Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

# КВАЛІФІКАЦІЙНА РОБОТА

на тему *Метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж*

Рівень вищої освіти: *другий (магістерський)*

Галузь знань: *12 – Інформаційні технології* (Шифр і найменування)

Спеціальність: *122 – Комп'ютерні науки* (Код і найменування)

Освітня програма: *Комп'ютерні науки* (Назва)

Виконав: *студент 2 курсу, група КНм-24-1* (Курс, група виконавця) _____ (Підпис) *Олександр БОНДАР* (Ім'я, ПРІЗВИЩЕ)

Керівник: *к.т.н., доцент кафедри КН* (Науковий ступінь, посада) _____ (Підпис) *Олександр ПАСІЧНИК* (Ім'я, ПРІЗВИЩЕ)

Нормоконтроль: *к.т.н., доцент кафедри КН* (Науковий ступінь, посада) _____ (Підпис) *Руслан БАГРІЙ* (Ім'я, ПРІЗВИЩЕ)

До захисту допускаю:
*Зав. кафедри КН, д.т.н., професор* _____ (Підпис) *Олександр БАРМАК* (Ім'я, ПРІЗВИЩЕ)

*17 грудня 2025* р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет *інформаційних технологій*
Кафедра *комп'ютерних наук*
Освітній ступінь *магістр*
Галузь знань *12 – Інформаційні технології*
Спеціальність *122 – Комп'ютерні науки*

ЗАТВЕРДЖУЮ
Завідувач кафедри *комп'ютерних наук*

_____
(підпис)
*д.т.н., професор Олександр БАРМАК*
«*29*» *08* *2025* року

**ЗАВДАННЯ**
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

1. Тема кваліфікаційної роботи: *«Метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж»*
2. Завдання видано студенту *Олександру БОНДАРУ*
(Ім'я, ПРІЗВИЩЕ)
3. Керівник роботи *доцент кафедри КН Олександр ПАСІЧНИК*
(Ім'я, ПРІЗВИЩЕ)
4. Затверджені наказом університету від «*25*» *08* *2025* р. № *65*
5. Дата видачі завдання студенту: «*28*» *08* *2025* р.
6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

*Мета роботи – підвищення точності виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж. Для досягнення мети слід виконати такі задачі: провести аналіз методів виявлення шахрайських транзакцій у фінансових операціях; провести аналіз можливостей, переваг та недоліків згорткових нейронних мереж для виявлення шахрайських транзакцій у фінансових операціях; спроєктувати метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж; виконати програмну реалізацію методу; виконати дослідження точності спроєктованого методу.*

*Ключові слова: згорткова нейронна мережа, шахрайські транзакції, глибоке навчання, фінансові операції, нормалізація, масштабування, балансування даних.*

7. Календарний план виконання кваліфікаційної роботи:

| № | Назва етапів (розділів) кваліфікаційної роботи бакалавра | Термін виконання | Примітка |
|---|---|---|---|
| *1* | *Вибір напряму дослідження та узгодження теми кваліфікаційної роботи з керівником, складання календарного графіка виконання роботи* | *вересень 2025* | *Виконано* |
| *2* | *Ознайомлення з предметною областю, аналіз існуючих методів і моделей, формулювання мети та завдань дослідження, визначення об'єкта й предмета дослідження* | *вересень 2025* | *Виконано* |
| *3* | *Розробка методу чи моделі для вирішення обраного завдання, опис архітектури рішення* | *жовтень 2025* | *Виконано* |
| *4* | *Програмна реалізація методу чи моделі* | *жовтень 2025* | *Виконано* |
| *5* | *Дослідження ефективності та експериментальна перевірка результатів, порівняння з відомими підходами* | *листопад 2025* | *Виконано* |
| *6* | *Написання пояснювальної записки, оформлення відповідно до вимог, врахування зауважень керівника* | *листопад 2025* | *Виконано* |
| *7* | *Підготовка презентаційних матеріалів та попередній захист* | *листопад 2025* | *Виконано* |
| *8* | *Перевірка пояснювальної записки на відповідність вимогам оформлення (нормоконтроль) та перевірка на академічну доброчесність. Отримання відгуку керівника та рецензії.* | *грудень 2025* | *Виконано* |
| *9* | *Публічний захист кваліфікаційної роботи* | *грудень 2025* | *Виконано* |

Виконавець: *студент групи КНм-24-1* _____ *Олександр БОНДАР*

Група виконавця — Підпис — Ім'я, ПРІЗВИЩЕ

Керівник: *доцент кафедри КН* _____ *Олександр ПАСІЧНИК*

Науковий ступінь, посада — Підпис — Ім'я, ПРІЗВИЩЕ

# Реферат

Кваліфікаційна робота магістра присвячена розробці методу виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж.

**Актуальність теми.** Сучасний етап розвитку технологій значною мірою зумовлений стрімким прогресом у сфері цифрових технологій, інформаційних систем та програмних рішень. Вони відіграють вирішальну роль у повсякденному житті та у різних галузях, включаючи фінанси та електронну комерцію. У зв'язку з цим, виявлення шахрайських транзакцій у фінансових операціях є важливою задачею для підвищення безпеки платежів та запобігання економічним втратам.

У цьому контексті актуальним є застосування методів інтелектуального аналізу даних та штучного інтелекту для автоматизованого та швидкого виявлення шахрайської діяльності. Згорткові нейронні мережі, поширені у задачах класифікації та розпізнавання патернів, демонструють високий потенціал у моделюванні складних залежностей у транзакційних даних, що робить їх доцільним інструментом для вирішення задачі детекції шахрайства.

**Мета і задачі роботи.** Метою кваліфікаційної роботи магістра є підвищення точності виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж.

Для досягнення поставленої мети визначені такі задачі дослідження:

– провести аналіз методів виявлення шахрайських транзакцій у фінансових операціях;

– провести аналіз можливостей, переваг та недоліків згорткових нейронних мереж для виявлення шахрайських транзакцій у фінансових операціях;

– спроєктувати метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж;

– виконати програмну реалізацію методу виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж;

– виконати дослідження точності виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж.

**Об'єкт дослідження.** Процес виявлення шахрайських транзакцій у фінансових операціях.

**Предмет дослідження.** Нейромережеві методи та технології для виявлення шахрайських транзакцій у фінансових операціях.

**Методи дослідження.** У роботі використано методи глибокого навчання, зокрема згорткові нейронні мережі (CNN) прямого поширення, а також метод SMOTE для формування збалансованого навчального набору даних.

**Наукова новизна одержаних результатів** – удосконалено метод виявлення шахрайських транзакцій у фінансових операціях, який відрізняється від існуючих застосуванням згорткової нейронної мережі з попередньою нормалізацією та масштабуванням ключових класифікаційних параметрів та балансування даних, що дозволяє підвищити точність класифікації фінансових транзакцій.

**Апробація результатів кваліфікаційної роботи та публікації**. Основні наукові та практичні результати пройшли апробацію на науково-практичній конференції – XVII Всеукраїнська науково-практична конференція "Актуальні проблеми комп'ютерних наук (АПКН – 2025)", м. Хмельницький, ХНУ, 14-15 листопада 2024 р. (Бондар О.П, Пасічник О.А., Скрипник Т.К., Петровський С.С. Метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж // Збірник наукових праць за матеріалами XVII Всеукраїнської науково-практичної конференції "Актуальні проблеми комп'ютерних наук (АПКН – 2025)".– Хмельницький: ХНУ, 2025. – С. 42 – 45.)

**Структура та обсяг роботи**. Кваліфікаційна робота магістра складається з вступу, чотирьох розділів, висновків, переліку посилань та додатків. Загальний

обсяг роботи становить 103 сторінок, з яких 70 сторінка основного тексту, і включає 23 рисунки та 8 таблиць.

**Ключові слова:** згорткова нейронна мережа, шахрайські транзакції, глибоке навчання, фінансові операції, нормалізація, масштабування, балансування даних.

**Зміст**

## Перелік скорочень

| Скорочення, термін, позначення | Пояснення |
|---|---|
| ШІ | Штучний інтелект |
| CNN | Convolutional neural network |
| ANN | Artificial neural network |
| ЗНМ | Згорткова нейронна мережа |
| AI | Artificial intelligence |
| AE | Auto Encoder |
| SMOTE | Synthetic Minority Over-sampling Technique |
| NN | Neural Network |

# Вступ

**Актуальність теми.** Сучасний етап розвитку технологій значною мірою зумовлений стрімким прогресом у сфері цифрових технологій, інформаційних систем та програмних рішень. Вони відіграють вирішальну роль у повсякденному житті та у різних галузях, включаючи фінанси та електронну комерцію. У зв'язку з цим, виявлення шахрайських транзакцій у фінансових операціях є важливою задачею для підвищення безпеки платежів та запобігання економічним втратам.

У цьому контексті актуальним є застосування методів інтелектуального аналізу даних та штучного інтелекту для автоматизованого та швидкого виявлення шахрайської діяльності. Згорткові нейронні мережі, поширені у задачах класифікації та розпізнавання патернів, демонструють високий потенціал у моделюванні складних залежностей у транзакційних даних, що робить їх доцільним інструментом для вирішення задачі детекції шахрайства.

**Мета і задачі роботи.** Метою кваліфікаційної роботи магістра є підвищення точності виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж.

Для досягнення поставленої мети визначені такі задачі дослідження:

- провести аналіз методів виявлення шахрайських транзакцій у фінансових операціях;
- провести аналіз можливостей, переваг та недоліків згорткових нейронних мереж для виявлення шахрайських транзакцій у фінансових операціях;
- спроєктувати метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж;
- виконати програмну реалізацію методу;
- виконати дослідження точності виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж.

**Об'єкт дослідження.** Процес виявлення шахрайських транзакцій у фінансових операціях.

**Предмет дослідження.** Нейромережеві методи та технології для виявлення виявлення шахрайських транзакцій у фінансових операціях.

**Методи дослідження.** У роботі використано методи глибокого навчання, зокрема згорткові нейронні мережі (CNN) прямого поширення, а також метод SMOTE для формування збалансованого навчального набору даних.

**Наукова новизна одержаних результатів** – удосконалено метод виявлення шахрайських транзакцій у фінансових операціях, який відрізняється від існуючих застосуванням згорткової нейронної мережі з попередньою нормалізацією та масштабуванням ключових класифікаційних параметрів та балансування даних, що дозволяє підвищити точність класифікації фінансових транзакцій.

**Апробація результатів кваліфікаційної роботи та публікації**. Основні наукові та практичні результати пройшли апробацію на науково-практичній конференції – XVII Всеукраїнська науково-практична конференція “Актуальні проблеми комп’ютерних наук (АПКН – 2025)”, м. Хмельницький, ХНУ, 14-15 листопада 2024 р. (Бондар О.П, Пасічник О.А., Скрипник Т.К., Петровський С.С. Метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж // Збірник наукових праць за матеріалами XVII Всеукраїнської науково-практичної конференції “Актуальні проблеми комп’ютерних наук (АПКН – 2025)”.– Хмельницький: ХНУ, 2025. – С. 42 – 45.)

**Структура та обсяг роботи**. Кваліфікаційна робота магістра складається з вступу, чотирьох розділів, висновків, переліку посилань та додатків. Загальний обсяг роботи становить 103 сторінок, з яких 70 сторінка основного тексту, і включає 23 рисунки та 8 таблиць.

# Розділ 1. Аналіз сучасного стану досліджуваної проблеми

Стрімкий розвиток комп'ютерних наук та технологій штучних нейронних мереж зумовлює появу нових інструментів для розв'язання складних обчислювальних задач. Сучасні методи забезпечують можливість ефективної роботи з великими обсягами даних, інтелектуального їх опрацювання та автоматизації значної частини рутинних або повторюваних процесів. Глибокі нейронні моделі, що здатні адаптуватися до змінних умов, поступово перетворюються на один із ключових компонентів сучасних інформаційних систем. Їх застосування відкриває перспективи для вирішення проблем, які тривалий час були недосяжними для традиційних алгоритмічних підходів. Одним із практично важливих напрямів використання таких технологій є виявлення фінансових операцій із ознаками шахрайства.

## 1.1 Характеристика задачі виявлення шахрайських транзакцій

Виявлення фінансового шахрайства, зокрема підозрілих транзакцій та несанкціонованих операцій з платіжними картками, є комплексною задачею, що потребує поєднання різнорідних джерел даних і застосування сучасних аналітичних технологій. У сучасних дослідженнях та практичних системах для розв'язання цієї проблеми використовуються кілька ключових підходів: від класичних методів аналізу транзакційної активності та правил виявлення аномалій до алгоритмів машинного навчання й глибоких нейронних моделей. Ці технології забезпечують більш точне, швидке й адаптивне розпізнавання шахрайських дій у фінансових системах.

Традиційні методи виявлення шахрайських транзакцій з платіжними картками спиралися переважно на ручні правила, базову статистику та простий аналіз поведінкових шаблонів користувачів. Такий підхід був поширений на ранніх етапах електронного банкінгу, коли обсяги транзакцій були невеликими, і

шахрайські схеми відносно стандартними. Проте з розширенням використання карток, появою інтернет-платежів та ростом числа транзакцій класичні рішення стали недостатньо ефективними, їм важко було вчасно реагувати на нові, нестандартні шахрайські моделі. Саме тому з’явилася потреба в більш гнучких і адаптивних системах, здатних аналізувати великі потоки даних, виявляти аномалії та складні патерни, які не вписуються у заздалегідь визначені правила [1].

Поява шахрайських транзакцій з банківськими картками збігається з розвитком електронних та безготівкових способів оплати. Збільшення кількості випадків таких дій підштовхнуло фінансові організації до впровадження механізмів підвищеної безпеки, в даному випадку систем спостереження та виявлення сумнівних транзакцій.

Кредитні картки можуть бути фізично викрадені з гаманця або отримані віртуально з незахищених веб-сайтів, через витік даних або схеми крадіжки особистих даних. Так, згідно із дослідженнями [2], у 2019 році хакери зламали бази даних Capital One та оприлюднили інформацію про кредитні картки понад 100 мільйонів людей.

Захист фінансової інформації є важливим але, попри захист, який вони пропонується банками, кредитні картки не є захищеними від шахрайства. Навпаки, вони постійно є головною мішенню для злодіїв особистих даних. Згідно із даними, наведеними в роботі [3], у 2024 році до Федеральної торгової комісії було подано майже 450 000 повідомлень про крадіжку особистих даних з кредитних карток.

Шахрайство з кредитними картками буває різних форм, кожна з яких має різний спосіб націлювання на користувачів. В роботі [4] наведено чіткий розбивку найпоширеніших типів, їх впливу та фінансових наслідків:

– шахрайство без пред'явлення картки при якому шахраї використовують вкрадену картку для онлайн-покупок або покупок по телефону без пред'явлення фізичної картки,

– шахрайство з втраченою або вкраденою карткою при якому фізична картка втрачена або вкрадена та використовується для покупок,

– шахрайство з заявкою на кредитну картку при якому шахраї використовують вкрадену особисту інформацію для відкриття нових рахунків кредитних карток на ім’я користувача,

– шахрайство із захопленням облікового запису при якому зловмисники захоплюють обліковий запис кредитної картки, змінюють паролі та здійснюють несанкціоновані транзакції,

– скімінг та клонування кредитних карток при якому пристрої фіксують дані картки, які потім використовуються шахраями для клонування картки.

Як зазначається в роботі [5], виявлення шахрайства з кредитними картками стосується поєднання інструментів, технологій та процесів для запобігання несанкціонованим покупкам як в онлайн-, так і в фізичному середовищі з метою перевірити, чи є власник картки тим, за кого він себе видає, та підтвердити легітимність кожної транзакції

Методи виявлення можуть охоплювати як базові механізми автентифікації, так і складні підходи, що використовують аналітику на основі машинного навчання. До таких методів належить багатофакторна автентифікація, яка передбачає перевірку особи за кількома рівнями, зокрема за допомогою SMS-кодів або підтверджень у спеціальних застосунках. Також застосовується технологія 3-D Secure, що передбачає введення додаткового пароля або коду перед підтвердженням транзакції.

Поширеним інструментом є біометрична перевірка, яка використовує відбитки пальців, розпізнавання обличчя чи голосу для підтвердження особи. Окрім цього, для одноразової автентифікації активно застосовуються одноразові паролі, що надсилаються користувачеві у вигляді унікальних кодів.Сучасні системи також відстежують аномалії транзакцій. Наприклад, замовлення з IP-адреси високого ризику або пристрою, який раніше не був пов'язаний з клієнтом, може викликати сповіщення. Залежно від встановленої системи виявлення шахрайства з кредитними картками, це може відбуватися в режимі реального часу або шляхом ретроспективного розслідування.

Виявлення шахрайства з використанням банківських карток у теорії виглядає як класична задача класифікації – визначити, чи є транзакція шахрайською. Однак з операційної точки зору це значно складніше завдання, оскільки йдеться про обробку потокових даних у реальному часі, високий рівень конкурентності, наявність дуже складних і нерівномірно представлених класів, слабке маркування та постійно змінювану поведінку зловмисників. Моделі, які демонструють високу точність в офлайн-умовах, можуть зіткнутися зі значними труднощами під час впровадження, якщо ці операційні аспекти не буде враховано окремо [6].

Шахрайські транзакції зазвичай складають незначну частку від загальної кількості транзакцій (часто менше 1%). Це негативно впливає на функціонування різноманітних алгоритмів машинного навчання та метрик оцінки: точність є малоефективною, у той час як важливими стають показники точності/повного зважування та чутливість до витрат. Дисбаланс також викликає проблему, при якій моделі можуть надмірно налаштовуватись під клас більшості, якщо не застосовуються методи повторної вибірки, відповідне зважування втрат, методи синтетичної меншості або підходи до виявлення аномалій.

Коли система обробляє транзакцію, що надходить з картки, у неї є секунди або менше, щоб вирішити, чи дозволити її, позначити позначкою чи заблокувати. Складні моделі стануть занадто повільними, якщо їх не оптимізувати або апроксимувати. Затримка також передбачає час перевірки людиною: занадто багато хибних спрацьовувань уповільнять операції та дратуватимуть клієнтів [7].

Незаконні методи швидко розвиваються; нові шахрайства, фальшиві ідентичності, видавання себе за іншу особу продавцем та скоординована діяльність ботів часто з'являються та зникають. Моделі, навчені на історичних даних, зазнають занепаду, якщо їх не перенавчити або не перекалібрувати. Важливо розпізнавати та реагувати на дрейф концепцій – з точки зору онлайн-ретаргетингу, інкрементного навчання або ансамблів, усвідомлюючи дрейф [8].

При отриманні транзакції з картки, система має лише декілька секунд (або менше) для визначення, чи може вона її обробити, позначити для подальшої

перевірки людиною або відхилити. Розширені моделі (такі як глибокі нейронні мережі або великі ансамблі) зазвичай демонструють значну повільність, якщо їх не оптимізувати чи апроксимувати. Затримка також включає час, витрачений на подальшу перевірку людиною: надто багато хибних тривог призводить до зупинки операцій і викликає роздратування у клієнтів. Для забезпечення реального часу необхідна масштабована архітектура потокової обробки [9].

Слідчим та командам з дотримання вимог потрібні чіткі пояснення. Моделі чорної скриньки з високою точністю, але низькою інтерпретаційністю перешкоджають розслідуванням, звітності та дотриманню нормативних вимог. Зазвичай використовуються гібридні моделі або пояснювальні системи, що не залежать від моделі [10].

Тактики шахрайства розвиваються з великою швидкістю. Нові схеми шахрайства, синтетичні ідентичності, підміна продавців та скоординована поведінка ботів постійно з'являються і зникають. Моделі, що навчалися на основі історичних даних, зазнають деградації у випадку, якщо їх не оновлюють або не адаптують. Важливо виявляти та реагувати на дрейф концепцій [11].

Глибоке проникнення фінансових інституцій в усі сфери життя людини з охопленням усіх верств населення, вікових і соціальних груп, у поєднанні з тотальною цифровізацією банківської та торговельної сфер, що постійно наростає, робить украй нагальним, необхідним і практично безальтернативним впровадження цифрових рішень, заснованих на найсучасніших досягненнях комп’ютерних наук, для виявлення шахрайських транзакцій з метою їх блокування та запобігання протиправним діям.

## 1.2 Аналіз публікацій та наукових підходів для виявлення шахрайських транзакцій у фінансових операціях

Роль ШІ у виявленні фінансового шахрайства суттєво змінилася, перейшовши від статичних систем, заснованих на правилах, до динамічних,

адаптивних алгоритмів. Структури раннього виявлення в основному залежали від заздалегідь визначених правил та ручного перегляду транзакцій, де сповіщення запускалися фіксованими параметрами, такими як сума транзакцій, частота або географічне розташування. Хоча ці методи пропонували базовий рівень захисту, їм бракувало можливості адаптуватися до нових та складних шахрайських схем. Як наслідок, такі системи часто видавали високий рівень хибнопозитивних результатів, що призводило до неефективних розслідувань та погіршення обслуговування клієнтів.

