася з примірників, які були розділені на навчальну та тестову підвибірки.

Графік EMBRI представлений на рис. 2.



Рисунок 2. Emerging Markets Bond total Return Index (EMBRI)

### 3.2. Експериментальні дослідження гібридних мереж DL

Першу серію експериментів проводили з гібридною мережею глибокого навчання з нео-фазі нейронами у якості вузлів. Під час експериментів змінювалися наступні параметри: співвідношення навчальна/тестова вибірка, кількість входів 3-5, кількість нечітких множин на змінну 3-5 та функції належності (MF): Bell, Gaussian та Triangular. Метою експериментів було знайти оптимальні значення параметрів.

Прогнозний період був прийнятий 5 днів, а показники точності – MSE та MAPE.

У першому експерименті було досліджено Bell MF. Після експерименту були знайдені оптимальні параметри гібридної мережі DL: кількість входів – 3, кількість нечітких множин – 3, співвідношення навчання/тест – 0,8. При цих значеннях параметрів була досягнута найкраща точність на дослідному зразку: MSE = 0,424 та MAPE = 0,155.

Наступні експерименти були проведені для гібридної мережі з Gaussian MF. Після проведення експериментів знайдено оптимальні параметри гібридної мережі DL: кількість входів – 3, кількість нечітких наборів – 4, співвідношення навчання/тест – 0,6. З цими параметрами результати прогнозування представлені в табл. 1.

Таблиця 1. Прогноз із Gaussian MF (кількість входів – 3, кількість нечітких множин – 4, співвідношення навчання/тест – 0,6)

Дата	Реальні значення	Прогноз	MSE	MAPE
17.08.2022	399,15	398,44	0,504	0,178
18.08.2022	399,12	398,17	0,902	0,238
19.08.2022	397,97	398,50	0,281	0,133
22.08.2022	396,82	397,40	0,336	0,146
23.08.2022	396,62	396,93	0,096	0,078
		Мінімум:	0,096	0,078
		Середнє:	0,424	0,155
		Максимум:	0,902	0,238

Наступний експеримент проводився з гібридною мережею DL з Triangular MF. Після проведення експерименту були знайдені оптимальні параметри та структура гібридної мережі: 5 входів, 3 MF та співвідношення 0,8. Після експериментів порівняли точність гібридних мереж з різними MF. Результати представлені на рис. 3.

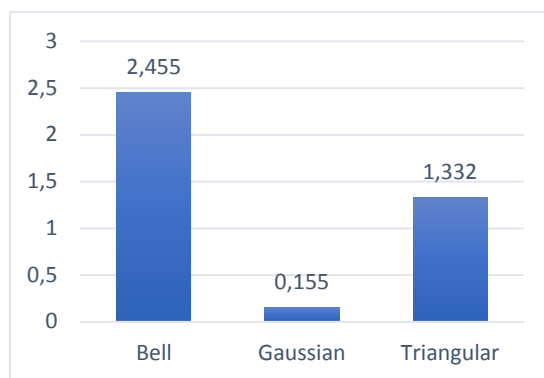


Рисунок 3. Порівняння найкращих значень MAPE (середнє) для різних MF

У наступній серії експериментів були досліджені мережі LSTM. Метою експериментів було знайти оптимальні параметри. Змінювалися такі параметри: кількість входів (3-5), співвідношення навчальна/тестова вибірка (0,6; 0,7; 0,8). Знайдено оптимальні значення параметрів: 3 входи, співвідношення 0,7.

У наступних експериментах було досліджено ефективність найкращих моделей гібридної мережі DL (МГУА-нф) і LSTM та порівняно з різними співвідношеннями навчальна/тестова вибірка. Відповідні результати представлені в табл. 2.

Таблиця 2. MSE для різного співвідношення навчання/тест та 3 входів

Вибірка даних	3 входи		4 входи		5 входів	
	МГУА-нф	LSTM	МГУА-нф	LSTM	МГУА-нф	LSTM
60/40	0,424	4,074	2,894	4,715	5,556	1,961
70/30	1,242	1,378	1,783	2,107	7,414	4,628
80/20	1,248	2,023	3,491	5,412	5,902	2,152

Аналізуючи ці результати, можна зробити висновок, що МГУА-нео-фазі мережа має кращу точність прогнозування на інтервалі 5, ніж LSTM для різних співвідношень.

У наступних експериментах було досліджено ефективність прогнозування гібридних мереж DL та LSTM на різних інтервалах прогнозування.

У табл. 3 представлені результати прогнозування МГУА-нео-фазі мережі та LSTM на інтервалі 7 днів.

Таблиця 3. Кращий прогноз на інтервалі 7 днів – MAPE

Дата	Реальні значення	МГУА-нео-фазі	LSTM
15.08.2022	400,32	0,412	1,806
16.08.2022	400,70	0,317	1,899
17.08.2022	399,15	0,704	1,518
18.08.2022	399,12	0,807	1,511
19.08.2022	397,97	0,859	1,224
22.08.2022	396,82	0,935	0,932
23.08.2022	396,62	0,630	0,877

На рис. 4 представлено порівняння точності прогнозів МГУА-нео-фазі мережі та LSTM на інтервалі 7 днів.

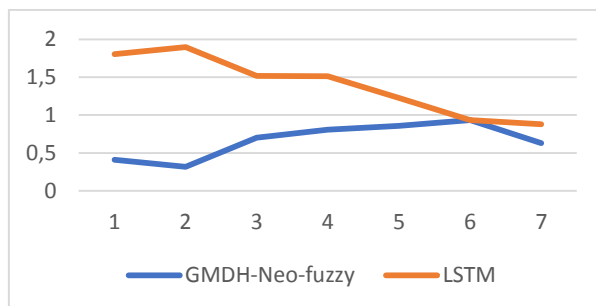


Рисунок 4. Порівняльний графік точності прогнозів на інтервалі 7 днів – MAPE

У наступних експериментах точність прогнозування обох мереж досліджувалася при середньостроковому прогнозуванні з інтервалом 20 днів. Точність за MAPE представлена на рис. 5.

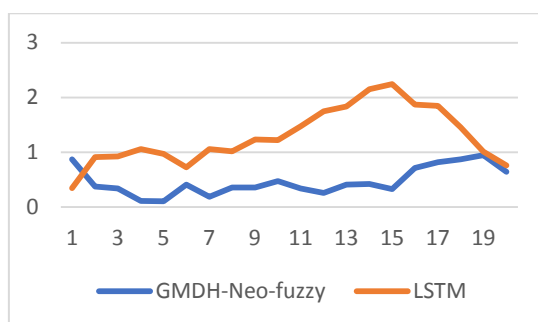


Рисунок 5. Найкращий прогноз з інтервалом 20 днів – MAPE

### 3.3. Порівняння результатів експериментів гібридних мереж DL та LSTM

У підсумкових експериментах порівнювали точність прогнозування гібридної мережі МГУА-нео-фазі та LSTM на різних інтервалах прогнозування (короткострокових і середньострокових). Відповідні результати за MSE та MAPE представлені в табл. 4.

Таблиця 4. Значення MSE та MAPE (середні) для різних інтервалів

Інтервал	Мережа	5 днів		7 днів		20 днів	
		MSE	MAPE	MSE	MAPE	MSE	MAPE
3	МГУА-нео-фазі	0,319	0,094	0,678	0,192	1,251	0,262
	LSTM	0,702	0,201	0,408	0,127	2,374	0,349
5	МГУА-нео-фазі	1,178	0,111	2,819	0,337	0,953	0,224
	LSTM	2,674	0,316	3,054	0,358	2,374	0,352
7	МГУА-нео-фазі	0,238	0,231	2,714	0,475	0,863	0,203
	LSTM	5,888	0,489	6,325	0,510	2,983	0,364
20	МГУА-нео-фазі	2,241	0,392	1,238	0,368	0,595	0,172
	LSTM	6,182	0,453	4,178	0,408	3,134	0,416

Аналіз цих результатів показує, що в цілому гібридна мережа DL має кращу точність, ніж LSTM на різних коротких та середніх інтервалах прогнозування.

## 4. ВИСНОВОК

У роботі розглянуто задачу прогнозування на фінансовому ринку з різними інтервалами прогнозування (короткострокове та середньострокове прогнозування). Для її вирішення було запропоновано застосувати гібридні мережі глибокого навчання (DL) на основі МГУА та LSTM.

Експериментальні дослідження проводились над задачею прогнозування Emerging Markets Bond Total Return Index (EMBRI) на фондовій біржі NASDAQ за період з січня по серпень 2022 року.

Під час експериментів проведено оптимізацію параметрів LSTM та гібридних мереж. Методом МГУА побудовано оптимальну структуру гібридної мережі DL.

Експериментальні дослідження оптимізованих LSTM та гібридних мереж проводились на різних інтервалах прогнозування та порівнювалася їх точність.

У результаті було встановлено, що застосування гібридних мереж DL має набагато кращу точність, ніж LSTM, у задачах короткострокового та середньострокового прогнозування на фондових біржах.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. S. Hochreiter and J. Schmidhuber Long short-term memory, *Neural Computation*, vol. 9, 1997, pp. 1735–1780. doi: 10.1162/neco.1997.9.8.1735.
2. B. Hammer, On the approximation capability of recurrent neural networks, *Neurocomputing*, vol. 31, 1998, pp. 107–123. doi: 10.1016/S0925-2312(99)00174-5.
3. C. Olah Understanding LSTM networks, 2020. Available from: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
4. A. Graves Generating sequences with recurrent neural networks, *CoRR*, vol. abs/1308.0850, 2013. doi: 10.48550/arXiv.1308.0850.
5. A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, Verlag, Berlin, Heidelberg: Springer, 2012. doi: 10.1007/978-3-642-24797-2.
6. S.S. Haykin, *Neural networks: a comprehensive foundation*, 2nd ed. Upper Saddle River, N.J: Prentice Hall, 1999.
7. Ossovsky S. *Neural networks for information processing*, transl. from Polish.–M. – Finance and Statistics, 2002 – 344p.
8. Wang F. *Neural Networks Genetic Algorithms and Fuzzy Logic for Forecasting* // Proc. Intern. Conf. Advanced Trading Technologies. – New York, 1992, pp. 504–532.
9. Yamakawa T., Uchino E., Miki T., Kusanagi H. A neo-fuzzy neuron and its applications to system identification and prediction of the system behavior // Proc. 2nd Intern. Conf. Fuzzy Logic and Neural Networks LIZUKA-92. – Lizuka, 1992, pp. 477–483.
10. I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT PRESS, 2016. Available from: <http://www.deeplearningbook.org>.
11. Yuriy Zaychenko, Yevgeniy Bodyanskiy, Oleksii Tyshchenko, Olena Boiko, Galib Hamidov Hybrid GMDH-neuro-fuzzy system and its training scheme. *Int. Journal Information theories and Applications*, 2018. vol.24, Number 2, pp. 156–172.
12. Yu. Zaychenko, Galib Hamidov The Hybrid Deep Learning GMDH-neo-fuzzy Neural Network and Its Applications. *Proceedings of 13-th IEEE Intern Conference Application of Information and Communication Technologies-AICT2019*. 23–25 October 2019, Baku, pp. 72–77.
13. Evgeniy Bodyanskiy, Yuriy Zaychenko, Olena Boiko, Galib Hamidov, Anna Zelikman Structure Optimization and Investigations of Hybrid GMDH-Neo-fuzzy Neural Networks in Forecasting Problems. *System Analysis & Intelligent Computing*. Ed. Michael Zgurovsky, Natalia Pankratova. *Book Studies in Computational Intelligence, SCI*, vol.1022. Springer, 2022, pp. 209–228.
14. Yuriy Zaychenko and Oleksii Kuzmenko Investigation of hybrid deep learning networks and LSTM in the short term forecasting in financial sphere, *IEEE 17th Int. Conference CSIT*, 11 November 2022, 4 p.

# РОЗПІЗНАВАННЯ ПОЖЕЖІ ЗА ДОПОМОГОЮ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

Маркін І.Д.<sup>1</sup>, Шаповал Н.В.<sup>2</sup>

КПІ ім. Ігоря Сікорського, Київ, Україна

<sup>1</sup>ivanmarkin2019@gmail.com, <sup>2</sup>shovgun@gmail.com

**У роботі розглянуто проблему виявлення пожежі за допомогою сучасних технологій та алгоритмів машинного зору. Для розв'язання поставленої задачі запропоновано використати згорткову нейронну мережу. Побудована згорткова нейронна мережа, проведено експерименти з побудованою нейронною мережею на вибірці зображень.**

**Ключові слова:** згорткова нейронна мережа, CNN, виявлення пожежі, класифікація.

## 1. ВСТУП

Виявлення пожежі на ранній стадії до її переростання в катастрофічну подію має вирішальне значення для нейтралізації пожеж та порятунку життя та майна людей. Через швидке поширення пожежі важливо виявити його на стадії перших проявів вогню.

Незважаючи на важливість цієї проблеми, вона залишається невирішеною на даний момент. Датчики, основним принципом яких є виявлення диму, добре працюють, коли пожежа триває протягом певного часу і утворилася достатня кількість диму для спрацьовування сигналізації [1]. Ці пристрої не можуть бути використані у великих масштабах на відкритому повітрі, наприклад, у лісі. Супутникові системи, стаціонарні віддалені камери та пілотовані літаки - це звичайні технології дистанційного моніторингу пожеж, проте ці технології мають низку обмежень з точки зору вартості, тимчасового та просторового дозволу.

Також для вирішення поставленої задачі пропонується використання згорткових нейронних мереж. На даний момент, технології нейронних мереж досягли високого рівня розвитку і вже дозволяють знаходити рішення специфічним завданням, працюючи за принципом, схожим на процес прийняття рішення людиною. Використання таких методів на даному етапі розвитку технологій не може зробити прийняття рішень повністю автоматизованим, проте воно суттєво знижує вплив людського фактора та забезпечує високу результативність за менших витрат часу та зусиль.

Метою даної роботи є розробка програмного забезпечення для виявлення лісової пожежі за допомогою нейронних мереж.

## 2. ОПИС ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

Відповідно до принципу алгоритмів виявлення об'єктів, алгоритм виявлення пожежі на основі згорткових нейронних мереж зображено на (Рис. 1) [2]. Згорткова нейронна мережа має функції пропозицій регіону, виділення ознак і класифікації. По-перше, ЗНМ приймає зображення як вхідні дані та виводить пропозиції регіонів шляхом згортання, об'єднання тощо. По-друге, система виявлення об'єктів на основі регіону визначає наявність чи відсутність вогню в регіонах пропозицій за допомогою згорткових шарів, шарів об'єднання, повноз'єднаних шарів. і т.д.



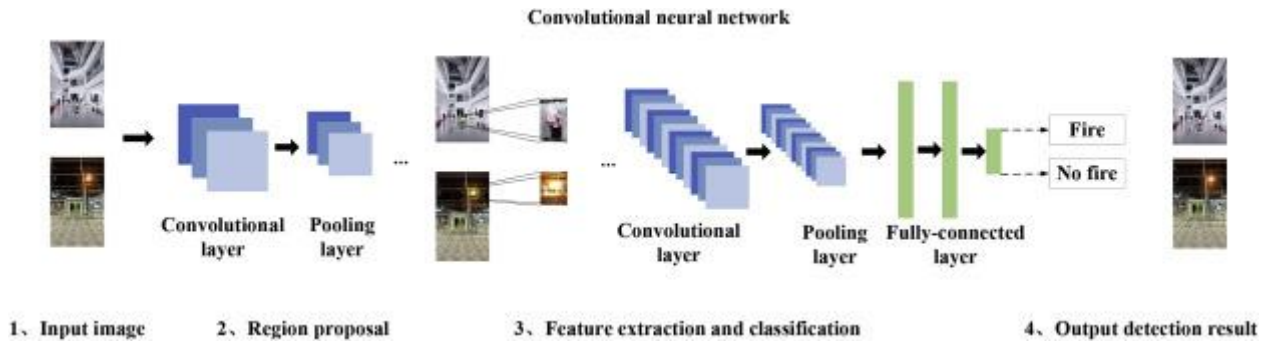


Рисунок. 1. Блок-схема алгоритмів виявлення пожежі за зображенням на основі CNN виявлення.

Згортковий рівень є основною частиною CNN. На відміну від інших нейронних мереж, які використовують ваги зв'язків і зважені суми, згортковий рівень використовує фільтри перетворення зображень, які називаються ядром згортки, для генерації карт властивостей вихідних зображень. Згортковий шар — це набір ядер згортки. Ядро згортки ковзає по зображеннях і обчислює новий піксель за зваженою сумою пікселів, над якими воно плаває, щоб створити карту функцій. Карта функцій відображає особливості вихідного зображення. Формула розрахунку шару згортки:

$$y = \sum_{j=0}^{J-1} \sum_{i=0}^{I-1} w_{ij} x_{m+1, n+j} + b, (0 \leq m \leq M, 0 \leq n \leq N)$$

де  $x$  позначає вхідне зображення певного розміру,  $w$  позначає ядро згортки певного розміру,  $b$  позначає зміщення, а  $y$  позначає вихідні карти ознак. На практиці значення  $w$  і  $b$  визначається шляхом навчання.

На рисунку 2 показано 32 ядра першого згорткового шару в Inception Resnet V2 [3] (сучасна ЗНМ) і 32 карти функцій зображення пожежі, згенерованого цими ядрами. Кількість карт ознак і ядер згортки однакова. Наприклад, у цьому шарі є три ядра згортки, тому генеруються три карти функцій. Колір пікселів відображає ступінь активації. Білі пікселі в певному місці на карті функцій вказують на те, що пікселі сильно позитивно активовані в тій же позиції на вихідному зображенні. Чорні пікселі вказують на сильно негативні активації. Сірі пікселі означають несильні активації. Порівняно з вихідним зображенням, карта функцій, згенерована згортковим ядром 14 цього шару, активована на краях. Темні/світлі області активуються позитивно на верхніх/нижніх краях, а світлі/темні області активуються негативно на верхніх/нижніх краях. Карта характеристик, створена згортковим ядром 26, активується на помаранчевих пікселях, оскільки більш білі пікселі на карті відповідають помаранчевим областям на вихідному зображенні. Це вказує на те, що ядра на попередніх рівнях в основному вивчають і витягують прості функції, такі як колір, краї тощо. Аналізуючи ці карти функцій, виявлено, що прості функції не можуть розрізнити пожежу та збурення, коли сцени складні, а також багато перешкод. Таким чином, необхідно розробити алгоритми виявлення пожежі за зображенням, які можуть отримувати складні характеристики зображення для виявлення пожежі в реальних умовах. Глибокі згорткові нейронні мережі кращі в цьому аспекті. На рис. 3 показані зразки ядер у першому, третьому та шостому шарі згортки Inception Resnet V2. Це вказує на те, що мережі виділяють складніші функції на наступних рівнях. Тому для вилучення складних характеристик зображення потрібна глибока мережа.

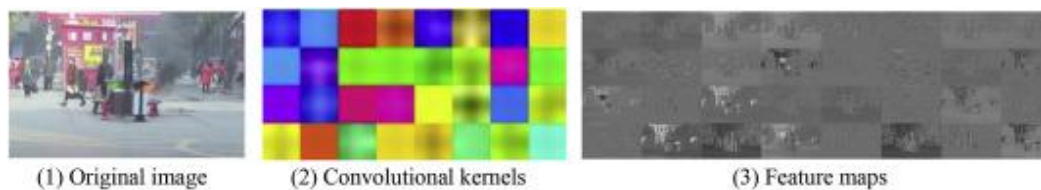


Рисунок 2. Об'єкти, виділені першим згортковим шаром

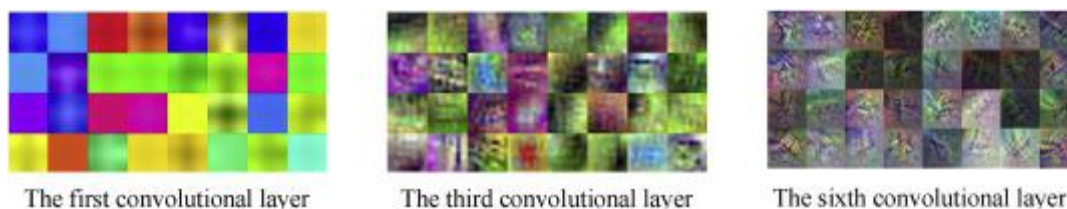


Рисунок 3. Зразки ядер у деяких згорткових шарах

### 3. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

#### 3.1. Вхідні дані

Набір даних був створений під час конкурсу NASA Space Apps Challenge у 2018 році. Метою було використання набору даних для розробки моделі, яка може розпізнавати зображення вогнем. Було зібрано дані які містять вогонь (зображення вогню), і звичайні зображення (зображення без вогню).

Дані поділено на 2 папки, папка fireimages містить 755 зображень пожежі (Рис. 4), деякі з них містять сильний дим, інша – зображення без пожеж (Рис. 5), які містять 244 зображення природи (наприклад: ліс, дерево, трава, річка, люди, туманний ліс, озеро, тварина, дорога та водоспад).



Рисунок 4. Приклад зображення з вогнем

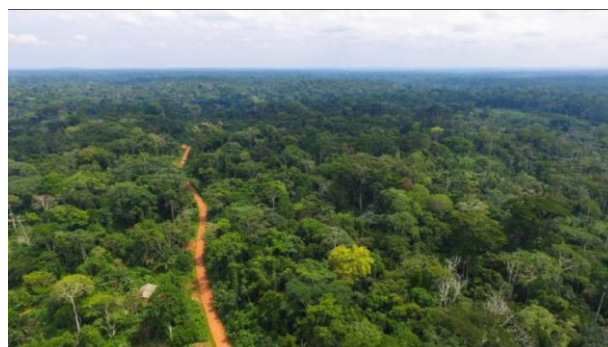


Рисунок 5. Приклад зображення без вогню

Дані будуть розділені на три різні категорії: навчання, перевірка та тестування. Навчальні дані використовуватимуться для навчання моделі CNN глибокого навчання, а її параметри будуть точно налаштовані за допомогою даних перевірки. Нарешті, продуктивність даних буде оцінена за допомогою тестових даних (даних, які модель раніше не бачила).

#### 3.2. Побудова згорткової нейронної мережі

Загальна схема побудованої нейронної мережі зображена на рисунку 6.

Тепер, коли ми вказали архітектуру моделі, ми скопіюємо модель для навчання. Для цього нам потрібно вказати функцію втрат, оптимізатор і метрику (як ми будемо оцінювати

продуктивність моделі). В якості оптимізатора використовуватимуться градієнтний спуск і стохастичний градієнтний спуск, в якості метрики – точність (accuracy).

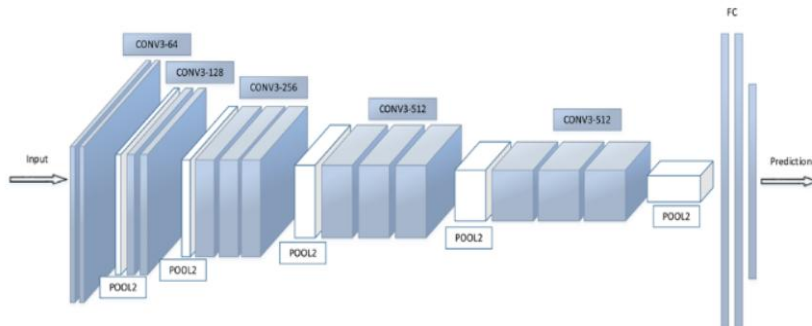


Рисунок 6. Структура побудованої мережі

### 3.3. Аналіз результатів

Нижче наведено графіки точності (Рис. 7) та функції втрат (Рис. 8) для розробленої нейронної мережі, а також матриця невідповідностей (Рис. 9).

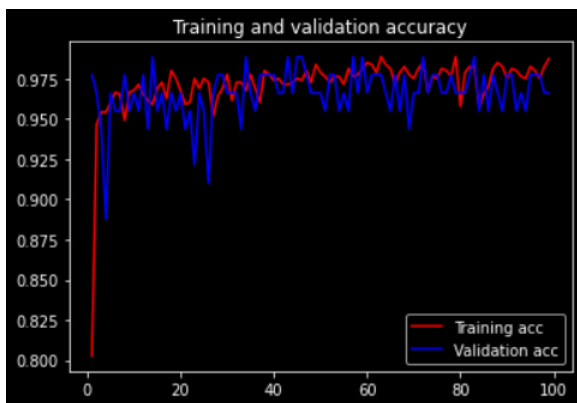


Рисунок 7. Точність

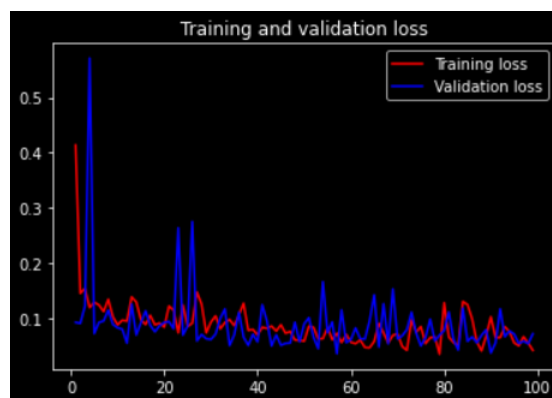


Рисунок 8. Функція втрат

Ми можемо спостерігати, як точність покращується з часом, зрештою стабілізуючись. Відповідно, втрати з часом зменшуються. Подібні графіки можуть допомогти діагностувати перенавчання. Якби ми побачили висхідну криву функції втрати (U-подібна форма на графіку), ми б запідозрили, що модель починає запам'ятовувати тестовий набір і погано узагальнювала б нові дані.

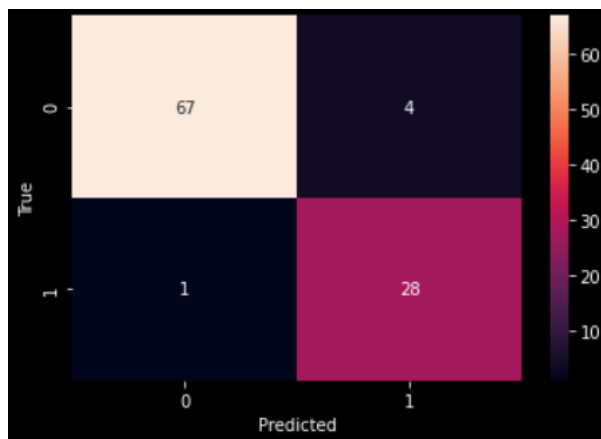


Рисунок 9. Матриця невідповідностей

Запропонована нейронна мережа показала високу точність у виявленні пожежі на зображенні. Досягнута точність 0,95.

### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Калиев Д. И., Швец О. Я. Свёрточные нейронные сети для решения задач обнаружения пожаров по данным аэрофотосъемки // Программные системы: теория и приложения. 2022. Т. 13, №1(52). С. 195–213. [http://psta.psiras.ru/read/psta2022\\_1\\_195-213.pdf](http://psta.psiras.ru/read/psta2022_1_195-213.pdf)
2. Pu Li, Wangda Zhao, Image fire detection algorithms based on convolutional neural networks, Case Studies in Thermal Engineering, Volume 19, 2020, pp 2-4.
3. S. Bianco, R. Cadene, L. Ceona, P. Napoletano Benchmark analysis of representative deep neural network architectures pp. 64270-64277.

# МЕТОДИ ПРОГНОЗУВАННЯ ІНДЕКСУ АКЦІЙ НА ОСНОВІ МЕХАНІЗМІВ ШТУЧНОГО ІНТЕЛЕКТУ

Міщенко Д.В.<sup>1</sup>, Мухін В.Є.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> mischenko\_dasha@ukr.net, <sup>2</sup> v.mukhin@kpi.ua [0000-0002-1206-9131]

Протягом останніх десятиліть об'єктом багатьох досліджень є прогнозування фондових ринків, яке, незважаючи на свою заплутаність, динамічність та нестабільність, є надзвичайно складною задачею. Для побудови ефективної моделі прогнозування необхідно враховувати різноманітні фактори, величезні обсяги даних та тривіальне співвідношення сигнал/шум, що значно ускладнює задачу прогнозування поведінки цін на фондовому ринку. Тим не менш, існує велика кількість різноманітних підходів, спрямованих на досягнення цієї мети. Метою роботи є аналіз фінансових даних, підбір моделей для прогнозування, реалізація методів прогнозування на основі механізмів штучного інтелекту та вибір найкращого методу. Результатом дослідження є програмний продукт, що видає поденний прогноз індексу акцій S&P 500.

**Ключові слова:** фондові ринки, методи прогнозування, нейронні мережі, S&P 500, авторегресія, індекси акцій.

## 1. ВСТУП

Фондовий ринок – це сукупність бірж, через які випускають, купують та продають акції публічних компаній.