У роботі [12] показано, що сучасне машинне навчання докорінно змінило підхід до виявлення шахрайства: системи на основі штучного інтелекту здатні в реальному часі аналізувати величезні обсяги даних, розпізнавати аномальні поведінкові патерни та постійно адаптуватися до нових загроз. Завдяки поєднанню глибокого навчання, нейронних мереж і поведінкової аналітики фінансові установи отримали значно швидше, точніше та гнучкіше виявлення шахрайських операцій, що суттєво підвищує рівень безпеки й операційної ефективності.

Ці переваги підтверджуються й конкретними цифрами. У 2023 році системи на основі ШІ запобігли втратам на суму понад 20 млрд доларів США [13]. Зокрема, модулі обробки природної мови виявили більше 60 % фішингових листів, технології комп’ютерного зору розпізнали підроблені документи на суму близько 3 млрд доларів, прогнозна аналітика скоротила середній час виявлення шахрайства на 85 %, а чат-боти на базі ШІ автоматизували обробку 75 % клієнтських звернень щодо підозрілої активності.

Практична реалізація таких систем найчастіше поєднує навчання з учителем і без учителя. У реальному розгортанні 2023 року гібридна архітектура XGBoost + One-Class SVM та Deep Autoencoder забезпечила загальне виявлення на рівні 94,7 % і скоротила кількість хибних спрацьовувань на 54 %, попри те, що компоненти без учителя потребують перенавчання кожні 7–14 днів через дрейф поведінки клієнтів [14].

Серед методів з учителем особливе місце посідають дерева рішень та їхні ансамблі — одні з найбільш досліджених і затребуваних алгоритмів, хоча їхній потенціал у фінансовій сфері часто недооцінюється [15]. У банківській практиці вони широко застосовуються для аналізу історичних даних, виділення ключових змінних і прогнозування кредитоспроможності та ризиків [16].

Найпопулярнішим ансамблевим методом є випадковий ліс, який створює багато дерев рішень, навчених бэггінгом на випадкових підмножинах даних і ознак, забезпечуючи високу стабільність і точність навіть за мінімального налаштування [17]. На наборі даних European Cardholders 2013 він досяг AUC-ROC 0,97 та F1-міри 0,89, перевершивши одиночне дерево на 15 % за повнотою виявлення рідкісних шахрайських транзакцій завдяки зменшенню перенавчання. Однак метод має високу обчислювальну складність для масивів понад 1 млн записів, що обмежує його використання в реальному часі без оптимізації чи паралелізації.

У сучасній банківській практиці машинне навчання визначається як підгалузь штучного інтелекту, що дозволяє створювати самонавчальні моделі, здатні без постійного людського втручання класифікувати транзакції та прогнозувати ризики [18]. Аналіз патернів (час, сума, геолокація, історія поведінки) дає змогу досягати точності 95–99 % у реальних системах фрод-детекції.

Згідно [19] при прогнозуванні кредитних дефолтів інтеграція кредитної історії, демографічних і поведінкових ознак дозволяє моделям – від логістичної регресії до AdaBoost – досягати практично ідеальної точності на тестових вибірках. Водночас залишаються актуальними проблеми сильного дисбалансу класів, швидкого старіння моделей і питання етичного використання конфіденційних даних, що стимулює перехід до гібридних архітектур із глибоким навчанням.

Особливе місце серед методів без учителя посідають автокодери. У роботі [20] автокодер навчався лише на нормальних транзакціях, а аномалії виявлялися за величиною помилки реконструкції. При сильному дисбалансі класів стандартний автокодер може добре відтворювати й шахрайські приклади, тому частіше використовують варіаційні, денойзингові або гібридні модифікації. Автокодери

особливо ефективні в напівконтрольованому сценарії з мінімальною кількістю маркованих аномалій: модель засвоює структуру нормальної поведінки та позначає значні відхилення як підозрілі. Це робить їх цінним інструментом у фрод-детекції, кібербезпеці та промисловому контролі якості [21].

Поряд з автокодерами популярним методом без учителя є кластерний аналіз. У роботі [22] його визначають як групування об’єктів, де елементи одного кластера максимально схожі між собою та відрізняються від інших. Оскільки алгоритми по-різному інтерпретують поняття «кластера», задачу розглядають як багатоцільову оптимізацію. Вибір алгоритму, метрики відстані та гіперпараметрів залежить від структури даних і аналітичних цілей. Процес кластеризації є ітеративним і часто вимагає неодноразового коригування попередньої обробки, моделі та параметрів для досягнення прийнятних результатів.

Водночас у задачах, що вимагають автоматичного вилучення ієрархічних ознак, провідну роль відіграють згорткові нейронні мережі (ЗНМ). Як показано в [23], ЗНМ є різновидом мереж із прямим поширенням сигналу, які вивчають ознаки шляхом оптимізації згорткових ядер. Завдяки параметричному спільному використанню ваг і локальній сприйняття вони ефективно працюють із просторово-структурованими даними – зображеннями, текстом, аудіо – і на сьогодні залишаються стандартом у комп’ютерному зорі та суміжних галузях.

Традиційні системи виявлення фінансового шахрайства, побудовані на жорстких експертних правилах і простих статистичних методах, дедалі частіше виявляються недостатніми через швидке ускладнення та мінливість шахрайських схем [24, 25]. Такі підходи не здатні забезпечувати необхідну точність і швидкодію при обробці мільйонів транзакцій у реальному часі. Натомість сучасні дослідження [26, 27] демонструють високу ефективність моделей глибокого навчання, зокрема згорткових нейронних мереж, які завдяки потужним механізмам автоматичного вилучення ознак успішно виявляють приховані аномалії та моделюють складні просторово-часові залежності у фінансових даних, що дозволяє суттєво підвищити точність і оперативність детекції шахрайства.

У дослідженні Faye та Zhang [28] згорткові нейронні мережі (CNN) показали значні переваги в виявленні шахрайських фінансових транзакцій, перевершивши логістичну регресію, SVM, Random Forest, KNN та градієнтний бустинг за точністю, precision, recall та F1-мірою завдяки потужному вилученню ознак із високовимірних даних. CNN адаптивні до нових схем шахрайства, ефективно обробляють великі обсяги даних і зменшують залежність від ручних правил, хоча вимагають значних обчислювальних ресурсів.

У дослідженні [29] (Security and Communication Networks) 1D-CNN досягли точності понад 98% на датасетах кредитних карток, перевершивши MLP, завдяки автоматичному вилученню патернів із послідовних даних, нормалізації пакетів та SMOTE для незбалансованих наборів. Переваги: менші обчислювальні вимоги, придатність для реального часу на CPU та зменшення ручного інжинірингу ознак порівняно з традиційними методами (логістична регресія, SVM). Недоліки — тривалий час навчання та ризик перенавчання, що потребує оптимізації.

Одним із ключових етапів підготовки даних у fraud detection є масштабування та нормалізація ознак. Дослідження на датасеті ULB 2013 (2022) показало, що без нормалізації великі ознаки (сума транзакції, баланс) домінують у логістичній регресії та SVM, знижуючи AUC до 0,67–0,71; після StandardScaler або MinMaxScaler показник зростав до 0,94–0,96. У реальному впровадженні в європейському банку (2023) відсутність масштабування в XGBoost знизила recall на 18 %, а RobustScaler відновив його до 93 % і скоротив час навчання на 11 % [30]. Таким чином, масштабування є обов’язковим для виробничих пайплайнів, оскільки його ігнорування значно погіршує якість моделей.

У роботі [31] нормалізацію визначають як приведення ознак до спільної шкали, включаючи центрування, масштабування та складніші методи, як квантильна нормалізація для узгодження розподілів.

Проблема дисбалансу класів критична для fraud detection, де шахрайські транзакції становлять <1 % даних. Для її вирішення застосовують undersampling (зменшення мажоритарного класу) та oversampling (збільшення міноритарного) [32].

Прості методи дублювання чи видалення поступаються синтетичним, зокрема SMOTE, який генерує нові приклади шляхом інтерполяції між сусідами, значно підвищуючи стійкість моделей на незбалансованих даних.

### 1.3 Огляд архітектур та методів виявлення шахрайських транзакцій у фінансових операціях

У предметній області виявлення шахрайських транзакцій у фінансових операціях широко застосовуються алгоритми машинного навчання з учителем, зокрема ансамблеві методи на основі дерев рішень. Серед них особливе місце посідають випадковий ліс (Random Forest Classifier) та методи градієнтного бустингу, такі як XGBoost і LightGBM. Ці алгоритми демонструють високу ефективність у задачах класифікації з сильно незбалансованими даними, типовими для fraud detection, і часто використовуються як базові (baseline) моделі перед переходом до глибокого навчання.

Класифікатор Random Forest [33] – це ансамблевий метод машинного навчання, який створює багато дерев рішень із бутстрап-вибіркою та випадковим відбором ознак, агрегує прогнози голосуванням більшістю (рисунок 1.1). Він зменшує перенавчання та дисперсію, забезпечує високу точність, стійкість до шуму, обробку змішаних ознак і оцінку їх важливості. Хоча й обчислювально вимогливий, він ефективний у біоінформатиці, кібербезпеці та аналізі зображень з мінімальним налаштуванням.

У дослідженні [34] Alexander Subagio та Ditdit Nugeraha Utama запропонували гібридний метод виявлення шахрайських транзакцій з кредитними картками, де MLP вилучає ознаки, а Random Forest класифікує. Ансамблевий підхід Random Forest забезпечує стійкі прогнози, а гібридна модель досягла найкращих результатів: точність 99,949%, precision 87,097%, recall 82,653% та F1-score 84,817%, перевершивши окремі MLP і Random Forest.

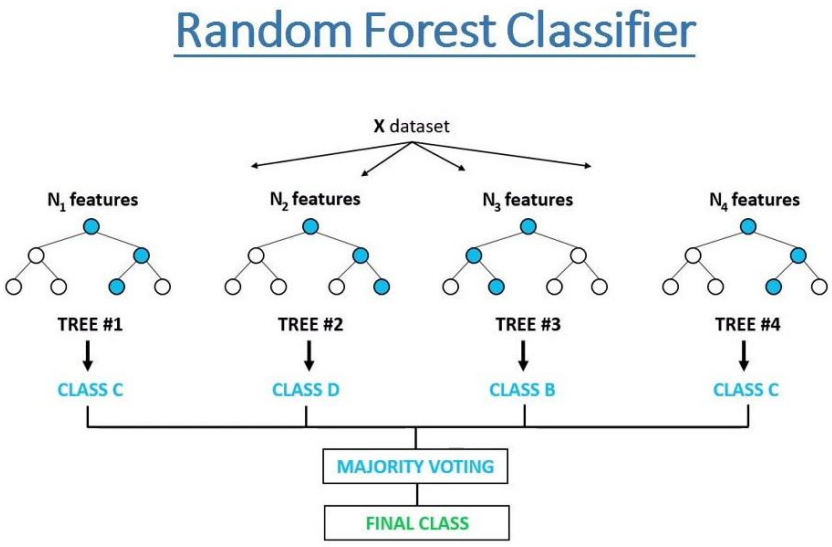



Рисунок 1.1 – Схема роботи методу Випадкового лісу

Gradient boosting [35] – це метод машинного навчання, що послідовно будує ансамбль слабких моделей (зазвичай дерев рішень), коригуючи помилки попередніх шляхом мінімізації диференційованої функції втрат за допомогою псевдо-решток (рисунок 1.2). Основні реалізації: GBM, XGBoost, LightGBM, CatBoost. Переваги: висока точність і гнучкість, хоча схильний до перенавчання (регуляризується learning rate та subsampling) та обчислювально вимогливий. Застосовується в пошуковому ранжуванні, фізиці високих енергій (відкриття бозона Хіггса) та геології.

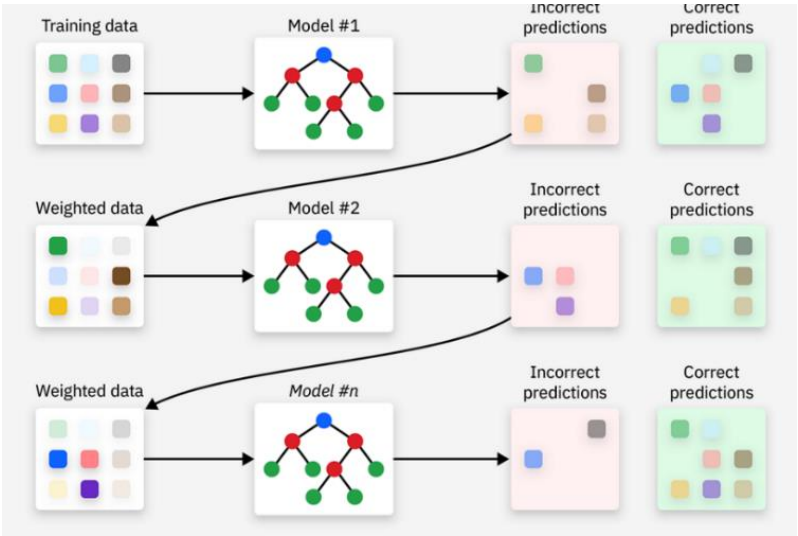



Рисунок 1.2 – Схема роботи Gradient boosting

У дослідженні [36] запропоновано гібридний метод AEELG (AutoEncoder Enhanced LightGBM) для виявлення шахрайських транзакцій з кредитними картками, де LightGBM – ефективна реалізація градієнтного бустингу з гістограмним біннингом, EFB та GOSS – є основним класифікатором. Комбінація з автоенкодером для вилучення ознак та SMOTE для балансування класів забезпечила високі метрики: recall 94,85%, BCR 97%, F-міра 80,27%, AUC 96,83%, перевершивши Random Forest, AdaBoost, Bagging та CNN за чутливістю до шахрайства на сильно незбалансованих даних. Автори висновують, що LightGBM з автоенкодером – потужне та масштабоване рішення для виявлення аномалій у фінансових даних.

### 1.4 Постановка задачі дослідження

Виконано огляд теоретичних підходів виявлення шахрайських транзакцій у фінансових операціях, проаналізовано можливості, переваги та недоліки згорткових нейронних мереж для виявлення шахрайських транзакцій у фінансових операціях, проаналізовано архітектури та методи виявлення шахрайських фінансових транзакцій.

Мета кваліфікаційної роботи магістра – підвищення точності виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж.

Для досягнення поставленої мети визначені такі задачі дослідження:

– провести аналіз методів виявлення шахрайських транзакцій у фінансових операціях;

– провести аналіз можливостей, переваг та недоліків згорткових нейронних мереж для виявлення шахрайських транзакцій у фінансових операціях;

– спроєктувати метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж;

– виконати програмну реалізацію методу;

– виконати дослідження точності виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж.

# Розділ 2 Проєктування методу виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж

## 2.1 Концепція та схема методу

Загальна схема методу зображена на рисунку 2.1. Як вхідні використовуються дані щодо фінансових транзакцій.


Рисунок 2.1 – Загальна схема методу

В межах єдиного покрокового послідовного процесу обробки даних відбувається отримання параметрів транзакції, їх попередня обробка та підготовка до подання в нейромережу, подальше автоматичне витягування інформативних ознак згортковою нейронною мережею, аналіз отриманих представлень із застосуванням бінарної класифікації, а також формування вихідних даних у вигляді результату перевірки транзакції на наявність шахрайства.

Спочатку виконується завантаження вхідних даних та вибір відповідного екземпляра з набору даних. З обраного запису виділяються ключові ознаки, які характеризують фінансову операцію, зокрема час виконання транзакції (Time), сума транзакції (Amount), а також набір похідних параметрів (V1, V2, ..., V28).

Ці ознаки об’єднуються у вхідний вектор ознак (feature vector), який складається з числових значень Time, V1, V2, ..., V28, Amount. Паралельно з формуванням вхідного вектора визначається мітка класу (Class), яка вказує, чи є транзакція легітимною (звичайною) чи шахрайською. Ця мітка використовується під час навчання моделі, тоді як на етапі інференсу (класифікації нової транзакції) вона відсутня або слугує для оцінки результату.

Сформований вхідний вектор разом із (за наявності) міткою класу передається на етап попередньої обробки.

Попередня обробка включає такі операції:

- нормалізації,
- масштабування,
- зміни розмірності.

Далі відбувається отримання ознак згортковою нейронною мережею з метою пошуку прихованих патернів. На цьому кроці використовується попередньо навчана модель згорткової нейронної мережі.

На етапі верифікації даних підготовлений вхідний вектор ознак (Time, V1-V28, Amount) подається на вхід заздалегідь завантаженої згорткової нейронної мережі (CNN). Модель CNN послідовно обробляє ці дані, поступово виділяючи та узагальнюючи ключові закономірності, характерні для фінансових транзакцій.

Обробка починається з першого згорткового шару (Convolutional layer), який застосовує фільтри до вхідних даних для виявлення локальних патернів і взаємозв’язків між ознаками. Далі йде шар максимального пулінгу (Max pooling layer), що зменшує просторову розмірність даних, зберігаючи при цьому найбільш значущі ознаки та знижуючи обчислювальну складність. Ця послідовність – згортковий шар з наступним шаром максимального пулінгу – повторюється, дозволяючи моделі на кожному рівні витягувати дедалі складніші та абстрактніші характеристики транзакції, відсіювати шум і підсилювати інформативні сигнали.

Після кількох таких блоків дані проходять через шар вирівнювання (Flatten layer), який перетворює багатовимірну карту ознак у одновимірний вектор. Далі слідує повнозв’язний шар (Dense layer), за яким розташовано шар відсікання (Dropout layer) для запобігання перенавчанню шляхом випадкового відключення частини нейронів під час навчання. На завершення розміщено ще один повнозв’язний шар (Dense layer), який формує остаточне представлення та видає прогноз.

На наступному кроці відбувається аналіз даних результатом якого є бінарна класифікація транзакій у фінансовій сфері на шахрайські та правомірні та обчислюється імовірність шахрайства.

Для бінарної класифікації використовується активаційна функція sigmoid:

$$p = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.1}$$

де:

- z – зважена сума ознак,
- p∈[0,1] – ймовірність того, що транзакція є шахрайською.
- Отримане значення p:
- p≈0 – транзакція майже напевно правомірна
- p≈1 – транзакція напевно шахрайська

Це значення і є оцінкою ризику шахрайства.

Для переходу від ймовірності до класу застосовується поріг τ =0.5:

$$class = \begin{cases} 1(Fraud), & p \geq T \\ 0(Non\ fraud), & p < T \end{cases} \quad (2.2)$$

Поріг може зменшений до 0.3 для мінімізації пропусків шахрайства або збільшений до 0.7 для зменшення хибних спрацювань.

На кроці формування вихідних даних (Output data) результати, отримані від згорткової нейронної мережі (CNN), узагальнюються та подаються у зрозумілій для користувача формі. Після завершення обробки вхідного вектора модель видає три ключові компоненти: значення прогнозу (prediction value), прогнозований клас (predicted class) та, за наявності, справжній клас транзакції (True class), який доступний лише під час тестування або оцінки моделі на відомих даних.

Вихідними даними є результат перевірки транзакцій с поділом їх на шахрайські та правомірні.

Функціонування методу ґрунтується на застосуванні попередньо навченої згорткової нейронної мережі, яка виконує аналіз транзакцій та оцінювання ймовірності шахрайства.

Процес створення ефективної моделі виявлення шахрайства розпочинається з отримання даних з датасету, які включають критичні ознаки, такі як час транзакції (Time), приховані ознаки, отримані методом PCA (V1...V28), та суму транзакції (Amount). Після отримання повного набору даних здійснюється ключовий підготовчий крок – розділення на навчальну та валідаційну вибірки. Навчальна вибірка виділяється для безпосереднього навчання нейронної мережі, а валідаційна вибірка використовується для моніторингу прогресу навчання, налаштування гіперпараметрів і запобігання перенавчанню.

Далі слідує попередня обробка даних. Для навчальної вибірки цей процес є комплексним і включає нормалізацію та масштабування числових ознак, що

забезпечує їхнє рівномірне входження до мережі, а також застосування методу балансування класів (SMOTE) критично важливого для вирішення проблеми суттєвого дисбалансу між незначною кількістю шахрайських та великою кількістю нормальних транзакцій. Крім того, відбувається зміна розмірності (решейпінг) даних, щоб підготувати їх до формату, необхідного для входу у згорткову нейронну мережу.

Попередня обробка даних з валідаційної вибірки обмежується лише нормалізацією та масштабуванням, використовуючи параметри, розраховані на навчальній вибірці, аби гарантувати, що валідаційна вибірка представляє реальні, необроблені вхідні дані.