Роль фондового ринку полягає в тому, щоб забезпечити компаніям спосіб залучення капіталу шляхом продажу акцій державним інвесторам. У той же час, фондовий ринок дозволяє приватним інвесторам купувати акції публічних компаній і ставати співвласниками їх бізнесу. Сукупна вартість всього фондового ринку часто відстежується та відображається за допомогою ринкових індексів, таких як промисловий індекс Доу-Джонса (Dow Jones Industrial Average) та індекс S&P 500 (S&P 500) [1].

Коли приватна компанія бажає стати публічною, вона, як правило, здійснює первинне публічне розміщення акцій (ІРО). Під час процесу ІРО компанія продає акції публічним інвесторам, щоб зібрати гроші, необхідні для погашення боргу або інвестування у свій бізнес. Після завершення ІРО ці акції починають торгуватися на одній або декількох фондових біржах, які складають фондовий ринок.

Інвестори купують акції в надії на те, що компанія, яка продає ці акції, з часом зросте і стане більш цінною, тим самим збільшуючи ціну кожної акції. Ціни на акції визначаються економічним законом попиту та пропозиції, і їх ціни часто коливаються щодня на основі змін у попиті інвесторів [2].

## 2. АКТУАЛЬНІСТЬ ПРОГНОЗУВАННЯ ФІНАНСОВИХ ДАНИХ

Прогнозування фінансових і економічних часових рядів завжди було і залишається важким завданням через його чутливість до політичних, економічних і соціальних факторів. Хорошими прикладами є пандемія у 2020 році та повномасштабне вторгнення Росії на територію України. У першому випадку, з середини лютого до середини березня 2020 ринки акцій у Європейському Союзі та Сполучених Штатах впали на 30 відсотків. Зміну очікуваного зростання дивідендів на 1 рік можна побачити на рисунку 1 та зміну очікуваного зростання ВВП на 1 рік, де А. – локдаун в Ухані, В. – карантин в Італії, С – заборона США на поїздки до ЄС та D – надзвичайний стан в США.

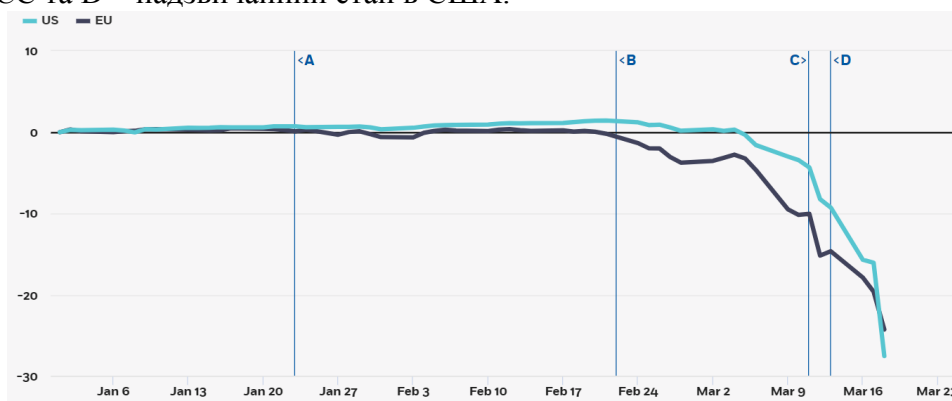


Рисунок 1. Зміна очікуваного зростання дивідендів на 1 рік [3]

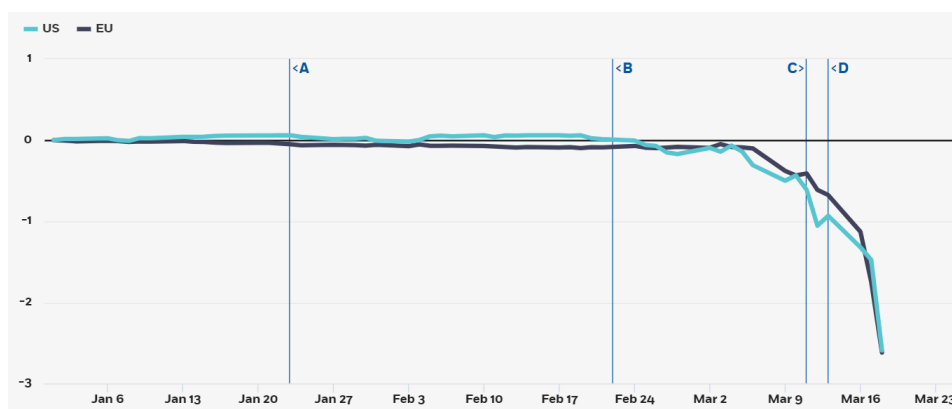


Рисунок 2. Зміна очікуваного зростання ВВП на 1 рік [3]

У другому випадку війна спричинила серйозні коливання ринку і привела до певної нестабільності. Ринки здебільшого проігнорували нещодавні конфлікти, пов'язані з Близьким Сходом та Іраном. Однак війна між Росією, Україною та союзниками по НАТО може мати серйозніший вплив, особливо на ціни на нафту та інші товари. Хоча фондові ринки відновилися до рівня до вторгнення лише за кілька тижнів після вторгнення Росії в Україну, перепади спричинили серйозні фінансові втрати для багатьох інвесторів.

У США індекс S&P 500 впав більш ніж на 7% за кілька днів і тижнів відразу після вторгнення, оскільки США та інші країни посилили жорсткі економічні санкції проти Росії, а інвестори стурбовані впливом цін на сировину. Але через місяць ринки відновилися, і S&P торгувався на рівні, вищому, ніж до вторгнення, навіть якщо ціна нафти залишалася вище 100 доларів за барель. На Рис. 3 зображений графік стану індексу S&P 500 як почалось повномасштабне вторгнення Росії в Україну та місяць після.

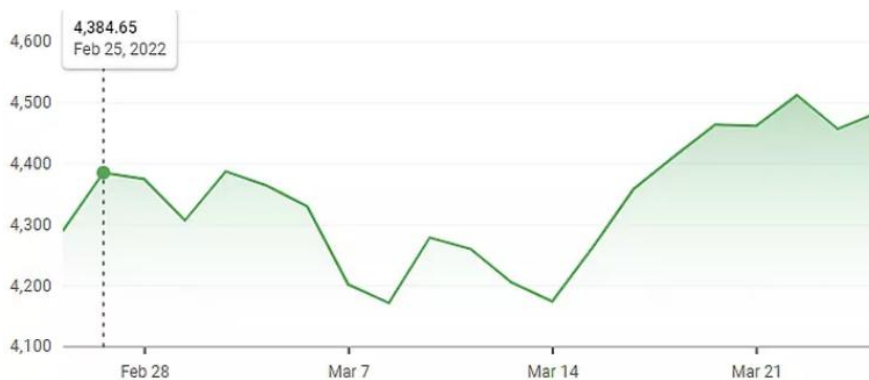


Рисунок 3. Ціна індексу S&P 500 з моменту вторгнення Росії в Україну та місяць потому [4]

Та навіть враховуючи ці випадки люди, які інвестують у фінансові ринки та обмін валюти, зазвичай шукають надійні моделі, які можуть гарантувати їм максимізацію свого профілю та мінімізацію втрат, наскільки це можливо в умовах постійної соціальної і економічної нестабільності.

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

В рамках дослідження було прийняте рішення побудувати моделі авторегресії, довгої короткострокової пам'яті та в подальшому розробка методу прогнозування на основі механізму уваги (Attention network) для порівняння та вибору найкращого методу для прогнозування.

Для дослідження та побудови прогнозу було обрано дані індексу акцій S&P 500, що є зваженим за ринковою капіталізацією індексом 500 провідних публічних компаній США, а саме за період з 31.12.2019 до 31.12.2021.

Спочатку було проведено низку маніпуляцій над вхідними даними, а саме – логарифмування, визначення порядку лагу регресії для побудови вектору ознак та розділення вибірки на навчальну та тестову з подальшою нормалізацією (стандартизацією) ряду.

Далі було реалізовано модель авторегресії порядку 3, у таблиці 1 представлено значення похибок прогнозу даної моделі. Для оцінки якості прогнозу було обрано метрики  $R^2$ , MSE, MAE та MAPE[5].

Таблиця 1. Рначення похибок прогнозу моделі лінійної регресії ARMA

	$R^2$	MSE	MAE	MAPE
Тренувальна	0.979615	0.000330	0.011232	0.001396
Тестова	0.957944	0.000067	0.006250	0.000745

Порівняємо прогноз моделі та справднні дані на рисунку 4. На верхньому графіку відображена візуалізація справжніх даних, а на нижньому отриманий прогноз за допомогою моделі лінійної регресії.



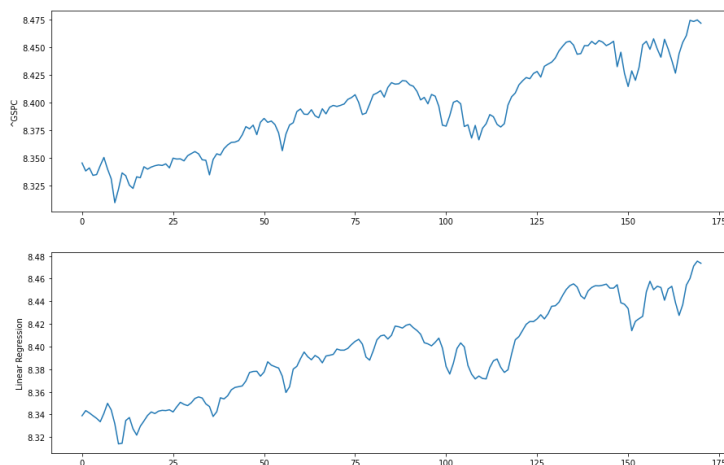


Рисунок 4. Прогноз лінійної авторегресійної моделі в порівнянні зі справжніми даними

Далі було реалізовано модель довгої короткострокової пам'яті в різних конфігураціях для знаходження найкращої моделі прогнозування ряду. У таблиці 2 показані значення похибок прогнозу LSTM моделі для різних конфігурацій.

Таблиця 1. Значення похибок прогнозу моделі LSTM

Конфігурації/ похибка	Датасет	$R^2$	MSE	MAE	MAPE
lstm_64-64_lstm_layers_128- 128_dense_layers	Train	0.965836	0.000553	0.016932	0.210003
	Test	-28.4967	0.046998	0.176939	2.100744
lstm_64-64_lstm_layers_64- 64_dense_layers	Train	0.940538	0.000962	0.024145	0.298472
	Test	-53.6261	0.087038	0.24359	2.892339
lstm_64-64_lstm_layers_32- 32_dense_layers	Train	0.917011	0.001343	0.026719	0.331763
	Test	-10.5617	0.018422	0.114673	1.361971
lstm_64-64_lstm_layers_ 128_dense_layers	Train	0.730207	0.004365	0.052797	0.653975
	Test	-23.7398	0.039419	0.178428	2.12056
lstm_64- 64_lstm_layers_64_dense_layers	Train	0.759237	0.003895	0.050692	0.627004
	Test	-41.0924	0.067067	0.225963	2.684566
lstm_64- 64_lstm_layers_32_dense_layers	Train	0.739095	0.004221	0.046994	0.584426
	Test	-17.2182	0.029028	0.143491	1.704148
lstm_32-32_lstm_layers_128- 128_dense_layers	Train	0.95694	0.000697	0.017505	0.217781
	Test	-9.48553	0.016707	0.108819	1.292486
lstm_32-32_lstm_layers_64- 64_dense_layers	Train	0.959036	0.000663	0.016207	0.202062
	Test	-3.75888	0.007583	0.069879	0.829604
lstm_32-32_lstm_layers_32- 32_dense_layers	Train	0.858923	0.002283	0.040241	0.496908
	Test	-40.314	0.065827	0.222979	2.649009
lstm_32-32_lstm_layers_128_ dense_layers	Train	0.897496	0.001658	0.033583	0.414701
	Test	-40.7559	0.066531	0.223151	2.650915
lstm_32- 32_lstm_layers_64_dense_layers	Train	0.914537	0.001383	0.028771	0.354616
	Test	-19.765	0.033086	0.164021	1.949447
lstm_32- 32_lstm_layers_32_dense_layers	Train	0.812266	0.003037	0.033089	0.413834
	Test	-3.43403	0.007065	0.071202	0.845728
lstm_128_lstm_layers_128- 128_dense_layers	Train	0.860072	0.002264	0.034473	0.427745
	Test	-61.991	0.100366	0.208407	2.470416



lstm_128_lstm_layers_64-64_dense_layers	Train	0.736589	0.004262	0.053753	0.664874
	Test	-12.6901	0.021813	0.13365	1.588622
lstm_128_lstm_layers_32-32_dense_layers	Train	0.841774	0.00256	0.037573	0.465034
	Test	-761.028	1.214169	0.879389	10.43801
lstm_128_lstm_layers_128_dense_layers	Train	0.588498	0.006658	0.059676	0.740511
	Test	-5.19191	0.009866	0.09324	1.108671
lstm_128_lstm_layers_64_dense_layers	Train	0.530279	0.0076	0.075688	0.93088
	Test	-251.953	0.40304	0.53179	6.315124
lstm_128_lstm_layers_32_dense_layers	Train	0.856315	0.002325	0.038668	0.476722
	Test	-44.1063	0.07187	0.23511	2.793376
lstm_64_lstm_layers_128-128_dense_layers	Train	0.817728	0.002949	0.045776	0.566415
	Test	-53.5476	0.086913	0.236663	2.809535
lstm_64_lstm_layers_64-64_dense_layers	Train	0.684549	0.005104	0.058813	0.726362
	Test	-146.799	0.235493	0.397531	4.719956
lstm_64_lstm_layers_32-32_dense_layers	Train	0.914261	0.001387	0.030858	0.380005
	Test	-279.403	0.446777	0.556264	6.605339
lstm_64_lstm_layers_128_dense_layers	Train	0.798097	0.003267	0.048157	0.593979
	Test	-296.757	0.474427	0.568438	6.749331
lstm_64_lstm_layers_64_dense_layers	Train	0.529415	0.007614	0.073541	0.90539
	Test	-271.97	0.434933	0.575537	6.837614
lstm_64_lstm_layers_32_dense_layers	Train	-0.82582	0.029541	0.147431	1.825889
	Test	-31.9915	0.052567	0.213531	2.538882
lstm_32_lstm_layers_128-128_dense_layers	Train	0.882336	0.001904	0.034477	0.427528
	Test	-103.438	0.166405	0.296107	3.512346
lstm_32_lstm_layers_64-64_dense_layers	Train	0.684455	0.005105	0.061368	0.757598
	Test	-103.955	0.167229	0.34645	4.11472
lstm_32_lstm_layers_32-32_dense_layers	Train	0.761386	0.003861	0.047534	0.587962
	Test	-53.5055	0.086846	0.257672	3.06136
lstm_32_lstm_layers_128_dense_layers	Train	-0.09325	0.017688	0.117866	1.454407
	Test	-91.1195	0.146778	0.345196	4.102582
lstm_32_lstm_layers_64_dense_layers	Train	0.872073	0.00207	0.025348	0.316033
	Test	0.786861	0.00034	0.014225	0.169246
lstm_32_lstm_layers_32_dense_layers	Train	-0.28382	0.020772	0.085497	1.067839
	Test	-5.26437	0.009981	0.097304	1.157964

З отриманих результатів робимо висновок, що найкраще себе показала модель з однією LSTM коміркою розміром 32 та одним прихованим повнозв'язним шаром розміром 64.

Порівнюємо прогноз реалізованих моделей та справжні дані на рисунку 5. На верхньому графіку відображена візуалізація справжніх даних, далі – лінійна регресія та на нижньому LSTM.

Отже, виходячи з візуалізацій та метрик якості моделей можна зробити висновок, що найкраще для прогнозування обраних даних показала модель довгої короткострокової пам'яті.

#### 4. ВИСНОВКИ

Мільйони людей по всьому світу щодня інвестують у фондовий ринок. Хороша модель прогнозування цін на акції допоможе інвесторам, керівництву та особам, які приймають рішення, у прийнятті правильних та ефективних рішень.

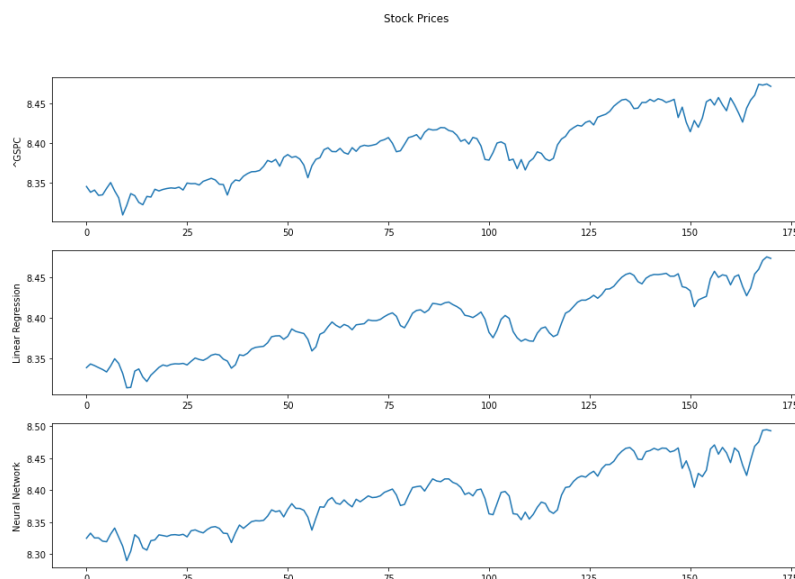


Рисунок 5. Прогноз реалізованих моделей в порівнянні зі справжніми даними

Метою дослідження було проаналізувати дані та реалізувати кілька методів прогнозування на основі механізмів штучного інтелекту та обрати найкращий з них для подальшого використання при вирішенні задачі прогнозування фінансових показників у короткостроковій перспективі.

У ході проведення дослідження були вивчені властивості досліджуваного часового ряду, а саме – показника індексу S&P 500 на період з 31.12.2019 до 31.12.2021, після чого було встановлено нестационарність часового ряду та зроблено висновок щодо автокорельованості ряду та доцільності застосування авторегресійних моделей, в якості референсних моделей.

Після виконання низки маніпуляцій над вхідними даними були проведені експерименти з побудовою та тестування моделей глибокого навчання з метою визначення найкращої моделі за показниками  $R^2$ , MSE, MAE та MAPE.

Як результат дослідження було встановлено, що модель з однією LSTM коміркою розміром 32 та одним прихованим повнозв'язним шаром розміром 64 має найкращі метрики як серед досліджуваних моделей глибокого навчання, так і в порівнянні з авторегресійними лінійними моделями.

Результат даної роботи можна застосувати при вирішенні подібних задач короткострокового прогнозування нестационарних часових рядів.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. S&P 500 Index: What It's for and Why It's Important in Investing URL: <https://www.investopedia.com/terms/s/sp500.asp> (дата звернення: 11.11.2022)
2. What Is the Stock Market and How Does It Work? URL: <https://learn.financestrategists.com/finance-terms/stock-market/> (дата звернення: 11.11.2022)
3. How coronavirus affects stock prices and growth expectations. URL: <https://www.chicagobooth.edu/review/how-coronavirus-affects-stock-prices-and-growth-expectations> (дата звернення: 10.11.2022)
4. How War Affects the Modern Stock Market. URL: <https://www.investopedia.com/solving-the-war-puzzle-4780889> (дата звернення: 11.11.2022)
5. A guide on regression error metrics. URL: <https://sefidian.com/2022/06/18/a-guide-on-regression-error-metrics-with-python-code/> (Дата звернення: 09.11.2022).

# **ЗОРОВИЙ ТРАНСФОРМЕР ДЛЯ ЗАДАЧІ КЛАСИФІКАЦІЇ РАКУ ШКІРИ**

Нікітін В.О.<sup>1</sup>, Шаповал Н.В.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>nvo63911@gmail.com,

<sup>2</sup>shovgun@gmail.com

**Автоматизація класифікації раку шкіри – актуальна проблема сучасності. Метою цієї роботи є класифікація зображень ракових пухлин на шкірі за допомогою зорового трансформера та аналіз можливості використання моделі на основі зорового трансформера як ефективного класифікатора патологічних утворень на шкірі.**

**На основі існуючих досліджень можна зробити припущення, що зоровий трансформер може мати кращі шанси для цієї задачі ніж традиційні методи, що використовують здебільшого згорткові нейронні мережі**

**Ключові слова: класифікація зображень, ViT**

## **1. ВСТУП**

Сьогодні в задачах обробки природної мови (NLP), трансформери стали стандартними моделями (наприклад, BERT, GPT-3 і т. д.). Тим не менш, можливості трансформерів у задачах комп'ютерного зору все ще є досить обмеженими. Більшість дослідників використовують згорткові мережі напряму, чи так чи інакше використовують згорткові шари у мережах

Із зображеннями трансформер використовувати дещо важче. Картинка складається з тисяч чи мільйонів пікселів. Таким чином, якщо трансформер виконує попарну операцію між кожною парою пікселів на зображенні, складність виростає до складно обчислювального рівня. Тому для картинок використовують дещо іншу форму локальної функції уваги замість глобальної.

Автори зорового трансформера використовують функцію уваги, не для всієї картинки, а для багатьох вхідних частин картинки. Тож, на першому етапі картинка розбивається на підкартинки розміром 16 на 16.

В цій роботі реалізований зоровий трансформер для задачі класифікації раку шкіри на прикладі меланом. Мета роботи полягає у тому, щоб побудувати ефективну модель класифікації на основі зорового трансформера за якої можна було б оцінювати чи варто людині із пухлиною на шкірі звернутися до лікаря та оцінити доцільність використання зорового трансформера, як моделі для вирішення цієї задачі [1].

## **2. ЗОРОВИЙ ТРАНСФОРМЕР**

Автори зорового трансформера використовують функцію уваги, не для всієї картинки, а для багатьох вхідних частин картинки. Тож, на першому етапі картинка розбивається на підкартинки розміром 16 на 16.



Рисунок 1. Розбиття картинки на підкартинки [2]

Потім створюється послідовність із цих зображень. Через нумерацію підкартинок моделі надається інформація про їх позиціонування. Це відбувається за допомогою пошукової таблиці, що зберігає вектор для кожного номеру позиції зображення. Тож, для першої підкартинки вектор з таблиці подається у модель з першим зображенням. Аналогічно з іншими підкартинками.

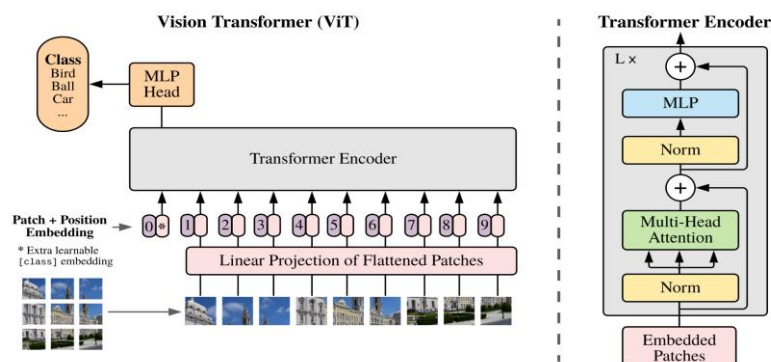


Рисунок 2. Загальна схема зорового трансформера [3]

Щоб створити надійну систему класифікації раку шкіри, потрібні різноманітні дані із усіма видами дерматологічних зображень. Оскільки потреба в ресурсах медичної візуалізації в академічних колах зростає, все більше і більше наборів даних стають загальнодоступними. Далі представлені теоретичні відомості про популярні та найбільш змістовні набори даних, що використовуються чи можуть бути потенційно використані для вирішення поставленої задачі.

Набір даних HAM100004 [4] було зібрано організацією International Skin Imaging Collaboration (ISIC) для вирішення проблеми дисбалансу та обмеження даних у наборах даних про захворювання шкіри. Він містить 10015 дермоскопічних зображень із сімома репрезентативними захворюваннями пігментних уражень шкіри. В роботі вони розділилися тільки на меланомні та здорові [5].

### 3. ТРЕНУВАННЯ МОДЕЛІ ТА РЕЗУЛЬТАТИ ГЕНЕРАЦІЇ

Зоровий трансформер навчався протягом 65 епох. В якості оптимізатора був використаний Адам із затуханням вагів. За метрику якості бралась ассурасу [3]. На рисунках 3 та 4 можна побачити графіки точності та функції втрат відповідно. Як бачимо, вони стабільно покращились протягом всього навчання.

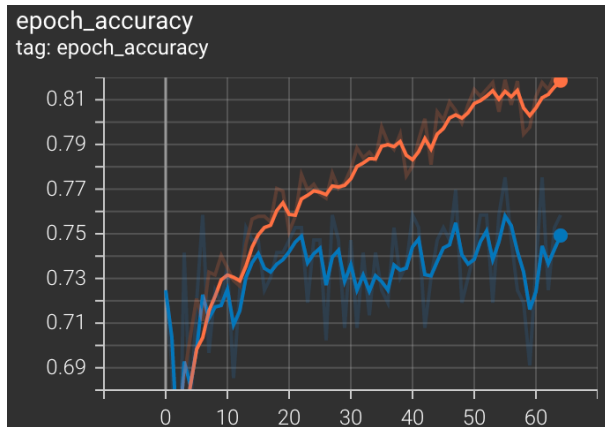


Рисунок 3. Графік ассурасу



Рисунок 4. Графік функції втрат

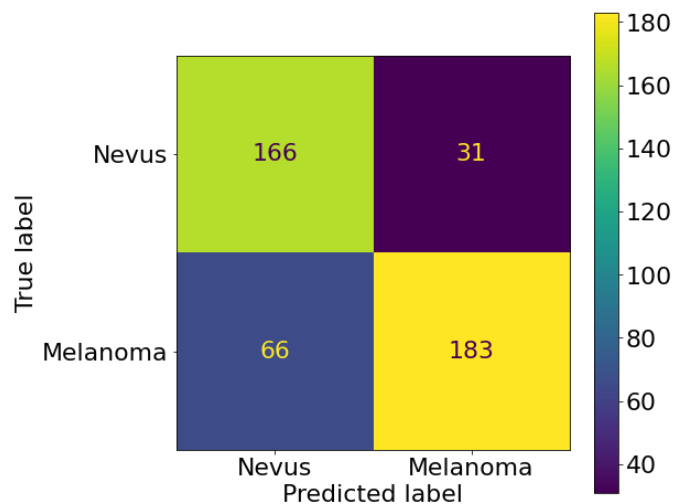


Рисунок 5. Матриця невідповідностей

### 4. ВИСНОВКИ

В результаті було отримано точність 78% на тестовому датасеті. Матриця невідповідностей (знову ж таки, на тестовому датасеті) показує, що більшість як позитивних так і негативних прикладів модель класифікувала правильно. Загалом з 446 тестових прикладів, маємо 66 помилок першого роду та 31 помилки другого роду.

Таким чином можна зробити висновок про придатність моделі зорового трансформера для поставленої задачі. Також, враховуючи тенденцію цієї моделі покращувати показники на більших датасетах краще ніж згортковій мережі [2], припущення про кращу роботу моделі зорового трансформера для класифікації раку шкіри ніж згорткової мережі є вірним.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ashish V., Ashish V., Noam S., Niki P., Jakob U., Llion J., Aidan N. G., Łukasz K.. Attention Is All You Need, 2017. 3–8 с.
2. Alexey D., Lucas B., Alexander K., Dirk W., Xiaohua Z., Thomas U., Mostafa D., Matthias M. An Image is worth 16X16 words: transformers for image recognition at scale, 2021, с 1–4.
3. Keras, ViT: документація. URL: [https://keras.io/examples/vision/image\\_classification\\_with\\_vision\\_transformer/](https://keras.io/examples/vision/image_classification_with_vision_transformer/)
4. HAM10000 : документація. URL: <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DBW86T>
5. Фаундація раку шкіри: вебсайт. URL: <https://www.skincancer.org/skin-cancer-information/>

# АЛГОРИТМИ СОРТУВАННЯ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ

Подколзін Г.Б.<sup>1</sup>, Павлюк В.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>podkolzin.gleb@lil.kpi.ua,

<sup>2</sup>pavliuk.vira@lil.kpi.ua

**Сортування є фундаментальною операцією в обчислювальній техніці. Однак, швидкість найсучасніших алгоритмів сортування в одному потоці досягли своїх меж. Тим часом глибоке навчання продемонструвало його потенціал для значного підвищення продуктивності задачі інтелектуального аналізу даних і машинного навчання. Метою роботи є дослідження доцільності та спроможності використання нейронних мереж у задачі сортування даних. Результатом дослідження є моделі sequence-to-sequence та pointer networks, що виконують сортування масивів чисел різної довжини. У роботі було використано теоретичні та емпіричні методи дослідження.**

**Ключові слова:** алгоритми сортування, нейронні мережі, глибоке навчання, sequence-to-sequence, pointer networks, сортування чисел.

## 1. ВСТУП

Сортування є основною операцією багатьох обчислювальних завдань. Воно впливає на найрізноманітніші параметри: від взаємодії користувачів із програмами до типів проблем, які ми можемо ефективно вирішити. Існує багато відомих послідовних алгоритмів сортування, таких як швидке сортування, сортування злиттям, пірамідалне сортування і інтросортування, які широко використовуються. Вони можуть задовольнити загальні вимоги до сортування в наукових дослідженнях та інженерних проектах. Однак ефективність алгоритмів послідовного знижується, коли вони стикаються з великомасштабними даними [1].

Основними цілями паралельного сортування є мінімізація часу сортування і навантаження процесорів, упорядковуючи дані елементи в бажаному порядку. Нейронна мережа - це відповідна архітектура в деяких паралельних алгоритмах, оскільки вона може обробляти дані одночасно.

## 2. МОДЕЛІ SEQUENCE-TO-SEQUENCE ТА POINTER NETWORKS

Однією з найбільш корисних властивостей, яку ми можемо мати, це здатність для моделі вивчати алгоритм, який узагальнює довші входні послідовності. Зокрема, фактична архітектура моделі легко потребуватиме здатності узагальнювати до більш довгих послідовностей, а алгоритм навчання, який її навчає, повинен буде відповідним чином упорядкувати, щоб модель не підганялася до коротших послідовностей, які вона бачить під час навчання. Ця проблема вимагає імплементацію функції, що відображає послідовність у послідовність. Очевидним шляхом вирішення цього є використання архітектури кодера-

декодера рекурентної нейронної мережі, яка використовується для задачі перекладу. З цієї причини для дослідження було обрано моделі Sequence-To-Sequence та Pointer Networks.

Моделі Sequence-To-Sequence (Seq2Seq) — це моделі глибокого навчання, які досягли значного успіху в таких завданнях, як машинний переклад, підсумовування тексту та створення підписів до зображень. Ця модель приймає послідовність елементів (слів, літер, часових рядів тощо) і виводить іншу послідовність елементів. Sequence-To-Sequence була запропонована дослідниками Іл'я Sutskever, Oriol Vinyals, Quoc V. Le для вирішення задачі перекладу тексту [2].

Модель складається з кодера і декодера (Рис 1.). Кодер фіксує контекст вхідної послідовності у вигляді прихованого вектора стану та надсилає його в декодер, який потім створює вихідну послідовність. Оскільки завдання базується на послідовності, і кодер, і декодер, як правило, використовують певну форму RNN, LSTM, GRU тощо. Прихований вектор стану може мати будь-який розмір, хоча в більшості випадків він приймається як ступінь 2 і велике число (256, 512, 1024), яке може певним чином представляти складність повної послідовності, а також домену.

RNN за дизайном приймають два входи, поточний приклад, який вони бачать, і представлення попереднього входу. Таким чином, вихід на кроці часу  $t$  залежить від поточного входу, а також входу на момент часу  $t-1$ . Саме тому вони працюють краще, коли ставляться до послідовних завдань. Послідовна інформація зберігається в прихованому стані мережі та використовується в наступному випадку.

Кодер, що складається з RNN, приймає послідовність як вхідні дані та генерує остаточне вбудовування в кінці послідовності. Потім це надсилається до декодера, який потім використовує його для передбачення послідовності, і після кожного наступного передбачення він використовує попередній прихований стан, щоб передбачити наступний екземпляр послідовності (Рис. 1) [3].

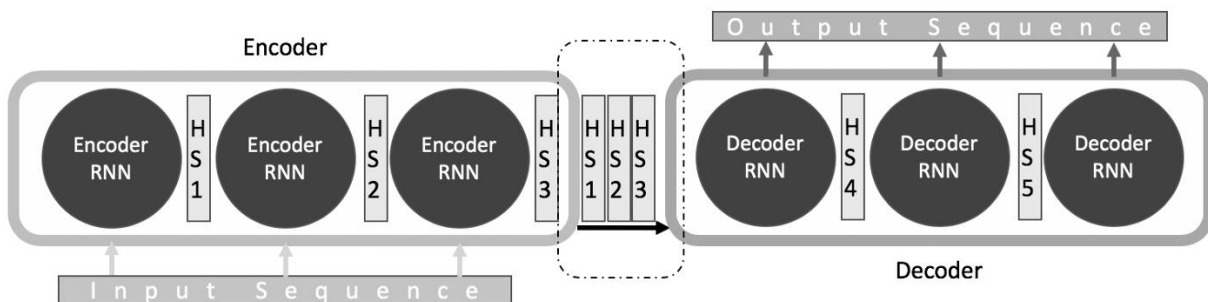


Рисунок 1. Модель кодера-декодера для моделювання Seq2Seq

Мережі вказівників (Pointer network, Ptr-Net) є різновидом моделі послідовності до послідовності з увагою. Замість перекладу однієї послідовності в іншу вони видають послідовність покажчиків на елементи вхідної серії. Основне використання цього — упорядкування елементів послідовності або набору змінної довжини [4].

Мережу вказівників запропонували Oriol Vinyals, Meire Fortunato, Navdeep Jaitly у 2017 році в своїй роботі [5]. Автори описали архітектуру мережі та запропонували її використання для вирішення таких складних геометричних задач, як: пошук плоских опуклих оболонок, обчислення триангуляцій Делоне та плоска задача комівояжера — лише з використанням навчальних даних.

Pointer networks (Рис. 2) ефективно створюють механізм уваги для змінної кількості токенів. Враховуючи послідовність токенів  $(t_0, t_1, \dots, t_{n-1})$ , мережа вказівників обслуговує вхідний словник кандидатів для наступного токена  $t_n$ . Кожен із цих кандидатів на вхід має



пов'язаний вектор вбудовування, створений кодувальником. Подібним чином, кожен з маркерів послідовності має власні вбудовані декодери. На відміну від інших моделей передбачення послідовності, мережі покажчиків побудовані таким чином, що кількість кандидатів на вхід може змінюватися під час висновку [6].

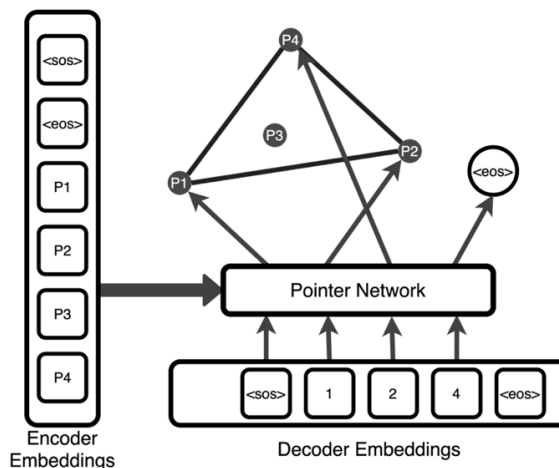


Рисунок 2. Архітектура Pointer Network

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Задля досягнення мети дослідження було виконано програмну реалізацію кодеру-декодеру Sequence-To-Sequence та Pointer networks. Написано скрипт для генерації навчального та тестового наборів даних. Також написано скрипти для тренування та оцінки моделей та репортигу метрик. Проведено навчання моделей з різними налаштуваннями параметрів та зроблено оцінку для кожної моделі. Для реалізації було обрано мову програмування Python, яка найчастіше використовується для задач машинного навчання та містить чимало бібліотек для роботи з нейронними мережами. В нашій роботі використовується бібліотека torch для реалізації моделей, numpy для різних обчислень та matplotlib для графічного представлення певних результатів.

Згенеровані дані для навчання моделей містять одновимірні масиви чисел різної довжини. Мінімальна довжина масива береться за 2, а максимальна – 100. В рамках даного експерименту розглядаються цілі числа, які не більше 32. Тестові дані для оцінки моделей згенеровані аналогічним чином. Навчальні дані містять 10000 екземплярів, тестові – 5000.

Під час навчання на кожній ітерації обчислюється функція втрат негативний логарифм правдоподібності втрати (NLLLOSS), приклад значень якої для однієї епохи зображено на рисунку 3 та рисунку 4.

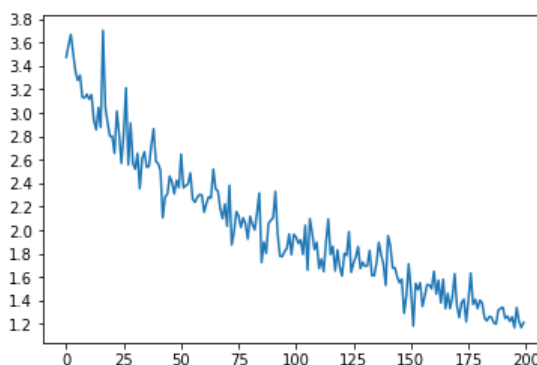


Рисунок 3. Значення NLLLOSS для першої епохи для моделі Pointer Network

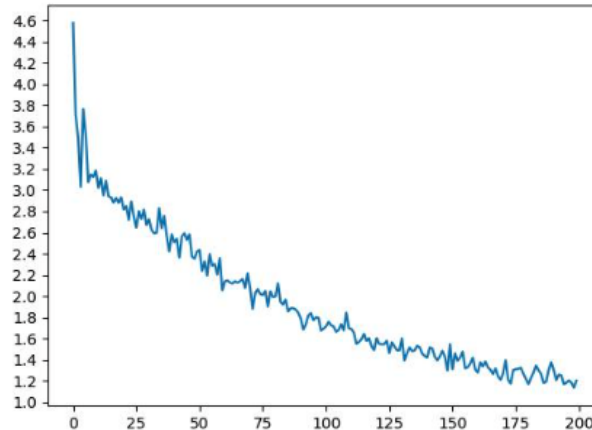


Рисунок 4. Значення NLLLOSS для першої епохи для моделі Seq2Seq

Результати тестування роботи моделей наведені у таблиці нижче (Табл. 1).

Таблиця 1. Робота сортування тестових даних моделей Sequence-To-Sequence та Pointer Networks

Оригінальний масив	Sequence-To-Sequence	Pointer Networks
[6, 23, 1]	[1, 6, 23]	[1, 6, 23]
[24, 7, 19, 28, 1]	[1, 7, 19, 24, 28]	[7, 19, 24, 28]
[22, 12, 12, 29, 5, 24, 1]	[5, 12, 22, 24, 29]	[1, 5, 12, 22, 24, 29]
[8, 10, 22, 19, 17, 6, 11, 1]	[6, 8, 10, 11, 17, 19, 22]	[1, 6, 8, 10, 11, 17, 19, 22]
[18, 11, 30, 15, 26, 11, 23, 4, 1]	[1, 4, 11, 11, 15, 18, 23, 26, 30]	[4, 11, 15, 18, 23, 26, 30]
[24, 2, 13, 21, 5, 6, 1]	[2, 5, 6, 13, 21, 24]	[1, 2, 5, 6, 13, 21, 24]
[25, 17, 5, 8, 18, 20, 18, 1]	[5, 8, 17, 18, 20, 25]	[1, 5, 8, 17, 18, 18, 20, 25]

Як видно з таблиці, модель періодично губить повторюванні значення (рядки 3 та 7 таблиці 1) та найменше значення 1. Тим не менш, немає порушення порядку сортування. Ми вирішили зробити модифікацію оригінального механізму уваги в моделі Pointer Networks, запропонованого авторами, з метою покращення результатів роботи. В таблиці 2 ми наводимо результати метрик, які дозволяють порівняти моделі. Метрика Permutation повертає долю, перестановок. Метрика Nondecreasing повертає долю елементів розташованих у порядку неспадання. Та, нарешті, метрика Accuracy повертає відсоток правильно відсортованих масивів.

Таблиця 2. Значення метрик Permutation, Nondecreasing та Accuracy

Метод	Sequence-To-Sequence	Pointer Networks	Modified Pointer Networks
Метрика			
Permutation	0.00	0.00	0.00
Nondecreasing	0.718	0.966	0.999
Accuracy	90.23%	92.81%	95.48%

Оскільки однією з причин проведення даного дослідження було бажання досягти мінімізації часу сортування даних великої розмірності, ми провели порівняння роботи моделі Pointer Network з послідовними алгоритмами сортування. Результати часу сортування для масиву довжиною 1000000 елементів наведені у таблиці нижче (Табл. 3).

Таблиця 3. Порівняння часу сортування

Алгоритм	Modified Pointer Network	Bubble Sort	Merge Sort	Heap sort	TimSort
Час	25 s	1 h 56 m	2.5 s	4 s	3.2 s

#### 4. ВИСНОВКИ

Сортування є фундаментальною операцією в обчислювальній техніці. Проблема сортування не є новою та має багато різних рішень за допомогою класичних алгоритмів сортування, які базуються на послідовному попарному порівнянні та перестановці елементів. Проте класичні алгоритми не є оптимальними у випадку великої розмірності даних. Це може стати серйозною проблемою при розробці складних рішень.

Тим не менш нейронні мережі, завдяки своїй архітектурі, здатні виконувати паралельні обчислення, що дозволяє оптимізувати витрати часу та навантаження на процесор під час сортування. Оскільки сортування чисел по суті відображає послідовність кластерів у послідовність такої самої розмірності, ми вибрали для дослідження моделі Sequence-To-Sequence та Pointer Networks. Перевагою моделей є здатність сприймати на вхід послідовності нефіксованої розмірності, саме тому їх використовують для проблеми перекладу.

Для дослідження було використано в якості навчальних даних списки цілочисельних значень різної розмірності. В результаті оцінки роботи моделей було виявлено, що вони виконують сортування чисел, проте періодично допускають похибку, зменшуючи довжину послідовності. Запропоновані метрики показали, що модель Pointer Networks краще впоралась з цією задачею.

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Zhao, Y., Dongli, H., Huang, D., Liu, Y., Yang, Z., Mao, L., Liu, C., & Zhou, F. (2020, July 27). *Sorting Data via a Look-Up-Table Neural Network and Self-Regulating Index*. <https://doi.org/10.1155/2020/4793545>
2. Sutskever, I., Vinyals, O., & V. Le, Q. (n.d.). Sequence to Sequence Learning with Neural Networks.
3. Dugar, P. (2019, July 13). Attention — Seq2Seq Models. Towardsdatascience. <https://towardsdatascience.com/day-1-2-attention-seq2seq-models-65df3f49e263>
4. Singh, A. (2020, May 24). Pointer Networks for Deep Learning. Towardsdatascience. <https://towardsdatascience.com/understanding-pointer-networks-81fbbc1ddbc8>
5. Vinyals, O., Fortunato, M., & Jaitly, N. (2017, January 2). *Pointer Networks*. arxiv.org. <https://arxiv.org/pdf/1506.03134v2.pdf>
6. McGough, M. (2021, June 18). Pointer Networks with Transformers. Towardsdatascience. <https://towardsdatascience.com/pointer-networks-with-transformers-81fbbc1ddbc8>
7. Singh, A. (2020, May 24). Pointer Networks for Deep Learning. Towardsdatascience. <https://towardsdatascience.com/understanding-pointer-networks-81fbbc1ddbc8>

# ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ТА GRADIENT BOOSTING ДЛЯ ОПТИМІЗАЦІЇ ЛОГІСТИЧНИХ ВИТРАТ

Скоробогатов С.Ю.<sup>1</sup>, Зайченко О.Ю.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup> iskor29@gmail.com [0000-0002-6532-0836],

<sup>2</sup> zaichenko.helen@iit.kpi.ua [0000-0002-4630-5155]

**Будь-яке комерційне підприємство перш ніж розпочати виробництво, аналізує ринкову кон'юнктуру. Знаючи обсяги попиту на продукцію, підприємство може планувати обсяги доходів та, відповідно, обсяг витрат. На сьогодні існує великий масив даних, які можна обробити, використовуючи методи математичного аналізу для підвищення точності прогнозування фінансових результатів підприємства на основі даних про планований обсяг доходів та витрат. Водночас, важливим є вирішення проблеми оптимізації логістичних витрат, тобто визначення оптимального варіанту доставки необхідного обсягу продукції кожному замовнику. Для вирішення даної проблеми пропонуємо використовувати нейронні мережі а також модель Gradient Boosting при прогнозуванні майбутнього попиту на продукцію, обсяги його реалізації та плануванні логістичних витрат. Вибір оптимального рішення пропонується здійснювати на основі аналізу метрик якості.**

**Ключові слова:** максимізація доходу, мінімізація витрат, прогнозування, нейронні мережі, регресійні моделі, оцінка попиту

## 1. ВСТУП

Завдяки достовірності оцінки попиту на продукцію та достатньому обсягу виготовлення продукції для задоволення даного попиту, власники та керівники підприємств можуть отримати інформацію для планування (прогнозування) відповідних доходів та витрат. Недостовірна оцінка попиту позначиться на неточності прогнозування обсягів виробництва. У свою чергу, це призведе до неточності прогнозування відповідних витрат підприємства, включаючи і логістичні. Тому в умовах високого ступеня невизначеності навколишнього середовища вважаємо необхідним підвищити ступінь достовірності прогнозів. Задля цього пропонуємо використовувати методи системного аналізу, включаючи нейронні мережі та регресійні моделі на основі Gradient Boosting.

Завдяки прогнозуванню попиту, компанії можуть виробляти достатню кількість продукції для того, щоб його задовольнити. Тому на даний момент використання аналізу даних і прогнозування є дуже поширене у великих компаніях. За допомогою прогнозування на основі нейронних мереж та регресійних моделей можна досягти високої точності оцінки попиту, що зменшить витрати компанії через похибку у аналізі даних та прийнятих на основі цього рішень. Саме шляхом прогнозування попиту на продукцію, компанії можуть оцінити свій подальший розвиток і зробити певні вкладення або обмеження на виготовлення продукції. Після оцінки попиту можна зробити замовлення на продукцію заздалегідь, що дозволить зменшити витрати на продукцію. Також можна зробити попередній план перевезень продукції, що збільшить швидкість її доставки та зменшить витрати компанії.

Коли підприємства країни отримують прибутки, це позначається на загальному стані економічного розвитку країни. Таким чином, оптимізуючи витрати підприємства, можемо отримати максимум прибутків та підвищити добробут населення країни.

## 2. МЕТОДИ МАКСИМІЗАЦІЇ ТОЧНОСТІ ОЦІНКИ МАЙБУТНЬОГО ПОПИТУ

На даний момент використовуються такі методи для оцінки попиту: опитування споживачів, лабораторні експерименти, рандомізоване дослідження та лінійна регресія. Опитування споживачів можуть бути виконані за допомогою прямих інтерв'ю або анкет з потенційними клієнтами, щоб отримати інформацію. Цей процес включає швидкі короткі інтерв'ю/опитування або поглиблені. Проте не дуже багато людей згодні відповісти на поставлені запитання, через що оцінювання попиту на основі опитів досить часто відхиляється від реальних результатів. Лабораторні експерименти мають на меті перевірити, як споживачі реагують на зміни змінних у функції попиту в гіпотетичній ситуації. Ці експерименти можна використовувати для перевірки поведінки споживачів різними способами, але вони не обов'язково мають відношення до попиту на певний продукт. Рандомізоване дослідження можна проводити, щоб усунути певні упередження, які можуть бути присутніми в іншому випадку, такі упередження можуть включати упередження відбору та упередження розподілу. Рандомізовані випробування проводяться шляхом випадкового розподілу досліджуваних на дві або більше груп. Регресія вимірює ступінь, в якому зміна в деякій залежній змінній спричинена однією або кількома незалежними змінними. Він адаптує лінійне рівняння до спостережених даних і вимірює, наскільки добре рівняння відповідає, або наскільки результати залежної змінної можна пояснити незалежною. Лінійна регресія є потенційним засобом для оцінки того, як попит (залежна змінна) змінюється з будь-якою незалежною змінною [1]. Проте лінійна регресія не є настільки точною, на відміну від повноцінної нейронної мережі, в якій є приховані шари (Рис. 1).

Все більше уваги приділяється оптимізації процесів, в основному, у вигляді зниження витрат та підвищення доходів. Зниження витрат можна досягти модернізацією обладнання, але даний підхід тягне за собою безліч витрат на проектування, купівлю, реконструкцію, а також супроводжується недоотриманим прибутком під час простою об'єкта, що реконструюється. Визначення обсягів доходів можна досягти за рахунок прогнозування. Використання нейронної мережі дозволить визначити обсяг попиту при мінімальній похибці. Зокрема, обсяг попиту на продукцію можна оцінити за допомогою нейронної мережі (рис. 1).

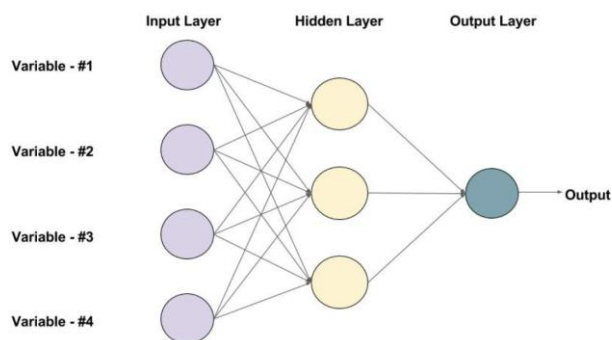


Рисунок 1. Зображення нейронної мережі [2]

Нейрон – базовий елемент нейронної мережі, одинична проста «обчислювальна комірка», здатний сприймати, перетворювати і поширювати сигнали. Тому об'єднання великої кількості нейронів в одну мережу дозволяє вирішувати досить складні завдання.

Навчання нейронної мережі насамперед полягає у зміні «сили» зв'язків між нейронами. Нейронні мережі масштабуються, вони здатні вирішувати досить складні завдання. Аналіз за допомогою нейронної мережі можна уявити як почерговий аналіз даних від менш складних до більш складних із проходженням шарів нейронної мережі [3].

Другий варіант підвищення точності прогнозування – використання регресійних моделей. Для таких цілей зазвичай використовується модель із Gradient Boosting, яка є ансамбль-моделлю. Моделі машинного навчання можна адаптувати до даних окремо або об'єднати в ансамбль. Ансамбль — це поєднання простих окремих моделей, які разом створюють більш потужну нову модель. Прискорення машинного навчання — це метод створення ансамблю. Він починається з підгонки початкової моделі до даних. Потім будується друга модель, яка фокусується на точному прогнозуванні випадків, коли перша модель працює погано. Очікується, що поєднання цих двох моделей буде кращим, ніж кожна з них окремо. Потім даний процес повторюється багато разів. Кожна наступна модель намагається виправити недоліки ансамблю всіх попередніх моделей. Ключова ідея полягає в тому, щоб встановити цільові результати для наступної моделі, щоб мінімізувати помилку (рис. 2).

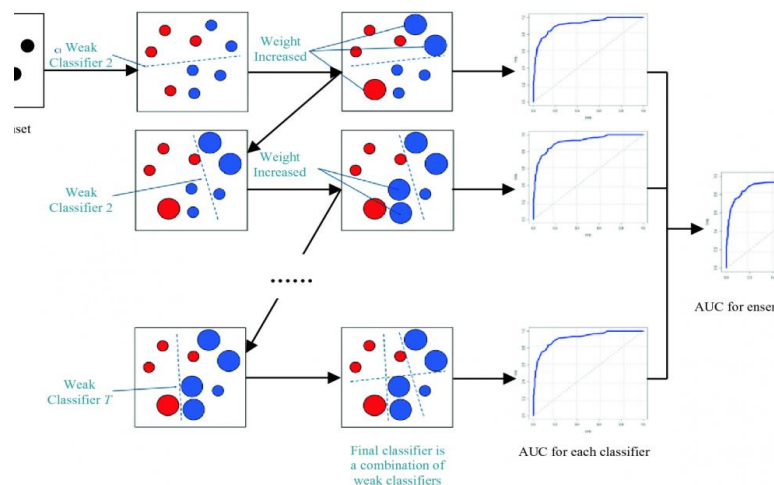


Рисунок 2. Зображення алгоритму оптимізації моделі Gradient Boosting [4]

Цільовий результат для кожного випадку в даних залежить від того, наскільки зміна прогнозу цього випадку впливає на загальну помилку прогнозу:

- Якщо невелика зміна прогнозу для випадку спричиняє значне зниження помилки, тоді наступним цільовим результатом випадку буде високе значення. Прогнози нової моделі, близькі до цілей, зменшать помилку.
- Якщо невелика зміна прогнозу для випадку не спричиняє зміни помилки, тоді наступний цільовий результат випадку дорівнює нулю. Зміна цього прогнозу не зменшує помилку.

Вважаємо доцільним для оцінки прогнозованого обсягу попиту на продукцію, який визначає обсяги доходів підприємства, використання двох методів – нейронні мережі та Gradient Boosting. В подальшому приймається для прийняття рішень про обсяги виробництва та варіанти доставки продукції до споживачів той варіант прогнозної моделі, який дає меншу похибку.

Використавши попередні вхідні дані критеріїв залежності попиту і вихідні змінні попиту, можна навчити моделі нейронної мережі і Gradient Boosting, після чого обрати найбільш точну модель. Є різні критерії визначення точності моделі, але найбільш популярними для оцінки

неперервних значень  $\epsilon$ : Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Accuracy.

Після оцінки майбутнього попиту за одним із варіантів, можна розв'язати задачу знаходження найбільш оптимального перевезення товару від складу товару до місця збуту товару. Для цього можемо використати метод потенціалів або угорський метод для вирішення отриманої транспортної задачі [5].

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

#### 3.1 Визначення точності прогнозування на основі нейронних мереж

За допомогою фреймворку tensorflow було побудовано нейронну мережу, в якій було 3 приховані шари. На першому шарі знаходилось 40 нейронів, на другому прихованому шарі знаходилось 50 нейронів і на останньому 1 нейрон. Модель навчено на 506 спостереженнях. Для навчання було обрано 80% даних для тренування і 20% даних для тестування навченої моделі (рис. 3).

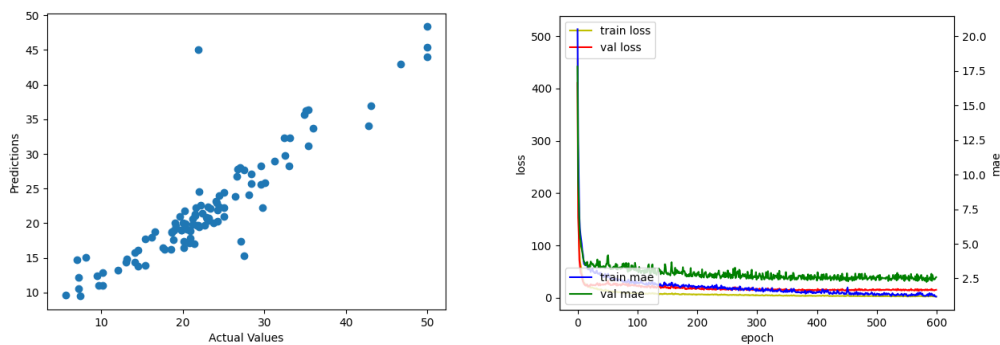


Рисунок 3. Навчання та тестування моделі із використанням нейронної мережі

За графіком помилок можемо побачити, що модель дуже швидко навчалась на початковому наборі даних, але із збільшенням кількості проходів модель зменшила швидкість зняження функції помилки. Перенавчання на наборі для тренування не відбулось, адже була використана регуляризація у вигляді Dropout, який дозволяє відключати нейрони на обраних шарах, щоб модель не орієнтувалась на конкретні нейрони. За графіком порівняння точних значень і оцінених за допомогою моделі можна зазначити, що модель дуже точно описала дані. Особливо це помітно, бо в ідеальному випадку повинна бути пряма по діагоналі. Є декілька викидів, але, зважаючи на досить маленький датасет, дані оцінки є досить точними також. За допомогою нейронної мережі можемо знайти майбутній попит на товар.

#### 3.2 Визначення точності прогнозування на основі Gradient Boosting Regressor

За допомогою бібліотеки Scikit-learn та зробленого в ній ансамблю моделей Gradient Boosting Regressor було зроблено навчання і прогнозування на 506 спостереженнях. Використано 80% даних для тренування і 20% даних для тестування навченої моделі (рис. 4).

За допомогою пошуку гіперпараметрів із GridSearchCV у бібліотеці Scikit-learn обрано використано такі параметри: learning\_rate=0.05, loss='ls', max\_depth=5, max\_features='log2', min\_samples\_leaf=5, min\_samples\_split=12, n\_estimators=200, subsample=0.8. Перенавчання у натренованої моделі відсутнє, як можна побачити за графіком точних значень і передбачених значень. Оскільки точки знаходяться майже по діагоналі, можна вважати, що ця модель виконала передбачення навіть краще, ніж просто нейронна мережа. Проте так може відбуватись не завжди, адже у нейронній мережі значено більше можливості в плані



налаштування шарів та функції, яку дані шари можуть виконувати. За допомогою Gradient Boosting Regressor можемо знайти майбутній попит на товар.