Підготовлені дані подаються на етап навчання нейронної мережі, яка використовує архітектуру згорткової нейронної мережі (CNN). Навчання включає кілька послідовних кроків: першим є витягування ознак, які автоматично ієрархічно вивчають складні патерни в даних. Далі відбувається зменшення розмірності, що допомагає скоротити обчислювальну складність і підвищити стійкість моделі до незначних змін вхідних даних. Подальше вирівнювання перетворює багатовимірний вихід згорткових шарів у плаский вектор. Фінальним етапом навчання є класифікація.

По завершенню навчання проводиться тестування результатів навчання на валідаційній вибірці для об'єктивної оцінки.

## 2.2 Архітектура нейронної мережі

Метод виявлення шахрайських транзакцій у фінансових операціях було реалізовано із використанням згорткової нейронної мережі (CNN), спеціально налаштованої та оптимізованої для задачі бінарної класифікації, де транзакції поділяються на шахрайські та легітимні. Такий підхід дозволяє моделі ефективно виявляти приховані закономірності та аномальні патерни у фінансових даних, які

можуть вказувати на шахрайство, навіть якщо вони не очевидні при традиційному аналізі.

Архітектура CNN складається з трьох типів шарів: згорткових, що виділяють локальні ознаки з вхідних даних; шарів об’єднання (pooling), які зменшують розмірність ознак, зберігаючи важливу інформацію та підвищуючи стійкість моделі; та повністю зв’язаних, що інтегрують ознаки для формування фінальних передбачень класу. Поєднання цих шарів створює мережу, здатну ефективно навчатися на великих обсягах даних, точно класифікувати транзакції та виявляти складні аномалії. На рисунку 2.2. наведено архітектуру CNN з порядком розташування та взаємодії шарів.

Основну функціональність згорткових нейронних мереж (CNN) розділено на чотири ключові складові.

1. Як і в інших формах штучних нейронних мереж (ANN), вхідний шар призначений для зберігання та прийому значень вхідних параметрів, які надходять у мережу.

2. Згортковий шар визначає вихід нейронів, які підключені виключно до локальних областей входу, шляхом обчислення скалярного добутку між вагами фільтрів та відповідною областю, підключеною до вхідного об'єму даних. Випрямлена лінійна одиниця (ReLU) має на меті застосувати поелементну нелінійну функцію активації, наприклад, max(0, x), яка замінює традиційну сигмоподібну, до виходу активації, створеної попереднім шаром, що сприяє кращій нелінійності та швидшому навчанню.

3. Шар об'єднання (pooling) потім просто виконує операцію зниження частоти дискретизації вздовж просторових розмірностей заданого входу, що дозволяє ще більше зменшити кількість параметрів у межах цієї активації та підвищити стійкість до невеликих зсувів.

4. Повністю зв'язані шари (fully connected layers) потім виконуватимуть ті ж самі завдання, що й у стандартних багатошарових ANN, інтегруючи інформацію та намагаючись отримати оцінки класів на основі отриманих активацій, які

безпосередньо використовуються для класифікації. Також рекомендується використовувати ReLU між цими шарами для суттєвого покращення загальної продуктивності моделі.

Завдяки цьому відносно простому, але ефективному методу послідовного перетворення, CNN здатні трансформувати оригінальні вхідні дані шар за шаром, застосовуючи методи згорткової обробки та даунсемплінгової дискретизації для отримання точних оцінок класів з метою класифікації та регресії [37].

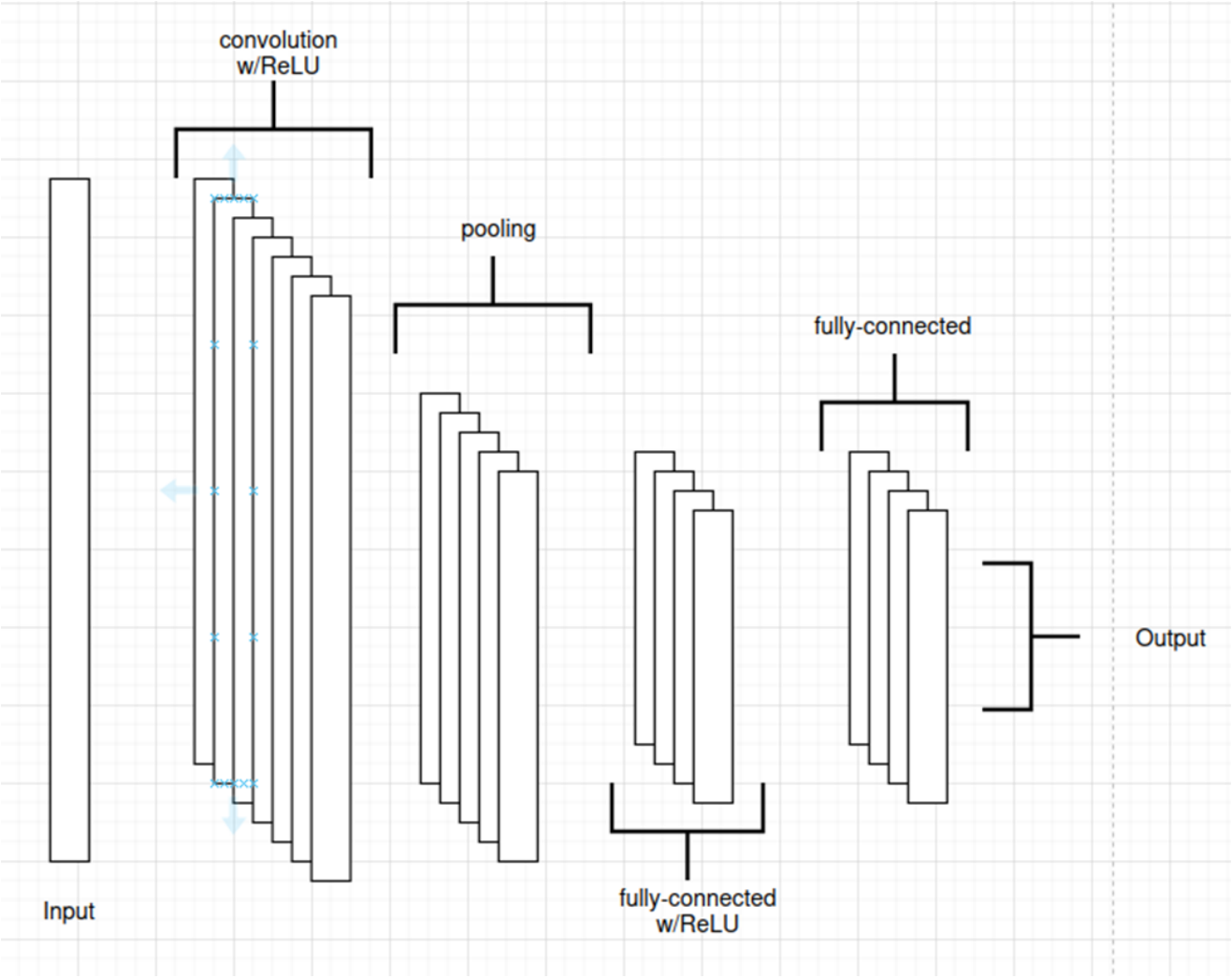



Рисунок 2.2 – Архітектура CNN

Операція згортки обчислює значення y[j] у позиції j на вихідній карті ознак, враховуючи вхідну послідовність x[i] та ядро w[k] [38].

Для дискретної вхідної послідовності x довжини M та ядра w довжини K,

вихід y з індексом j обчислюється як сума добутків:

$$y[j] = (x \times w)[j] = \sum_{k=0}^{K-1} x[j+k] \times w[K-1-k] + b \quad (2.3)$$

де:

- y[j] – вихідне значення в позиції j (на карті ознак).
- x[i] – вхідна послідовність.
- w[k] – ваги ядра (фільтра).
- b – член зміщення, що використовується для навчання, який часто додається до суми.
- w[K-1-k] являє собою стандартну згортку, де ядро перевертається. У глибокому навчанні ядро часто не перевертається, що спрощує формулу до:

$$y[j] = \sum_{k=0}^{K-1} x[j+k][w] + b \quad (2.4)$$

Врахувується, як змінюються розмірності після згортки. Довжина вихідної послідовності (Lout) залежить від довжини вхідного сигналу, розміру ядра, кроку та доповнення:

$$L_{out} = \left\lfloor \frac{L - K + 2P}{S} \right\rfloor + 1 \quad (2.5)$$

де:

- L in – довжина вхідної послідовності.
- K – розмір ядра.
- S – крок (на скільки кроків рухається ядро за раз).

– P – доповнення

– ⌊ ⌋ – функція підлоги, округлена до найближчого цілого числа.

На рисунок 2.3 — Зображено схему роботи згорткового шару.

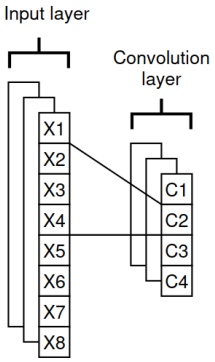



Рисунок 2.3 – Схема роботи згорткового шару

Шар об'єднання зменшує довжину карти ознак. Для максимального об'єднання з розміром вікна Psize та кроком Pstride:

$$y_{pool}[j] = max\ (y_{conv}[j \cdot P_{stride} : j \cdot P_{stride} + P_{size} - 1] \qquad (2.6)$$

де:

– y pool[j] – це вихід шару об'єднання в позиції j.

– y conv – це вхідна карта ознак (вихід згортки).

Схема роботи шару максимального об'єднання зображено на рисуноку 2.4.

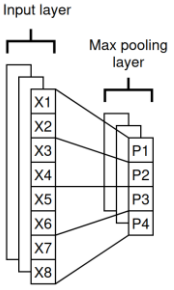



Рисунок 2.4 – Схема роботи шару  максимального об'єднання

У повністю з’єднаному шарі кожен нейрон з’єднаний з усіма нейронами попереднього шару, що дозволяє моделі враховувати глобальні залежності між всіма виділеними ознаками (Рисунок 2.5).

Математично робота повністю з’єднаного шару описується як:

$$y = f(W_x + b) \tag{2.7}$$

де:

- Wx – матриця ваг повністю з’єднаного шару вхідного вектору ознак;
- b – вектор зміщень (bias);
- f – нелінійна функція активації;
- y – вихідний вектор шару.

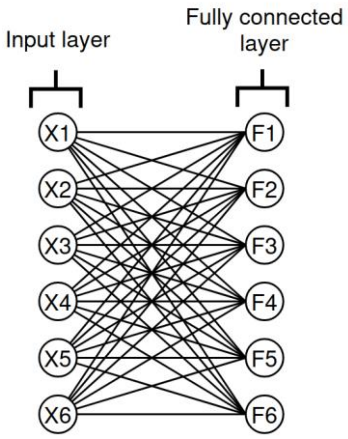



Рисунок 2.5 – Схема роботи повністю з’єднаного шару

Архітектура нейронної мережі була модифікована для ефективного виявлення шахрайських транзакцій у фінансових операціях. Основний акцент зроблено на автоматичному вилученні релевантних і інформативних ознак із вхідних даних з подальшим виконанням завдання бінарної класифікації – визначенням легітимності чи шахрайського характеру кожної транзакції.

Модель поєднує згорткові, об’єднувальні (pooling) та повністю зв’язані шари, де кожен елемент виконує свою чітко визначену функцію в багатоступеневому

процесі обробки інформації. Завдяки такій багаторівневій ієрархічній структурі забезпечується поступове перетворення низькорівневих вхідних сигналів у високорівневі абстрактні представлення.

Додаткове включення механізмів регуляризації, таких як відсіювання, а також нелінійних функцій активації суттєво підвищує узагальнюючу здатність моделі, роблячи її більш стійкою до шумів у даних та ефективно запобігаючи перенавчанню.

Модифікована архітектура мережі складається з двох згорткових шарів, двох шарів об’єднання, шару вирівнювання, двох щільних шарів та шару відсіювання (рисунок 2.6).

Перший одновимірний згортковий шар відповідає за вилучення локальних ознак із послідовних даних. Він застосовує 32 фільтри, кожен з розміром ядра три, що дозволяє мережі виявляти короткострокові закономірності у вхідній послідовності, такі як локальні залежності між послідовними елементами. Функція активації ReLU вводить нелінійність, дозволяючи мережі навчатися складним представленням. Параметр форми вхідних даних визначає розмірну структуру вхідних даних і забезпечує сумісність моделі з навчальним набором даних.

Шар максимального об'єднання виконує операцію зниження роздільної здатності, яка зменшує розмірність карт ознак, створених попереднім згортковим шаром. Вибираючи максимальне значення в кожному вікні з двох елементів, шар зберігає найважливіші активації, відкидаючи менш релевантну інформацію. Ця операція не тільки підвищує обчислювальну ефективність, але й зменшує ймовірність перенавчання, забезпечуючи просторову інваріантність у вивчених ознаках.

Другий згортковий шар суттєво поглиблює здатність моделі формувати ієрархічні представлення вхідних даних. Оснащений 64 фільтрами та ядром розміром 3×3 (або аналогічно для 1D-згортки), цей шар фокусується на захопленні більш складних, абстрактних і вищого порядку шаблонів, які базуються на локальних ознаках, виявлених першим згортковим шаром. Функція активації ReLU,

застосована й тут, продовжує сприяти ефективному моделюванню нелінійних залежностей між ознаками, значно підвищуючи виразність усієї мережі, полегшуючи поширення градієнтів під час навчання та загальну продуктивність у автоматичному вилученні релевантних ознак.

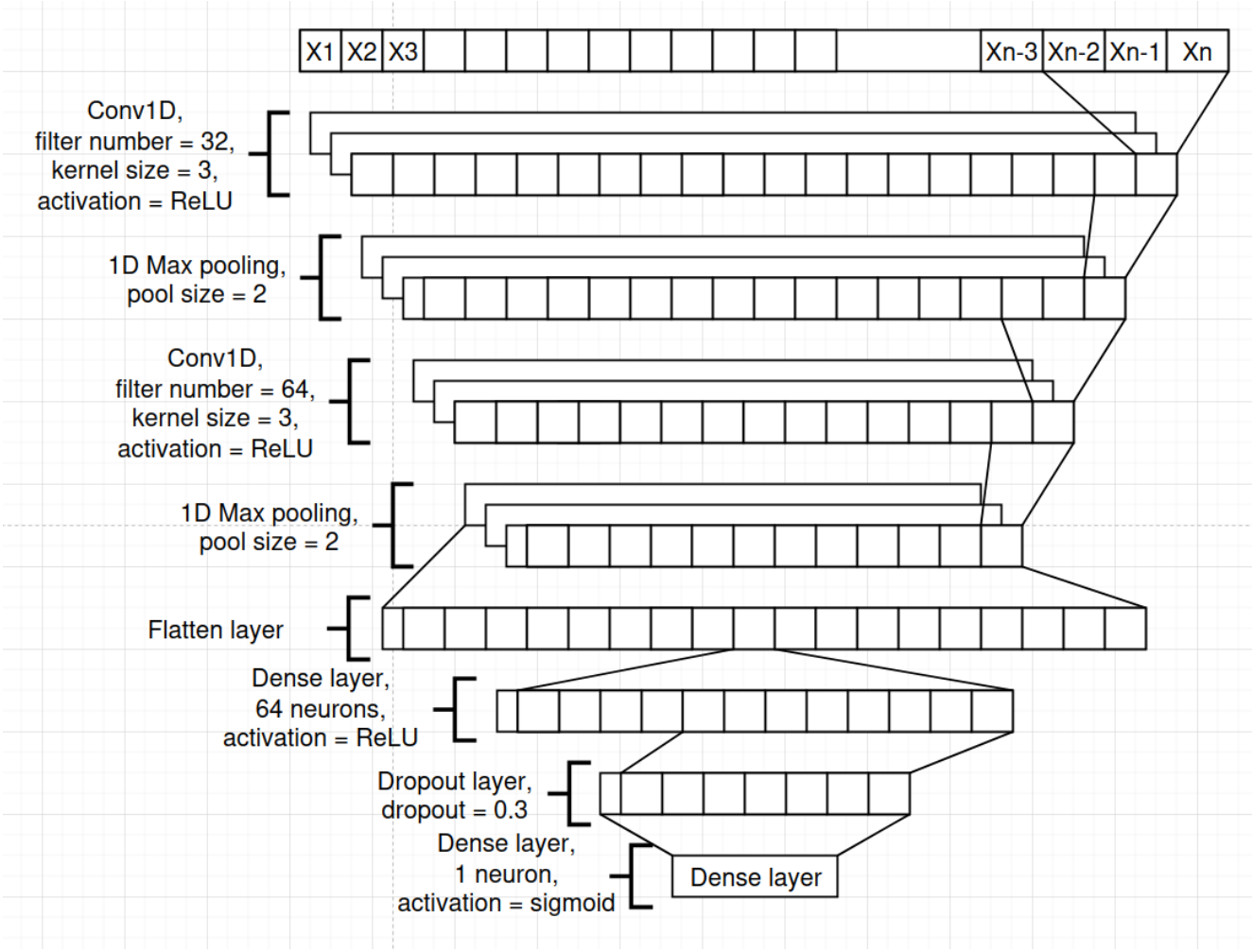



Рисунок 2.6 – Модифікована архітектура нейронної мережі

Другий шар об’єднання (max pooling) продовжує процес цілеспрямованого зменшення розмірності даних, застосовуючи операцію максимального об’єднання до вікон розміром 2 елементи. Цей етап додатково стискає карти ознак, сформовані попереднім згортковим шаром, зберігаючи лише найсильніші активації та ефективно пригнічуючи надлишкові чи менш інформативні сигнали, що часто містять шум або несуттєві варіації. Завдяки такому підходу модель набуває більшої

стійкості до незначних зсувів і трансформацій у вхідних даних, одночасно забезпечуючи високу обчислювальну ефективність і швидкість обробки великих обсягів транзакційної інформації.

Шар вирівнювання (Flatten) виконує важливе перетворення багатовимірного тензора, отриманого на виході згорткових та об'єднувальних шарів, в одновимірний вектор фіксованої довжини. Це перетворення є необхідним технічним мостом, який з'єднує етап автоматичного вилучення просторово-ієрархічних ознак із наступними повністю зв'язаними шарами, що традиційно працюють виключно з плоскими векторними представленнями. Завдяки цьому кроку мережа отримує можливість глобально інтегрувати всі вилучені локальні та абстрактні ознаки, створюючи основу для точної та обґрунтованої остаточної класифікації транзакцій.

Щільний шар виконує роль повністю зв'язаного нейронного шару, який інтегрує та глибоко інтерпретує всі ознаки, автоматично вивчені на попередніх етапах мережі. Складаючись із 64 нейронів, він з'єднує кожен свій вузол з усіма вхідними сигналами, що дозволяє формувати складні абстрактні представлення високого рівня, безпосередньо орієнтовані на цільове завдання бінарної класифікації шахрайських транзакцій. Застосування функції активації ReLU забезпечує високу обчислювальну ефективність, сприяє нелінійності перетворень та ефективно пом'якшує проблему зникнення градієнта, що в підсумку підтримує стабільне й ефективне навчання всієї моделі на глибоких архітектурах.

Шар відсіву (Dropout) є одним із найефективніших методів регуляризації, спеціально призначеним для запобігання перенавчанню моделі на тренувальних даних.

Під час кожної ітерації процесу навчання він випадковим чином деактивує приблизно 30 % нейронів, тим самим змушуючи мережу формувати надлишкові, більш робастні та узагальнені представлення даних замість залежності від фіксованих, стабільних зв'язків між окремими нейронами. Цей механізм суттєво покращує здатність моделі до узагальнення, робить її стійкішою до варіацій у

вхідних даних і значно підвищує продуктивність на невидимих або зовсім нових прикладах, що особливо важливо в реальних сценаріях виявлення шахрайства, зменшуючи ризик перенавчання та забезпечуючи вищу надійність прогнозів.

Останній щільний шар мережі функціонує як вихідний і містить лише один нейрон, який за допомогою сигмоїдної функції активації генерує скалярне значення ймовірності в діапазоні від 0 до 1. Це значення прямо відображає рівень впевненості моделі в тому, що поданий вхід належить до позитивного класу, тобто є шахрайською транзакцією.

Використання саме сигмоїдної активації виявляється особливо доцільним і ефективним для завдань бінарної класифікації, оскільки воно забезпечує інтуїтивно зрозумілу інтерпретацію вихідних ймовірностей, спрощує застосування порогових значень для остаточного класового рішення та сприяє кращій калібрації прогнозів моделі в цілому.