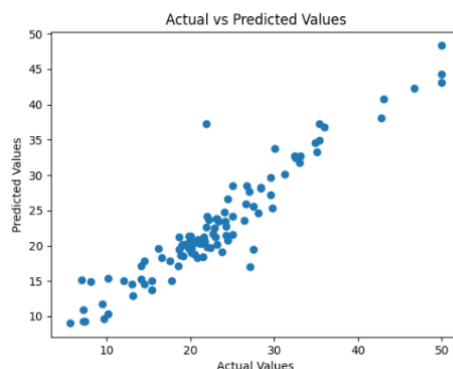


Рисунок 4. Навчання та тестування моделі із використанням Gradient Boosting Regressor

## 4. ВИСНОВКИ

На сьогодні одним із найбільш важливих критеріїв ефективності витрат компанії є оцінка попиту і подальша закупка і перевіз матеріалів з місця виготовлення на місця продажі або подальшого виготовлення іншої продукції. Правильний підхід до вирішення проблеми оцінки попиту надає можливості для прийняття правильних рішень у розподілі продукції, отриманні максимального прибутку та досягненні максимальних результатів у максимізації доходів у майбутньому.

Завдяки високоточному прогнозуванню попиту за допомогою нейронних мереж та регресійних моделей на прикладі Gradient Boosting Regressor, можна отримати дуже високу точність попиту в наступні рази на основі попередньо зібраних даних. Це дозволить компанії отримати наглядну картину, щоб розуміти фінансову ситуацію і потім зробити необхідні дії для досягнення максимального заробітку та стабільності.

За результатами метрик якості та візуального аналізу прогнозованих значень побудованими моделями, можна отримати попит із точністю більше ніж 95%. Це дозволяє замовити продукцію у виробників заздалегідь та із відповідністю до обмежень потужностей виробника розрахувати оптимальний план для побудованої транспортної задачі.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wilkinson, N. Управлінська економіка: підхід до вирішення проблем. Кембридж: Cambridge University Press, стор.125.
2. What is a Neural Network? [Електронний ресурс]. – Режим доступу: <https://www.projectpro.io/article/neural-network-projects/440> (дата доступу 19.11.2022)
3. Zell, Andreas. Simulation Neuronaler Netze [Simulation of Neural Networks] (1st ed.). Addison-Wesley. p. 73. ISBN 3-89319-554-8.
4. Gradient Boosting – What You Need to Know [Електронний ресурс]. – Режим доступу: <https://datascience.eu/machine-learning/gradient-boosting-what-you-need-to-know/> (дата доступу 19.11.2022)
5. Ю. П. Зайченко. Дослідження операцій. Підручник. Сьоме видання, перероблене та доповнене. – К.:Видавничий Дім «Слово», 2006. – 816 с.



# МЕТОДИ БАГАТОФАКТОРНОЇ АВТЕНТИФІКАЦІЇ ДО ВЕБ ДОДАТКІВ ЗА ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ ТА ТЕХНОЛОГІЇ БЛОКЧЕЙН

Славінський В.О.<sup>1</sup>, Гуськова В.Г.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>sevoslav@gmail.com

У інтернеті практично всі веб-сайти вимагають підтвердження особи, однак поточні методи авторизації недосконалі, оскільки вони покладаються на застарий метод ідентифікації через e-mail і пароль, який легко зламати або ненавмисно передати зловмисникам. У роботі розглянуто альтернативні методи авторизації, такі як авторизація через біометрію, а також новітній підхід - автентифікація через блокчейн. Результатом дослідження є порівняння існуючих методів авторизації та автентифікації, а також перспективи автентифікації за допомогою блокчейну та біометрії у вебі 3.0.

**Ключові слова:** автентифікація, мультифакторна автентифікація, блокчейн, web3.0, розпізнавання облич.

## 1. ВСТУП

Об'єктом дослідження є мультифакторна авторизація та аутентифікація в мережі Web3.0.

Предметами дослідження визначено нейронні мережі для розпізнавання облич та блокчейн для роботи з розподіленими ідентифікаторами. Задача роботи полягає у створенні прикладу використання мультифакторної автентифікації на основі нейронних мереж і блокчейну в мережі Web3.0. Отримані результати надалі можна буде розвинути в самостійну бібліотеку, яку зможуть використовувати незалежні розробники для своїх додатків в розподіленому інтернеті. Дана робота допоможе зробити систему безпечнішою та зручнішою для користувачів, а також дасть змогу перестати зберігати свої дані у великих корпорацій, таких як Google або Facebook.

## 2. МЕТОДИ ІДЕНТИФІКАЦІЇ ОСОБИ

Для ідентифікації особи може використовуватися документ, що посвідчує особу, наприклад, посвідчення особи (також відоме як ІС, ID-картка, картка громадянина), або паспортна картка (якщо вона має невеликий формат, як звичайна кредитна картка). Деякі країни також видають офіційні документи, що посвідчують особу, такі як національні посвідчення особи, які можуть бути обов'язковими або необов'язковими, в той час як інші можуть покладатися на регіональні посвідчення особи або неофіційні документи для підтвердження особи.

Існують і деякі інші прийнятні форми посвідчення особи:

**Щось, що особа знає (SYK):** пароль, PIN-код, дівоче прізвище матері або кодова комбінація замка. Ідентифікація особи за допомогою чогось, що вона вже знає, є, ймовірно, найпростішим варіантом, але одним з найменш безпечних.

**Щось, що є у особи (SYH):** ключ, свайп-карта, картка доступу або бейдж - все це приклади предметів, якими може володіти особа. Цей метод зазвичай використовується для отримання доступу до таких об'єктів, як банки та офіси, але він також може бути використаний для отримання доступу до конфіденційних місць або перевірки системних облікових даних. Це також простий варіант, але ці предмети легко вкрати.

**Щось, чим є людина (SYA):** біометричні дані особи є унікальними і не можуть бути втрачені або викрадені. Використання біометричних даних для ідентифікації особи є найбільш точним і безпечним варіантом.

Кіберзлочинці постійно вдосконалюють свої системні атаки. Як наслідок, команди безпеки мають справу з безліччю постійно мінливих проблем з автентифікацією. Ось чому компанії починають розгортати більш складні плани, які включають автентифікацію. Деякі з найпоширеніших методів аутентифікації, що використовуються для захисту сучасних систем, включають в себе:

**Face Recognition.** Аутентифікація за обличчям - це технологія, яка дозволяє людям отримувати доступ до онлайн-сервісів, фізичних параметрів та інших ресурсів, використовуючи зображення свого обличчя.

Аутентифікація за обличчям, яка також називається розпізнаванням обличчя, спирається на вбудовану технологію зондування мобільних та інших пристроїв. На відміну від інших рішень ідентифікації, таких як паролі, верифікація за допомогою електронної пошти, селфі або зображень, або ідентифікація за відбитками пальців, біометричне розпізнавання обличчя використовує унікальні математичні та динамічні шаблони, які працюють як сканер обличчя, що робить цю систему однією з найбезпечніших та найефективніших.

**Fingerprint Recognition.** Автентифікація за відбитками пальців - це акт перевірки особи на основі одного або декількох відбитків пальців. Ця концепція використовується протягом десятиліть у різних сферах, включаючи цифрову ідентифікацію, кримінальне правосуддя, фінансові послуги та охорону кордонів.

**FIDO 2.** Fast Identity Online (FIDO) - це відкритий стандарт для безпарольної автентифікації. FIDO дозволяє користувачам і організаціям використовувати цей стандарт для входу на свої ресурси без імені користувача або пароля за допомогою зовнішнього ключа безпеки або ключа платформи, вбудованого в пристрій.

Ці ключі безпеки FIDO2, як правило, є USB-пристроями, але можуть також використовувати Bluetooth або NFC. Завдяки апаратному пристрою, який обробляє автентифікацію, безпека облікового запису підвищується, оскільки немає пароля, який можна було б розкрити або вгадати.

**SMS OTP.** Це зручний метод, який не вимагає від користувачів встановлення будь-яких додатків. Замість цього, для аутентифікації на зареєстрований телефон користувача надсилається одноразовий пароль за допомогою SMS, який використовується для аутентифікації.

Загальні проблеми:

- Проблеми з користувацьким інтерфейсом - OTP часто мають обмеження за часом, а обмежений прийом мобільних операторів може спричинити проблеми для користувачів у віддалених районах.
- Вразливість до шкідливого програмного забезпечення, атак SS7 та підміни SIM-карт.

**OATH OTP (Soft Tokens).** Це програмне забезпечення, яке може бути вбудоване в мобільні додатки і використовує криптографічні операції для автентифікації користувача та пристрою. Ці рішення зазвичай забезпечують більш плавний користувацький інтерфейс; немає необхідності перемикатися між додатками або покладатися на апаратний пристрій. З

точки зору безпеки, вони мають значні переваги, оскільки SDK з м'якими токенами підтримують просунуту криптографію, наприклад, цифрові підписи.

Загальні проблеми:

- Поганий користувацький досвід (UX) - користувачі повинні постійно перемикатися між додатками для підтвердження особи/транзакції; користувач втрачає доступ з кожною зміною/втратою/оновленням свого смартфона; немає безпечних варіантів резервного копіювання

**Blockchain.** Блокчейн-аутентифікація відноситься до системи, розробленої для підвищення безпеки користувачів та перевірки особи користувача і дозволяє користувачам підключатися до ресурсів, заснованих на технологіях цифрової валюти, транзакцій, криптовалют тощо.

Вона використовує технологію розподіленого реєстру блокчейн та методи аутентифікації для підвищення конфіденційності та безпеки систем аутентифікації. Вся мережа, заснована на блокчейні, здатна мати власну цілісність даних.

Персональна інформація, яка використовується для перевірки особи користувача, зберігається в хеші блоку, наприклад, ім'я користувача або пароль. Це допоможе досягти суверенної ідентичності.

**Passwords.** Паролі є найпоширенішим методом автентифікації. Паролі можуть бути у вигляді рядка букв, цифр або спеціальних символів. Для захисту необхідно створювати надійні паролі, які включають комбінацію всіх можливих варіантів.

Однак паролі схильні до фішингових атак, що послаблює їх ефективність. Середньостатистична людина має близько 25 різних облікових записів в Інтернеті, але тільки 54% користувачів використовують різні паролі для своїх облікових записів.

Справа в тому, що існує дуже багато паролів, які потрібно пам'ятати. Як наслідок, багато людей обирають зручність, а не безпеку. Більшість людей використовують прості паролі замість того, щоб створювати надійні паролі, тому що їх легше запам'ятати.

**Secret Questions.** Безпечні питання є поширеним методом автентифікації особи, з яким ви, ймовірно, стикалися раніше. Створюючи обліковий запис або реєструючись на послугу в Інтернеті, користувачі конфіденційно передають відповіді на секретні запитання провайдеру.

Як правило, ці питання безпеки та відповіді на них використовуються для самообслуговування відновлення пароля - введення правильної відповіді перевіряє користувача та дозволяє йому скинути пароль - хоча ви також можете використовувати питання безпеки як додатковий фактор автентифікації для входу в систему.

**Personal Info.** Персональна аутентифікаційна інформація - ПІН-код або будь-який інший пароль чи інформація, яку користувачі створюють або приймають для використання з метою аутентифікації своєї особи у додаток. Інші приклади Персональної автентифікаційної інформації можуть включати ім'я, дату народження, поштовий індекс, місце проживання, які можуть використовуватися або вимагатися для здійснення Інтернет-транзакцій або інших операцій.

**Email OTP.** Метод Email OTP дозволяє авторизуватися за допомогою одноразового пароля (OTP), який надсилається на зареєстровану адресу електронної пошти. При спробі авторизації на будь-якому сервісі сервер надсилає OTP на зареєстровану електронну адресу користувача.

Щоб використовувати функцію Email OTP, необхідно спочатку зареєструвати альтернативний ідентифікатор електронної пошти. Реєстрація альтернативного ідентифікатора електронної пошти необхідна для того, щоб OTP могло бути надіслано на цей ідентифікатор, оскільки ви не зможете отримати доступ до основного ідентифікатора

електронної пошти, якщо обліковий запис буде заблоковано або якщо ви забудете пароль до облікового запису.

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Кожен метод автентифікації користувача, як правило, може бути оцінений за допомогою наступних трьох ключових параметрів:

**Зручність:** Наскільки природно і без проблем для кінцевого користувача використовувати цю автентифікацію?

**Безпека:** наскільки складно зловмиснику обдурити автентифікацію?

**Розгортання:** Наскільки легко розгортати для всіх користувачів на різних платформах, пристроях, географічних регіонах тощо? Важливим фактором є те, чи є сценарій використання B2C або B2E, тобто чи є кінцевий користувач споживачем або працівником.

Тож для дослідження візьмемо описані вище методи автентифікації і проаналізуємо їх за обраними факторами та наведемо табличний огляд деяких з найпопулярніших методів автентифікації з оцінками за кожним з трьох параметрів. Зверніть увагу, що ці бали є простими відносними показниками, які дають загальне уявлення про відносні сильні та слабкі сторони кожного методу.

Таблиця 1. Порівняння існуючих методів автентифікації

Методи автентифікації	Вид	Відомості			Коментарі
		Зручність	Безпека	Розгортання	
Face Recognition	SYA	9	7	2	Загальне розпізнавання облич - непослідовна безпека, сильно залежить від апаратного та програмного забезпечення пристрою. Наприклад, безпека розпізнавання облич на Android значною мірою залежить від марки пристрою.
Fingerprint Recognition	SYA	7	7	4	Загалом безпечніший і більш розповсюджений, ніж розпізнавання обличчя, і з кожним роком стає все більш розповсюдженим. Практично всі смартфони мають досить пристойний сканер відбитків пальців.
FIDO 2	SYH	6	8	3	Використовує біометричні автентифікатори і намагається зробити їх сумісними на різних пристроях, працюючи з усіма гравцями галузі, але має деякий час, щоб дозріти в плані підтримки.
SMS OTP	SYH	3	5	8	Не дуже безпечний, оскільки схильний до атак на підміну SIM-карт, але активно використовується як другий фактор у сценаріях B2C, оскільки кожен користувач має смартфон.
OATH OTP (Soft Tokens)	SYH	3	8	5	Досить безпечний, але потребує програми-аутентифікатора, і користувач повинен знати, як її налаштувати. Це може бути корисно, якщо кінцевий користувач технічно підкований, як правило, у сценаріях B2E.

Blockchain	SYH	8	10	5	Дуже безпечний спосіб аутентифікації, і в той же час досить зручний оскільки не вимагає носіння спеціальних пристроїв. Досить мати доступ до гаманця з будь-якого пристрою будь то персональний комп'ютер або телефон. Також легко вбудовується в додатки оскільки спочатку був розроблений в середовищі web і має хорошу інтеграцію з будь-якими веб додатками.
Passwords	SYK	4	3	6	Найбільш розповсюджений через історію та можливість розгортання має добре відомі проблеми з безпекою. Всі фахівці з безпеки хочуть, щоб вона зникла якомога швидше.
Secret Questions	SYK	5	2	7	Можливо, це гірше, ніж паролі, особливо в епоху соціальних мереж, оскільки багато інформації про користувачів є загальнодоступною.
Personal Info	SYK	6	0	8	Запитувати особисту інформацію, таку як дата народження, є жахливим методом в епоху соціальних мереж. Він навіть не повинен бути вказаний як метод аутентифікації, але я бачив, що деякі сервіси використовують його.
Email OTP	SYK	4	5	7	Гідний метод для сценаріїв B2C, таких як SMS, і не залежить від іншого пристрою, але в значній мірі залежить від ефективності та безпеки електронної пошти користувача.

Якщо зобразити ці оцінки на бульбашковій діаграмі, де розмір бульбашки відображає розгортання, безпеку по осі Y та зручність використання по осі X, то вона виглядатиме як зображено на рисунку 1:

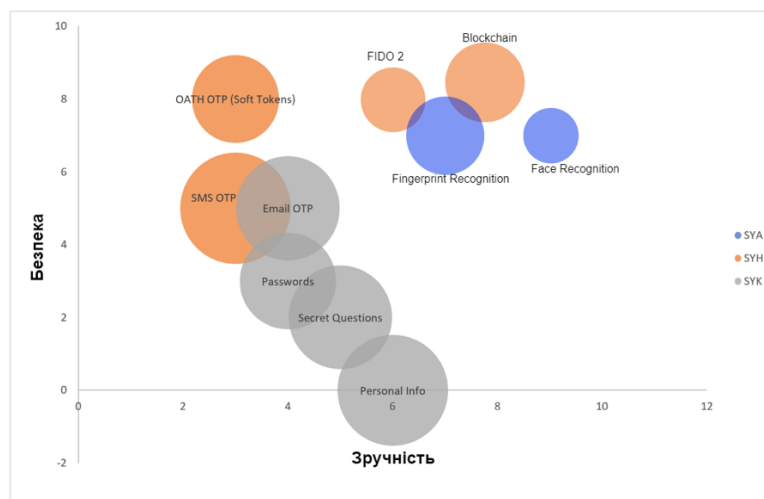


Рисунок 1. Порівняння методів автентифікації користувачів за 3 параметрами - безпека, зручність та розгортання. Розмір бульбашки вказує на можливість розгортання цього методу автентифікації

## 4. ВИСНОВКИ

Дивлячись на отримані результати можна легко побачити, що в більшості ситуацій єдиний метод аутентифікації користувача не може бути панацеєю. Саме для цього на допомогу приходять мультифакторна автентифікація, вона може допомогти убезпечити вхід користувача, використовуючи другий фактор, бажано іншої категорії. Як ми бачимо з результатів, фактор знання найслабший у безпеці та зручності, оскільки вимагають постійного введення даних і їх запам'ятовування. Якщо ж говорити про фактор того, хто ти є, то найпростіше використовувати розпізнавання облич, оскільки воно вимагає найменше дій від користувача і є більш унікальним, оскільки відбиток пальця простіше підробити і складніше розпізнати. І нарешті останній фактор - це те що людина має. З них найкращі показники у блокчейна, оскільки він є найнадійнішим, оскільки блокчейн гарантує, що дані користувача не будуть змінені, оскільки вони вже записані в блокчейн, а також немає необхідності мати спеціальний пристрій, достатньо мати доступ до свого криптогаманця.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is Multi-Factor Authentication (MFA) and How Does it Work? URL: <https://www.onelogin.com/learn/what-is-mfa>
2. Two-Factor Authentication (2FA) URL: <https://duo.com/product/multi-factor-authentication-mfa/two-factor-authentication-2fa>
3. Identity and Web3 URL: <https://auth0.com/blog/identity-and-web3/>
4. Cryptocurrency Explained With Pros and Cons for Investment URL: <https://www.investopedia.com/terms/c/cryptocurrency.asp>
5. Advantages and disadvantages of biometrics URL: <https://www.miteksystems.com/blog/advantages-and-disadvantages-of-biometrics>
6. The 5 most common multi-factor authentication (MFA) methods URL: <https://securitybrief.asia/story/the-5-most-common-multi-factor-authentication-mfa-methods>
7. Comparison of User Authentication Methods on Three Parameters [URL: <https://www.cyberark.com/resources/blog/comparison-of-user-authentication-methods-on-three-parameters>

# АГЕНТНА АРХІТЕКТУРА МЕХАНІЗМІВ КОНСЕНСУСУ В БЛОКЧЕЙНІ

Тимошук О.Л.<sup>1</sup>, Лисов Б.С.<sup>2</sup>

<sup>1,2</sup> Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>o.tymoshchuk@kpi.ua [0000-0003-1863-3095], <sup>2</sup>lysov.bohdan@lll.kpi.ua

**Блокчейн технологія створила твердий фундамент для децентралізації проведення фінансових операцій без єдиного контролюючого органу за правилами мережі. «The Merge» - операція переходу з proof-of-work на proof-of-stake механізм консенсусу - створює нове підґрунтя для дослідження ефективності використання ресурсів в технологіях. Мета дослідження полягає у розробці агентної архітектури для аналізу блокчейну Ethereum, роботи його транзакційного механізму в умовах обраного механізму консенсусу та порівняння цих механізмів у вигляді зібраних метрик та як вони впливають на кількість витраченої електроенергії, а отже і впливу на навколишнє середовище. Новизна роботи полягає у підборі специфічних параметрів для забезпечення оптимального використання децентралізованої мережі відповідно до використання електроенергії за допомогою агентної архітектури.**

## 1. ВСТУП

З появою таких P2P платіжних систем, як Bitcoin і Ethereum, збільшується кількість людей, які зацікавлені в покупці та продажі цифрових активів. Інфраструктура не стоїть на місці: з'являються різні аналоги з додатковим функціоналом, але всі вони підпорядковуються суворим стандартам.

Для валідації стану мережі існують механізми консенсусу, під якими мається на увазі досягнення загальної згоди її членами. Що стосується блокчейну Ethereum, процес є формалізованим, і досягнення консенсусу означає, що принаймні 66% вузлів у мережі погоджуються щодо глобального стану мережі.

До 15 вересня 2022 року Ethereum використовував механізм proof-of-work, але зрештою перейшов на proof-of-stake концепт. Операція переходу усунула потребу в енергоємному майнінгу, що натомість дозволило захистити мережу за допомогою накопиченого ETH. Це важливо, оскільки знижується кількість використаного ресурсу для проведення транзакції, що позитивно впливає на довколишнє середовище в порівнянні з минулими роками.

За допомогою штучного інтелекту, а саме інтелектуальних агентів та архітектур, що спираються на це поняття, можна протестувати та проаналізувати якість змін, навантаживши мережу і конфігуруючи її параметри.

Агентом вважаємо все, що може розглядатися як сприймаюче своє середовище за допомогою датчиків і впливає на це середовище за допомогою виконавчих механізмів [1].

Основним напрямком цієї роботи є застосування методів штучного інтелекту для налаштування мережі блокчейну і підбору найефективніших комбінацій параметрів з точки зору швидкості та задіяного ресурсу обчислень. Головною метою виступає збір та аналіз метрик. Такий аналіз надасть можливість зробити висновки, наскільки безпечною та ефективною є децентралізована система, чи правильно обрано вектор розвитку сучасних блокчейнів. Висновок дозволить з більшою впевненістю інвестувати чи утриматись від інвестування в цю сферу, а також запропонувати краще рішення, якщо воно існує.

## 2. ВПЛИВ КРИПТОВАЛЮТ ТА ДЕЦЕНТРАЛІЗОВАНИХ СИСТЕМ НА НАВКОЛИШНЄ СЕРЕДОВИЩЕ

Найвідоміший вплив технології блокчейн на навколишнє середовище пов'язаний із споживанням енергії та, отже, можливим негативним впливом на клімат. Поточний стандартний процес перевірки транзакцій, заснований на алгоритмі підтвердження роботи, є «надзвичайно енергоємним», оскільки вимагає величезної кількості обчислювальних потужностей і, отже, електроенергії для роботи пов'язаних комп'ютерних розрахунків. Більш широке використання технології блокчейн могло б протидіяти зусиллям із пом'якшення кліматичних змін, оскільки електроенергія в усьому світі в основному виробляється з викопного палива. У 2016 році виробництво з горючих видів палива все ще становило 67,3% від загального світового валового виробництва електроенергії.

Приклад Bitcoin є особливо показовим. Порівняно з альтернативними методами оплати, біткоїн був у 20 000 разів більш енергоємним, ніж Visa. У 2019 році, за даними деяких аналітиків, енергія, споживана для кожної транзакції біткоїну, зросла до 635 кВт/год, що дорівнює електроенергії, яка могла б забезпечити енергією приблизно 21 домогосподарство США протягом 1 дня. Згідно з останніми дослідженнями, споживання електроенергії біткоїнами становить від 20 до 80 ТВт-год на рік, або приблизно 0,1-0,3% світового споживання електроенергії. Наприклад, Cambridge Bitcoin Electricity Consumption Index (CBECI) наводить цифру в 64 ТВт-год на рік (CBECI, 2019), що перевищує річне споживання електроенергії Швейцарією або всіма електромобілями в усьому світі (58 ТВт-год) у 2018 році. Ці оцінки слід тлумачити з обережністю через методологічні проблеми, обмежену доступність даних і дуже різноманітні умови в галузі. Крім того, ці цифри все ще набагато нижчі, ніж інші кінцеві споживачі, такі як охолодження, яке спожило 2 020 ТВт-год електроенергії в 2016 році.

Тим не менш, біткоїн - це лише одна криптовалюта, яка є лише одним із застосувань блокчейну. Важливо знайти спосіб зменшити споживання енергії для перевірки транзакцій. Заміна оригінального механізму консенсусу, тобто алгоритму підтвердження роботи, на інші підходи (такі як алгоритми «підтвердження частки», «підтвердження повноважень» або «підтвердження вичерпаного часу») було запропоновано. Деякі криптовалюти та блокчейн-програми вже покладаються на ці альтернативи. Проте все ще потрібна ретельна оцінка кожного механізму та його енергетичного впливу та енергоефективності. Перехід на більш екологічні джерела енергії та розробка обчислень, які потребують менше енергії, є іншими варіантами, які слід. Також можна було б зменшити енергоспоживання біткоїна шляхом реструктуризації способу стимулювання обслуговування блокчейну.

Як і інші нові технології на основі ІКТ, блокчейн також викликає занепокоєння щодо електронних відходів (електронних відходів). Майнерам-конкурентам потрібне все більш ефективне обладнання для майнінгу, що призводить до швидкого старіння приблизно кожні 1,5 року. З моменту свого створення апаратне забезпечення для майнінгу біткоїнів вже перейшло від використання центральних процесорів до графічних процесорів, програмованих вентильних матриць та інтегральних схем для конкретних програм. Приблизні оцінки показують, що біткоїн створює 135 г електронних відходів за транзакцію, що в 30000 разів більше, ніж транзакція Visa [2].

15 вересня 2022 року сталася подія «The Merge», що означало фактично зміну механізму консенсусу в блокчейні Ethereum [3]. Злиття було об'єднанням початкового рівня виконання Ethereum (основної мережі, яка існувала з моменту створення) з його новим консенсусним рівнем підтвердження частки, Beacon Chain. Це усунуло потребу в енергоємному майнінгу, а натомість дозволило захистити мережу за допомогою



розставленого EТН. Це був справді захоплюючий крок у реалізації бачення Ethereum — більше масштабованості, безпеки та стійкості. Тому актуальність дослідження процесу переходу з одного механізму на інший є високою.

Протестувати систему глобально та порівняти результати досліджень не є можливим в масштабах одного дослідника, тому було прийнято рішення працювати з системою локально на машині, імітуючи реальне навантаження. Змінити значення параметрів мережі для знаходження найефективніших комбінацій може допомогти агентна архітектура.

З використанням показників продуктивності, які втілюють в собі критерії оцінки успішної поведінки агентів, можна визначити яким чином використовується обчислювальний ресурс, що в свою чергу призводить до споживання електроенергії.

### **3. ТЕХНІЧНІ ІНСТРУМЕНТИ ДЛЯ РЕАЛІЗАЦІЇ АГЕНТНОЇ АРХІТЕКТУРИ В МЕРЕЖІ БЛОКЧЕЙН**

Для того, щоб розгорнути локальну версію блокчейну були використані наступні технології: Puppeth, Vagrant, Nodejs [4-6].

Створення приватного блокчейну зараз не є тривіальним завданням. Щоб розгорнути мережу, потрібно багато часу та зусиль. Хоча з'являються сервіси, які значно спрощують це завдання, але ми хочемо контролювати приватну мережу та мати повну можливість запускати та підтримувати її всередині. Зробити це на 100% вручну було б надзвичайно трудомістким, якби не зручний інструмент від групи Ethereum під назвою puppeth.

Puppeth — це майстер CLI, який допомагає створити нову мережу Ethereum аж до генезису, завантажувальних вузлів, підписувачів, ethstats, крана, інформаційної панелі тощо, без клопоту, який зазвичай потребує налаштування всіх цих служб одну за одною. Puppeth використовує ssh для підключення до віддалених серверів і створює свої мережеві компоненти з контейнерів докерів за допомогою docker-compose. Користувач керується процесом за допомогою майстра командного рядка, який автоматично виконує важку роботу та конфігурує топологію.

Vagrant — це програмний продукт із відкритим вихідним кодом для створення та підтримки портативних віртуальних середовищ розробки програмного забезпечення, який допоможе нам з розташуванням відповідних вузлів мережі. Він використовує «Provisioners» і «Providers» як будівельні блоки для керування середовищами розробки. Провайдери — це інструменти, які дозволяють користувачам налаштовувати конфігурацію віртуальних середовищ. Puppeth і Chef є двома найпоширенішими провайдерами в екосистемі. Постачальники — це служби, які Vagrant використовує для налаштування та створення віртуальних середовищ. Підтримка віртуалізації VirtualBox, Hyper-V і Docker поставляється з Vagrant, тоді як VMware і AWS підтримуються через плагіни. Для контролювання і автоматизації процесу було обрано варіант написання свого агенту та агентної архітектури, використовуючи мову програмування Node.js.

### **4. АНАЛІЗ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ**

Вирішення задачі полягало в декількох етапах, тому опишемо їх детально.

Перший етап – створення десяти віртуальних машин, щоб запустити на них програму та додати в спільну мережу. Сконфігуровано та налаштовано порти, які будуть приймати і з яких будуть віддаватись команди та дані.

Другий етап полягає у створенні вузлів блокчейну Ethereum на кожній віртуальній машині. Для цього було застосовано майстра CLI Puppeth, який дав готовий інтерфейс.

Також програма згенерувала свої унікальні адреси в мережі. Коли програми були готові до отримання вказівок, а акаунти ініціалізовані – зроблено додавання їх у мережу.

Третій етап – створення такої ж мережі, але з proof-of-stake механізмом консенсусу. Для цього розподілення було 5 master вузлів та 5 slave вузлів.

Етап номер чотири заключає в собі написання інтелектуального агента, який наче ляльковод буде смикати за ниточки – відправляти запити на мережу, створювати блоки, після чого змінюючи параметри системи в залежності від результату (Рис. 1).

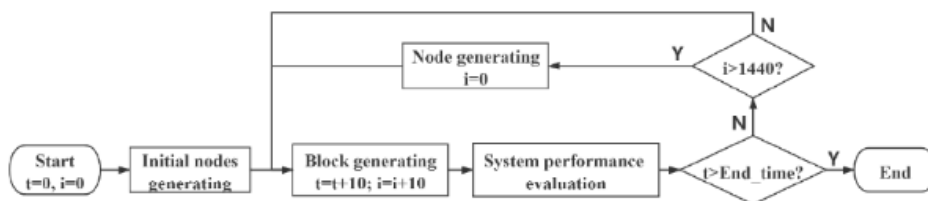


Рисунок 1. Структура агентної моделі

Для того, щоб результат був більш наочним – було додано ще огляд proof-of-authority механізму і сконфігуровано за прикладом proof-of-work.

Ефективність трьох консенсусних протоколів порівнюється на основі наступних параметрів: Transaction Cost and Gas Cost; Transactions Per Second (TPS); Electricity usage per transaction.

Зібрані метрики було оформлено у вигляді таблиць та графіків для наочності. Інструмент для моніторингу стану мережі Ethstats одразу показував кількість транзакцій на секунду для кожного алгоритму (Табл. 1). Також мануально було відображено графік порівняння Total Gas Cost, який показує ефективність того чи іншого підходу (Рис. 2).

Таблиця 1. Кількість транзакцій на секунду в залежності від механізму консенсусу

Механізм консенсусу	Кількість транзакцій на секунду
PoA	2380
PoW	395
PoS	4613

Фінальним порівнянням виступили результати метрик, отримання яких власне і ставилось за завдання: кількість витраченої електроенергії (Табл. 2).

Таким чином можемо сказати, що proof-of-stake концепт дійсно є значно енергоефективним та перехід на нього є доцільним, оскільки шкоди навколишньому середовищу завдається менше. Мінусом цього підходу є більша складність в реалізації та відсутність довготривалих випробувань в реальних умовах, тобто практично не доведено стійкість до зовнішніх або внутрішніх атак, лише описано теоретично.

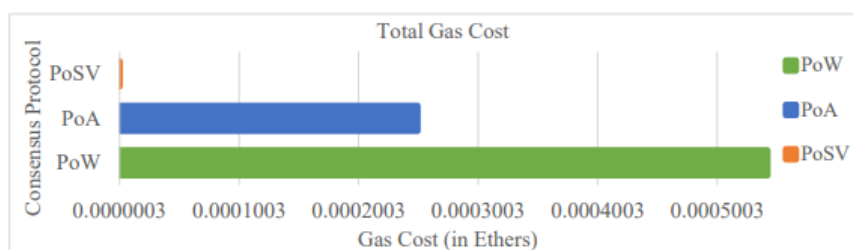


Рисунок 2. Порівняння Total Gas Cost кожного механізму консенсусу

Таблиця 2. Кількість електроенергії для проведення однієї транзакції в залежності від механізму консенсусу

Механізм консенсусу	Кількість використаної електроенергії (КВт-годин/транзакція)
PoA	1.1783
PoW	1.2321
PoS	0.0201

## 5. ВИСНОВКИ

Поставлена задача є досить складною, оскільки аналіз великою мережі за рахунок екстраполяції локальних результатів може містити похибку. Експеримент проводиться на одній машині, імітуючи багато з'єднань, але реальна мережа набагато більша, з більшими складовими як смарт контракти. Також на отримані результати впливає кількість вузлів та електронних гаманців у системі. Логічно, що реальні умови відрізняються від «лабораторних», але навіть з отриманих результатів можемо побачити різницю у споживанні ресурсів. Офіційні цифри, які наводить головний сайт Ethereum, зі зменшенням кількості витраченої електроенергії кажуть про скорочення витрат на 99.8% [3]. Тобто за словами розробників Ethereum, споживання має зменшитись у 50 разів. Отримані результати кажуть про зменшення споживання майже в 60 разів. Враховуючи похибку «стерильності» умов, можна зробити висновок, що офіційні цифри досить правдиві. Агентна архітектура показала себе як життєздатна для проведення таких експериментів, збору метрик та налаштування мережі, автоматизувавши роботу та значно зменшивши втручання людини, залишаючи їй лише порівняння кінцевих чисел.

## ПЕРЕЛІК ВИКРИСТАНИХ ДЖЕРЕЛ

1. Stuart J.Russel and Peter Norvig. Artificial Intelligence. A modern approach. Second edition. 2006. С. 75-440, 863-1010.
2. Mastercard [Електронний ресурс]. URL: <https://www.mastercard.com/news/perspectives/2021/why-mastercard-is-bringing-crypto-onto-our-network/>
3. Ethereum платформа [Електронний ресурс] – URL: <https://ethereum.org>
4. Official golang implementation of the Ethereum protocol [Електронний ресурс]. - URL: <https://github.com/puppeth>
5. Офіційна сторінка програмного забезпечення Vagrant [Електронний ресурс] – URL: <https://www.vagrantup.com/>
6. Офіційна сторінка мови програмування Node.js [Електронний ресурс] – URL: <https://nodejs.org/en/>

# ВИЯВЛЕННЯ АНОМАЛІЙ МЕТОДАМИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

Хомич О.Р.<sup>1</sup>, Недашківська Н.І.<sup>2</sup>

КПІ ім. Ігоря Сікорського, Київ, Україна

<sup>1</sup>khomych.alexander@iit.kpi.ua [0000-0002-1977-3973],

<sup>2</sup>nedashkovskaya.nadezhda@iit.kpi.ua [0000-0002-8277-3095]

**Проведено порівняльний аналіз результатів виявлення різних видів аномалій різними методами інтелектуального аналізу даних. Колективні аномалії описуються групою векторів даних, де кожен окремий вектор може бути допустимим. Контекстні аномалії часто описуються метаданими, ними можуть бути часові, просторові чи просторово-часові ознаки навколишнього середовища. Показано, що моделі на основі рекурентних нейронних мереж LSTM та гібридні з елементами LSTM більш точно виявляють контекстні, колективні аномалії, викиди у часових рядах порівняно з іншими методами інтелектуального аналізу даних.**

**Ключові слова:** розпізнавання аномалій, рекурентні нейронні мережі, LSTM, ансамбль нейронних мереж LSTM, штучний інтелект

## 1. ВСТУП

Аномалії створюють проблему в різних областях застосування. Вони часто призводять до зниження продуктивності системи і можуть викликати нестабільність і збої. Часто причинами аномалій є невідомі ефекти в складних системах. Завдяки прогресу досліджень у сфері машинного навчання останніми роками було запропоновано широкий спектр нових підходів до виявлення аномалій. Різні архітектури глибоких нейронних мереж, зокрема архітектури на основі довготривалої короткострокової пам'яті (LSTM), були розроблені і виявилися здатними вирішувати різноманітні складні задачі розпізнавання [1, 2].

На сьогоднішній день аномалії виявляють в основному методами теорії статистики, інформації, алгоритмами класифікації та кластеризації [1, 3]. Зокрема, для цієї задачі застосовують аналіз головних компонент (PCA), машини опорних векторів (SVM), алгоритм k найближчих сусідів (kNN), різні типи кореляційного аналізу, а також додаткові методи, такі як ковзні вікна, які поєднуються з вищезгаданими для виявлення змінних у часі аномалій.

Для вирішення проблеми розпізнавання складних контекстних аномалій з характеристиками, що змінюються в часі, в останні роки застосовують нові архітектури рекурентних нейронних мереж [3 – 8], нейронні мережі типу машин Больцмана та автокодувальників [6]. Глибокі нейронні мережі використано при виявленні аномалій у медицині [5], системах автоматизації виробництва [4,8], системах інтернету речей (IoT) [7].

Цю роботу присвячено дослідженню та порівняльному аналізу результатів розпізнавання різних видів аномалій різними методами інтелектуального аналізу даних.

## 2. КЛАСИФІКАЦІЯ АНОМАЛІЙ

Точкові аномалії (Рис. 1 ліворуч) можуть виявлятися методами статистики: для цільових параметрів рахуються значення щільності ймовірності та аналізуються процентилі.

Колективні аномалії характеризуються групою векторів даних, де кожен окремий вектор допустимий, але група цих векторів вказує на нерегулярність. Тому виявлення колективних аномалій базується на аналізі послідовностей даних. У [2] запропоновано метрику, яка об'єднує ступені відхилення кожного окремого вектору з групи, для випадку виявлення колективних аномалій у багатовимірних часових рядах. Якщо значення цієї сукупної метрики перевищує поріг, причому поріг змінюється з часом, то робиться висновок про присутність такої аномалії.

Контекстні аномалії (Рис. 1 праворуч) описуються як окремими векторами даних, не будучи при цьому точковими аномаліями, так і групами векторів даних. В другому випадку група вказує на нерегулярність у межах навколишніх векторів даних або груп векторів даних, які називаються контекстом. В цьому полягає відмінність від колективної аномалії.

Також, на відміну від колективних аномалій, які описуються своєю внутрішньою структурою або змістом, виявлення контекстних аномалій сильно залежить від короткострокових і довгострокових характеристик навколишніх зовнішніх структур даних [3]. У випадку багатовимірних часових рядів, контекстом вектору даних або групи векторів даних є об'єднання всіх навколишніх векторів даних, які лежать у визначеному часовому горизонті. Контекстні аномалії також часто описуються метаданими, які називають атрибутами аномалії [5, 6]. Ними можуть бути будь-які часові, просторові чи просторово-часові ознаки навколишнього середовища. У системах автоматизації виробництва атрибутами аномалії розглядають дані про виробничий процес і пов'язані з ним параметри процесу [4,8].

Оцінювання контекстної аномалії проводять на основі показників, що засновані на відстані, та методом ковзного вікна, коли значення відстані до попереднього вікна перераховується за кожною новою вибіркою даних [7]. Якщо показник перевищує динамічний (змінний у часі) поріг, робиться висновок про присутність контекстуальної аномалії.

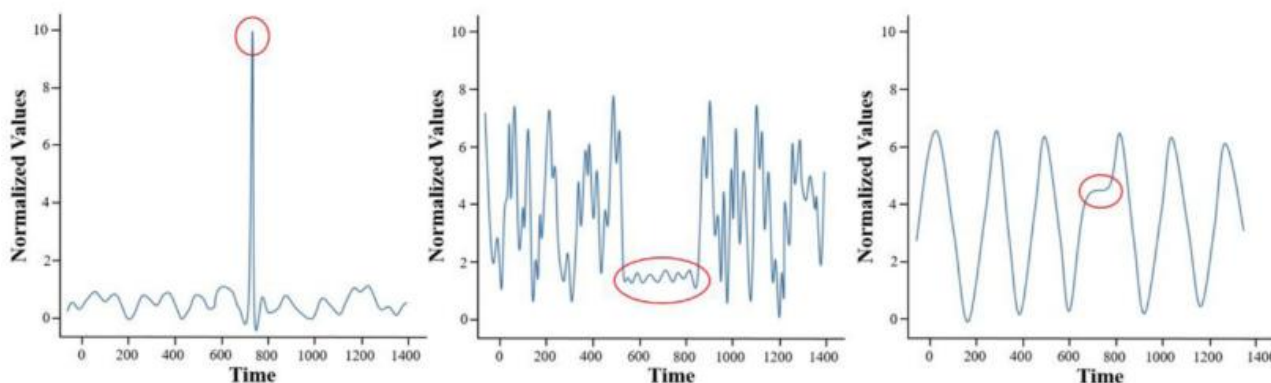


Рисунок 1. Точкова аномалія (ліворуч), колективна аномалія (посередині) і контекстна аномалія (праворуч) [9]

### 3. РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ ТИПУ LSTM

Складову мережі LSTM – комірку пам'яті (Рис. 2) – розроблено в [10] для вирішення проблеми зникаючого градієнту, яка виникає зі звичайними рекурентними мережами і призводить до того що частина ваг припиняє оновлюватися під час процесу навчання, в результаті стає важко вивчати довгострокові залежності. В комірці LSTM вектор стану розщеплюється на два вектори:  $h(t)$  – короткостроковий і  $z(t)$  – довгостроковий стани. Тому мережа LSTM може дізнатися яку інформацію слід зберігати в довгостроковому стані, а яку відкидати. Під час перетину мережі зліва направо довгостроковий стан  $z(t - 1)$  спочатку

проходить через шлюз забування forget gate, де відкидаються деякі спогади, і потім до нього додаються (у шлюзі add gate) нові спогади, що були обрані вхідним шлюзом input gate. Результат  $z(t)$  подається на вихід комірки без будь-яких подальших перетворень.

В комірці LSTM на кожному часовому кроці одні спогади відкидаються, а інші додаються. Ця процедура зумовлена обставиною, що жодна пам'ять не є нескінченною і що людська пам'ять як взірець також має обмежені можливості. Після операції додавання (add gate) довгостроковий стан  $z(t)$  копіюється і пропускається через функцію гіперболічного тангенсу, після чого результат фільтрується вихідним шлюзом. Остаточним результатом є короткостроковий стан  $h(t)$ , який дорівнює виходу комірки  $y(t)$  для даного часового кроку  $t$ .

Отже, стан комірки LSTM вміщує зменшене та зважене представлення історичної вхідної інформації. Останні введені дані не обов'язково домінують при створенні вихідних сигналів, оскільки при створенні виходу враховуються також важливі минулі події.

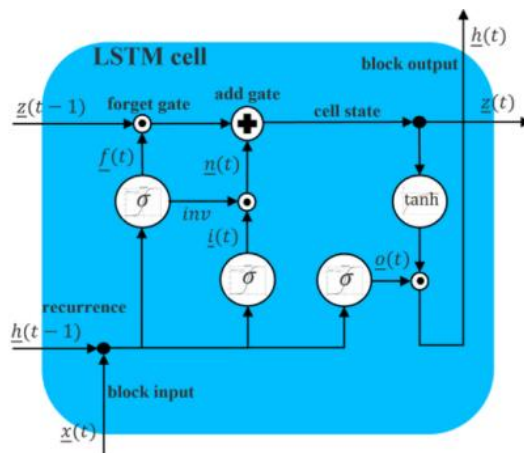


Рисунок 2. Архітектура комірки LSTM [10]

#### 4. ПОРІВНЯННЯ РЕЗУЛЬТАТІВ РОЗПІЗНАВАННЯ АНОМАЛІЙ РІЗНИХ ТИПІВ

Проведено наступні експерименти.

**Експеримент 1.** На вхід подавався багатовимірний часовий ряд без розмітки. Досліджувалися контекстні аномалії. В результаті ансамбль нейронних мереж LSTM [11] показав кращі результати за просту рекурентну нейронну мережу RNN за показниками точності (precision), повноти (recall) та F1-score.

**Експеримент 2.** На вхід подавався багатовимірний часовий ряд з розміткою на класи «нормальний» і «аномалія». Досліджувалися колективні та контекстні аномалії. Аналізувалася гібридна модель LSTM – SVM [12], яка складається з двох частин: 1) мережа LSTM – прогнозує динаміку системи, 2) модель опорних векторів (SVM) – використовується як класифікатор аномалій. Ця гібридна модель реалізує адаптивний і самонавчаючийся механізм виявлення. Модель LSTM – SVM показала кращі результати ніж традиційна SVM. Якість оцінювалася за показником AUC та ROC-кривою.

**Експеримент 3.** На вхід подавався одновимірний часовий ряд без розмітки. Досліджувалися колективні аномалії та викиди. В результаті дуальний LSTM [13] показав кращі результати ніж інші розглянуті методи за точністю (precision), повнотою (recall) та F1-score.

При наявності часового ряду з розміткою на класи, аналізувалися відхилення виходів, спрогнозованих мережею, від відомих реальних значень цільової змінної, і на основі цього приймалося рішення про присутність аномалії.

## 5. ВИСНОВКИ

Проведено порівняльний аналіз результатів розпізнавання різних видів аномалій різними методами інтелектуального аналізу даних. Розглянуто такі нові розробки як ансамбль нейронних мереж LSTM, гібридна модель LSTM – SVM, дуальний LSTM. Показано, що ці моделі на основі рекурентних нейронних мереж LSTM та гібридні з елементами LSTM більш точно виявили аномалії всіх розглянутих типів завдяки здатності моделей вивчати часові залежності та фіксувати їх у представленні стану низької розмірності, моделювати стаціонарні і нестаціонарні часові залежності.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ahmed M., Naser Mahmood A., Hu J. A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* 2016. Vol. 60. P. 19–31, <http://dx.doi.org/10.1016/j.jnca.2015.11.016>.
2. Akoglu L., Tong H., Koutra D. Graph based anomaly detection and description: a survey. *Data Min. Knowl. Disc.* 2015. Vol. 29. P. 626–688, <http://dx.doi.org/10.1007/s10618-014-0365-y>.
3. Bontemps L., Cao V.L., McDermott J., Le-Khac N.A. Collective anomaly detection based on long short term memory recurrent neural networks. *International Conference on Future Data and Security Engineering*. 2016. P. 141–152, [http://dx.doi.org/10.1007/978-3-319-48057-2\\_9](http://dx.doi.org/10.1007/978-3-319-48057-2_9).
4. Canizo M., Triguero I., Conde A., Onieva E. Multi-head CNN–RNN for multi time series anomaly detection: an industrial case study. *Neurocomputing*. 2019. Vol. 363. P. 246–260, <http://dx.doi.org/10.1016/j.neucom.2019.07.034>.
5. Chalapathy R., Chawla S. Deep Learning for Medical Anomaly Detection – A Survey. 2019. P. 1–50, arXiv:1901.03407.
6. Chandola V., Banerjee A., Kumar V. Deep learning for anomaly detection: a survey. *ACM Comput. Surv.* 2009. Vol. 41 (3). P. 1–58, <http://dx.doi.org/10.1145/1541880.1541882>.
7. Cook A.A., Misirlı G., Fan Z. Anomaly detection for IoT time-series data: a survey. *IEEE Internet Things J.* 2019. Vol. 7, no. 7. P. 6481–6494, <http://dx.doi.org/10.1109/JIOT.2019.2958185>.
8. Ding S., Morozov A., Vock S., Weyrich M., Janschek K. Model-based error detection for industrial automation systems using LSTM networks. *IMBSA: Model-Based Saf. Assess.* 2020. P. 212–226, [http://dx.doi.org/10.1007/978-3-030-58920-2\\_14](http://dx.doi.org/10.1007/978-3-030-58920-2_14).
9. Kim T.Y., Cho S.B. Web traffic anomaly detection using C-LSTM neural networks. *Expert Systems With Application.* 2018. Vol.106. P. 66–76, <http://dx.doi.org/10.1016/j.eswa.2018.04.004>.
10. Hochreiter S., Schmidhuber J. Long short-term memory. *Neural Comput.* 1997. Vol. 9 (no. 8). P. 1735–1780, <http://dx.doi.org/10.1162/neco.1997.9.8.1735>.
11. Malhotra P., Vig L., Shrof G., Agarwal P. Long short term memory networks for anomaly detection in time series. In: *Proceedings of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. 2015. P. 89–94.
12. Ergen T., Mirza A.H., Kozat S.S. Unsupervised and Semi-supervised Anomaly Detection with LSTM Networks. 2017. arXiv:1710.09207.
13. Lee M.-C., Lin J.-C., Gan E.G. ReRe: a lightweight real-time ready-to-go anomaly detection approach for time series. In: *2020 IEEE 4th Annual Computers, Software and Applications Conference (COMPSAC), Madrid, Spain.* 2020. P.322–327, <http://dx.doi.org/10.1109/COMPSAC48688.2020.0-226>.



# **КЛАСИФІКАЦІЯ ВАКАНСІЙ ВІДПОВІДНО ДО КЛАСИФІКАТОРА ПРОФЕСІЙ МЕТОДАМИ ШТУЧНОГО ІНТЕЛЕКТУ**

Цимбал Ю.О.<sup>1</sup>, Недашківська Н.І.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>tsymbal.julia@lil.kpi.ua, <sup>2</sup>nedashkovskaya.nadezhda@lil.kpi.ua [0000-0002-8277-3095]

**Сучасні способи пошуку роботи значно змінили ринок праці України за останні роки. Інформація, що зазвичай збирається держстатом, є доцільною, але неповною. Значна частина інформації про потреби роботодавців, запит конкретних професій та сфер діяльності не збирається на даний момент в Україні. Мета роботи полягає у створенні комп'ютерної системи для збору, підготовки та класифікації зібраних вакансій згідно до класифікатора професій. Розроблено гібридний підхід до обробки неструктурованих текстових даних щодо ринку праці за допомогою методів машинного навчання.**

**Ключові слова: ринок праці, онлайн сегмент ринку праці, великі дані, інтелектуальний аналіз, обробка природної мови, класифікація.**

## **1. ВСТУП**

Основним напрямком цієї роботи є застосування різних методів штучного інтелекту та обробки природної мови як потужного інструменту інтелектуального аналізу даних у демографічній сфері населення, а саме ринку праці. Розвиток економічних відносин стрімко зростає, відповідно і робочі місця масштабуються, змінюються вимоги до працівників та шукачів роботи, створюються нові професії та трансформуються існуючі. Наразі в демографічних дослідженнях в Україні науковці тільки починають використовувати різноманітні інструменти машинного навчання та штучного інтелекту, тому дана тема є дуже актуальною. Такий автоматизований метод аналізу ринку праці дасть змогу більш детально та прицільно досліджувати професійну сферу. Очищена та належним чином систематизована інформація онлайн джерел щодо вакансій статистично значуще доповнює наявну інформацію про офіційні вакансії Державного центру зайнятості, їх вимог щодо знань і навичок робочої сили, типу трудових контрактів, досвіду, трудових обов'язків та особистих якостей. Головна мета та ціль даного продукту – це можливість оцінити та аналізувати ринок праці в Україні у реальному часі та базуючись на реальних даних. Такий аналіз надасть змогу ширше та глибше зрозуміти ситуацію працевлаштування.

## **2. СТАН РИНКУ ПРАЦІ УКРАЇНИ ТА ЙОГО ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ**

Інформаційне забезпечення дослідників, управлінців та учасників ринку праці в Україні надається кількома ресурсами, зокрема державними та недержавними інституціями та нерегулярно міжнародними дослідниками. Переважна більшість даних щодо ринку праці збирається та агрегується спеціальними державними інституціями, а саме Державною службою статистики України (далі ДССУ) та Державним центром зайнятості (ДЦЗ).



ДССУ проводить свою діяльність згідно державного законодавства за усіма нормами та правилами ведення статистичних оцінок. Тому дані, що подаються як результат опитування та агрегації є достовірними та релевантними, надійними та достатньо точними. Статистична методологія публікується у відкритий доступ, тому з цими методами та підходами можна вільно ознайомитися. У сфері ринку праці ДССУ збирає дані підприємств та організацій щодо середньооблікової кількості штатних працівників, обсягів прийнятих та звільнених, розмір використаного та невикористаного робочого часу, обсяги вимушеної неповної зайнятості, середню заробітну плату, фонд оплати праці та його структуру, розмір заборгованості працівникам заробітної плати. Всі ці показники можуть бути кореспондованими між собою та представлені за видами економічної діяльності, регіонами, організаційно-правовими формами господарювання, формами зайнятості. Опубліковані дані обов'язково класифікуються згідно різних видів державних класифікаторів відповідно до тематики даних. Використовують зокрема класифікатор видів економічної діяльності (КВЕД) ДК 009:2010, організаційно-правових форм господарювання (КОПФГ) ДК 002:2004, кодифікатор адміністративно-територіальних одиниць та територіальних громад (КАТОТТГ) [1].

Отже, ДССУ забезпечує велику кількість коректно зібраних та оброблених даних щодо ринку праці по підприємствах та організаціях. Проте залишається відкрите питання щодо професійної кваліфікації працівників, кількості відкритих вакантних робочих місць на підприємствах та на ринку праці загалом.

Цю прогалину частково заповнює Державний центр зайнятості – установа, що веде перелік непрацевлаштованих громадян, допомагає їм підвищити кваліфікацію та набути потрібні професійні навички. Також ДЦЗ має відкритий інформаційний портал для публікації та поширення даних про вакантні робочі місця по Україні [2]. Державний центр зайнятості – популярний ресурс для розміщення пропозицій робочих місць роботодавцями та пошуку роботи безробітними. Основна перевага вищезгаданого порталу в тому, що дані про вакансії уніфіковані, структуровані та найголовніше назви публікуються згідно з Державним класифікатором професій. З іншого боку на цьому ресурсі можна знайти переважно прості професії, що не потребують спеціальних навичок та підготовки та мало привабливі професії.

В той самий час безліч запитів та пропозицій різнопланових робочих місць можна знайти на недержавних онлайн ресурсах пошуку роботи. Станом на сьогодні це єдине найвагомніше джерело інформації про актуальний стан ринку праці. Відповідно до цього без використання таких даних інформаційна картина українського ринку праці є неповною та викривленою.

Отже, провівши дослідження якості та способів інформування ринку праці України очевидним стає те, що наявні на сьогодні дані та підходи їх збору не дають можливості повно та цілковито оцінити стан ринку праці. Наразі ніяка структура чи державна організація не збирає та не надає повноцінно дані, що стосуються вільних робочих місць, запиту роботодавців, кількості шукачів роботи та найцікавіше – розподіл актуальних сфер діяльності та професій.

Збір даних з порталів онлайн пошуку роботи та класифікації згідно державного класифікатора професій дозволить побудувати на об'єднаних даних показники українського ринку праці, які точніше показують ринкову ситуацію, перспективи його розвитку. Крім того, забезпечення єдиної класифікаційної платформи для державних і недержавних даних дозволить доповнити знання експертів та державного регулювання новою інформацією щодо нових навичок і компетенцій, що стають актуальними на ринку праці. Це дозволить більш обґрунтовано формувати систему професійної підготовки та вищої освіти, планувати державне замовлення на фахівців, вчасно заповнювати прогалини дефіцитних спеціалістів.

### **3. ТЕХНІЧНІ ІНСТРУМЕНТИ ДЛЯ РЕАЛІЗАЦІЇ АЛГОРИТМУ КЛАСИФІКАЦІЇ ПРИРОДНОЇ МОВИ**

Обробка природної мови (Natural Language Processing, NLP) – перетин машинного навчання та математичної лінгвістики, спрямований на вивчення методів аналізу та синтезу природної мови. NLP дозволяє краще розуміти запити користувачів та аналізувати корпоративну інформацію. Використання даної технології гарантує, що у кожного користувача є доступ до найбільш актуальних, корисних джерел інформації, які б інакше залишилися прихованими у величезних обсягах даних.

Робота була сконцентрована на використанні методів та підходів, що походять з розділу лінгвістики – дистрибутивної семантики. Дистрибутивна семантика (distributional semantics) займається обчисленням ступеня семантичної близькості між лінгвістичними одиницями на основі їх розподілу у великих масивах лінгвістичних даних або так званих текстових корпусах [3]. Одне з основних понять – розподільна семантична модель (distributional semantic model - DSM) – це масштабована та/або трансформована матриця спільного входження  $M$ , у якій кожен рядок  $x$  представляє розподіл цільового терміна в різних контекстах. Головний принцип використання методів дистрибутивної семантики полягає у тому, що кожному слову надається свій контекстний вектор. Множина векторів формує словесний векторний простір. Як спосіб представлення моделі використовуються векторні простори з лінійної алгебри. Інформація про дистрибуцію лінгвістичних одиниць подається у вигляді багаторозрядних векторів, які утворюють векторний словесний простір. Вектори відповідають лінгвістичним одиницям (словам чи словосполученням), а виміри – контекстам. Координати векторів – це числа, що показують скільки разів слово або словосполучення зустрілося в даному контексті. Семантична відстань між поняттями, вираженими словами природної мови, зазвичай обчислюється як косинусна відстань між векторами словесного простору [4]. Після проведення такого аналізу виявляють найближчі за змістом слова стосовно досліджуваного слова.