### 2.3 Навчання нейронної мережі

Навчання згорткової нейронної мережі спрямоване на знаходження такого набору вагових коефіцієнтів, який забезпечує мінімальну похибку класифікації фінансових транзакцій на шахрайські та легальні. Процес навчання реалізується як ітеративна оптимізаційна процедура, під час якої модель поступово покращує свою здатність до узагальнення на основі навчальних та валідаційних даних.

Задача навчання нейронної мережі формулюється як задача мінімізації функції втрат, що оцінює різницю між фактичними мітками класів та прогнозованими ймовірностями приналежності транзакцій до класу шахрайських. Оскільки розв’язується задача бінарної класифікації, у роботі використовується функція бінарної перехресної ентропії:

$$L = \frac{-1}{N}\sum_{i=1}^{N}[y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)] \qquad (2.8)$$

де:

– yi – істинна мітка класу,

– $\hat{y}_i$ – прогнозована ймовірність,

– N – кількість зразків у навчальній вибірці.

Мінімізація цієї функції дозволяє моделі підвищувати точність передбачення ймовірності шахрайства для кожної транзакції.

Процес навчання нейронної мережі зображено на рисунку 2.7.

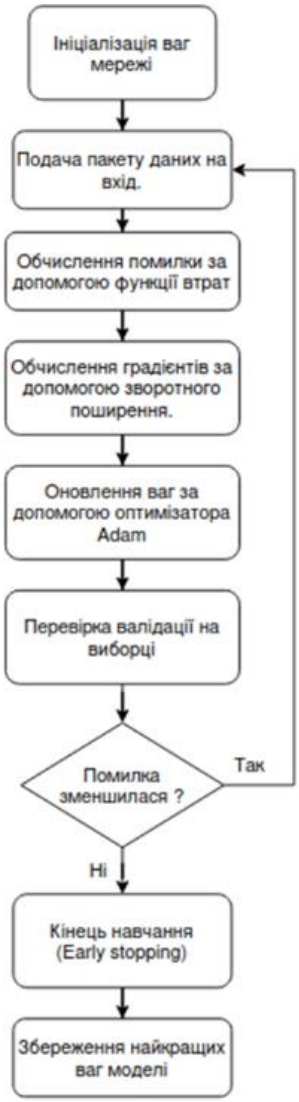



Рисунок 2.7 – Схема процесу навчання нейронної мережі

Для оновлення ваг нейронної мережі використовується оптимізатор Adam, який поєднує переваги Momentum та RMSProp і автоматично налаштовує швидкість

навчання для кожного параметра, забезпечуючи швидку та стабільну збіжність. Навчання проводиться по епохах із пакетною подачею даних та оцінюванням на валідаційній вибірці. Для запобігання перенавчанню застосовується рання зупинка, а після завершення зберігаються ваги з мінімальною валідаційною помилкою.

Процес навчання нейронної мережі є ітеративним і керується механізмом ранньої зупинки (early stopping). Спочатку ваги мережі ініціалізуються випадковими значеннями, після чого навчальні дані подаються пакетами на вхід моделі. Для кожного пакета обчислюється помилка прогнозу за допомогою функції втрат, градієнти поширюються назад (backpropagation), а ваги оновлюються оптимізатором (наприклад, Adam). Після обробки пакета оцінюється помилка на валідаційній вибірці; якщо вона зменшилася, навчання продовжується з наступними даними. У разі відсутності покращення протягом певного часу навчання зупиняється, а зберігаються найкращі ваги моделі, які показали найнижчу помилку на валідації. Такий підхід запобігає перенавчанню та забезпечує отримання оптимальної моделі для виявлення шахрайських транзакцій.

## 2.4 Формування та підготовка даних

### 2.4.1 Навчальний набір даних

Вхідними даними для тренування згорткової нейронної мережі (CNN) з метою виявлення шахрайських транзакцій у фінансових операціях слугує відомий датасет Credit Card Fraud Detection [39], доступний на платформі Kaggle за посиланням: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud. Цей набір даних містить реальні транзакції, здійснені кредитними картками європейських власників у вересні 2013 року, і охоплює операції протягом двох днів. Загалом датасет включає 284 807 транзакцій, серед яких лише 492 випадки шахрайства, що робить його надзвичайно незбалансованим — позитивний клас (шахрайські операції) становить всього 0,172 % від загальної кількості записів. Така сильна

незбалансованість є типовою для реальних задач виявлення фінансового шахрайства та вимагає спеціальних технік обробки під час моделювання.

Датасет містить виключно числові вхідні змінні, які є результатом попереднього перетворення за методом головних компонент (PCA) для забезпечення анонімізації. Через конфіденційні обмеження оригінальні ознаки залишаються прихованими та недоступними. Ознаки V1, V2, … V28 представляють собою головні компоненти, отримані за допомогою PCA, тоді як єдині змінні, що не пройшли таке перетворення – це Time (Час) та Amount (Сума). Ознака Time фіксує кількість секунд, що минули між поточною транзакцією та першою в наборі даних, дозволяючи враховувати тимчасові залежності. Ознака Amount вказує на суму транзакції та може бути корисною для навчання, чутливого до вартості операцій (наприклад, cost-sensitive learning). Цільова змінна Class (Клас) є відгуком моделі: вона приймає значення 1 для шахрайських транзакцій та 0 – для легітимних, що робить задачу класичною бінарною класифікацією.

### 2.4.2 Методи попередньої обробки набору даних

Дані щодо фінансових транзакцій потребують попередньої обробки.

Для реальних транзакцій виконується операції нормалізації, масштабування та зміни розмірності.

Для навчальної виборки виконується операції нормалізації, масштабування, балансування та зміни розмірності.

Для валідаційної виборки виконується операції нормалізації, масштабування та зміни розмірності.

Попередня обробка даних (нормалізація, масштабування, балансування (виконується лише для навчальної вибірки), зміна розмірності) забезпечує коректність, стабільність і ефективність роботи алгоритму, сприяючи кращій узагальнюючій здатності та точності прогнозів.

На етапі попередньої обробки даних ознака Amount(Сума) масштабована за допомогою RobustScaler. Цей метод базується не на мінімальних та максимальних

значеннях, а на медіані та інтерквартильному розмаху, що робить його стійким до наявності викидів у даних – поширеного явища у фінансових транзакціях, де окремі операції можуть мати надзвичайно великі суми. Формула перетворення має вигляд:

$$x' = \frac{x - median(x)}{IQR(x)} \tag{2.9}$$

де:

- median – медіана,
- IQR – міжквартильний розмах.

У результаті більшість значень суми зосереджуються поблизу нуля, тоді як екстремальні транзакції не мають надмірного впливу на навчання моделі. Такий підхід забезпечує стабільнішу збіжність нейронної мережі та підвищує її стійкість до нетипових значень у даних.

Ознаку Time(Час) було нормалізовано до інтервалу [-1,1] за допомогою лінійного перетворення:

$$x' = \frac{x - min\ (x)}{max(x) - min\ (x)} \tag{2.10}$$

де:

- min(x) – мінімальне значення x,
- max(x) – максимальне значення x.

Таким чином, найменше значення часу отримує -1, найбільше +1, а всі проміжні значення рівномірно розподіляються між ними.

Для балансування вибірки було застосовано метод SMOTE (Synthetic Minority Oversampling Technique), який штучно генерує нові синтетичні зразки міноритарного класу шляхом інтерполяції між існуючими прикладами шахрайських транзакцій, тим самим збільшуючи їхню кількість без простого дублювання.

У сильно незбалансованих наборах даних мажоритарний клас домінує під час процесу навчання, що призводить до формування упередження в моделі: вона

схильна переважно прогнозувати більш частий клас. Як наслідок, показники ефективності для міноритарного класу виявляються незадовільними, навіть якщо цей клас представляє критичні та важливі сценарії, такі як виявлення фінансового шахрайства, рання медична діагностика рідкісних захворювань або контроль якості у промисловому виробництві, де помилки можуть мати серйозні наслідки.

Класичним прикладом є набір даних для виявлення шахрайства, де шахрайські транзакції становлять лише близько 0,1–1 % від загального обсягу. У такому разі наївний класифікатор, який завжди прогнозує «не шахрайство», міг би легко досягти 99 % загальної точності, але при цьому пропустив би всі реальні випадки шахрайства (recall для міноритарного класу дорівнюватиме 0). Це повністю нівелює практичну цінність моделі, оскільки основна мета саме в надійному виявленні рідкісних, але критичних аномалій.

SMOTE (Synthetic Minority Oversampling Technique) є просунутим методом передискретизації, який ефективно усуває дисбаланс класів шляхом генерації синтетичних прикладів для міноритарного класу. На відміну від простих технік надмірної вибірки, що лише дублюють існуючі зразки меншості (що часто призводить до перенавчання), SMOTE створює абсолютно нові, штучні точки даних, розташовані вздовж лінійних відрізків, які з’єднують випадково обраного представника міноритарного класу з його найближчими k-сусідами в просторі ознак.

Цей метод оперує саме в просторі ознак, а не в оригінальному просторі даних, що дозволяє враховувати внутрішні кореляції та взаємозв’язки між ознаками, роблячи синтетичні приклади більш реалістичними та репрезентативними. Такий підхід не лише запобігає перенавчанню, характерному для базових методів дублювання, але й забезпечує значно більшу різноманітність навчальних прикладів для міноритарного класу, сприяючи кращій узагальнюючій здатності моделі на невидимих даних [40]. Алгоритм SMOTE дотримується систематичного підходу до створення синтетичних вибірок класів меншин у наборі даних, що допомагає усувати дисбаланс класів та підвищувати ефективність моделей машинного навчання.

Схема роботи алгоритму зображена на рисунку 2.8.

Крок 1: Визначення вибірок класів меншин. Алгоритм спочатку ідентифікує всі наявні вибірки, що належать до класів меншин у наборі даних. Ці вибірки виступають основою для створення синтетичних прикладів і забезпечують репрезентативність нових даних.

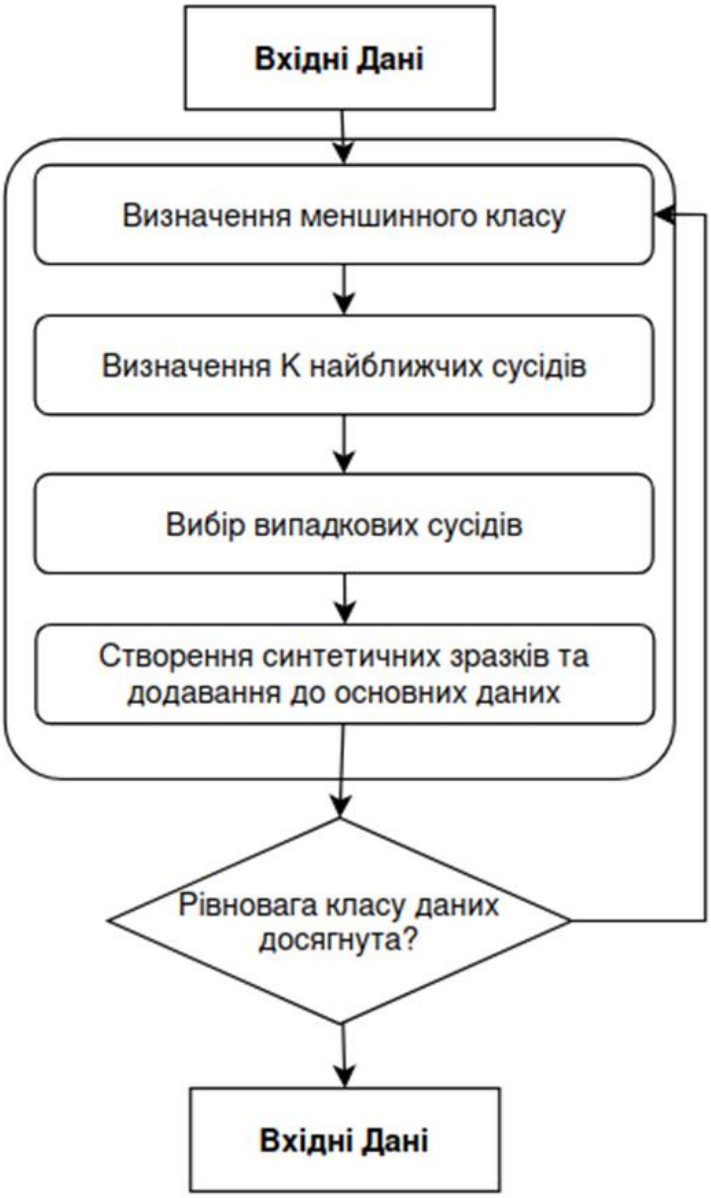



Рисунок 2.8 – Схема роботи алгоритму SMOTE

Крок 2: Пошук K найближчих сусідів. Для кожної вибірки класу меншин SMOTE знаходить k найближчих сусідів (зазвичай k=5) у межах одного класу,

використовуючи евклідову відстань або інші відповідні метрики відстані. Ця ідентифікація сусідства є критично важливою для підтримки локальної структури даних і гарантує, що створені синтетичні вибірки будуть відповідати природній геометрії простору ознак.

Крок 3: Вибір випадкових сусідів. З k найближчих сусідів алгоритм випадковим чином обирає одного сусіда для кожної синтетичної вибірки, що планується створити. Такий підхід забезпечує різноманітність синтетичних прикладів та запобігає надмірній концентрації нових точок даних у вузьких областях.

Крок 4: Створення синтетичних вибірок. Нові синтетичні вибірки генеруються вздовж відрізка прямої, що з'єднує початкову вибірку меншин та її обраного сусіда, шляхом випадкового вибору точки на цьому відрізку. Це дозволяє створювати реалістичні, проміжні приклади, які зберігають властивості класу меншин.

Крок 5: Повторювати до балансування. Процес повторюється для всіх вибірок класу меншин доти, доки не буде досягнуто бажаного рівня балансу класів. Зазвичай це відбувається до моменту, коли розмір класів меншин дорівнює розміру класу більшості або досягається задане цільове співвідношення, що дозволяє значно покращити якість навчання моделі на збалансованому наборі даних.

Після масштабування ознак було виконано балансування вибірки, оскільки у вихідних даних спостерігався суттєвий дисбаланс між кількістю шахрайських та не шахрайських транзакцій. У більшості реальних фінансових наборів даних кількість легітимних операцій значно перевищує кількість шахрайських, що може призвести до переважного навчання моделі на більший клас і, відповідно, погіршення здатності виявляти шахрайство.

Для усунення цієї проблеми кількість зразків шахрайських транзакцій було зрівняно до рівня кількості не шахрайських транзакцій. У результаті сформовано збалансований набір даних, у якому обидва класи мають однакову кількість прикладів.

## 2.5 Критерії та метрики оцінювання ефективності

Для оцінки ефективності виявлення шахрайських транзакцій у фінансових операціях було обрано наступні метрики: accuracy, precision, recall, F1 score, confusion matrix.

Accuracy – це фундаментальна метрика, яка використовується для оцінки ефективності моделі класифікації. Вона показує частку правильних прогнозів, зроблених моделлю, від усіх прогнозів [41].

$$Accuracy = \frac{NumberofCorrectPredictions}{TotalNumberofPredictions} \quad (2.11)$$

де:

- Number of Correct Predictions – число коректних прогнозів
- Total Number of Predictions – тотальне число прогнозів

Хоча Accuracy забезпечує швидкий знімок, вона може вводити в оману у випадках незбалансованих наборів даних. Наприклад, у наборі даних з 90% класу A та 10% класу B, модель, яка передбачає лише клас A, все одно досягне 90% точності, але не зможе виявити жодних випадків класу B.

Accuracy хороша, але вона дає хибнопозитивний ефект досягнення високої точності. Проблема виникає через те, що можливість неправильної класифікації вибірок другорядних класів є дуже високою.

2. Precision

Precision вимірює, скільки позитивних прогнозів, зроблених моделлю, насправді є правильними. Це корисно, коли вартість хибнопозитивних результатів висока, наприклад, у медичних діагнозах, де прогнозування захворювання, коли його немає, може мати серйозні наслідки.

$$Precision = \frac{TP}{TP + FP} \tag{2.12}$$

де:

– TP – Істиннопозитивні результати

– FP – Хибнопозитивні результати

Точність допомагає гарантувати, що коли модель прогнозує позитивний результат, він, ймовірно, буде правильним.

3. Recall

Recall або чутливість вимірює, скільки фактично позитивних випадків було правильно ідентифіковано моделлю. Це важливо, коли пропуск позитивного випадку (хибнонегативний) коштує дорожче, ніж хибнопозитивні результати.

$$Recall = \frac{TP}{TP + FN} \tag{2.13}$$

де:

– FN – Хибнонегативні результати

У сценаріях, де важливо відстежувати всі позитивні випадки (наприклад, виявлення захворювання), відповідність є ключовим показником.

4. F1 Score

F1 Score – це гармонійне середнє точності та повноти. Вона корисна, коли нам потрібен баланс між точністю та повнотою, оскільки об'єднує їх в одне число. Високий показник F1 означає, що модель добре працює за обома показниками. Його діапазон становить [0,1].

Нижча повнота та вища точність дають нам велику точність, але тоді вона пропускає велику кількість випадків. Чим кращий показник F1, тим краща продуктивність. Математично її можна виразити таким чином:

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.14}$$

5. Confusion Matrix

У галузі машинного навчання, зокрема у вирішенні задачі статистичної класифікації, матриця плутанини, також відома як матриця помилок, являє собою специфічний макет таблиці, що дозволяє наочно оцінити та візуалізувати ефективність алгоритму, зазвичай у контексті контрольованого навчання [42].

Кожен рядок матриці зазвичай представляє екземпляри, що належать до фактичного класу, тоді як кожен стовпець відображає екземпляри, які були передбачені моделлю як належні до певного класу, або навпаки – у літературі зустрічаються обидва варіанти відображення. Така структура дозволяє легко визначити кількість правильно та неправильно класифікованих прикладів для кожного класу. Діагональ матриці, яка з'єднує верхній лівий та нижній правий кути, відображає всі екземпляри, що були передбачені коректно, тобто для яких прогноз моделі збігся з фактичним класом (рисунок 2.9)

| | | Predicted condition | |
|---|---|---|---|
| | Total amount = P + N | Positive (P) | Negative (N) |
| Actual condition | Positive (P) | True positive | False negative |
| | Negative (N) | False positive | True negative |

Рисунок 2.9 – Матриця плутанини

**Висновки до розділу 2**

Спроєктовано метод виявлення шахрайських транзакцій у фінансових операціях з використанням згорткових нейронних мереж. Описано загальну ідею методу та архітектуру моделі згорткової нейронної мережі.

Наведено послідовність навчання нейронної мережі з використанням датасету Credit Card Fraud Detection.

Описано загальну схему спроєктованого методу, який поєднує попередню обробку вхідних даних з нормалізацією та масштабуванням ключових класифікаційних параметрів та балансування даних й згорткової нейронної мережі.

Визначено критерії для оцінювання точності спроєктованого методу, а саме точність класифікації (Accuracy), повнота (Recall), точність передбачення (Precision), 1-міра (F1 Score), матриця плутанини(Confusion Matrix).

# Розділ 3 Програмна реалізація методу виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж

## 3.1 Засоби програмної реалізації методу

Для програмної реалізації розробленого методу виявлення шахрайських транзакцій було обрано мову програмування Python . Python є широковживаним у сфері машинного навчання завдяки простоті синтаксису, розвиненій екосистемі бібліотек та активній спільноті розробників. Для створення програмної реалізації методу виявлення шахрайських транзакцій у фінансових операціях ііз використанням  згорткових нейронних мереж, необхідно обрати програмні та апаратні засоби, які забезпечать ефективність, масштабованість і високу продуктивність. Вибір технологічного стеку ґрунтується на вимогах до системи, що включають підтримку сучасних алгоритмів глибокого навчання, обробку великих обсягів транзакційних даних.

Для реалізації згорткових нейронних мереж обрано бібліотеку TensorFlow. Для обробки фінансових і транзакційних записів використовуються бібліотеки NumPy, Pandas, Scikit-learn і Matplotlib.

TensorFlow [43] є високопродуктивною платформою для розробки, оптимізації та розгортання моделей у промислових середовищах, включно з серверними інфраструктурами та мобільними пристроями. Разом з іншими фреймворками він підтримує апаратне прискорення (GPU), що значно скорочує час навчання моделей виявлення шахрайства.

NumPy забезпечує швидке виконання векторизованих математичних операцій над великими масивами даних, тоді як Pandas [44] використовується для структурування, очищення та агрегації табличної інформації, зокрема під час формування наборів ознак і статистичних звітів.

Для попередньої обробки, нормалізації та балансування даних, а також для розрахунку базових метрик точності застосовується Scikit-learn.

Matplotlib слугує інструментом для візуалізації результатів навчання моделей, зокрема побудови графіків втрат, точності та порівняльного аналізу моделей.