У обробці природної мови (NLP) word embedding – термін, який використовується для представлення слів для аналізу тексту, як правило, у формі дійсного вектора, який кодує значення слова таким чином, що слова, розташовані ближче у векторному просторі, є подібними за значенням. Тобто word embedding – це техніка отримання DSM-вектору.

Найпопулярніший алгоритм обробки природної мови – word2vec [5, 6]. Він використовує модель нейронної мережі для вивчення асоціацій слів із великого корпусу тексту. Після навчання така модель може виявляти слова-синоніми або пропонувати додаткові слова для окремого речення. Модель word2vec представляє кожне окреме слово певним вектором, який підбирається, щоб відображати семантичні та синтаксичні якості слів. Рівень семантичної подібності між словами, представленими цими векторами, зазвичай визначається на основі косинусної відстані. У word2vec представлено дві модельні архітектури для створення цих розподілених представлень слів: модель неперервного пакету слів (Continuous Bag Of Words, CBOW) та неперервного скіп-граму (Skip-gram) [5]. В обох цих архітектурах виконується ітераційне проходження по всьому корпусу тексту і розглядаються як окремі слова, так і ковзне вікно контекстних слів, що оточують обрані слова.

### **4. РОЗРОБКА АЛГОРИТМУ КЛАСИФІКАЦІЇ ТА РЕЗУЛЬТАТИ**

В роботі пропонується комплексний метод вирішення поставленої задачі, який складається з кількох етапів. Перший етап – збір даних. Наразі немає спільної бази даних

вакансій українського ринку праці. Кожен приватний інтернет-портал з пошуку роботи має свою базу, яка не узгоджена з іншими аналогічними посередниками пошуку роботи. Тому перший крок пропонованого методу – це визначити як можна зібрати дані з кількох онлайн джерел, а також встановити критерії вибору таких інтернет-порталів. Критерії необхідні для того, щоб зменшити розбіжність зібраних даних, оскільки кожен вебсайт має свою власну структуру та види, і формат опублікованої інформації відрізняється.

Другий крок – провести аналіз можливих методів збору інформації з вебресурсів та обрати підхід, який буде більш універсальним та не потребуватиме великих витрат: часу, знань, фінансів. Базуючись на цих критеріях та беручи до уваги особливості онлайн порталів було обрано методологію вебскрепінгу. Вебскрепінг використовується для отримання тексту з веб-сторінок. Програмне забезпечення для сканування веб-сторінок розробляється для розпізнавання різних типів вмісту на веб-сайті та для отримання та зберігання лише типів вмісту, вказаних користувачем, наприклад назви статей чи авторів із веб-сайту новин або ціни й описи продуктів із комерційного веб-сайту. Можна використовувати комерційне програмне забезпечення або мови програмування. Для цієї роботи був розроблений програмний продукт для збору даних, написаний на мові програмування Python та з використанням бібліотеки Selenium для імітування дії користувача у мережі. Особливість розробленої програми в тому, що алгоритм збору даних універсальний, але практична реалізація повинна коригуватися відповідно до вебресурсу. Це зумовлено тим, що кожен онлайн портал пошуку роботи зроблено за різними архітектурами та з використанням різних веб елементів. У результаті застосування розробленого інструменту вдалося зібрати дані щодо актуальних вакансій.

Третій етап – підготовка даних. Підготовка або препроцесинг – важлива та невід’ємна складова будь-якого циклу розробки програмного забезпечення. Від якості даних залежить результат класифікації чи прогнозу, відповідно до задачі. Для зібраного набору даних було використано усі основні елементи препроцесингу тестової інформації: вилучення спеціальних символів та стоп-слів української мови, вилучення пустих або незадовільно заповнених даних, та сортування даних згідно мови. Для публікації вакансій в Україні часто використовується не тільки українська мова, але й англійська та російська. Аналогічної підготовки потребує класифікатор професій, згідно з яким необхідно розділити зібрані вакансії.

Четвертий етап – розробка алгоритму класифікації вакансій. Для класифікації професій розроблено гібридний підхід з метою покращення ефективності класифікації та зменшення втручання людини. Використано різні методології автоматизації та штучного інтелекту, що включають методи обробки природної мови. Створено систему, яка використовує вказаний гібридний підхід для поєднання потужності онтологій і словників із гнучкістю алгоритмів машинного навчання (Рис. 1).



Таблиця 1. Відсоткове співвідношення співпадінь назви вакансії та назви професії в класифікаторі згідно різних методів перевірки

№	Метод визначення семантичної подібності	Відсоткове відношення співпадінь
1	Exact match	12%
2	Direct match	65%
3	Regular expression match	3%
4	FastText match (unsupervised)	16%

Отримані результати показують, що значну частину вакансій можна класифікувати різними комбінаціями порівнянь назв, але в будь-якому разі залишаються назви вакансій, які не узгоджені з класифікатором професій. Саме тому є сенс використовувати методи машинного навчання для досягнення повноцінного результату. Надалі та частина класифікованих вакансій може бути використана як навчальна вибірка для методів навчання з учителем. Це розширить спектр алгоритмів, які можна застосувати, та дасть змогу порівняти результати. Наприклад, навчання без учителя за алгоритмом FastText знаходить смислові схожості, використовуючи word embedding та обчислення косинусної відстані між закодованими словами. Таким чином є можливість встановити схожість між частинами тексту, навіть написані різним лексиконом.

## 5. ВИСНОВКИ

Аналіз світового досвіду засвідчує можливість використання великих даних для дослідження в реальному часі широкого кола проблем функціонування ринку праці: від оцінки загальної онлайн пропозиції робочих місць і їх класифікації за професіями та видами діяльності, до більш специфічних питань таксономії навичок, виявлення нових професій та розробки модельних траєкторій побудови кар'єри.

Загалом складається враження, що інформування зацікавлених осіб про український ринок праці є досить насиченим і різноманітним, торкається різних аспектів та відбувається досить регулярно. Однак, аналіз згаданих джерел щодо переліку та формату даних дає підстави для висновку, що дані цих джерел мало узгоджуються між собою, часто надають неактуальну інформацію, багато в чому дублюються, в той час як важливі моменти ринку праці залишаються без відповіді. В результаті, сьогодні неможливо сформулювати єдину і не суперечливу картину українського ринку праці, якою б могли одночасно користуватися державні структури, дослідники, населення та підприємства. Формування єдиної інформаційної системи ринку праці, де усі могли б знайти актуальну для себе інформацію, вимагає значних методичних та організаційних зусиль, подолання численних інституційних бар'єрів, узгодження методології збору та обробки даних, їх таксономії, а також застосування нових наукових інформаційних методів та технологій.

В цій роботі розроблено систему класифікації вакансій згідно класифікатора професій, яка базується на запропонованому гібридному підході. Така система наразі дає можливість регулярно, комплексно і в реальному часі отримувати актуальну інформацію щодо ринку праці в Україні.

Пропонований алгоритм класифікації включає етапи збору власних актуальних даних, що задовольняють умовам поставленого завдання, підготовки та попередньої обробки цих даних, моделі онтологій та методи машинного навчання для отримання фінального

результату – присвоєння кожній назві вакансії мітки класу згідно класифікатора. Складність цієї роботи полягає у тому, що, по-перше, немає готового набору даних, а отже необхідно попередньо збирати дані на різних онлайн-ресурсах та агрегувати. По-друге, немає розмічених даних, тому потрібно було створювати їх самостійно. По-третє, реальна оцінка результату класифікації може бути здійснена тільки експертом.

### **ПЕРЕЛІК ВИКРИСТАНИХ ДЖЕРЕЛ**

1. Державна служба статистики України. URL: <https://ukrstat.gov.ua/> (дата звернення 07.10.2022)
2. Державна служба зайнятості. URL: <https://www.dcz.gov.ua/> (дата звернення 07.10.2022)
3. Шарнін М. М., Сомін Н. В., Кузнецов І. П., Морозова Ю. І., Галина І. В., Козеренко Є. Б. Статистичні механізми формування асоціативних портретів предметних областей на основі природно-мовних текстів великих обсягів для систем здобуття знань. Інформатика та її застосування: журнал. - 2013. - Т. 7, вип. 2. - С. 92–99.
4. Schutze Н. Dimensions of meaning Proceedings of Supercomputing'92. — 1992. — С. 787—796.
5. Mikolov, Tomas. Efficient Estimation of Word Representations in Vector Space, 2013.
6. Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning. The MIT Press Cambridge, Massachusetts London, England, 2017.

# ІДЕНТИФІКАЦІЯ ПСИХІЧНИХ ЗАХВОРЮВАНЬ МЕТОДАМИ ОБРОБКИ ПРИРОДНОЇ МОВИ НА ОСНОВІ ДОПИСІВ КОРИСТУВАЧІВ У СОЦМЕРЕЖАХ

Цупрун І.Ю.<sup>1</sup>, Стусь О.В.<sup>2</sup>

<sup>1,2</sup> Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>itsuprik@gmail.com, <sup>2</sup>o.stus@kpi.ua

**Методи обробки природної мови в останні роки отримали великий поштовх для розвитку та на сьогоднішній день, їх використання призвело до приголомшливих досягнень у різних сферах, від медицини до науки та розваг. Метою даної роботи є створення системи для ідентифікації психічних захворювань на основі дописів користувачів у соцмережах, тобто вирішити задачу класифікації текстів у вказаній предметній області. У роботі було описано та використано методи необхідні для розв'язання задачі класифікації тексту, а також оглянуто вже існуючі дослідження, що стосуються даної теми.**

**Ключові слова:** обробка природної мови, тексти, психічні захворювання, класифікація, ідентифікація, соцмережі.

## 1. ВСТУП

У сучасному світі, де розвиток технологій біжить нелюдськими темпами, все більше і більше людей страждає від психічних захворювань. Так, за даними Всесвітньої Організації Охорони Здоров'я (далі ВООЗ), у 2019 р. кожна восьма людина планети, тобто. загалом 970 мільйонів чоловік, страждають на психічні розлади, причому найбільш поширеними є тривожні та депресивні розлади. У 2020 р. на тлі пандемії COVID-19 кількість людей, які страждають на тривожні та депресивні розлади, значно зросла. За попередніми оцінками лише за рік поширеність тривожних і серйозних депресивних розладів збільшилася на 26% та 28% відповідно. Незважаючи на наявність ефективних методів профілактики та лікування, більшість людей із психічними розладами не мають доступу до ефективної медичної допомоги. Багато хто з них також стикається зі стигматизацією, дискримінацією та порушенням прав людини. Крім того, є люди які бояться звернутись по допомогу або просто не розуміють що потребують її. [1]

Метою даної роботи є дослідження методів машинного навчання, які допоможуть створити систему для ідентифікації психічних розладів на основі публікацій людини в соцмережах та повідомлень у месенджерах. Таким чином, після виявлення того чи іншого психічного захворювання система матиме можливість у не нав'язливій формі запропонувати людині допомогу, що дуже важливо з огляду на страх людей звертатись по допомогу та ділитись своїми проблемами.

Об'єктом дослідження стали дописи користувачів що мають психічні розлади у соцмережах. Предметом дослідження є підходи та методи вирішення задачі класифікації текстів.

## 2. МЕТОДИ ОБРОБКИ ПРИРОДНОЇ МОВИ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ ТЕКСТІВ

Класифікація тексту - це процес розуміння сенсу неструктурованого тексту та організації його за задалегідь визначеними категоріями (тегами). Одним з найпопулярніших завдань класифікації тексту є аналіз настроїв, який має на меті класифікувати неструктуровані дані за настроями. Інші завдання класифікації включають виявлення намірів, моделювання тем і виявлення мови. В даній роботі вирішується задача відношення тексту до мітки того чи іншого психічного розладу. Всього було розглянуто 8 класів, які відповідають восьми основним психічних розладам що виділяє ВООЗ: тривожні розлади, депресія, біполярні розлади, посттравматичні стресові розлади, шизофренія, розлади харчової поведінки, асоціальна поведінка та дисоційовані розлади, порушення розвитку центральної нервової системи (аутизм).

Процес вирішення будь-якої задачі класифікації текстів зазвичай складається з трьох основних етапів: попередня обробка тексту, векторизація слів, вирішення задачі класифікації.

Попередня обробка даних - це заключний етап збору та підготовки даних, який включає в себе трансформацію даних у доступний для розуміння формат. В обробці природної мови виділяють такі етапи попередньої обробки даних: чистка даних, зміна регістру, токенизація, видалення стоп-слів, стематизація, лематизація, N-грами. [2]

Багато алгоритмів машинного навчання і майже всі моделі глибокого навчання не здатні обробити простий текст в сирому вигляді. Для виконання будь-яких завдань, таких як класифікація, регресія, кластеризація і т.д., їм потрібні числові дані в якості вхідних даних. Крім того, з величезної кількості даних, які присутні в текстовому форматі, необхідно витягти якісь знання. Для перетворення текстових даних у числові були вигадані методи векторизації тексту.

Методи векторизації можна поділити на дві категорії: ті, що працюють на базі частоти та ті, що враховують контекст в якому вжито слово. Найкращим частотним методом вважається векторизація на базі метрики TF-IDF. В методі TF-IDF генерується матриця токенів документа, кожен стовпець якої представляє окреме унікальне слово. Кожна комірка не вказує частоту токена, а містить значення ваги, яке означає, наскільки важливим є слово для окремого текстового повідомлення або документа. Крім того, цей метод відрізняється від інших тим, що враховує не тільки входження слова в окремому документі, але і в усьому корпусі та надає більшої ваги подіям, що трапляються рідше, і меншої ваги очікуваним подіям. TF-IDF слова вказує на те, як часто лексема зустрічається в документі і наскільки унікальною є ця лексема для всього корпусу документів. [3]

Найбільш відомим способом контекстної векторизації став метод word2vec відкритий у 2013 році Томашом Міколовим. Ідея метода word2vec полягає у визначенні слова за його контекстом або контексту який відповідає слову за допомогою використання нейронних мереж. Ці два протилежні принципи отримали назви Skip Gram та Continuous Bag Of Words (CBOW) відповідно. Модель CBOW використовує контекст кожного слова як вхідні дані і намагається передбачити слово, що відповідає контексту. Розглянемо як приклад речення "Бажаю чудового дня". Нехай на вхід нейронної мережі подається слово "чудовий". У такому випадку, в якості цільового слова для нейронної мережі буде слово "день", використовуючи єдине контекстне вхідне слово "чудовий". Якщо бути точніше, на вхід в мережу подається вектор слова "чудовий" закодований за допомогою частотного методу векторизації. В процесі прогнозування цільового слова буде отримано векторне представлення цільового слова [4].



Алгоритми класифікації з використанням машинного навчання є останнім етапом у вирішенні задачі класифікації тексту, після попередньої обробки даних та векторизації, тобто приведення текстових даних до числових векторів. Після того, як було сформоване векторне представлення всіх розмічених текстових документів, їх можна використовувати для навчання класифікатора. Векторне представлення текстових документів передається класифікатору з правильними категоріями. Модель вивчає асоціації між різними лінгвістичними ознаками в тексті та категоріями.

Вибір моделі класифікації залежить від розміру корпусу та кількості слів у документах текстів. Зазвичай, вибір моделі повинен бути заснованим на відповідях на наступні питання: "Як представлені текстові дані алгоритму, який очікує на числовий вхід?", "Який тип моделі має бути використаним?", "Які параметри конфігурації мають бути використані для моделі?". На щастя, завдяки десятиліттям досліджень, на сьогоднішній день є доступ до великої кількості варіантів попередньої обробки даних і конфігурації моделі. Найкраще такі дані зведені у блок-схемі від компанії Google, яка допомагає звизити вибір моделей для експериментів базуючись на згаданих вище питаннях. Якщо відношення кількості документів до середньої кількості слів у документі менше за 1500, то рекомендується та є більш ефективним використання звичайних алгоритмів машинного навчання, таких як метод опорних векторів або градієнтний бустинг, та частотної векторизації текстів для підготовки даних, наприклад, TF-IDF. В іншому випадку більш ефективним буде використання моделі глибокого навчання, наприклад згорткової нейронної мережі та векторизації контекстним методом, наприклад за допомогою мережі Skim Gram.

### 3. ДОСЛІДЖЕННЯ ТА РЕЗУЛЬТАТИ

Для дослідження були зібрані публікації людей у спільнотах пов'язаних з психічним здоров'ям в соціальній мережі Reddit. Її особливістю, яка задовольняє потреби даного дослідження є те, що вона налічує безліч тематичних каналів (англ. subreddit) в яких люди діляться своїми думками, проблемами, почуттями згідно обраної теми. Головне припущення, яке було використано для збору даних можна сформулювати наступним чином: людина, яка хвора на певне психічне захворювання, пише пост у канал, в якому люди діляться як жити з цим психічним захворюванням. Якщо людина ділиться тим, що її також турбує ця проблема - всі її пости позначаються міткою цього захворювання. Всього було зібрано 740 345 публікації, всі англійською мовою.

Після збору даних кожен було виконано попередню обробку даних. Дописи поганих користувачів, тобто такі що є спамом, флудом або заблокованих користувачів - було видалено. Що стосується обробки тексту публікації - були видалені непотрібні знаки пунктуації та пробіли для кожного допису, символи приведені до нижнього регістру. Після цього був використаний набір інструментів природної мови (NLTK), реалізований на Python, щоб токенізувати дописи користувачів і фільтрувати часто вживані слова (стоп-слова). Для приведення токенізованих слів до їх кореневої форми було використано алгоритм Портера Стеммера, який застосовує послідовний ряд правил, що відсікає закінчення та суфікси та не містить бази слів, тому спрацьовує швидко, але не завжди безпомилково. Після цієї процедури для аналізу були використані дані 130 060 користувачів із 588 472 публікаціями. Фінальним етапом попередньої обробки стала розмітка даних для бінарної класифікації, тобто у відповідність кожному тексту була поставлена мітка 1 або 0 для кожного психічного розладу.

Набір даних було розбито на навчальну та тренувальну вибірку у відношенні 80% та 20% відповідно. Для вирішення задачі класифікації було порівняно два алгоритми:

класифікація за допомогою градієнтного бустингу та та класифікація за допомогою згорткових нейронних мереж (CNN).

Для класифікатора XGBoost було використано векторизатор на основі метрики TF-IDF для перетворення слів у n-вимірні вектори, тобто частотний метод. У випадку згорткової нейронної мережі було застосовано контекстний метод векторизації, а саме модель Skim Gram парадигми word2vec.

Рисунок 1 демонструє архітектуру нейронної мережі, що була розроблена під час дослідження. Архітектура моделі організована послідовністю шарів, що включає шар векторизації, декілька шарів згортки (Convolutional), шар максимального об'єднання (Max pooling), декілька щільних шарів (Dense), що представляють повнозв'язну нейронну мережу та вихідний шар з функцією активації сигмоїд для бінарної класифікації. Для навчання нейронної мережі було використано бінарну функцію втрат - перехресну кросенторпію та оптимізатор Адам. Гіперпараметр швидкості навчання було встановлено на рівні 0,001. Модель навчалася протягом 50 епох, а розмір батчу був встановлений на рівні 64 екземпляри.

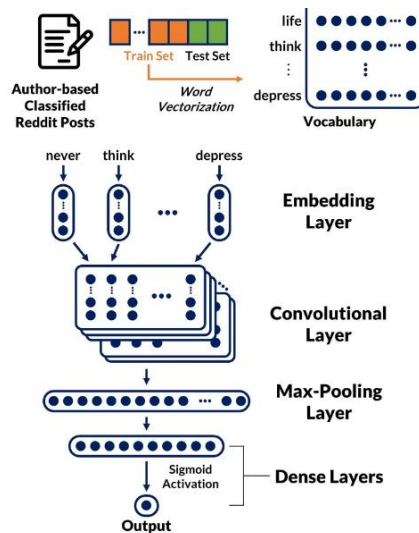


Рисунок 1. Архітектура згорткової нейронної мережі

Для перевірки ефективності моделей було використано чотири метрики оцінки: точність (accuracy), влучність (precision), повнота (recall) та міра F1 (F1-Score). TP, FN, TN та FP означають відповідно істинно-позитивний, хибно-негативний, істинно-негативний та хибно-позитивний результати.

Рисунки 2 та 3 підсумовують результати роботи восьми моделей бінарної класифікації. Серед восьми різних підкласів аутизм показав найвищу точність (96,96%) у згортковій нейронній мережі, але мав найнижчий показник F1 (XGBoost: 38,31%, CNN: 48,73%), що пов'язано з проблемою дисбалансу класів. Загалом, моделі CNN показали вищу точність, ніж моделі XGBoost, за всіма показниками. Для класу депресія були отримані найвищі показники точності (89,10%), відгуку (71,75%) та оцінки F1 (79,49%). Це пов'язано з тим, що він є найбільш збалансованим. Інші підкласи також показали високу точність з моделями CNN, а їхній показник F1 у визначенні психічних захворювань коливалися від сорока до п'ятдесяти (%), що є відносно нижчими, ніж у збалансованих за класами каналах. Таким чином, запропонована модель може точно виявляти потенційних користувачів, які можуть мати психічні розлади. Для покращення результатів необхідно провести додатковий збір даних та вирішити проблему незбалансованості даних. Це призведе до кращої продуктивності моделей.

Психічний розлад	Клас	XGBoost класифікатор	
		F1-Score	Точність (%)
депресія	1	58.02	71.69
	0	78.65	
тривожні розлади	1	55.92	70.41
	0	77.73	
біполярні розлади	1	53.59	85.53
	0	91.43	
посттравматичні розлади	1	46.43	85.14
	0	91.37	
шизофренія	1	40.97	86.72
	0	92.52	
аутизм	1	38.31	94.91
	0	97.35	
асоціальна поведінка	1	52.74	84.13
	0	90.34	
розлади харчової поведінки	1	34.25	92.44
	0	94.43	

Рисунок 2. Результати класифікації методом XgBoost

Психічний розлад	Клас	Згортова нейронна мережа CNN			
		Precision	Recall	F1-Score	Точність (%)
депресія	1	89.1	71.75	79.49	75.13
	0	58.66	82.04	68.41	
тривожні розлади	1	87.54	41.44	56.25	77.81
	0	75.92	96.91	85.14	
біполярні розлади	1	87.22	38.02	52.95	90.2
	0	90.4	99.05	94.53	
посттравматичні розлади	1	91.84	32.69	48.21	90.49
	0	90.42	99.54	94.76	
шизофренія	1	81.16	24.87	38.07	94.33
	0	94.62	99.56	97.03	
аутизм	1	48.08	49.39	48.73	96.96
	0	98.48	98.4	98.44	
асоціальна поведінка	1	86.17	36.52	53.25	90.1
	0	91.4	98.45	93.62	
розлади харчової поведінки	1	45.34	47.43	47.38	95.32
	0	97.32	96.65	96.23	

Рисунок 3. Результат класифікації за допомогою згорткових нейронних мереж.

## 4. ВИСНОВКИ

Методи обробки природної мови в останні роки отримали великий поштовх для розвитку та на сьогоднішній день, їх використання призвело до приголомшливих досягнень у різних сферах, від медицини до науки та розваг.

Однією з найбільш актуальних задач обробки природної мови є класифікації тексту. Основними етапами для її розв'язку є попередня обробка текстових даних, векторизація текстових даних та вирішення класичної задачі класифікації.

У ході дослідження було розроблено та порівняно вісім моделей машинного навчання для бінарної класифікації дописів у соцмережах з метою ідентифікувати психічні захворювання. Виконано порівняння ефективності звичайних алгоритмів класифікації та згорткових нейронних мереж. Дійшли до висновку, що згорткові нейронні мережі краще розв'язують поставлену задачу. Проведено тренування гіперпараметрів згорткової моделі. Найкращі результати для кожного з розладів було обгорнуто програмним інтерфейсом з метою надати користувачу можливість використовувати модель на практиці.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Institute of Health Metrics and Evaluation. Global Health Data Exchange (GHDx). URL: <https://vizhub.healthdata.org/gbd-results/> (дата звернення: 10.08.2022).
2. Amal Menzli. Tokenization in NLP: Types, Challenges, Examples, Tools. URL: <https://neptune.ai/blog/tokenization-in-nlp/> (дата звернення: 14.10.2022).
3. Chirag Goyal. Step by Step Guide to Master NLP – Word Embedding and Text Vectorization. URL: <https://www.analyticsvidhya.com/blog/2021/06/part-5-step-by-step-guide-to-master-nlp-text-vectorization-approaches/> (дата звернення: 10.09.2022).
4. Tomas Mikolov. Distributed Representations of Words and Phrases and their Compositionality. URL: <https://arxiv.org/pdf/1310.4546.pdf> (дата звернення: 20.10.2022).

# ГІБРИДНА НЕЙРОННА МЕРЕЖА ДЛЯ ВИЯВЛЕННЯ І ВІДСТЕЖЕННЯ ЛЮДИНИ У ВІДЕОПОТОЦІ

Шаповал Н.В.<sup>1</sup>, Русакова Л.О.<sup>1</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>shovgun@gmail.com, <sup>2</sup>rusakova.larisa@iit.kpi.ua

**Аналіз відеоконтенту (Video content analysis, VCA) або відеоаналітика (VA) на сьогоднішній день використовується у охоронних системах, роздрібній торгівлі, автомобільній промисловості, технологіях розумного будинку тощо. До задач відеоаналітики відносяться виявлення і відстеження об'єктів. Метою дослідження є підвищення ефективності виявлення і відстеження людини у відеопослідовностях. Було використано емпіричні та теоретичні методи досліджень. В результаті розроблено гібридну нейронну мережу для виявлення і відстеження людини: удосконалено згорткову нейромережу типу YOLO.**

**Ключові слова: виявлення об'єктів, відстеження об'єктів, згорткова нейронна мережа, YOLO, HOG.**

## 1. ВСТУП

На сьогоднішній день системи відеоспостереження використовуються майже у всіх сферах людської діяльності. Системи відеоаналітики надають можливість автоматичного аналізу відео для виявлення та визначення часових і просторових подій [1]. До них відносяться системи розпізнавання і відстеження осіб, що застосовуються у автоматичному керуванні для виявлення пішоходів на дорогах, у магазинах для підрахунку кількості та аналізу поведінки клієнтів, на зупинках громадського транспорту і в аеропортах для аналізу пасажиропотоку, у офісних приміщеннях і системах розумного будинку тощо.

Програмні підходи до виявлення і відстеження людини досить різноманітні, створюються спеціальні застосунки і веб-додатки чи окремі програмні модулі. Останнім часом широкого поширення набули нейронні мережі. Але при такому підході можуть виникнути проблеми з виявленням дрібних чи неповних об'єктів, стійкістю до зашумлених чи погано освітлених зображень. Багато часу займає навчання мереж як і сам процес виявлення. Деякі мережі обробляють відео зі швидкістю менше 10 кадрів за секунду що не прийнятно для застосування у режимі реального часу. Тому виникає необхідність створення ефективного алгоритму виявлення і відстеження людини, який дозволить значно точніше і швидше обробляти відеокадри.

## 2. МЕТОДИ ВИЯВЛЕННЯ І ВІСТЕЖЕННЯ ЛЮДИНИ

Існує багато різних методів виявлення об'єктів на зображеннях. Відео є складнішим за зображення, оскільки воно має ще один вимір – час. Його можна представити у вигляді серії (набору) зображень. І для його обробки достатньо прокрутити всі кадри у відеофайлі, застосувати відповідні методи розпізнавання об'єктів для кожного із кадрів. Але оскільки застосування одного і того самого алгоритму кожного разу не є ефективним з точки зору обчислень і не гарантує, що певний виділений об'єкт буде знайдено на кожному з кадрів (не

всі детектори є стійкими до різних поз та ракурсів), додатково використовуються алгоритми відстеження об'єктів.

Значного успіху в задачі виявлення об'єктів досягнув метод Віоли-Джонса [2], але він навчається повільніше у порівнянні з іншими існуючими методами, оскільки використовує значну кількість ознак. Гістограма орієнтованих градієнтів (Histogram of Oriented Gradients, HOG) працює з локальними комірками [3]. Тому він інваріантний до геометричних і фотометричних перетворень об'єкта. Але все одно традиційні методи виявлення об'єктів краще підходять для задач, де набір тренувальних прикладів невеликий. А для роботи з відеопослідовностями краще застосовувати згорткові нейронні мережі.

Найважливішою особливістю нейронних мереж, яка дозволяє успішно використовувати їх у різноманітних задачах (кластеризація, класифікація, розпізнавання образів, прогнозування і т. д.), полягає у паралельній обробці інформації усіма ланками, що дозволяє значно прискорити процес обробки інформації. Інша, не менш важлива властивість, – здатність до навчання. Згорткові нейронні мережі ефективно застосовуються у задачі розпізнавання об'єктів у режимі реального часу завдяки можливості до узагальнення накопичених знань і врахуванню інформації про співвідношення частин зображення між собою.

Згорткові мережі, які застосовуються для виявлення об'єктів базуються на двох підходах – одно і двоступеневому виявленні об'єктів. Двоступеневі детектори спочатку визначають регіони (області), у яких можуть знаходитись об'єкти, а потім шукають їх саме у цих регіонах. Такі мережі мають високу точність розпізнавання. Двоступеневими детекторами є мережа R-CNN (Region-based Convolutional Neural Networks) та її похідні (Fast R-CNN, Faster R-CNN, Mask R-CNN, Mesh R-CNN). А одноступеневі детектори, що одночасно виконують локалізацію та класифікацію об'єктів у всіх частинах зображення за один прохід, демонструють високу швидкість передбачення (висновку). Серед прикладів можна навести YOLO, SSD і RetinaNet, а також CenterNet, CornerNet та FCOS, що не використовують прив'язки.

В рамках даного дослідження було проведено порівняльний аналіз мереж Faster R-CNN, YOLO, SSD, RetinaNet з метою оцінки точності й швидкості виявлення об'єктів. Для цього використано датасет Human-Aligned Bounding Boxes from Overhead Fisheye cameras (HABBOF) [4]. Розроблений в лабораторії обробки візуальної інформації (Visual Information Processing, VIP) Бостонського університету та опублікований у вересні 2019 року. Набір даних містить 4 відеозаписи, записані камерами типу «риб'яче око» (fisheye cameras) у двох різних кімнатах (комп'ютерна лабораторія та невеликий конференц-зал) та відповідні анотації загальною кількістю 5837 кадрів.

Таблиця 1. Порівняння мереж на датасеті HABBOF

Мережа	Основа	mAP, %	recall, %	fps
Faster R-CNN	ResNet50	51,3	31,7	9
YOLOv3	DarkNet-53	37,5	26,4	39
SSD300	VGG16	39,7	27,6	36
RetinaNet	ResNet50	41,2	33,1	28

В цілому, мережа Faster R CNN досить точно виявляє об'єкти (51,3% на HABBOF), але її не варто застосовувати для роботи з відео (лише 9 кадрів за секунду). SSD і RetinaNet можуть бути застосовані тоді, коли важливою є точність виявлення об'єктів. Хоча вони поступаються швидкістю YOLO. SSD при цьому виявляє об'єкти у різних масштабах, а RetinaNet володіє високим значенням повноти (33,1%), що вказує на правильність класифікації об'єктів. YOLO працює значно швидше (39 кадрів за секунду) і підходить для

роботи в режимі реального часу, але показує відносно невелику точність (37,5%). Виявляє більше помилок локалізації, ніж Faster R CNN, та має труднощі з виявленням дрібних та об'єднаних у групи об'єктів. Тому її було обрано для подальшого удосконалення.

### 3. РОЗРОБЛЕННЯ ГІБРИДНОЇ МЕРЕЖІ

Запропонована у ході дослідження мережа складається з мережі YOLOv3 як базової для вирішення підзадачі виявлення об'єктів. Оскільки побудова такої моделі «з нуля» вимагає багато обчислювальних ресурсів, було вирішено завантажити модель, попередньо натреновану на датасеті COCO [5].

Структуру мережі подано на рисунку 1.



Рисунок 1. Структура гібридної мережі

Дескриптор HOG витягує ознаки з вхідного зображення і подає їх на вхід мережі YOLO для виявлення об'єктів. Далі за інформацією про місцезположення центральної точки об'єкта на кожному з кадрів реалізується відстеження цього об'єкта. Для цього у програмному коді створено окремий клас PntTracking, який знаходить послідовні пари найближчих сусідів, використовуючи двостороннє зіставлення дескрипторів точок.

Навчання мережі тривало 1 годину, що зумовлено комбінованою архітектурою мережі. Готова мережа досягла точності 39,2 % при 43 кадрах за секунду.

Таким чином було встановлено, що використання дескриптора HOG для даного набору даних дало вираш у точності на 1,7% і на чотири кадри за секунду швидшу роботу. Проте навчання зайняло на 20 хвилин більше часу.

### 4. ОЦІНКА ЕФЕКТИВНОСТІ СТВОРЕНОЇ МЕРЕЖІ

Надалі проведено кілька експериментів із запропонованою нейронною мережею з метою оцінки ефективності її роботи в різних умовах. Спершу на вхід мережі було подано відео із конференц залу в офісному приміщенні. Кадр з результатами виявлення і відстеження представлено на рисунку 2.

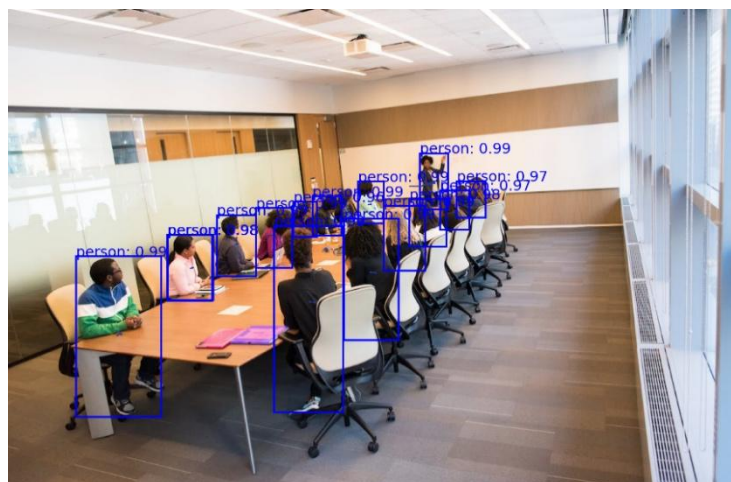


Рисунок 2. Результати роботи мережі



Як і було очікувано, мережа досить точно виявила і відстежила присутніх у відеофрагменті людей.

Другим випробуванням стала робота мережі в умовах поганого освітлення. На вхід подано відеофрагмент із пішоходами, знятий у вечірній час. Кадр результуючого відео наведено на рисунку 3.

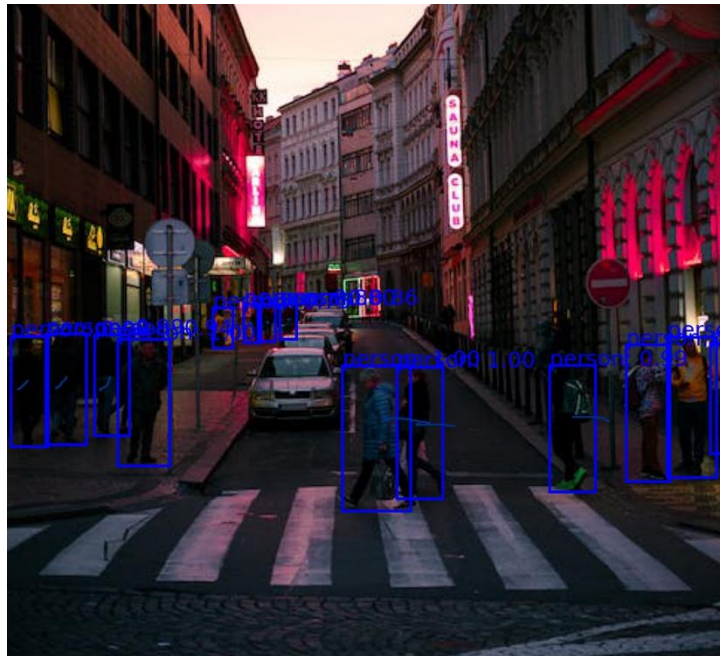


Рисунок 3. Результати роботи мережі в умовах поганого освітлення

Мережа досить точно впоралася із завданням. Змогла навіть виявити осіб, що знаходилися подалі від камери і майже зливалися із заднім фоном.

Насамкінець на вхід мережі подано відео з камер спостереження магазину. Наявна велика кількість дрібних об'єктів. Частина людей перекривається. Приклад кадру результату мережі представлено на рисунку 4.

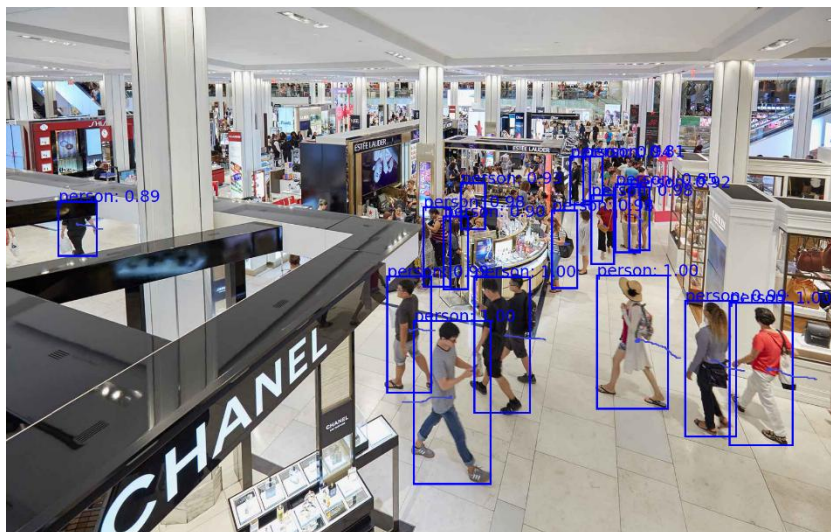


Рисунок 4. Результати роботи мережі



Мережа впоралася відносно добре навіть при перекритті об'єктів. Проте проблема виявлення дрібних об'єктів все ж залишається не вирішеною до кінця.

## 5. ВИСНОВОК

Таким чином, у даній роботі було досліджено використання згорткових нейронних мереж у задачі виявлення та відстеження людини в режимі реального часу.

Отримані результати наступні. Були розглянуті методи виявлення і відстеження об'єктів на зображеннях, зокрема: алгоритм Віоли-Джонса, гістограми орієнтованих градієнтів, нейронні мережі, відстеження на основі виявлення. Проведено порівняльний аналіз швидкості й точності нейронних мереж для виявлення об'єктів Faster R-CNN, YOLO, SSD та RetinaNet.

Найбільшу збалансованість між швидкістю й точністю продемонструвала мережа YOLO. Тому у ході дослідження реалізовано гібридну нейронну мережу на її основі. Поєднання YOLO з дескриптором HOG дозволило підвищити точність й швидкість виявлення і відстеження людини у відеопотоці. До того ж мережа виявила гарні результати і при слабкому освітленні. Відкритим залишається питання виявлення дрібних об'єктів.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Introduction to Video Analytics. [Електронний ресурс]. URL: <https://www.eetimes.com/Introduction-to-video-analytics/> (дата звернення: 06.06.22).
2. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition Conference (CVPR). 2001.
3. Dalal N., Triggs B. Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition Conference (CVPR). 2005.
4. Li S., Tezcan M. O., Ishwar P., Konrad J. Supervised people counting using an overhead fisheye camera. In: International Conference on Advanced Visual and Signal-Based Surveillance (AVSS). 2019.
5. Redmon J., Farhadi A. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767. 2018.

# ЗГОРТКОВА НЕЙРОННА МЕРЕЖА ДЛЯ ЗАДАЧІ КЛАСИФІКАЦІЇ ЗАХВОРЮВАННЯ ЛИСТЯ РОСЛИН

Шевченко В.В.<sup>1</sup>, Шаповал Н.В.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>nvo63911@gmail.com, <sup>2</sup>shovgun@gmail.com

**Актуальність теми обумовлена розвитком сільськогосподарства. Кількість нових видів захворювання рослин збільшується, що негативно впливає на врожайність і як наслідок на економіку країни. Метою цієї роботи є класифікації зображень вірусних захворювань листків рослин за допомогою згорткової нейронної мережі. Результат дослідження можна використовувати для подальшого аналізу під час створення автоматизованих теплиць.**

**Ключові слова:** класифікація зображень, згорткова нейронна мережа

## 1. ВСТУП

Згорткові нейронні мережі (CNN, ConvNet) – це клас глибинних штучних нейронних мереж прямого поширення, який застосовується до аналізу зображень. Використання згортки дозволяє зменшити кількість інформації, що дозволяє краще обробляти зображення більш високої роздільної здатності та виділяти опорні ознаки зображення, а саме: ребра, контури, грані. На наступному рівні обробки з цих ознак можна розпізнати повторювані фрагменти текстур, які далі можуть скластися в фрагменти зображення.

В цій роботі реалізовано декілька моделей згорткових нейронних мереж для задачі класифікації захворювань листків рослин. Дана модель допоможе фахівцям точніше аналізувати ситуацію із захворювання та приймати більш підходящий сценарій лікування.

## 2. ЗГОРТКОВА НЕЙРОННА МЕРЕЖА

Набір даних було зібрано організацією AIR Lab Makerere University. Дана вибірка[0] містить зображення двох типів захворювання листків та здорові листки рослин. Кількість зображень – 1034.

Для створення моделі була проведена попередня обробка зображень, стандартизація.

Першим дослідженням було створення власної шестишарової згорткової мережі. Точність розпізнавання зображень – 70%.

Тому для покращення результатів було використано мережу VGG16[0], який вважається однією з найкращих моделей комп'ютерного зору на сьогоднішній день. Дана модель використовує архітектуру з дуже малими (3 на 3) згортковими фільтрами. Глибина становить до 16-19 вагових шарів, що значно покращує результат. Щоб отримати оптимальний набір параметрів, необхідно мати багато даних та велику обчислювальну потужність.

Fine-tuning [0] дозволяє уникнути цих обмежень. Завдання тонкого налаштування мережі полягає в налаштуванні параметрів вже навченої мережі, щоб вона адаптувалась до нового завдання. Початкові рівні вивчають дуже загальні особливості, кожен наступний рівень має тенденцію вивчати моделі, більш специфічні для завдання, якому вони

навчаються. Таким чином, для точного налаштування за допомогою Fine-tuning початкові шари залишаються недоторканими, а наступні шари перенавчаються для нашого завдання. Час навчання буде менший, оскільки кількість параметрів зменшиться.

Також було використано мережу Inceptionv3[0], але отримано гірший результат. Порівнюючи ці дві згорткові мережі, можна зрозуміти, що вони працюють приблизно однаково, але результат залежить від вибірки даних, її об'єму попередньої обробки.

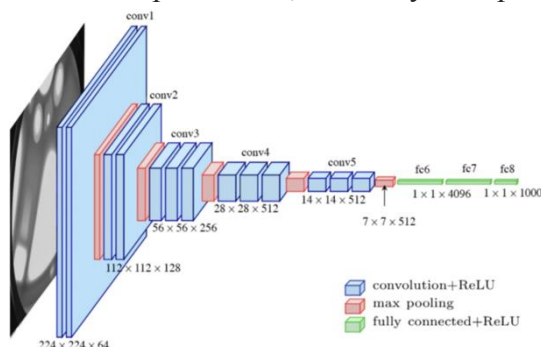


Рисунок 1. Архітектура VGG16

### 3. ТРЕНУВАННЯ МОДЕЛІ ТА РЕЗУЛЬТАТИ ГЕНЕРАЦІЇ

Різні моделі згорткових нейронних мереж навчались протягом 10 епох. У якості оптимізатора був використаний Adam із затуханням вагів. За метрику якості було вибрано accuracy.

Таблиця 1. Порівняння результатів роботи різних згорткових мереж

Модель	Втрати	Точність
Власна модель (6 вагових шарів)	0.42	0.7
Модель з використанням VGG16	0.35	0.85
Модель з використанням VGG16 та fine-tuning	0.23	0.91

### 4. ВИСНОВКИ

У результаті модель з використанням VGG16 та fine-tuning показала найкращий результат, а саме точність 91% на тестовому датасеті. Точність розпізнавання захворювання «bean rust» – 92%, «angular leaf spot» – 83% та здорових листків – 95%.

Отже, отриману модель можна використовувати для класифікації здорових та хворих рослин.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Датасет зображень листя рослин: вебсайт. URL: <https://github.com/AI-Lab-Makerere/ibean.git> (дата звернення: 14.11.2022).
2. Keras, VGG16, Inceptionv3: документація: URL: <https://keras.io/api/applications/> (дата звернення: 14.11.2022).
3. Keras, Sequential: документація: URL: <https://keras.io/api/models/sequential/> (дата звернення: 14.11.22).
4. Keras, fine-tuning: документація: URL: [https://keras.io/guides/transfer\\_learning/](https://keras.io/guides/transfer_learning/) (дата звернення: 16.11.22).

# РОЗРОБКА МОДЕЛЕЙ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ АНАЛІЗУ ТА ФОРМУВАННЯ ІНВЕСТИЦІЙНОГО ПОРТФЕЛЮ

Шевчук О.С.<sup>1</sup>, Кузнєцова Н.В.<sup>2</sup>

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>1</sup>oleksii.shevchuk@ukr.net, <sup>2</sup>natalia-kpi@ukr.net [0000-0002-1662-1974]

У сучасному світі важливим є навички правильного фінансового управління, ціллю якого є приведення активів у рух з метою їх збереження та примноження. Ключовим елементом фінансового менеджменту є інвестування, зокрема у формі інвестиційного портфелю. Для ефективного формування та управління інвестиційного портфелю було розроблено велика кількість математичних методів та моделей, зокрема модель Марковіца. Останнім часом все більшої популярності набувають моделі штучного інтелекту. Їх застосування розповсюджується на різноманітні сфери, фінансова не стала виключенням. У даній роботі будуть розглянуті класичні методи формування інвестиційного портфелю, розроблено нову модель з використанням штучного інтелекту та проведено їх порівняння на конкретних моделях.

**Ключові слова:** інвестиційний аналіз, формування інвестиційного портфелю, методи штучного інтелекту.

## 1. ВСТУП

Поява поняття інвестування бере свій початок ще з часів стародавнього Вавилону. Саме тими часами датуються перші писемні згадки, про інвестування та грамотне управління своїми фінансами. З плином часу розроблялися нові способи інвестування, які підвищували прибутки та зменшували ризики. Одним із таких інструментів ефективного інвестування є інвестиційний портфель.

Інвестиційний портфель – це цілеспрямовано сформована сукупність вкладень в інвестиційні об'єкти, яка відповідає певній інвестиційній стратегії інвестора. Це означає, що основною ціллю формування інвестиційного портфелю є забезпечення реалізації розробленої інвестиційної політики шляхом підбору найбільш ефективних та надійних інвестицій [1].

Зважаючи на це, ключовим аспектом у формування інвестиційного портфелю є вибір оптимальних інвестиційних об'єктів, що обумовлює доцільність вивчення способів їх аналізу та моделювання їх поведінки. При цьому, важливим є врахування усіх особливостей фінансової системи, у якій знаходиться інвестиційний об'єкт, та чинників, що впливають на неї. Для підвищення ефективності аналізу, використовують математичні моделі, що описують фінансові процеси, а також стає все більш популярним використання моделей зі штучним інтелектом для вирішення інвестування.

## 2. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Метою даної роботи є розробка методу формування інвестиційного портфелю з використанням штучного інтелекту. Крім цього будуть розглянуті й існуючі методи, та

проведене їх порівняння між собою, та із методом штучного інтелекту. Результатом роботи буде розробка методу штучного інтелекту для формування інвестиційного портфелю, а також конкретних моделей, на основі математичних методів, та методів зі штучним інтелектом. Таким чином, об'єктом дослідження даної роботи є інвестиційні портфелі, а також способи їх аналізу та формування. Предметом дослідження є математичні методи та методи штучного інтелекту, що допомагають у формуванні та аналізі інвестиційних портфелів.

### **3. ОГЛЯД КЛАСИЧНИХ МЕТОДІВ ФОРМУВАННЯ ІНВЕСТИЦІЙНОГО ПОРТФЕЛЮ**

Формування інвестиційного портфелю можна розглядати як задачу вибору декількох оптимальних альтернатив серед множини усіх можливих інвестиційних об'єктів. Саме у таких випадках може допомогти метод прийняття рішень при нечіткому відношенні переваги на множині альтернатив. Даний метод базується на теорії Люфті Заде про нечіткі множини та їх властивості, яка була опублікована ще у 1965 році. Дана теорія допомагає формально описувати нечіткі поняття, якими користується людина, описуючи свої бажання, цілі та уявлення про систему [2].

Суть методу прийняття рішень полягає у порівнянні альтернатив за їх критеріями, і вибору оптимального або оптимальних із усієї множини. Зокрема цінність кожного з критеріїв встановлюється відповідно до цілей та вподобань. У випадку формування інвестиційного портфелю, альтернативами є об'єкти інвестування, а критеріями можуть бути різноманітні метрики, наприклад, для акцій компанії це може бути капіталізація компанії, її борги, вид діяльності компанії, ціна самої акції та інше. Таким чином, метод прийняття рішень допомагає інвестору обґрунтовано обрати об'єкти інвестування, зважаючи на свої цілі та вподобання.

Найбільш розповсюдженим і, певною мірою, класичним способом формування інвестиційного портфелю є спосіб із використанням моделі Марковіца. Теорія формування інвестиційного портфелю за Г. Марковіцем засновується на поведінковій специфіці інвестора, який хоче вкласти свої ресурси в компанії із найменшим ризиком, і за цей ризик отримати певні дивіденди. Підхід Марковіца передбачає, що інвестор враховує лише два параметри: ризикованість та величину доходів [3]. Модель оптимального портфелю за Г. Марковіцем ґрунтується на наступних принципах:

- інвестор хоче максимізувати дохід за певного заданого рівня ризику;
- інвестори завжди намагаються уникати ризику. Із двох активів із однаковою прибутковістю обирається той, ризикованість якого нижча;
- ризик є невизначеністю майбутнього результату;
- портфель інвестора складається з всіх його активів та зобов'язань;
- інвестори приймають рішення про інвестиції, засновуючись на очікуваних доходах та ризикованості інвестицій.

Таким чином, основна ідея моделі Марковіца полягає у тому, щоб статистично визначити майбутній дохід та ризик кожного інвестиційного об'єкта. Вибір об'єктів портфелю та їх частини може варіюватися від стратегії інвестора. Можна мінімізувати ризики портфеля або максимізувати його прибутковість. З розвитком теорії Марковіца з'явився інший спосіб вибору об'єктів інвестування – максимізація коефіцієнта Шарпа. Даний коефіцієнт допомагає вирішити проблему вибору між двома об'єктами інвестування, шляхом порівняння їх дохідності відносно безризикового об'єкту інвестування. Коефіцієнт Шарпа показує відносну прибутковість об'єкта на одиницю ризику [4]. Таким чином,

максимізація цього значення допомагає сформувати портфель із максимальною «прибутковістю» ризику.

Ще один метод, який допоможе визначити ймовірність певної події за наявними вхідними даними є мережі Байєса. Мережі Байєса допомагають встановити причинно-наслідкові зв'язки між певними ознаками та висновком, що отримується при таких умовах. У випадку задачі формування інвестиційного портфелю, на основі різноманітних вхідних параметрів, таких як зміна ціни акції, зміна капіталу, виду діяльності компанії, можна встановити, що чекає на ціну акції в майбутньому – падіння чи зростання.

Мережу Байєса можна розглядати як модель представлення взаємозв'язків між вершинами ациклічного графа, що представлені у вигляді ймовірнісних залежностей, тобто зв'язки між вершинами графа є причинними. Зв'язок буде називатися причинним лише тоді, коли подія є причиною виникнення події, тобто подія певною мірою впливає на виникнення події [5]. Теоретичним підґрунтям для мереж Байєса є теорія ймовірностей і, зокрема, теорема Байєса яку можна описати формулою \_\_\_\_\_.

Ця формула дає змогу визначити ймовірність залежної події, базуючись на наявній статистичній інформації.

Таким чином, метод мереж Байєса, дає можливість побудувати модель, яка буде визначати ймовірність кожного результату подій, із множини всіх результатів, базуючись на отриманих знаннях.

#### **4. ОГЛЯД МЕТОДУ ШТУЧНОГО ІНТЕЛЕКТУ ФОРМУВАННЯ ІНВЕСТИЦІЙНОГО ПОРТФЕЛЮ**

Метод з використанням штучного інтелекту є досить універсальним щодо даних. На вхід йому подаються різноманітні дані про акції компанії та саму компанію, як місячні, так і поквартальні. З цих даних збираються статистичні метрики для кожного параметру, такі як максимальне та мінімальне значення, середнє значення на проміжку, стандартне відхилення, а також ці самі ж метрики для першої різниці та інші параметри. Також використовується базова інформація про компанії, така як сектор та індустрія в якій працює компанія.

Сам метод базується на моделях градієнтних бустингів для визначення «справжньої» ціни компанії. Метод вивчає всі компанії, що відносяться до певних сектору та індустрії, їх метрики та ціни на акції, і встановлює де ціна завищена через певні чинники, а де навпаки – акції недооцінені. Сама суть градієнтних бустингів полягає у пошуку функції, яка описує набір вхідних та відповідних цільових даних. Навчання такої моделі – процес ітеративний, ідеєю якого є знайти таке рівняння, для якого деяка функція похибки на всьому наборі тренувальних даних буде оптимальним [6].

Бустингові алгоритми мають можливість формувати ансамблі – коли декілька моделей об'єднуються в єдине ціле. Такий підхід часто використовується для того, щоб підсилити позитивні якості окремих алгоритмів, які самі по собі можуть проявлятися слабо, а в групі показують чудовий результат. При використанні ансамблевих методів, алгоритми навчаються одночасно і можуть виправляти помилки один одного. Ансамблі моделей зазвичай будуються на основі моделі дерева рішень. Дерева додаються по одному в ансамбль і навчаються для взаємного виправлення помилок прогнозування, що роблять попередні моделі. Таким чином кінцевий результат покращується.

#### **5. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ**

Для дослідження використовувалися різноформатні дані про різноманітні компанії їх ціни на акції, зокрема базова інформація про компанію, їх поквартальні звіти, а також денні

дані про компанію та акції. Цільовим значенням, було значення курсу закриття ціни на акції, а прогноз робився на наступний квартал. Портфель формувався не з усіх можливих акцій компаній, а лише із компаній, що відносяться до п'яти обраних індустрій. Таким чином, портфель формувався із акцій наступних компаній: NKLA, TSLA, RACE, STLA, F, TM, MSFT, ADBE, ORCL, PAYO, BB, DBX, AAPL, NVDA, AMD, SONY, KOSS, BOX, NFLX, DIS, IMAX, WWE, CNK, WMG. Першою побудованою моделлю була модель для методу прийняття рішень при нечіткому відношенні переваги на множині альтернатив. Портфель, сформований цією моделлю, складається із семи об'єктів, таке значення було обране інвестором. Результати (Табл. 1) даного портфелю незадовільні, зокрема через наявність акцій TSLA, які дуже сильно впали за останній час, а саме їх в портфелі найбільший відсоток. Сумарний прибуток портфелю складає -1291,63.

Таблиця 1. Інвестиційний портфель моделлю прийняття рішень

Компанія	Різниця ціни закриття	Відсоток	Прибуток
TSLA	-64.24	21.74	-1396.52
NKLA	0.29	16.30	4.73
WWE	12.68	15.22	192.96
BOX	-1	13.04	-13.04
STLA	-4.77	11.96	-57.03
BB	-0.76	10.87	-8.26
AMD	-1.33	10.87	-14.46

Наступною побудованою моделлю була модель, що базується на теорії Марковіца. Вона показала значно кращі результати (Табл. 2, Табл. 3) ніж попередня і її можна назвати ефективною. Особливо це стосується портфеля, що збирався зо максимальним коефіцієнтом Шарпа, адже його прибутковість становить 96.75 у порівнянні із -105.67 для портфелю мінімального ризику.

Таблиця 2. Інвестиційний портфель моделлю Марковіца з мінімальним ризиком (найбільш значущі об'єкти)

Компанія	Різниця ціни закриття	Відсоток	Прибуток
TSLA	-64.24	1.4	-89.94
STLA	-4.77	2.3	-10.97
TM	-8.15	33.3	-271.40
MSFT	6.19	2.8	17.33
ADBE	-87.1	1.4	-121.94
ORCL	8.36	16.9	141.28
NVDA	-37.17	0.9	-33.45
DIS	7.22	2.6	18.77
WWE	12.68	16.4	207.95

Таблиця 3. Інвестиційний портфель моделлю Марковіца з максимальним коефіцієнтом Шарпа (найбільш значущі об'єкти)

Компанія	Різниця ціни закриття	Відсоток	Прибуток
TSLA	-64.24	3.9	-250.54
TM	-8.15	2.7	-22.01
ADBE	-87.1	2.1	-182.91
ORCL	8.36	1.3	10.87
NVDA	-37.17	2.7	-100.36
WWE	12.68	48.9	620.05

Модель мереж Байєса показала невдалі результати (Табл. 4), як і модель прийняття рішень. Прибуток портфелю, створеного цією моделлю, складає -740.65.