Для експериментальної розробки, навчання моделей і попереднього аналізу даних використовується Jupyter Notebook, що дозволяє виконувати код поетапно та відображати результати у реальному часі. Для інтеграції системи, налагодження програмних модулів і підготовки до розгортання застосовується Visual Studio Code.

Програмна реалізація спроєктована з урахуванням використання графічних процесорів (GPU), зокрема серій NVIDIA з підтримкою технології CUDA, що дозволяє значно прискорити навчання згорткових моделей на великих обсягах даних. Для середовищ із обмеженими ресурсами передбачено можливість використання хмарних платформ – Google Colab, AWS або Microsoft Azure, які надають доступ до високопродуктивних обчислювальних вузлів і масштабованих сховищ.

### 3.2 Архітектура програмної реалізації

Функціональне призначення розробленої програмної реалізації методу полягає в автоматичному виявленні, класифікації та відстеженні шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж. Метод забезпечує обробку великих обсягів транзакційних даних у реальному часі або на основі історичних записів, що дає змогу оперативно реагувати на підозрілі дії. Вона характеризується високою точністю та надійністю, зберігаючи ефективність навіть за умов наявності шумів, аномалій або спроб маскування шахрайських дій. Основне завдання методу полягає у забезпеченні безперервного моніторингу фінансових потоків, виявленні потенційно небезпечних операцій та аналізі поведінкових патернів користувачів і рахунків.

Для реалізації зазначених завдань програмна реалізація методу включає набір модулів, які відповідають за попередню обробку даних, виявлення та класифікацію транзакцій, аналіз часових залежностей, оцінювання якості роботи та

візуалізацію отриманих результатів. Сукупність цих компонентів забезпечує повний цикл виявлення шахрайства у фінансових системах, включно з банківськими операціями, електронними платежами та транзакціями електронної комерції.

Програмна реалізація має модульну архітектуру, у межах якої кожен компонент виконує окрему функцію та взаємодіє з іншими для забезпечення цілісності процесу виявлення шахрайських транзакцій.

Програмна реалізація розробленого методу включає в себе 4 модуля:

Модуль A***pp*** є центральною складовою програмної реалізації та виконує роль інтеграційного ядра, що забезпечує узгоджену взаємодію трьох допоміжних модулів. Він слугує платформою для їхнього об'єднання, координації та демонстрації їх функціональних можливостей у рамках єдиної програмної реалізації (рисунок 3.1).

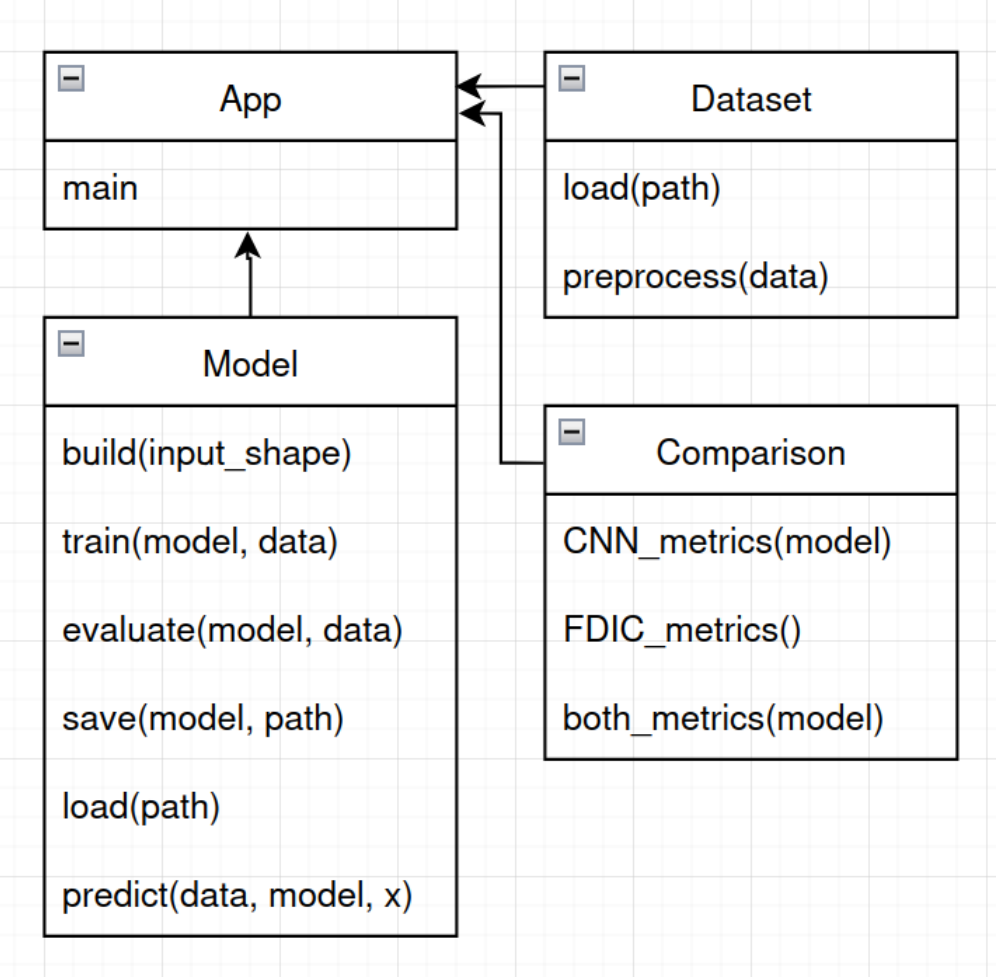



Рисунок 3.1 – Схема взаємодії модулів

Модуль Dataset відповідає за підготовку та обробку даних для подальшого навчання моделі машинного навчання на основі транзакцій кредитних карток. Він забезпечує завантаження необробленого набору даних із CSV-файлу, організовує структуру даних у зручний для аналізу формат та розділяє їх на ознаки та цільову мітку. Ознаки включають числові параметри транзакцій, такі як сума, час та анонімізовані компоненти PCA, а цільова мітка визначає, чи є транзакція шахрайською. Модуль робить роботу з даними зручною та стандартизованою, дозволяючи подальшим компонентам безпосередньо використовувати їх для побудови та навчання моделей.

Модуль також виконує попередню обробку даних, включно з нормалізацією та балансуванням. Значення ознаки Amount(Сума) масштабується за допомогою робастного скалера, щоб зменшити вплив викидів, а ознака Time(Час) приводиться до інтервалу [0;1], щоб уникнути диспропорцій у внеску різних ознак у процес навчання. Для вирівнювання класів застосовується алгоритм SMOTE, який генерує синтетичні приклади для менш представленного класу шахрайських транзакцій. В результаті модуль повертає оброблений та збалансований набір даних, готовий для подальшого використання у тренуванні моделей, забезпечуючи їхню стабільність, коректність та підвищену точність прогнозування.

Модуль Model відповідає за побудову, налаштування та навчання моделі машинного навчання, призначеної для класифікації фінансових транзакцій на шахрайські та легітимні. Основна мета модуля полягає у створенні обчислювальної структури, здатної ефективно виявляти приховані закономірності в оброблених даних і приймати обґрунтовані рішення на основі цих закономірностей. Модель, реалізована в модулі, інтегрує різноманітні алгоритмічні підходи та компоненти, що дозволяють аналізувати великі обсяги даних, визначати ключові характеристики транзакцій, які впливають на ймовірність шахрайства, і адаптуватися до складних та динамічних структур даних.

Модуль забезпечує повний процес навчання моделі на підготовлених даних, включно з розподілом набору даних на тренувальну та тестову частини, що дозволяє

оцінити ефективність та здатність моделі узагальнювати отримані знання на нових прикладах. Під час навчання модуль відстежує різні метрики продуктивності та прогрес моделі, надаючи можливість детально аналізувати, наскільки добре вона справляється із завданням класифікації, і визначати потенційні проблеми або точки для оптимізації. Візуалізація результатів навчання та ключових метрик забезпечує користувачеві інтуїтивне розуміння якості, стабільності та точності моделі, а також дозволяє порівнювати її ефективність із іншими підходами або попередніми версіями моделі.

Окрім навчання, модуль надає розширений функціонал для оцінювання та тестування моделі на нових, невідомих даних, включно з обчисленням ключових показників ефективності, таких як точність, Precision, Recall, F1-score, та побудовою матриці помилок. Це дає змогу проводити детальний аналіз сильних і слабких сторін моделі, ідентифікувати потенційні напрямки для вдосконалення та робити обґрунтовані висновки щодо її придатності для реальних умов роботи.

Модуль також дозволяє зберігати навчану модель та надалі завантажувати її для повторного використання без необхідності повторного навчання, що значно спрощує інтеграцію моделі у виробничі системи та робочі процеси, підвищує зручність її застосування та забезпечує ефективне й довготривале використання у практичних завданнях.

Модуль Model забезпечує повноцінну інтерактивну роботу з окремими прикладами транзакцій у реальному часі, що дозволяє миттєво перевіряти прогноз моделі для конкретної операції, порівнювати його з фактичною міткою класу та аналізувати причини можливих помилок. Завдяки цьому модуль виконує комплексну роль центрального компонента системи, об’єднуючи етапи створення архітектури, навчання, тонкого налаштування, всебічної оцінки та практичного застосування моделі для виявлення шахрайських транзакцій. Така інтеграція гарантує максимальну гнучкість, прозорість і повний контроль на всіх стадіях роботи з даними – від первинної обробки до фінального розгортання.

Модуль Comparison виконує функцію аналітичного компонента, що

відповідає за комплексну оцінку якості моделей, які застосовуються для розв’язання задачі класифікації. Його робота охоплює повний цикл методів валідації – від підготовки вхідних даних та приведення їх до узгодженого формату до отримання підсумкових метрик, які відображають ефективність моделей у різних сценаріях. Завдяки цьому модуль формує цілісне уявлення про поведінку моделі на збалансованих і реалістичних вибірках, що дає змогу більш обґрунтовано інтерпретувати результати.

У межах загальної архітектури програмної реалізації цей компонент виступає інструментом інтегрованого аналізу: він порівнює результати роботи власної моделі з орієнтовними показниками відомих рішень. Такий підхід забезпечує можливість не лише оцінити рівень точності чи стабільності методів, а й визначити відносні переваги кожного підходу в контексті поставленої задачі. Модуль виконує зіставлення характеристик, таких як точність, повнота чи збалансованість класифікації, представляючи їх у зручній для дослідника формі.

Окреме значення має роль модуля у візуалізації результатів. Він надає наочні графічні інструменти, які дають змогу інтерпретувати метрики не лише у вигляді числових показників, а й як елементи порівняльного аналізу. Така подача підтримує процес прийняття рішень і сприяє глибшому розумінню впливу обробки даних, збалансування вибірки чи архітектури моделі на кінцеві результати.

Особливості взаємодії між модулями:

- послідовність обробки,
- вхідні та вихідні формати.

Дані передаються між модулями у визначеному порядку, забезпечуючи логічний та структурований процес роботи. Кожен модуль отримує дані у форматі, який відповідає його специфікації, забезпечуючи безперебійний обмін інформацією

### 3.3 Реалізація програмних модулів

Функція load виконує роль базового механізму доступу до даних. Вона

забезпечує стандартизоване завантаження вхідного набору даних із зовнішнього джерела, представленого у вигляді табличного файлу. Застосування єдиного інтерфейсу для отримання даних сприяє підвищенню узгодженості роботи всієї підсистеми обробки, оскільки незалежно від конкретного джерела або шляху зберігання дані надходять у структурованому форматі, придатному для подальших аналітичних або обчислювальних операцій.

Функція preprocess виконує комплекс процедур первинної обробки даних, спрямованих на підвищення їхньої якості та забезпечення коректності подальшого моделювання. На першому етапі вона формує копію вхідного набору даних, що дозволяє уникнути небажаних змін у вихідному джерелі. Один з ключових показників Amount(Сума) піддається нормалізації за допомогою методу RobustScaler, який зменшує вплив викидів та забезпечує більш стабільне масштабування для розподілів із сильними перекосами. Це особливо важливо для фінансових даних, де значення суми транзакції можуть коливатись у широкому діапазоні.

Паралельно відбувається нормування ознаки Time(Час) до інтервалу [0, 1] шляхом лінійного перетворення відносно мінімального та максимального значень. Такий підхід уніфікує часову шкалу й робить модель менш чутливою до абсолютних величин, зосереджуючи увагу на відносній позиції транзакцій у часовій послідовності.

Подальший етап полягає в корекції дисбалансу класів, характерного для задач виявлення шахрайства. З цією метою застосовується метод SMOTE (Synthetic Minority Over-sampling Technique), який синтетично збільшує представлення рідкісного класу (Class = 1), створюючи нові зразки на основі найближчих сусідів. Це дає змогу покращити здатність моделі розпізнавати аномальні транзакції та запобігає домінуванню більшості у процесі навчання.

Після застосування SMOTE результати ресемплінгу перетворюються назад у структуру таблиці, де повністю відновлюється початковий набір специфікацій ознак, зокрема їхні назви, типи та порядок розташування, а також додається

оновлена та коректно сформована цільова змінна. Такий підхід забезпечує збереження логічної узгодженості даних і дозволяє уникнути можливих порушень структури, що можуть виникати під час генерування синтетичних вибірок. Повернений набір даних стає збалансованим, належним чином нормалізованим і структурно погодженим з вимогами подальшого аналізу, що значно підвищує його придатність для ефективного використання в моделях машинного навчання та підвищення їх загальної продуктивності.

Функція build відповідає за створення та початкову конфігурацію нейронної мережі, що застосовується для розв’язання задачі бінарної класифікації, зокрема в контексті виявлення аномальних чи шахрайських транзакцій. Вона формує послідовну архітектуру (Sequential model), у якій обчислювальні шари виконуються один за одним, забезпечуючи прямий потік даних від вхідних ознак до кінцевого прогнозу.

Структура моделі складається з кількох ключових компонентів. Перші два блоки: Conv1D з 32 та 64 фільтрами виконують одновимірну згортку, що дає змогу виявляти локальні закономірності у структурі ознак. Використання активації ReLU сприяє стабільному проходженню градієнтів і дозволяє моделі ефективно працювати з нерівномірно розподіленими даними. Після кожного згорткового шару застосовується MaxPooling1D, який зменшує розмірність проміжних представлень і водночас підсилює виділення найбільш значущих сигналів, що покращує узагальнюючу здатність моделі.

Після згорткових блоків дані перетворюються у вектор за допомогою шару Flatten, що забезпечує перехід від просторової структури до повнозв’язних представлень. Подальший Dense-шар із 64 нейронами виконує високорівневу обробку ознак, а застосування Dropout із ймовірністю 0.3 слугує механізмом регуляризації, знижуючи ризик перенавчання шляхом випадкового вимкнення частини нейронів під час навчання. Завершальний шар із активацією sigmoid формує значення ймовірності належності до позитивного класу, що є стандартним підходом для бінарної класифікації.

Фінальний етап – компіляція моделі. Використовується оптимізатор Adam зі швидкістю навчання 0.001 для стабільного оновлення ваг. Функція втрат — binary_crossentropy, оптимальна для бінарної класифікації. Як метрики задано accuracy, precision та recall для комплексної оцінки моделі за умов дисбалансу класів.

Функція train реалізує повний цикл навчання нейронної мережі, включаючи підготовку даних, формування навчальної та тестової вибірок, налаштування механізмів контролю якості навчання та побудову графічних візуалізацій ключових метрик. Тренування CNN відбувається в 4 етапи (рисунок 3.2).


Рисунок 3.2 – Етапи тренування CNN

Першим етапом є завантаження даних з датасету, на якому здійснюється імпорт необхідних бібліотек для роботи з табличними даними, після чого ініціюється процес читання вхідного файлу, який містить усі транзакції у форматі CSV.

Після завантаження даних відбувається логічне розділення структури датасету на два основні компоненти: вхідні ознаки, що описують характеристики транзакцій, та цільову змінну, яка вказує на належність операції до шахрайської або легальної категорії (рисунок 3.3).


Рисунок 3.3 – Етап завантаження даних

Другим етапом є побудова згорткової нейронної мережі для виявлення прихованих закономірностей у послідовних фінансових даних. Архітектура містить згорткові та підвибіркові шари для виділення важливих ознак і зменшення шуму, після чого дані передаються до повнозв’язаних шарів для класифікації та оцінювання ймовірності шахрайства (рисунок 3.4).

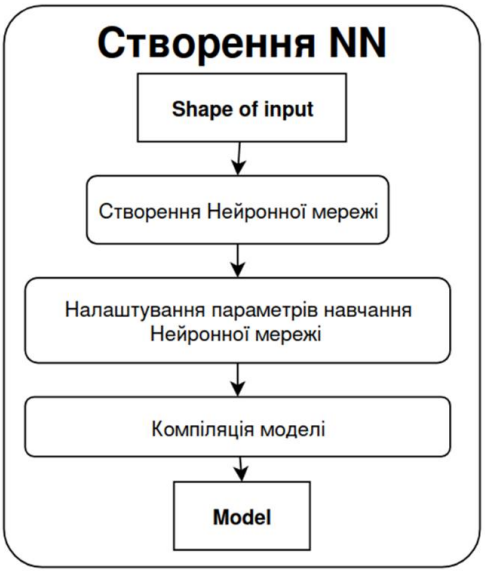



Рисунок 3.4 – Етап побудови CNN

Третім етапом є безпосереднє навчання CNN та оцінювання результатів навчання, що забезпечує адаптацію нейронної мережі до характеристик фінансових транзакцій і перевірку її ефективності на тестових даних (рисунок 3.5).

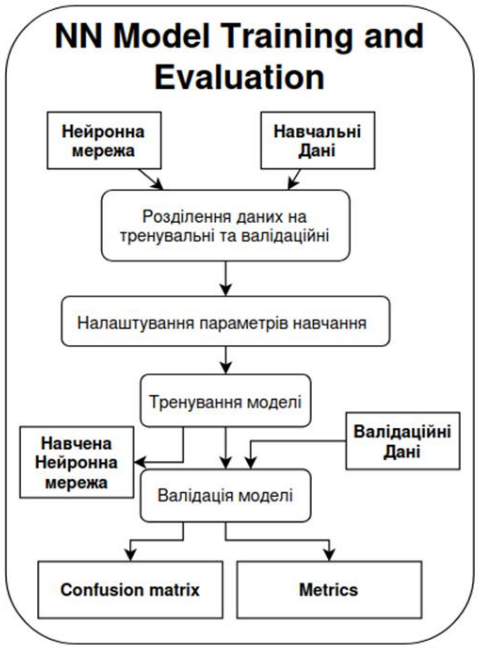



Рисунок 3.5 – Етап навчання та оцінювання CNN

На даному етапі вхідний набір даних поділяється на дві частини: тренувальну та валідаційну вибірки. Таке розбиття дозволяє уникнути перенавчання моделі,

забезпечує об’єктивну оцінку її продуктивності та гарантує надійну перевірку узагальнюючої здатності. Для контролю процесу навчання застосовується механізм ранньої зупинки (Early Stopping), який автоматично припиняє тренування, якщо протягом заданого числа епох (наприклад, 10–15) не спостерігається покращення обраної метрики похибки на валідаційній вибірці. Це ефективно запобігає деградації узагальнюючої здатності моделі та економить обчислювальні ресурси.

У процесі тренування модель ітеративно оптимізує свої вагові параметри, мінімізуючи значення функції втрат та одночасно підвищуючи точність класифікації на обох вибірках. Після завершення навчання або спрацьовування ранньої зупинки проводиться підсумкове оцінювання моделі на незалежній тестовій вибірці з обчисленням показників ефективності.

Модель компілюється з використанням адаптивного алгоритму оптимізації Adam з початковим коефіцієнтом навчання 0,001, який динамічно коригує швидкість навчання для кожного параметра окремо, що сприяє швидшій і стабільнішій конвергенції. Як функція втрат застосовується бінарна перехресна ентропія (Binary Cross-Entropy), яка кількісно оцінює розбіжність між прогнозованими ймовірностями приналежності до позитивного класу та фактичними бінарними мітками, ефективно керуючи процесом оновлення ваг. Під час навчання одночасно відстежуються метрики точності та повноти, що дає змогу всебічно оцінити поведінку моделі, особливо в умовах значного дисбалансу класів, коли проста точність може вводити в оману.

Функція evaluate виконує комплексну та багатогранну оцінку продуктивності побудованої моделі класифікації шахрайських транзакцій, охоплюючи кілька важливих і послідовних етапів: ретельну підготовку даних, обчислення основних метричних показників ефективності, формування інтегральної F1-міри як гармонічного середнього точності та повноти, а також детальний візуальний аналіз результатів за допомогою матриці змішування (confusion matrix).

На початковому етапі функція здійснює вилучення ознак (features) та цільової змінної (target) з переданого тестового набору даних. Вхідні матриці

попередньо перетворюються у спеціальний формат, повністю сумісний зі згортковою архітектурою моделі, а саме – у тривимірний тензор виду (кількість зразків, кількість ознак, 1), де останній вимір додається для імітації каналу зображення. Це перетворення забезпечує повну узгодженість структури даних із тим форматом, який очікує модель під час виконання прогнозування, уникаючи можливих помилок несумісності.

Після цього дані поділяються на навчальну та тестову підмножини із фіксованою часткою тестових прикладів (25%), що реалізується за допомогою стандартної функції train_test_split. Тестова вибірка відіграє ключову роль незалежного набору даних, на якому об’єктивно оцінюється здатність моделі узагальнювати набуті знання, ефективно виявляти шахрайські транзакції та уникати перенавчання на даних, що використовувалися під час тренування.

Основні кількісні характеристики продуктивності моделі отримуються шляхом виклику вбудованого методу model.evaluate, який повертає значення функції втрат (loss), загальної точності (accuracy), точності позитивного класу (precision) та повноти (recall). Додатково, на основі цих показників розраховується F1-міра – інтегрована метрика, яка є гармонічним середнім між precision та recall і виявляється особливо корисною у випадках сильної асиметрії класів, типової для задач детекції шахрайства. Усі отримані результати виводяться у чіткому та зрозумілому формалізованому вигляді для зручного аналізу.

Далі функція переходить до побудови та візуалізації матриці змішування, що надає можливість детально дослідити, як саме модель класифікує окремі зразки та де саме виникають помилки. Спершу обчислюються прогнозні ймовірності для тестової вибірки, які потім перетворюються у бінарні передбачення на основі стандартного порогу 0.5. Матриця змішування чітко показує кількість правильних і помилкових класифікацій для кожного класу, зокрема справжні позитиви (True Positives), хибні позитиви (False Positives), справжні негативи (True Negatives) та хибні негативи (False Negatives). Її графічна візуалізація за допомогою інструменту ConfusionMatrixDisplay з бібліотеки scikit-learn дає змогу наочно оцінити сильні й

слабкі сторони моделі, наприклад, схильність до надмірних помилкових спрацьовувань (false alarms) або небезпечних пропусків реальних випадків шахрайства, що є критичним для практичного впровадження системи.

Функція predict виконує локалізований і детальний аналіз роботи побудованої моделі шляхом здійснення прогнозування класу для одного конкретного екземпляра даних, що дозволяє глибше зрозуміти її поведінку в реальних умовах. На відміну від процедур повноцінного навчання моделі чи комплексної оцінки її загальної точності на великому наборі даних, даний метод повністю зосереджений на індивідуальному передбаченні, що відкриває можливість проводити точкову діагностику та аналіз поведінки моделі саме на рівні окремої транзакції, виявляючи потенційні сильні чи слабкі сторони в конкретних випадках.

Функція спершу виокремлює ознаки (features) та цільову змінну (class) безпосередньо з підготовлених даних без будь-якої додаткової обробки, нормалізації чи масштабування, що спрощує процес і робить його швидшим для одиночного прогнозу. Хоча в коді функції присутній крок поділу вибірки на тренувальну й тестову частини за допомогою стандартних інструментів, цей етап тут практично не використовується для фактичного навчання чи оцінки і виконується переважно для забезпечення повної узгодженості з загальним робочим процесом та структурою інших функцій проєкту. Основне ж завдання полягає в тому, щоб сформувати точний прогноз саме для зразка з заданим індексом x: його вектор ознак ретельно приводиться до спеціального формату тривимірного тензора виду (1, кількість_ознак, 1), який є повністю сумісним зі згортковою архітектурою моделі та імітує структуру вхідних даних під час тренування. Після цього, за допомогою виклику методу model.predict, обчислюється ймовірність належності до класу шахрайства, а функція додатково виводить фактичну реальну мітку класу для цього зразка, що дає змогу користувачеві негайно оцінити коректність отриманого передбачення, порівняти прогноз з дійсністю та провести якісний аналіз помилок чи успішних класифікацій у конкретному випадку.

Функція both_metrics є комплексним інструментом для оцінки та порівняння продуктивності моделі глибокого навчання на задачі бінарної класифікації, зокрема у контексті виявлення фінансового шахрайства. Основна мета цієї функції полягає у підготовці даних, обчисленні ключових метрик продуктивності моделі та їх візуальному порівнянні з заздалегідь визначеними контрольними значеннями.

На першому етапі функція завантажує датасет через метод load, що повертає структуру даних із ознаками транзакцій і класовими мітками (шахрайська чи легальна транзакція). Після завантаження дані копіюються для подальшої обробки, що дозволяє уникнути зміни оригінального набору. Далі виконується масштабування ключових числових ознак: стовпець Amount нормалізується за допомогою RobustScaler, що робить його нечутливим до викидів, а стовпець Time приводиться до інтервалу [0,1] методом мінімакс нормалізації. Ці кроки є критично важливими для забезпечення коректного навчання моделі, оскільки вхідні ознаки з різними масштабами можуть впливати на процес оптимізації і призводити до домінування одних ознак над іншими.

Після попередньої обробки формуються матриці ознак та міток. Ознаки X отримуються видаленням стовпця Class, тоді як мітки y відповідають значенням цього стовпця. Далі дані переформатовуються у тривимірну структуру X.shape[0]xX.shape[1]x1, що необхідно для подання даних у згорткову нейронну мережу (CNN), оскільки такі моделі очікують тривимірний ввід, де третій вимір відображає канал, подібно до кольорових каналів у зображеннях.

Наступний етап полягає у розбитті даних на навчальну і тестову вибірки за допомогою функції train_test_split, де 25% даних відводяться на тестування, а залишок – на навчання. Фіксація random_state гарантує відтворюваність результатів.

Після підготовки даних відбувається безпосередня оцінка моделі за допомогою методу model.evaluate(), який повертає список метрик, включаючи loss, accuracy, precision і recall. На основі precision та recall обчислюється F1-score за стандартною формулою.

Далі результати моделі зберігаються у списку cnn_vals, а контрольні значення FDIC – у списку fdic_vals. Ці значення використовуються для порівняння продуктивності власної моделі з результатами еталонного методу, що дозволяє швидко оцінити ефективність і надійність нової моделі.

Останній етап функції відповідає за візуалізацію результатів. Для кожної з метрик (Accuracy, Precision, Recall, F1-score) створюється окремий стовпчиковий графік, на якому відображаються порівняння двох методів: власного та FDIC. Значення метрик підписуються безпосередньо на стовпчиках, що підвищує наочність графіка і полегшує порівняння. Окрім цього, графіки містять заголовки, підписи осей та сітку, що робить їх більш інформативними та читабельними.

Таким чином, функція both_metrics інтегрує ключові аспекти оцінки моделі: підготовку даних, масштабування та нормалізацію ознак, розбиття на тренувальні та тестові набори, обчислення критичних метрик продуктивності та наочне порівняння результатів з еталонними даними. Вона дозволяє швидко оцінити ефективність моделі, визначити сильні та слабкі сторони та прийняти рішення щодо можливих покращень, наприклад оптимізації архітектури CNN або зміни підходу до балансування класів. Функція корисна у задачах фінансового шахрайства, де висока точність та своєчасне виявлення аномалій є критично важливими, а наочне порівняння з існуючими методами дозволяє швидко оцінити переваги нового алгоритму.

## Висновки до розділу 3

Виконано програмну реалізацію методу виявлення шахрайських транзакцій, описано її функціональні складові та структуру. Програмна реалізація виконана за модульним принципом, який забезпечує гнучкість, ефективність та можливість інтеграції з іншими платформами. Модулі взаємодіють у певній послідовності, починаючи з завантаження та обробки даних і завершуючи виведенням результатів.

Для програмної реалізації використано мову Python, фреймворки TensorFlow, а також NumPy і Pandas із частковим використанням хмарних сервісів Google Colab, та використання GPU з підтримкою CUDA, що забезпечує достатню обчислювальну потужність для навчання й тестування моделі.

Описано функції модулів програмної реалізації. Модуль Dataset забезпечує завантаження та обробку вхідної інформації з датасету, модуль Model виконує створення, навчання та валідацію моделі, яка відповідає за класифікацію транзакцій а модуль Comparison відповідає за виведення метрик створеного методу та методу з яким порівняно створений метод.

# Розділ 4 Експериментальне тестування методу виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж

## 4.1 Характеристика експериментального датасету

Для проведення дослідження з виявлення шахрайських фінансових операцій було використано сформований набір даних, отриманий із публічного ресурсу Credit Card Fraud Detection Dataset на платформі Kaggle (Див пункт 2.3). Для отримання більш ґрунтовного уявлення про структуру датасету було побудовано гістограми для кожної ознаки (рисунок 4.1).

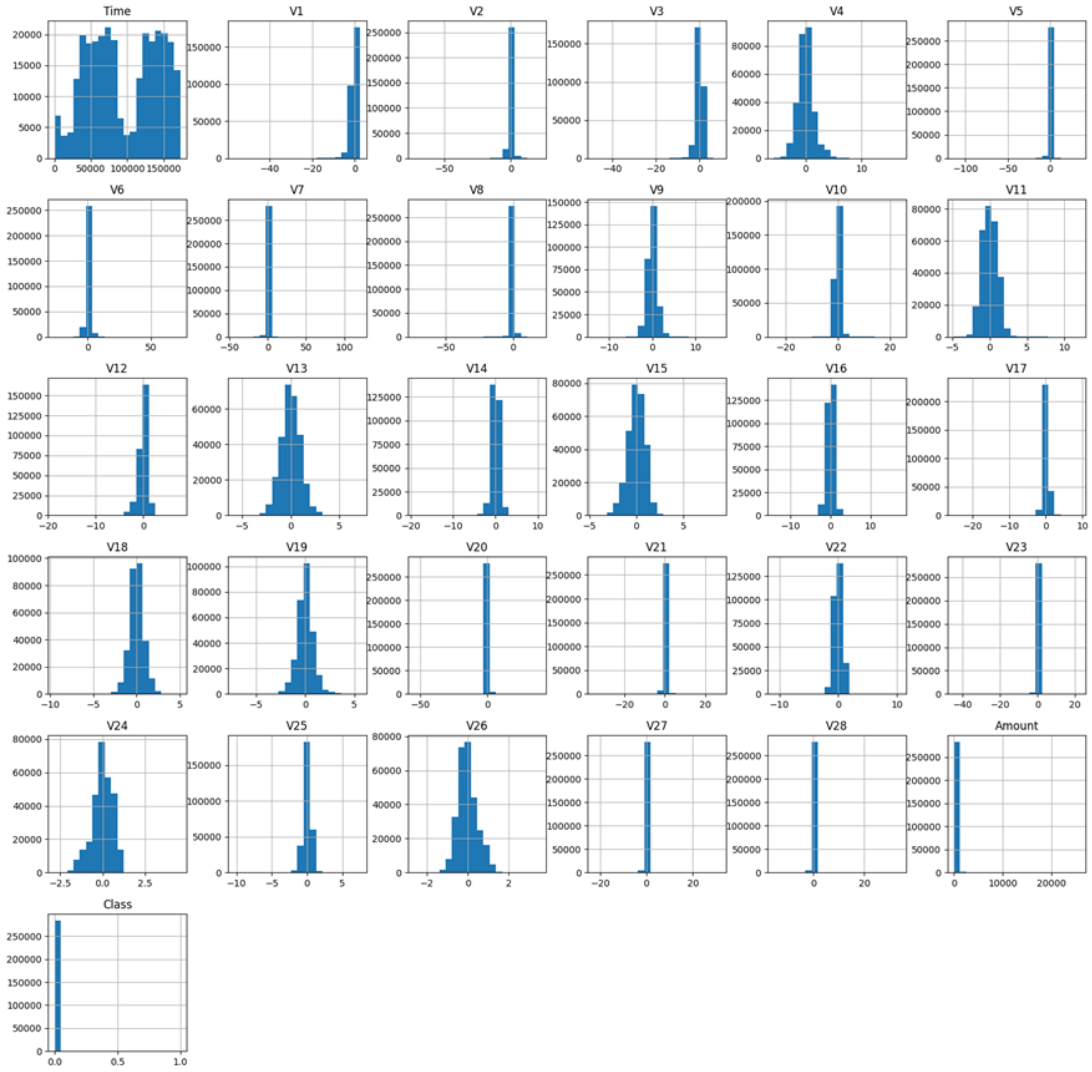



Рисунок 4.1 – Гістаграми ознак транзакцій

Значення ознаки Amount(Сума) у 75% випадків становить 77 або менше, а середнє значення дорівнює 88. За умов максимально зафіксованого значення 25691 це свідчить про суттєву асиметрію розподілу та наявність поодиноких великих транзакцій, які істотно впливають на масштаб ознаки (таблиця 4.1).

Таблиця 4.1 – Характеристики ознаки Amount(Сума)

| Характеристка | Значення |
|---|---|
| Count | 284807 |
| Mean | 88,349619 |
| Std | 250,120109 |
| Min | 0 |
| 25% | 5,6 |
| 50% | 22 |
| 75% | 77,165 |
| max | 25691,16 |

Для усунення проблеми асиметричного розподілу та впливу поодиноких великих транзакцій, що суттєво спотворюють масштаб ознаки, параметр Amount(Сума) було приведено до стабільнішого діапазону шляхом застосування робастного масштабування. Такий підхід зменшує вплив викидів і забезпечує рівномірніший внесок ознаки в подальше моделювання (таблиця 4.2).

Таблиця 4.2 – Характеристики ознаки Amount(Сума) після масштабування

| Характеристка | Значення |
|---|---|
| Count | 284807 |
| Mean | 0,927124 |
| Std | 3,495006 |
| Min | -0,307412 |
| 25% | -0,229162 |
| 50% | 0 |
| 75% | 0,77838 |
| max | 358,683155 |

Значення ознаки Time(Час) представляє з себе секунду в яку було зроблено транзакцію. Цей показник використовується для аналізу часових закономірностей у поведінці користувачів та виявлення потенційних аномалій, пов’язаних із нетиповою активністю у певні проміжки часу (таблиця 4.3).

Таблиця 4.3 –Характеристики ознаки Time(Час)

| Характеристка | Значення |
|---|---|
| Count | 284807 |
| Mean | 94813,859575 |
| Std | 47488,145955 |
| Min | 0 |
| 25% | 54201,5 |
| 50% | 84692 |
| 75% | 139320,5 |
| max | 172792 |

Оскільки всі транзакції були здійснені протягом двох днів, ознака Time(Час) не має суттєвої практичної цінності для розв’язання задачі класифікації. У зв’язку з цим її було нормалізовано та приведено до інтервалу [0; 1], що дозволило мінімізувати її вплив на модель та забезпечити узгодженість масштабів між усіма ознаками (таблиця 4.4).

Таблиця 4.4 – Характеристики ознаки Time(Час) після Нормалізації

| Характеристка | Значення |
|---|---|
| Count | 284807 |
| Mean | 0,548717 |
| Std | 0,274828 |
| Min | 0 |
| 25% | 0,31681 |
| 50% | 0,490138 |
| 75% | 0,806290 |
| max | 1 |

Було встановлено, що ознака Class створює значний дисбаланс у вибірці, оскільки кількість шахрайських транзакцій суттєво поступається числу валідних операцій. Така нерівномірність негативно впливає на здатність моделі коректно розпізнавати рідкісні випадки шахрайства, оскільки алгоритм схильний орієнтуватися на домінуючий клас. Для усунення цієї проблеми було застосовано метод SMOTE, який штучно генерує нові зразки міноритарного класу шляхом інтерполяції між наявними прикладами.

Кожна транзакція в наборі супроводжується докладним набором параметрів, які відображають різні аспекти фінансової операції. Зокрема, для кожного запису вказано час здійснення транзакції, суму платежу, а також 28 анонімізованих змінних, отриманих у результаті перетворення методом головних компонент (PCA), що приховують початкові чутливі фінансові дані. Такі змінні описують приховані закономірності у поведінці користувачів і дозволяють моделі виявляти відхилення від типових шаблонів проведення платежів. Наявність цих параметрів забезпечує більш точне навчання моделей машинного навчання та створює можливість для глибокого аналізу структури як звичайних, так і потенційно шахрайських транзакцій.

Датасет було розділено на підмножини для забезпечення об’єктивності та достовірності оцінювання роботи моделі. Навчальна підмножина (75%) використовується для безпосереднього навчання нейронної мережі, підбору оптимальних ваг і налаштування параметрів моделі. Валідаційна підмножина (25%) слугує для контролю якості моделі під час навчання, даючи змогу своєчасно виявляти ознаки перенавчання та коригувати гіперпараметри.Таке співвідношення забезпечує збалансоване поєднання достатнього обсягу навчальних даних із надійною перевіркою узагальнювальної здатності моделі, що є особливо важливим у задачах виявлення шахрайських транзакцій, де точність і стабільність результатів мають критичне значення.

Сформований набір даних є оптимальною базою для навчання та тестування методів, орієнтованих на виявлення шахрайських фінансових операцій. Завдяки

високій структурованості, збалансованості після застосування методу SMOTE та наявності ретельно підготовлених ознак, він забезпечує високу точність і надійність побудованих моделей. Використання такого набору сприяє створенню систем, здатних ефективно розпізнавати навіть незначні відхилення у поведінці користувачів, що є критично важливим для фінансової безпеки.

Крім того, застосування цього датасету відкриває широкі можливості для подальших досліджень у сфері інтелектуального аналізу транзакцій, розробки адаптивних алгоритмів виявлення шахрайства та підвищення стійкості фінансових систем до кібератак і зловживань.

Перед початком експериментів було здійснено налаштування параметрів моделі, зокрема визначено кількість епох навчання, швидкість навчання та метрики оцінки. Для забезпечення достовірності результатів набір даних було поділено на навчальну, валідаційну та тестову підмножини.

Тестування проводилося виключно на тестовій множині, яка не використовувалася під час процесу навчання, що забезпечило об’єктивність оцінки. Для підвищення швидкості обробки та імітації умов реального часу тестування здійснювалося на графічному процесорі (GPU) з підтримкою технології CUDA.

### 4.2 Результати експериментальної перевірки

Навчання нейронної мережі було проведено на попередньо обробленому та збалансованому наборі даних, який містив приклади шахрайських і не шахрайських транзакцій. Процес навчання здійснювався з використанням навчальної підмножини, що становила 75% від загального обсягу даних, із застосуванням методу ранньої зупинки для запобігання перенавчанню. В результаті навчання моделі було отримано метрики accuracy, precision, recall та графіки метрик accuracy, precision та recall на кожній епохі навчання (рисунок 4.2-4.4).

Показники accuracy, precision, recall на першій епосі становлять 97,6%, 97,9%, 97,3% відповідно, а до десятої епохи зростають до 99.9%, що свідчить про

стабільне та поступове покращення навчання моделі.