Таблиця 4. Інвестиційний портфель моделлю прийняття рішень

Компанія	Різниця ціни закриття	Відсоток	Прибуток
MSFT	6.19	11.41	70.63
ADBE	-87.1	11.41	-993.42
ORCL	8.36	11.41	95.38
PAYO	2.08	11.41	23.72
BB	-0.76	11.41	-8.67
DBX	2.41	11.41	27.49
SONY	2.76	20.15	55.62
BOX	-1	11.41	-11.41

Найбільш ефективною моделлю стала модель побудована із використанням штучного інтелекту. Сумарний дохід створеного портфелю складає 195.93. Модель виключила акціях, які впали найбільше за останній час, а саме ADBE та TSLA, і додала до портфелю найбільш прибуткові об'єкти з непоганим відсотком. Складові портфелю та їх відсотки в портфелі представлені у таблиці 5.

## 5. ВИСНОВКИ

Задача фінансового менеджменту завжди була актуальною. Особливо це важливо в кризові часи, коли ситуація в світовій економіці дуже нестабільні і більшість фінансових активів зазнають падіння в ціні.

В результаті даного дослідження було розглянуто класичні методи формування інвестиційного портфелю та побудовано відповідні моделі. Також, було розроблено новий метод формування, що використовує штучний інтелект. Було проведено порівняння створених моделей та визначено їх ефективність.



Таблиця 5. Інвестиційний портфель моделлю зі штучним інтелектом

Компанія	Різниця ціни закриття	Відсоток	Прибуток
NKLA	0.29	8.96	2.60
RACE	-0.71	6.08	-4.31
STLA	-4.77	10.72	-51.13
ORCL	8.36	9.50	79.43
PAYO	2.08	13.80	28.71
AMD	-1.33	9.81	-13.05
SONY	2.76	9.61	26.53
BOX	-1.00	8.62	-8.62
WWE	12.68	7.20	91.35
CNK	16.75	6.49	10.19
WMG	3.72	9.20	34.23

Найменш ефективною була модель прийняття рішень. Дійсно, даний метод допомагає лише обрати найбільш оптимальні альтернативи серед всіх можливих, але він ніяк не аналізує фінансові процеси. Також погано себе показав метод мереж Байєса. Йому не вдалося не те що отримати прибуток, але хоча б не зазнати збитків. Щодо методу Марковіца, то він показав відносно непогані результати. При виборі стратегії мінімального ризику, збитки будуть значно меншими, ніж для попередніх двох методів. Для стратегії максимізації коефіцієнта Шарпа був отриманий позитивний результат із прибутковим портфелем. Найбільш ефективним виявився метод із штучним інтелектом. Портфель, сформований даною моделлю, був найбільш прибутковим і він виключи акції з найбільшим падінням курсу.

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Замковий О.І. Портфельні теорії інвестування. метод.-наук. посіб. Дніпро: НТУ «ДП», 2020. – 70 с.
2. Зайченко Ю.П. Портфельні теорії інвестування. метод.-наук. посіб. Дніпро: НТУ «ДП», 2020. – 70 с.
3. Скрипниченко М.В. Портфельные инвестиции: Учебное пособие. - СПб: Университет ИТМО, 2016 - 40с.
4. Шарп У., Инвестиции. / У. Шарп, Г. Александер, Дж. Бейли. – М.: ИНФРА-М, 1997. – 172 с.
5. Бідюк П.І. Коршевніюк Л.О. Проектування комп'ютерних інформаційних систем підтримки прийняття рішень: Навчальний посібник. — Київ: ННК „ІПСА” НТУУ „КПІ”, 2010. — 340 с.
6. Открытый курс машинного обучения. Тема 10. Градиентный бустинг. URL: <https://habr.com/ru/company/ods/blog/327250/> (дата звернення: 29.10.2022)
7. Galkina S. Optimal Portfolio Selection Using Machine Learning Techniques / International Journal of Open Information Technologies ISSN: 2307-8162 vol. 2, no. 6, 2014.

# MAKING DECISIONS IN THE SERVICE BUSINESS SPHERE USING ARTIFICIAL INTELLIGENCE

I.A. Shubenkova.<sup>1</sup>, M.S. Marenych<sup>2</sup>

National Technical University of Ukraine  
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine

<sup>2</sup> mariiamarenych@gmail.com

**Current business conditions, featuring an unstable economic situation and growing fierce competition, put higher requirements on the efficiency and quality of decision-making at all levels of enterprise management. The research aims to analyze the business processes of the service sector businesses and develop software to improve the efficiency of increasing sales. Based on the researched material, I created the decision support system (DSS) to predict sales of service sector businesses (coffee, restaurant). The main idea of product development is free of charge and mass use. In work were used both theoretical and empirical research methods.**

**Keywords: decision-making, service sector business, business process analysis, sales prediction.**

## 1. INTRODUCTION

Today, making weighted and high-quality decisions is one of the main challenges for managers. Decision support provides up-to-date global information about the current situation and business development tendencies through data mining methods and tools. At the same time, the amount of information is constantly growing. It leads to a situation when it becomes impossible to effectively manage the company without using business analysis automation methods and tools [1].

The implementation of data mining in the practical work of enterprises to resolve specific issues of improving management effectiveness in the majority of cases is relatively expensive and time-consuming. The key challenges are the need for a formal statement of the problem and strategy of searching for information, the heuristic approach of most intelligent models, and high dimension and low-quality data. Therefore, developing new systems and implementing data mining methods to solve specific problems of increasing management efficiency in social and economic sectors is an urgent research and technical task.

## 2. PROBLEM STATEMENT

Based on the review of available scientific and technical sources, we can define the research goal: to develop a system for predicting demand for the assortment of products of the service sector business using artificial neural networks. For this, it is necessary to complete the following tasks:

- To analyze and propose a list of service sector business parameters those affect the value of demand and direct them to the neural network inputs;
- To justify the type and structure of a neural network for predicting the demand for products;
- To implement a system for predicting demand for products of the restaurant and coffee business in the service sector;
- To find relevant information in open sources and create a pool of training information based on which to train the neural network;

- To test the system and conclude the effectiveness of the developed product.

As inputs for predicting demand will be used:

- A list of names and economic efficiency values of restaurants and coffee shops that may affect the demand for the range of their products;
- A list of values of the selected values and the relevant value of the upcoming period demand, which makes up the dataset for training the neural network;
- Parameters of available system software development tools that enable to choice of the best option for the implementation of a software based on neural networks.

The output information will be the value of demand for specific products for a certain time period.

### 3. PREDICTION METHODS FOR DEMAND FOR PRODUCT SALES

Ideally, demand prediction should include both qualitative and quantity prediction. Both of them require collecting information from different data sources. Qualitative data can be extracted from external sources such as news reports, cultural and social media trends, and competitor and market research. Information from internal sources, such as customer feedback and preferences, also significantly increases the accuracy of the prediction. Quantitative information is usually internal and can be collected from sales figures, periods of peak purchases, and web and search analytics. Current technologies use advanced analytics, powerful databases, artificial intelligence, and machine learning to analyze and process highly complex data sets. When modern technology is applied to qualitative and quantitative prediction and predictive analytics, supply chain managers can provide ever-increasing levels of accuracy and resilience.

According to the industry, customer database, and product volatility, demand planners use the following prediction methods [2]:

1. Predicting demand at the macro level;
2. Predicting demand at the micro level;
3. Short-term demand prediction;
4. Long-term demand prediction.

To understand it better, we provide a relative analysis of prediction methods among themselves (Table 1).

Table 1. Comparative analysis of prediction methods

Comparison parameters	Timeline analysis method	Casual method	Expert assessment methods
Term	Short-term, mid-term	medium-term, long-term	long-term
Requirements for implementing the method	Quantitative parameters	Quantitative parameters	Experts and qualified specialists
Speed	+	+	+
Complexity of method	Medium	High	High
Credibility and reliability	Depends on the quality and quantity of data	Depends on the prediction	Depends on the expert's competence
Advantages	Simplicity, clarity of results, easy implementation		Systematic approach
Disadvantages	Only quantitative predictions, availability of data for a long-term perspective	Practical implementation of the method for some areas may be incorrect	The credibility and reliability of the research results depend on the expert's competence

After a detailed analysis of each method and considering the specifics of the service sector business, this method is used when predicting for the short and medium term period, allowing to obtain sufficient accuracy of predicted values. Also, this method is not only the most popular method of all listed but also rather simple and effective. It is easy enough to implement using a computer. Using this method will allow us to consider the effect of seasonality and tendency factors to identify the months of the highest and lowest activity in sales and, therefore, to plan stocks correctly. Based on the previous analysis of time series reflecting the dynamics of sales of products of the service sector business, linear and polynomial functions were chosen as the trend model. These are the most simple and commonly used functions in sales prediction. The terms of these functions can be easily calculated using the least squares method. Let's create a mathematical model:

Let  $x = \{x_1, x_2, x_3 \dots x_n\}$  – a set of parameters applied to the inputs of the neuron, where  $n$  is the number of input parameters.

$y_{true} = \{y_1, y_2, y_3 \dots y_m\}$  – a training set of valid values of demand used in the neural network, where  $m$  is the number of output parameters;

$\omega_1, \omega_2, \omega_3 \dots \omega_n$  – a set of synaptic weights

$$S = \sum_{i=1}^n \omega_i \cdot X_i,$$

Where  $S$  is an adder designed to produce a weighted sum.

$$y_{pred} = f(S) = F\left(\sum_{i=1}^n \omega_i \cdot X_i\right),$$

Where  $y_{pred}$  is the predicted demand value for the product,  $f$  is the sigmoid activation function. The target function will be the following formula:

$$L = \frac{1}{2} \sum_{i=1}^m (y_{true} - y_{pred})^2 \rightarrow \min,$$

Where  $L$  is the loss function.

By using neural networks prediction demand for a range of things in a supermarket solve, it is necessary to train the neural network so that the prediction accuracy is as high as possible. To implement the algorithm of the training process, you must initially import the training data from the database. The file holds the parameters that affect the demand and the demand values for the selected products for a given time term using simulated sales for the selected products. Figure 1 illustrates the neural network training process for the service sector business. To start training, the user needs to select the necessary product for which the sales simulation for a specific time was performed and begin the neural network training process. After the training process completing and the weights are corrected, the program will write all the training data of the neural network to the database file. By using the training, the user will be able to predict the sales of products for a selected time. Neural network training is the loss function minimization problem. To solve this problem, we use the stochastic gradient descent algorithm. Using this algorithm, you can adjust the synapse weights of the network. The following steps carry out the implementation of the stochastic gradient descent algorithm:

1. Select one parameter from the data set.
2. Find all partial derivatives of the weight or displacement loss function.
3. Update all weights and displacements.
4. Repeat the cycle from step 1.

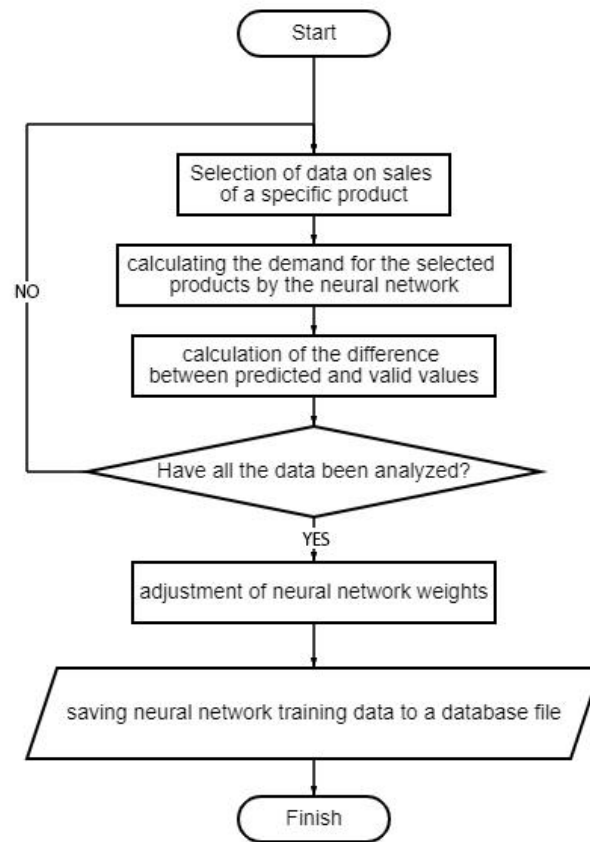


Figure 1. The algorithm of the neural network training process for service sector business

#### 4. SOLUTION

The developed product must comply with quality features such as sustainability, usefulness, accessibility, scalability, openness, flexibility, and testability. It requires additional constraints/rules from the development process, such as:

- Compliance with templates and styles;
- Documentation of development at various levels;
- Testing of components, specific modules, and subsystems;
- Project and process management.

By considering the requirements to ensure the sustainability and flexibility of the system, we chose the Layers pattern for its development, which divides the system into two parts: client and server. The system's design will rely on the subject area (DDD approach) and SOLID principles [3]. When developing the customer, we chose the Windows Forms user interface. The server, in turn, will consist of the following modules:

1. Business Logic Layer– logic and all necessary calculations of the application in the language of business;
2. Data Access Layer – the level of access to data.
3. Data Base – database for data storage.

By taking into account all the above templates, the project will look as follows (Figure 2):

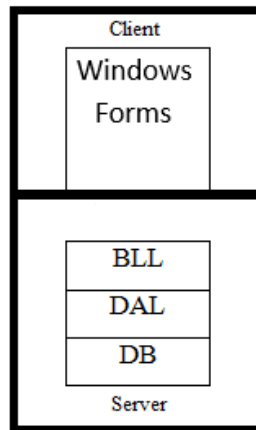


Figure 2. Project structure

Each component will implement a contract (interface) that provides flexibility to the element. Since the business is constantly changing relatively quickly, each part must soon adapt to these changes. For this, we will use the Dependency Injection pattern.

## 5. CONCLUSIONS

We have reviewed the main methods of prediction of economic processes and conducted their comparative analysis. We chose the development tools to implement the decision support system for the service sector business and justified the prediction method.

An algorithm for building a prediction and estimating we developed its error based on the chosen method. We implemented the developed algorithm in software and the information system of the service business. The program showed good forecasting accuracy results and fully met the requirements. In further research, it is advisable to introduce such elements that take into account at the probability level rare events that sometimes occur in the world (for example, pre-holidays, quarantine, etc.). It can radically change the real numbers compared to the forecasts of the best economic mathematical models.

## REFERENCES

1. "Business intelligence (BI): the definitive guide". MicroStrategy. <https://www.microstrategy.com/en/resources/introductory-guides/business-intelligence-the-definitive-guide>.
2. Project Analysis. Kyiv: Libra, 1999.
3. Karpychev, V. Yu, та Yu P. Shal'nova. "Big Data Conceptual Model". Proceedings of R.E. Alekseev NSTU, № 2 (2021): 16–23. [https://doi.org/10.46960/1816-210x\\_2021\\_2\\_16](https://doi.org/10.46960/1816-210x_2021_2_16).

# ВИКОРИСТАННЯ МОДЕЛІ BI-LSTM ДЛЯ ПІДВИЩЕННЯ ТОЧНОСТІ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ

Яцько Я.В.<sup>1</sup>, Терентьєв О.М.<sup>2</sup>

<sup>1</sup> Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

<sup>2</sup> Інститут телекомунікацій і глобального інформаційного простору  
Національної академії наук України, Київ, Україна

<sup>1</sup>yayatskaa@gmail.com, <sup>2</sup>o.terentiev@gmail.com

**Криптовалюта – цифрова валюта, яка використовується у фінансових системах і використовує технологію блокчейн і криптографічні функції для досягнення прозорості та децентралізації. Оскільки ціни на криптовалюту сильно коливаються, необхідні інструменти для їх моніторингу та прогнозування. Довга короткочасна пам'ять (LSTM) – це модель глибокого навчання, яка здатна точно прогнозувати часові ряди даних. У цьому дослідженні використовується двонаправлена LSTM. (Bi-LSTM) для підвищення точності і нормалізації оцінки середньоквадратичної помилки (RMSE) з використанням чотирьох криптовалют: Bitcoin, Ethereum, Ripple і Binance (BNB).**

**Ключові слова:** криптовалюта, Bi-LSTM, блокчейн, Bitcoin.

## 1. ВСТУП

На ринок криптовалют впливають фактори невизначеності, такі як політичні та економічні проблеми на глобальному рівні. Тому точна інтерпретація прогнозів є складним завданням. Іншою проблемою, яка є предметом цього дослідження, є щоденні коливання курсу криптовалют, які необхідно вирішити за допомогою прикладного інструменту, який може відстежувати та запобігати невизначеності в транзакціях [3]. На ринку криптовалют є багато популярних монет, таких як Bitcoin (BTC), Ethereum (ETH), Binance Coin (BNB) і Ripple (XRP).

Попереднє дослідження включало в себе використання алгоритму нейронної мережі з довгою короткочасною пам'яттю (LSTM). При цьому, значення RMSE мало значний розбіг (дисперсію) від 2 000 до 50 000 доларів. Тому ціллю експерименту було використання моделі Bi-LSTM для підвищення точності прогнозування.

## 2. ПОСТАНОВКА ЗАДАЧІ

Прогнозування курсу криптовалют має незначні відмінності від методу прогнозування акцій. Кращі результати можна отримати, поєднуючи різні методи, такі як аналіз даних часових рядів, технічний аналіз фондового ринку та історичні дані з ціни, з декількома алгоритмами [4]–[6]. Це дослідження робить певний внесок у область прогнозування криптовалют. Оцінкою якості моделі обрано RMSE та MAPE. У попередніх дослідженнях для прогнозування використовувалися лише алгоритми нейронної мережі з довгою короткочасною пам'яттю (LSTM). Для вдосконалення попередньої моделі запропоновано використати гібридну нейронну мережу, що складається з рекурентних блоків (GRU) і двонаправленої LSTM (Bi-LSTM), оскільки остання обробляє набори даних за допомогою багаторазового циклу, що збільшує передбачувану здатність моделі. В цьому дослідженні

було використано дані з Binance за останні 3 роки та використано модель Bi-LSTM, структура мережі якої зображена на рис.1:

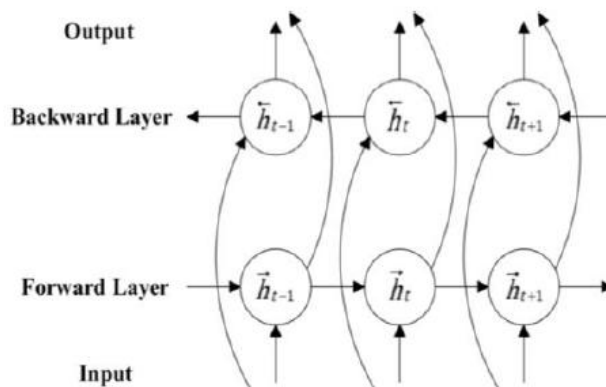


Рисунок 1. Структура мережі Bi-LSTM [7]

### 3. ОПИС ЗНАЙДЕНОГО РІШЕННЯ

Слід зазначити, що результати є кращими, ніж в попередньому експерименті з використанням простої моделі LSTM (RMSE менше на 20 % пунктів). Після нормалізації результати стають ближчими до 0, а оцінка MAPE становить менше 10% за RMSE [4], [6]. На рис. 2 зображено порівняльну тенденцію реальних даних та прогнозу за допомогою моделі експерименту.

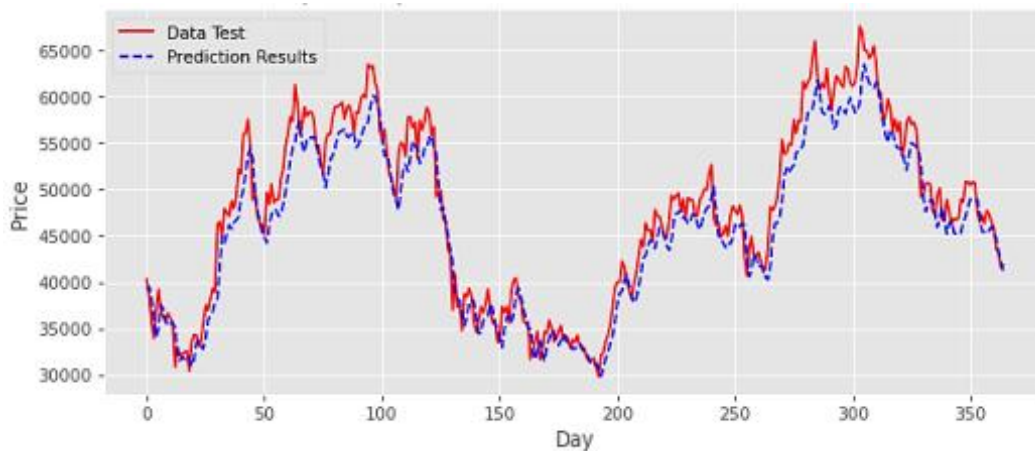


Рисунок 2. Порівняння фактичних даних і прогнозу курсу Bitcoin



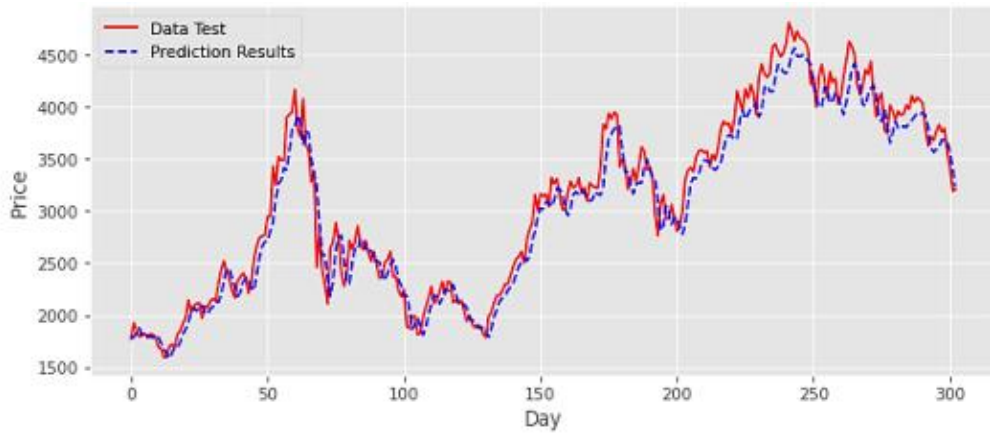


Рисунок 3. Порівняння фактичних даних і прогнозу курсу Ethereum

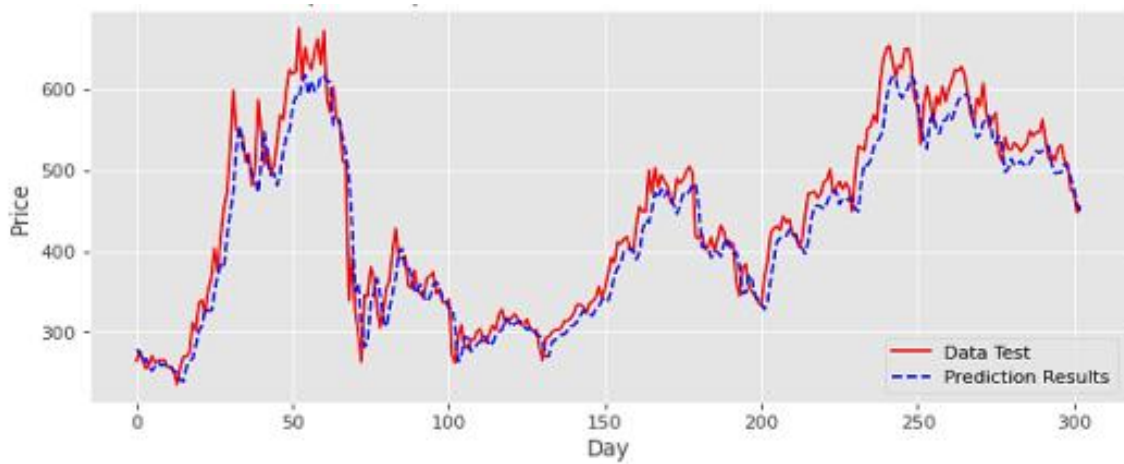


Рисунок 4. Порівняння фактичних даних і прогнозу курсу Binance

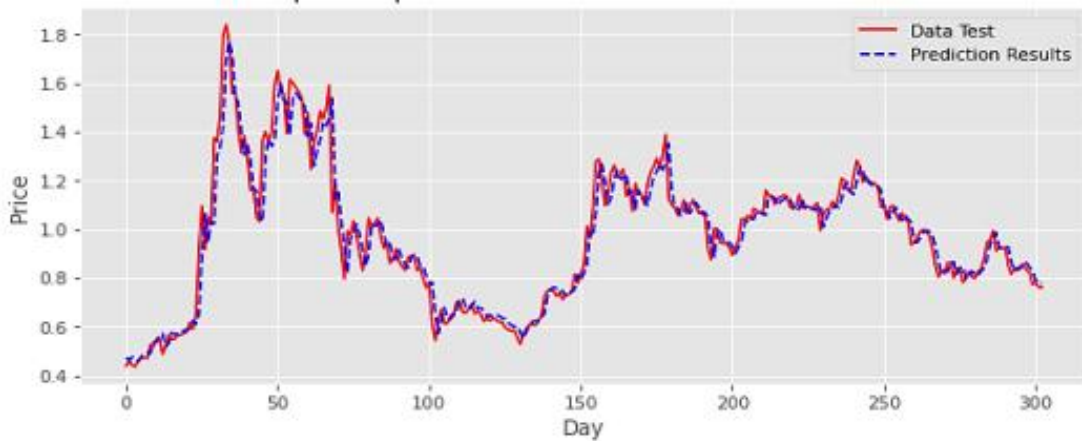


Рисунок 5. Порівняння фактичних даних і прогнозу курсу Ripple

В табл. 1 розміщено результати оцінки моделі на основі значень MAPE і RMSE.

Таблиця 1. Метрики якості моделі

Монета	RMSE	Normalize RMSE	MAPE (%)
Bitcoin	2343.2200	0.062	4.0
Ethereum	203.8900	0.063	5.31
Binance	404.1800	0.073	5.64
Ripple	0.0933	0.066	5.82

#### 4. ВИСНОВКИ

В ході дослідження було розглянуто гібридну модель GRU Bi-LSTM для покращення прогнозування курсу криптовалют. Запропонована модель дала значно кращі результати, у порівнянні зі звичайним LSTM. У дослідженні було застосовано кілька підходів для прогнозування курсу криптовалют з використанням історичних даних, отриманих за допомогою Binance API за 3 роки. Чотири монети (Bitcoin, Ripple, Binance та Ethereum) були використані для перевірки точності запропонованої моделі щодо забезпечення оптимальних результатів. Завданням для наступних досліджень є покращення точності прогнозування за рахунок додання в модель нових даних, таких як політичні настрої, природні умови і т.д. з зовнішніх джерел інформації.

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu, and R. Brooks, “A brief survey of cryptocurrency systems,” 2016 14th Annual Conference on Privacy, Security and Trust, PST 2016, pp. 745–752, 2016, doi: 10.1109/PST.2016.7906988.
2. E. Pintelas, I. E. Livieris, S. Stavroyiannis, T. Kotsilieris, and P. Pintelas, “Investigating the problem of cryptocurrency price prediction: a deep learning approach,” IFIP Advances in Information and Communication Technology, vol. 584 IFIP, pp. 99–110, 2020, doi: 10.1007/978-3-030-49186-4\_9.
3. J. Eapen, D. Bein, and A. Verma, “Novel deep learning model with CNN and bi-directional LSTM for improved stock market index prediction,” in 2019 IEEE 9th Annual Computing and Communication Workshop and Conference, CCWC 2019, 2019, pp. 264–270, doi: 10.1109/CCWC.2019.8666592.
4. V. B. Kamble and S. N. Deshmukh, “Comparision between accuracy and MSE, RMSE by using proposed method with imputation technique,” Oriental journal of computer science and technology, vol. 10, no. 04, pp. 773–779, 2017, doi: 10.13005/ojst/10.04.11.
5. M. De Caux, F. Bernardini, and J. Viterbo, “Short-term forecasting in Bitcoin time series using LSTM and GRU RNNs,” in Anais do VIII Symposium on Knowledge Discovery, Mining and Learning, SBC, 2020, pp. 97–104.
6. T. O. Hodson, “Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not,” Geoscientific Model Development, vol. 15, no. 14, pp. 5481–5487, Jul. 2022, doi: 10.5194/gmd-15-5481-2022.
7. Tavakoli, N. (2019). Modeling Genome Data Using Bidirectional LSTM. 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), 2, 183.