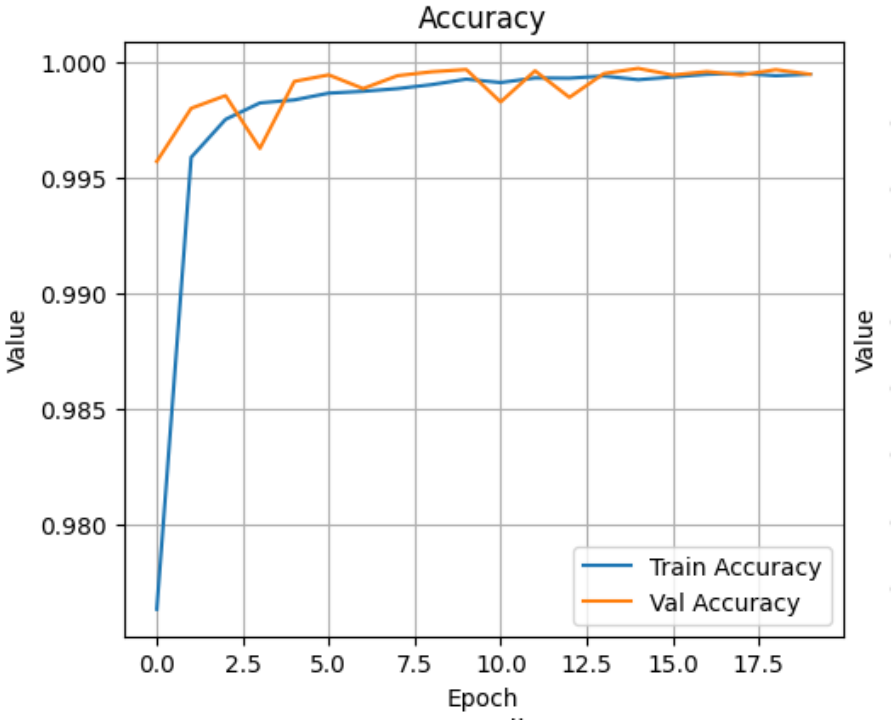



Рисунок 4.2 – Графік метрики accuracy протягом процесу навчання

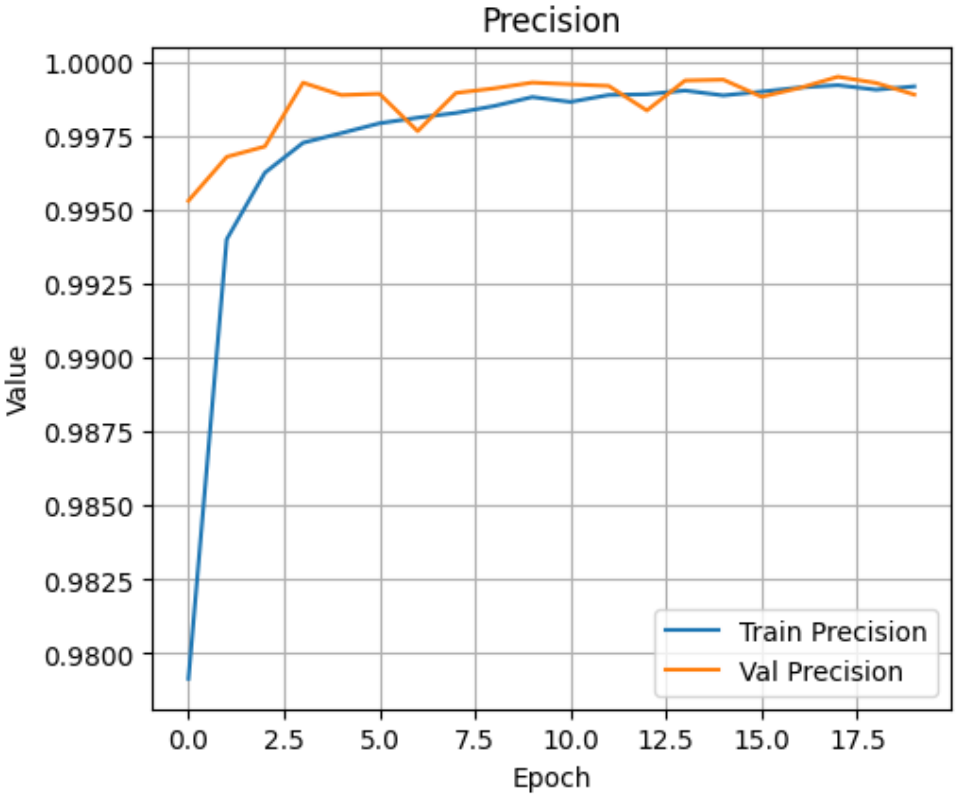



Рисунок 4.3 – Графік метрики precision протягом процесу навчання

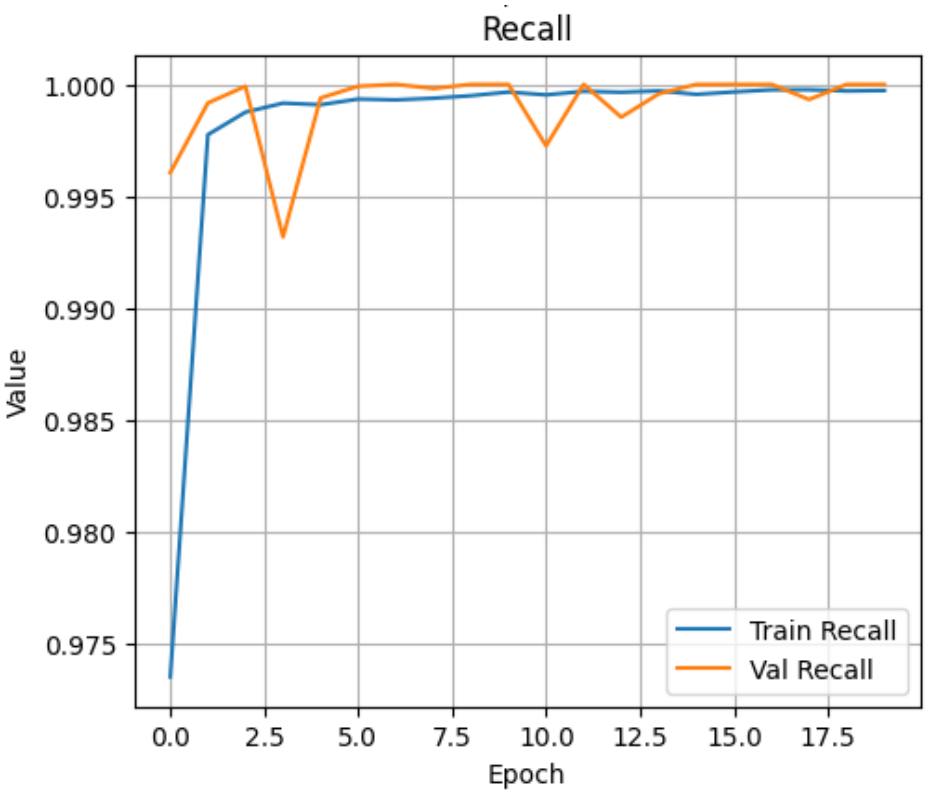



Рисунок 4.4 – Графік метрики recall протягом процесу навчання

Навчання нейронної мережі та тестування результатів навчання виконувалося відповідно на навчальній та валідаційній підмножинах, що становили 75% та 25% від усього датасету. Метрики Accuracy, Precision, Recall, F1 Score, Confusion Matrix тестування методу на навчальній підмножині наведені на таблиці 4.5, рисунку 4.5. та в таблиці 4.7.

Таблиця 4.5 – Метрики оцінки на навчальній підмножині

| Метрика | Значення |
|---|---|
| Accuracy | 0,9996 |
| Precision | 0,8660 |
| Recall | 0,8753 |
| F1 Score | 0,8706 |

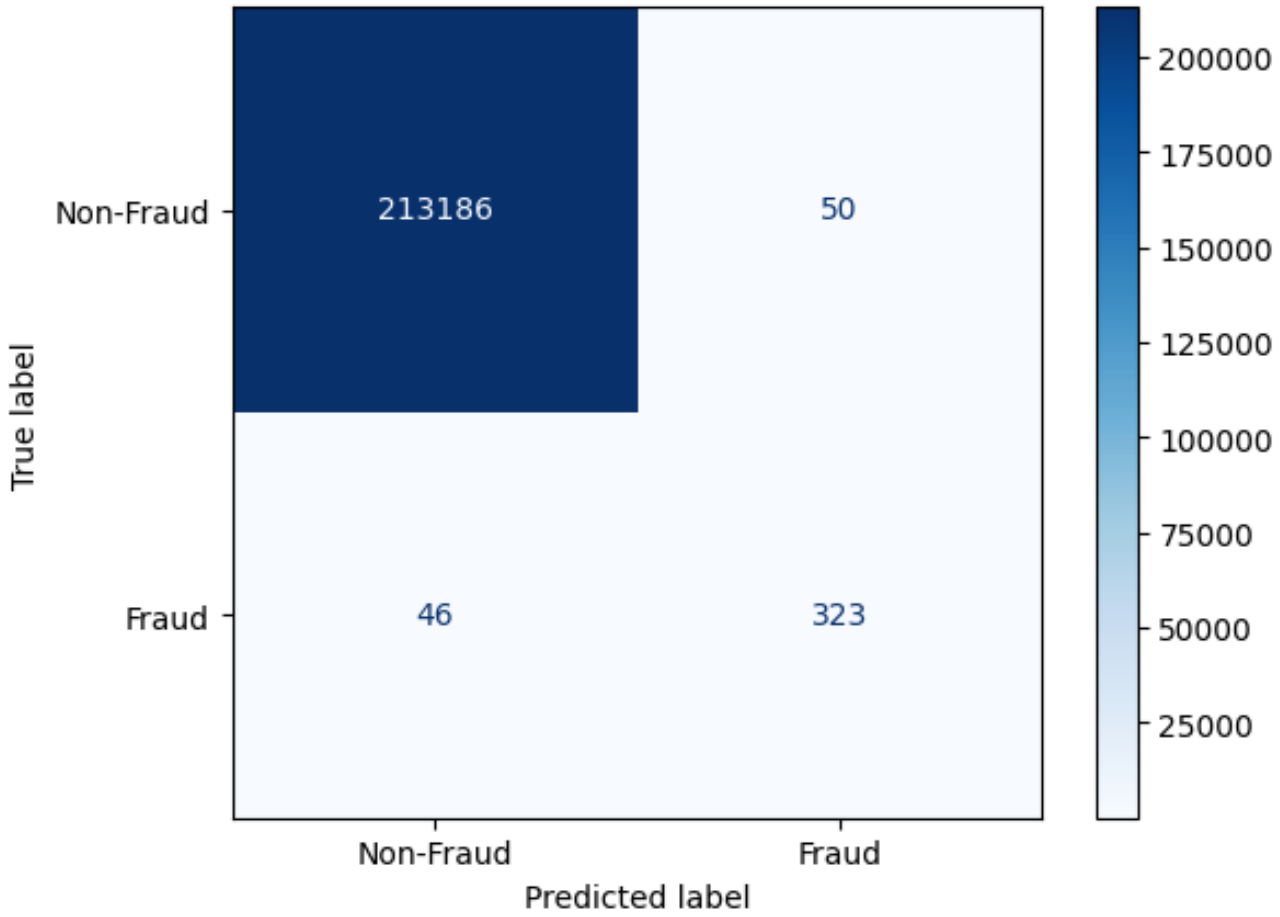



Рисунок 4.5 – Confusion Matrix на навчальній підмножині

Метрики Accuracy, Precision, Recall, F1 Score, Confusion Matrix тестування результатів навчання нейронної мережі на валідаційній підмножині наведені на таблиці 4.6, рисунку 4.9. та в таблиці 4.7.

Таблиця 4.6 – Метрики оцінки на валідаційній підмножині

| Метрика | Значення |
|---|---|
| Accuracy | 0,9996 |
| Precision | 0,8833 |
| Recall | 0,8618 |
| F1 Score | 0,8724 |

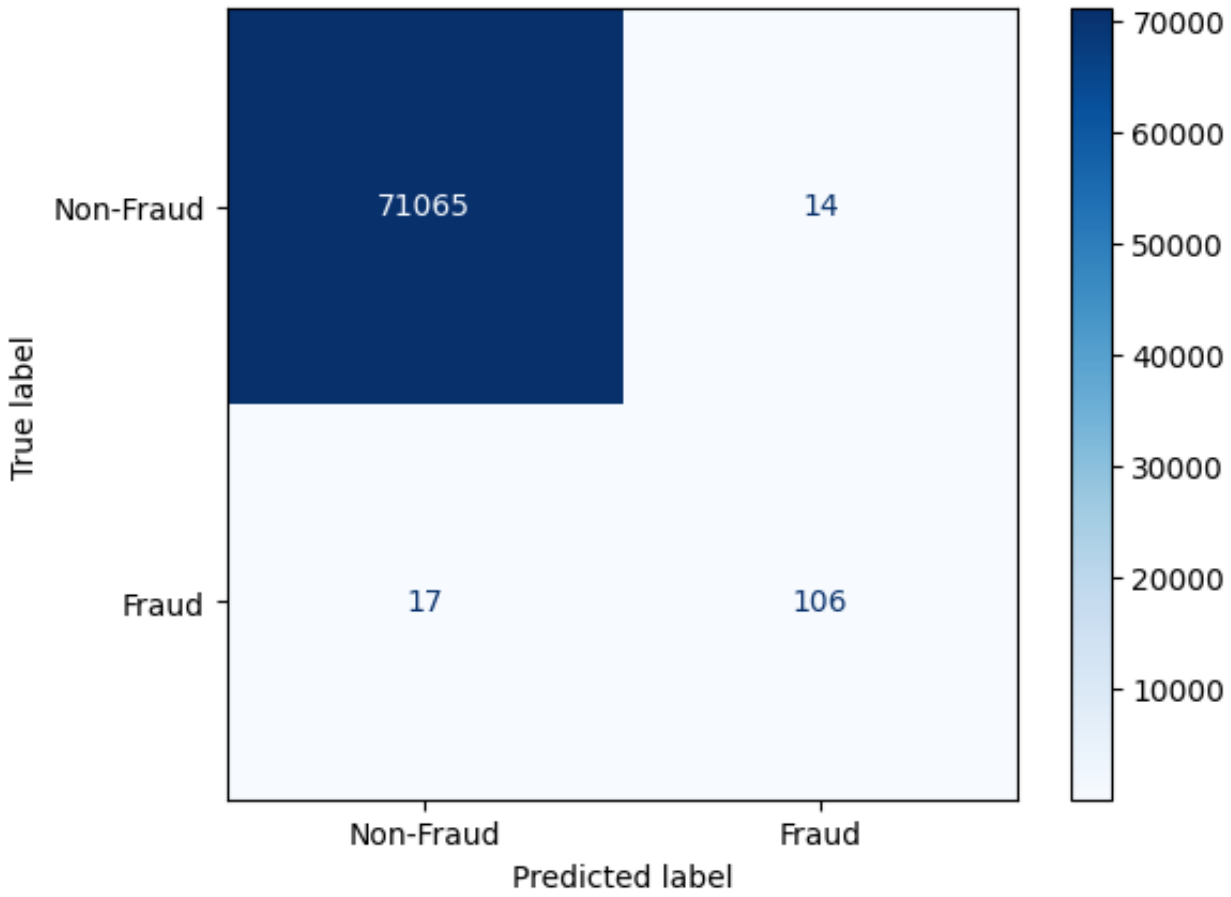



Рисунок 4.6 – Матриця неточностей методу на валідаційній підмножині

На основі аналізу отриманих результатів можна з упевненістю зробити висновок, що запропонований метод демонструє надзвичайно високий рівень ефективності в задачі виявлення шахрайських транзакцій.

Значення метрик, оцінених на навчальній підмножині (таблиця 4.5), складають: Accuracy – 99,96%, Precision – 88,33%, Recall – 86,18% та F1-score – 87,24%. На валідаційній підмножині (таблиця 4.6) відповідні значення метрик становлять: Accuracy – 99,96%, Precision – 86,60%, Recall – 87,53% та F1-score – 87,06%.

Такі високі та стабільні значення метрик на обох підмножинах свідчать про відмінну точність і збалансованість моделі. Вона ефективно виявляє шахрайські транзакції, правильно класифікує більшість легітимних операцій, мінімізує помилки та демонструє добру узагальнювальну здатність.

Таблиця 4.7 – Метрики оцінювання для навчальної та валідаційної вибірок

| Вибірка | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| навчальна | 0,9996 | 0,8660 | 0,8753 | 0,8706 |
| валідаційна | 0,9996 | 0,8833 | 0,8618 | 0,8724 |

Значення метрики Accuracy перевищує 99,9 %, що пояснюється сильно незбалансованими даним і є типовим для шахрайства. Незважаючи на високе значення Accuracy, дана метрика не є достатньою для оцінки якості моделі в умовах значного дисбалансу класів, тому додатково аналізуються Recall, Precision та F1-score для міноритарного класу.

### 4.3 Порівняння результатів з іншими методами

Створений метод було порівняно з методом FDIC [45]. У FDIC методі транзакційні дані (30 ознак, включно з PCA-компонентами) спочатку перетворюються на зображення розміром 5×6 пікселів, після чого для класифікації застосовуються згорткові нейронні мережі (CNN), зокрема модифіковані архітектури AlexNet та ResNet-50. Автори досягли точності (accuracy) 99.26 % та AUC 0,98.

Як демонструють результати, наведені в таблиці, розроблений підхід забезпечив найвищі значення всіх оцінюваних метрик, що свідчить про його перевагу у виявленні шахрайських транзакцій порівняно з FDIC методом (таблиця 4.8).

Таблиця 4.8 – Результати порівняння двох методів

| Метод | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| FDIC | 0,9995 | 0,9146 | 0,8035 | 0,8549 |
| Створений метод | 0,9996 | 0,8660 | 0.8753 | 0,8706 |

Створений метод та метод FDIC демонструють дуже близькі результати, з незначною перевагою FDIC за метрикою Precision – 0,9146. Створений метод має вищий Recall – 0,8753 та вищий F1-Score 0,8706, а також має невелику перевагу за величиною Accuracy – 0,9996.

Це свідчить про те, що створений метод краще балансує між виявленням шахрайських транзакцій та мінімізацією помилкових спрацьовувань, пропускаючи менше шахрайства порівняно з FDIC. Водночас FDIC точніше класифікує легітимні транзакції, але може пропускати більшу частку шахрайських випадків.Таким чином, створений метод є більш надійним у контексті повного виявлення шахрайства за рахунок кращого балансу метрик, хоча й з дещо вищим ризиком помилкової класифікації деяких легітимних операцій як шахрайських.

Метрики Accuracy, Precision, Recall, F1 Score для створеного методу та методу FDIC наведені на на рисунку 4.10.

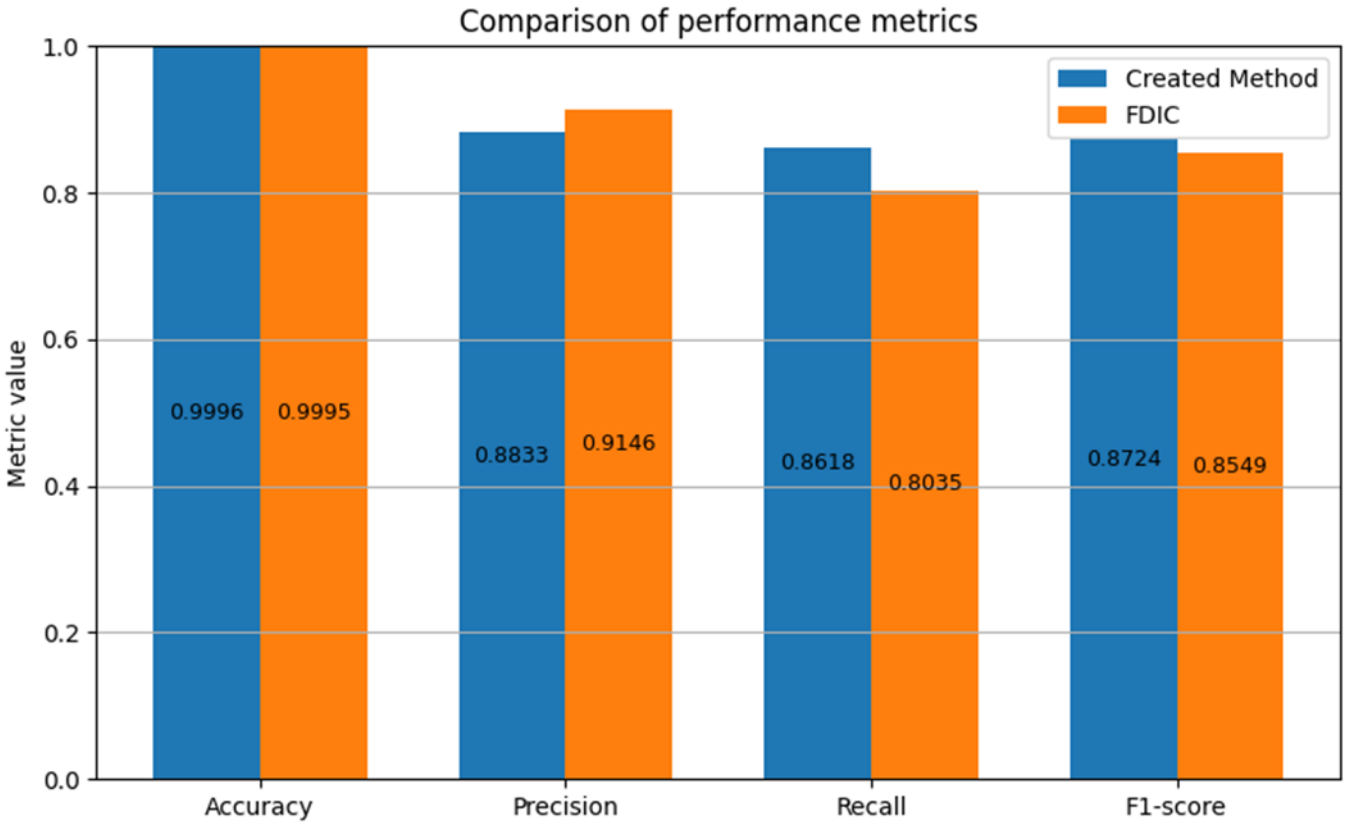



Рисунок 4.7 – Метрики сроєктованого методу та методу FDIC

## Висновки до розділу 4

Виконано дослідження точності методу виявлення шахрайських операцій у фінансових транзакціях.

Для навчання та валідації згорткової нейронної мережі, що використовується у спроєктованому методі, використовувався Credit Card Fraud Detection Dataset, що містить 284807 транзакцій. Навчальна та валідаційна вибірки формувалися у співвідношенні 75/25. Значення метрик, оцінених на навчальній складають: Accuracy – 99,96%, Precision – 88,33%, Recall – 86,18% та F1-score – 87,24%. На валідаційній підмножині відповідні значення метрик становлять: Accuracy – 99,96%, Precision – 86,60%, Recall – 87,53% та F1-score – 87,06%. Значення метрик оцінених на валідаційній підмножині складають значення метрик оцінених на валідаційній підмножині

Для оцінювання точності спроектованого методу проведено порівняння з дослідженнями, що використовують модель FDIC. Спроєктований метод та метод FDIC демонструють дуже близькі результати, з незначною перевагою FDIC за метрикою Precision – 0,9146. Створений метод має вищий Recall – 0,8753 та вищий F1-Score 0,8706, а також має невелику перевагу за величиною Accuracy – 0,9996.

Це свідчить про те, що створений метод краще балансує між виявленням шахрайських транзакцій та мінімізацією помилкових спрацьовувань, пропускаючи менше шахрайства порівняно з FDIC. Водночас FDIC точніше класифікує легітимні транзакції, але може пропускати більшу частку шахрайських випадків. Таким чином, створений метод є більш надійним у контексті повного виявлення шахрайства за рахунок кращого балансу метрик, хоча й з дещо вищим ризиком помилкової класифікації деяких легітимних операцій як шахрайських.

Отже, хоча FDIC демонструє вищу метрику Precision, що робить його кращим для виявлення легітимних транзакцій, але він є менш надійним для задачі виявлення шахрайства. Запропонований метод забезпечує вищу ймовірність виявлення шахрайських транзакцій, але з вищою ймовірністю помилкового позначення деяких легітимних операцій як підозрілих.

# Загальні висновки

У процесі виконання кваліфікаційної роботи магістра було успішно досягнуто мету, визначену на етапі постановки задачі, а саме – підвищення точності виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж.

Для досягнення поставленої мети виконані такі задачі:

– проведено аналіз методів виявлення шахрайських транзакцій у фінансових операціях;

– проведено аналіз можливостей, переваг та недоліків згорткових нейронних мереж для виявлено шахрайських транзакцій у фінансових операціях;

– спроєктувано метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж;

– виконано програмну реалізацію методу;

– виконано дослідження точності виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж.

Спроєктовано метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж. Описано загальну схему методу та архітектуру моделі згорткової нейронної мережі.

Наведено послідовність навчання нейронної мережі з використанням датасету Credit Card Fraud Detection та обробкою даних з нормалізацією та масштабуванням ключових класифікаційних параметрів та балансування даних.

Визначено критерії для оцінювання точності спроєктованого методу, а саме точність класифікації (Accuracy), повнота (Recall), точність передбачення (Precision), 1-міра (F1 Score), матриця плутанини(Confusion Matrix).

Виконано програмну реалізацію методу на мові програмування Python та фреймворку TensorFlow.

Виконано дослідження точності методу виявлення шахрайських операцій у фінансових транзакціях.

Для навчання та валідації згорткової нейронної мережі що використовується у спроєктованому методі використовувався Credit Card Fraud Detection Dataset, що містить 284807 транзакцій. Навчальна та валідаційна вибірки формувалися у співвідношенні 75/25. Значення метрик, оцінених на навчальній складають: Accuracy – 99,96%, Precision – 88,33%, Recall – 86,18% та F1-score – 87,24%. На валідаційній підмножині відповідні значення метрик становлять: Accuracy – 99,96%, Precision – 86,60%, Recall – 87,53% та F1-score – 87,06%. Значення метрик оцінених на валідаційній підмножині складають значення метрик оцінених на валідаційній підмножині

Для оцінювання точності спроектованого методу проведено порівняння з дослідженнями, що використовують модель FDIC.

Створений метод та метод FDIC демонструють дуже близькі результати, з незначною перевагою FDIC за метрикою Precision – 0,9146. Створений метод має вищий Recall – 0,8753 та вищий F1-Score 0,8706, а також має невелику перевагу за величиною Accuracy – 0,9996.

Отже, хоча FDIC демонструє вищу метрику Precision, що робить його кращим для виявлення легітимних транзакцій, але він є менш надійним для задачі виявлення шахрайства. Запропонований метод забезпечує вищу ймовірність виявлення шахрайських транзакцій, але з вищою ймовірністю помилкового позначення деяких легітимних операцій як підозрілих.

## Перелік посилань

1. Sorournejad, S., Zojaji, Z., Atani, R. E., & Monadjemi, A. H. (2016). A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective. arXiv. https://doi.org/10.48550/arXiv.1611.06439.

2. Tietsort J. R. Credit Card Fraud Detection: How To Spot & Avoid Fraud. Identity Guard. URL: https://www.identityguard.com/news/credit-card-fraud-detection.

3. Credit card scams that are commonly used today. (2025). LifeLock. https://lifelock.norton.com/learn/fraud/credit-card-scams.

4. SEON. (2025). Credit card fraud detection: How it works prevention tips. https://seon.io/resources/credit-card-fraud-detection/.

5. Cherif, A., Badhib, A., Ammar, H., Alshehri, S., Kalkatawi, M., & Imine, A. (2023). Credit card fraud detection in the era of disruptive technologies: A systematic review. Journal of King Saud University – Computer and Information Sciences, 35(1), 203–221. https://doi.org/10.1016/j.jksuci.2022.10.006.

6. Breskuvienė, D., & Dzemyda, G. (2024). Enhancing credit card fraud detection: Highly imbalanced data case. Journal of Big Data, 11, Article 182. https://doi.org/10.1186/s40537-024-01059-5.

7. Verma, S., & Dhar, J. (2024). Credit Card Fraud Detection: A Deep Learning Approach. arXiv. https://doi.org/10.48550/arXiv.2409.13406.

8. Alrasheedi, M. A. (2025). Enhancing Fraud Detection in Credit Card Transactions: A Comparative Study of Machine Learning Models. Computational Economics. Advance online publication. https://doi.org/10.1007/s10614-025-11071-3.

9. Hilal, W., Gadsden, S. A., & Yawney, J. (2022). Financial fraud: A review of anomaly detection techniques and recent advances. Expert Systems with Applications, 193, Article 116429. https://doi.org/10.1016/j.eswa.2021.116429.

10. Mienye, I. D., Esenogho, E., & Modisane, C. (2025). Detecting Imbalanced Credit Card Fraud via Hybrid Graph Attention and Variational Autoencoder Ensembles. AppliedMath, 5(4), 2745–2772. https://doi.org/10.3390/appliedmath5040131

11. Breskuvienė D., Dzemyda G. Enhancing credit card fraud detection: highly imbalanced data case. J Big Data. 2024;11:182. DOI: https://doi.org/10.1186/s40537-024-01059-5. URL: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-024-01059-5.

12. Khandaker, H. (2025, May 27). Financial fraud detection in the AI era: Best practices for compliance and risk management. Avahi. https://avahi.ai/blog/financial-fraud-detection-in-the-ai-era.

13. Burnett, S. (2025, June 16). Banking fraud detection statistics 2025: Prevalence, impact, and prevention strategies. CoinLaw. https://coinlaw.io/banking-fraud-detection-statistics.

14. Decision trees: From efficient prediction to responsible AI. (2023). Frontiers in Artificial Intelligence, 6, Article 1124553. https://doi.org/10.3389/frai.2023.1124553.

15. Bettaieb, O. AI fraud detection: How banks are fighting financial crime in 2025. Aloa. https://aloa.co/ai/resources/industry-insights/ai-fraud-detection-banking.

16. Mienye I. D., Jere N. A Survey of Decision Trees: Concepts, Algorithms, and Applications. IEEE Access. 2024;12:86716–86727. DOI: https://doi.org/10.1109/ACCESS.2024.3416838.

17. Random forest: A complete guide for machine learning. (n.d.). Built In. https://builtin.com/data-science/random-forest-algorithm.

18. What is machine learning? Definition, types, and examples. (2025, October 15). Coursera. https://www.coursera.org/articles/what-is-machine-learning.

19. Tufail, Sh., Riggs, H., Tariq, M., & Sarwat, A. I. (2023). Advancements and challenges in machine learning: A comprehensive review of models, libraries, applications, and algorithms. Electronics, 12(8), Article 1789. https://doi.org/10.3390/electronics12081789.

20. Bank, D., Koenigstein, N., & Giryes, R. (2023). Autoencoders. In L. Rokach, O. Maimon, & E. Shmueli (Eds.), Machine Learning for Data Science Handbook (pp. 353–374). Springer. https://doi.org/10.1007/978-3-031-24628-9_16.

21. Berahmand, K., Daneshfar, F., Salehi, E. S., Li, Y., & Xu, Y. (2024). Autoencoders and their applications in machine learning: A survey. Artificial Intelligence Review, 57, Article 28. https://doi.org/10.1007/s10462-023-10662-6.

22. Cluster analysis. (n.d.). In Wikipedia. https://en.wikipedia.org/wiki/Cluster_analysis.

23. Convolutional neural network. (n.d.). In Wikipedia. https://en.wikipedia.org/wiki/Convolutional_neural_network.

24. Karthika, J., & Senthilselvi, A. (2023). Smart credit card fraud detection system based on dilated convolutional neural network with sampling technique. Multimedia Tools and Applications, 82(20), 31691–31708. https://doi.org/10.1007/s11042-023-15730-1.

25. Ming, R., Abdelrahman, O., Innab, N., & Ibrahim, M. H. K. (2024). Enhancing fraud detection in auto insurance and credit card transactions: A novel approach integrating CNNs and machine learning algorithms. PeerJ Computer Science, 10, Article e2088. https://doi.org/10.7717/peerj-cs.2088.

26. Baria J. B., Baria V. D., Bhimla S. Y., Prajapati R., Rathva M., Patel S. Deep Learning based Improved Strategy for Credit Card Fraud Detection using Linear Regression. Journal of Electrical Systems. 2024;20(10s). URL: https://journal.esrgroups.org/jes/article/view/5257.

27. Arslan E., Güneş A. Fraud detection in enterprise resource planning systems using one-class support vector machine combined with convolutional neural network: The case of Spor Istanbul. Annals of Applied Sport Sciences. 2023;11(S1). DOI: http://dx.doi.org/10.61186/aassjournal.1222.

28. Faye E., Zhang W. Enhancing Financial Fraud Detection: The Efficacy of Convolutional Neural Networks // Journal of computer science and software applications. 2024. Vol. 4, No. 7. P. 25–35. [Електронний ресурс]. Режим доступу: https://www.mfacademia.org/index.php/jcssa.

29. Odeniyi O., Oyinloye O., Thompson A. Fraud Detection Using Multilayer Perceptron and Convolutional Neural Network // International Journal on Advances in Security. 2021. Vol. 14, no. 1 & 2. [Електронний ресурс]. Режим доступу:

https://personales.upv.es/thinkmind/dl/journals/sec/sec_v14_n12_2021/sec_v14_n12_2021_1.pdf.

30. De Amorim L. B. V., Cavalcanti G. D. C., Cruz R. M. O. The choice of scaling technique matters for classification performance [Електронний ресурс] / Lucas B. V. de Amorim, George D. C. Cavalcanti, Rafael M. O. Cruz. – arXiv, 2022. – 37 с. – DOI: 10.48550/arXiv.2212.12343. – URL: https://arxiv.org/pdf/2212.12343.

31. Normalization (statistics). (n.d.). In Wikipedia. https://en.wikipedia.org/wiki/Normalization_(statistics).

32. Oversampling and undersampling in data analysis. (n.d.). In Wikipedia. https://en.wikipedia.org/wiki/Oversampling_and_undersampling_in_data_analysis.

33. Random Forest Classifier - an overview [Електронний ресурс] // ScienceDirect Topics. – Elsevier. – URL: https://www.sciencedirect.com/topics/computer-science/random-forest-classifier.

34. Fraud credit card transaction detection using hybrid multilayer perceptron-random forest method [Електронний ресурс] // Learning Gate. – 2023. – URL: https://www.learning-gate.com/index.php/2576-8484/article/view/5823/2085.

35. Gradient boosting [Електронний ресурс] // Wikipedia. – URL: https://en.wikipedia.org/wiki/Gradient_boosting.

36. Ding L. An AutoEncoder enhanced light gradient boosting machine method for credit card fraud detection [Електронний ресурс] / L. Ding, L. Liu, Y. Wang, P. Shi, J. Yu // PeerJ Computer Science. – 2024. – Vol. 10 : e2323. – DOI: 10.7717/peerj-cs.2323. – URL: https://peerj.com/articles/cs-2323/.

37. Introduction to Convolution Neural Network. GeeksforGeeks. URL: https://www.geeksforgeeks.org/introduction-convolution-neural-network/.

38. Cacciari, I., & Ranfagni, A. (2024). Hands-on fundamentals of 1D convolutional neural networks: A tutorial for beginner users. Applied Sciences, 14(18), Article 8500. https://doi.org/10.3390/app14188500.

39. Singh, Y. (2022, March 22). Robust scaling: Why and how to use it to handle outliers. Proclus Academy. https://proclusacademy.com/blog/robust-scaler-outliers/.

40. What is SMOTE & how does it work? (2025, September 8). ML Journey. https://mljourney.com/what-is-smote-how-does-it-work/.

41. Credit Card Fraud Detection. Kaggle. Machine Learning Group - ULB. URL: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud.

42. Evaluation metrics in machine learning. (2025, October 29). GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/metrics-for-machine-learning-model/.

43. Confusion matrix. (n.d.). In Wikipedia. https://en.wikipedia.org/wiki/Confusion_matrix.

44. TensorFlow Core guide. (2023, March 2). TensorFlow. https://www.tensorflow.org/guide.

45. Pandas documentation (Version 2.3.3). (2025, September 29). https://pandas.pydata.org/docs/.

46. Terzi D. S. Explainable Credit Card Fraud Detection with Image Conversion [Електронний ресурс] / D. S. Terzi, U. Demirezen, Ş. Sağiroğlu // ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal. – 2021. – Vol. 10, iss. 1. – P. 63–76. – DOI: https://doi.org/10.14201/ADCAIJ20211016376. – Режим доступу: https://pdfs.semanticscholar.org/e437/d9ffa86ca415aeefcbed3703ed6d602598db.pdf.

# ДОДАТКИ

# Додаток А

# Наукова публікація

*Актуальні проблеми комп'ютерних наук*

УДК 004

Бондар О.П, Пасічник О.А., Скрипник Т.К., Петровський С.С.

*Хмельницький національний університет*

## МЕТОД ВИЯВЛЕННЯ ШАХРАЙСЬКИХ ТРАНЗАКЦІЙ У ФІНАНСОВИХ ОПЕРАЦІЯХ З ЗАСТОСУВАННЯМ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

*Розглянуто прикладні аспекти розробки інформаційної системи для виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж. Запропонована система забезпечує ефективну обробку та аналіз великих обсягів транзакційних даних, виявляючи приховані закономірності, характерні для шахрайської поведінки.*

*The applied aspects of developing an information system for detecting fraudulent transactions in financial operations using convolutional neural networks are considered. The proposed system provides effective processing and analysis of large volumes of transaction data, revealing hidden patterns characteristic of fraudulent behavior.*

Поява шахрайських транзакцій з банківськими картками збгається з розвитком електронних та безготівкових способів оплати. Шахрайські дії полягають у незаконному виведенні коштів з користувачів шляхом підробки даних картки або отримання несанкціонованого доступу до рахунків. Збільшення кількості випадків таких дій підштовхнуло фінансові організації до впровадження механізмів підвищеної безпеки, в даному випадку систем спостереження та виявлення сумнівних транзакцій.

Кредитні картки можуть бути фізично викрадені з гаманця або отримані віртуально з незахищених веб-сайтів, через витік даних або схеми крадіжки особистих даних. Наприклад, у 2019 році хакери зламали бази даних Capital One та оприлюднили інформацію про кредитні картки понад 100 мільйонів людей [1].

Захист фінансової інформації є важливим але, попри захист, який вони пропонується банками, кредитні картки не є захищеними від шахрайства. Навпаки, вони постійно є головною мішенню для злодіїв особистих даних, причому у 2024 році до Федеральної торгової комісії було подано майже 450 000 повідомлень про крадіжку особистих даних з кредитних карток [2].

Роль ШІ у виявленні фінансового шахрайства суттєво змінилася, перейшовши від статичних систем, заснованих на правилах, до динамічних, адаптивних алгоритмів. Структури раннього виявлення в основному залежали від заздалегідь визначених правил та ручного перегляду транзакцій, де сповіщення запускалися фіксованими параметрами, такими як сума транзакцій, частота або

географічне розташування. Хоча ці методи пропонували базовий рівень захисту, їм бракувало можливості адаптуватися до нових та складних шахрайських схем. Як наслідок, такі системи часто видавали високий рівень хибнопозитивних результатів, що призводило до неефективних розслідувань та погіршення обслуговування клієнтів.

Досягнення в машинному навчанні змінили цей процес, дозволивши системам аналізувати величезні набори даних та виявляти поведінкові моделі, що свідчать про шахрайську діяльність. Завдяки впровадженню таких методів, як поведінкова аналітика, глибоке навчання та нейронні мережі, сучасні моделі виявлення шахрайства тепер оцінюють численні точки даних у режимі реального часу. Ці системи на основі ШІ не лише оцінюють особу та наміри користувача, але й постійно адаптуються до нових та мінливих загроз. В результаті фінансові установи можуть досягти швидшого, точнішого та адаптивнішого виявлення шахрайства, значно підвищуючи як безпеку, так і операційну ефективність [3].

Розглянемо метод виявлення шахрайства з кредитними картками за допомогою перетворення зображень [4]

Автори розглядають зростаючу складність даних про транзакції з кредитними картками, що характеризуються високою швидкістю, серйозним дисбалансом класів та зміною моделей шахрайства, та пропонують нову структуру під назвою «Виявлення шахрайства за допомогою перетворення зображень» (FDIC) для покращення ефективності виявлення. У FDIC транзакції обробляються як часові ряди та перетворюються на двовимірні зображення за допомогою таких методів, як кутові поля Грама (GAF), поля переходів Маркова (MTF) та діаграми повторення (RP). Ці зображення потім класифікуються за допомогою згорткової нейронної мережі (CNN), тим самим фіксуючи часові та двосторонні зв'язки між характеристиками транзакцій, які може бути важко фіксувати за допомогою звичайних табличних або послідовних моделей. Для підвищення прозорості та інтерпретованості автори застосовують підхід штучного інтелекту (XAI) на основі теплових карт (через Grad-CAM та «сфокусований» варіант), щоб візуалізувати, які області перетворених зображень впливають на прийняття рішень моделлю.

Хоча запропонований підхід FDIC демонструє інновації у представленні даних про транзакції для виявлення шахрайства, певні обмеження та наслідки заслуговують на розгляд. Автори повідомляють про F1-бал 85,49% та повноту 80,35% для класу шахрайства, що демонструє конкурентоспроможну продуктивність порівняно з попередніми дослідженнями.

Мета: розробка та програмна реалізація методу виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж.

На етапі попередньої обробки даних ознака Amount масштабована за допомогою RobustScaler. Цей метод базується не на мінімальних та максимальних значеннях, а на медіані та інтерквартильному розмаху, що робить його стійким до

наявності викидів у даних – поширеного явища у фінансових транзакціях, де окремі операції можуть мати надзвичайно великі суми. Формула перетворення має вигляд:

$$x' = \frac{x - \text{median}(x)}{\text{IQR}(x)} \quad (1)$$

У результаті більшість значень суми зосереджуються поблизу нуля, тоді як екстремальні транзакції не мають надмірного впливу на навчання моделі. Такий підхід забезпечує стабільнішу збіжність нейронної мережі та підвищує її стійкість до нетипових значень у даних.

Ознаку Time було нормалізовано до інтервалу [-1,1] за допомогою лінійного перетворення:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2)$$

Таким чином, найменше значення часу отримує -1, найбільше +1, а всі проміжні значення рівномірно розподіляються між ними.

Після масштабування ознак було виконано балансування вибірки, оскільки у вихідних даних спостерігався суттєвий дисбаланс між кількістю шахрайських та не шахрайських транзакцій. У більшості реальних фінансових наборів даних кількість легітимних операцій значно перевищує кількість шахрайських, що може призвести до переважного навчання моделі на більший клас і, відповідно, погіршення здатності виявляти шахрайство.

Після тренування моделі було проведено оцінювання роботи моделі та отримано основні показники оцінювання роботи моделі (Риснок 1).

```
Epoch 1/5
3110/3110 ━━━━━━━━━━━━━━━━━━━━ 19s 6ms/step - accuracy: 0.9817 - loss: 0.0502 - precision: 0.9831 - re
call: 0.9802 - val_accuracy: 0.9957 - val_loss: 0.0144 - val_precision: 0.9916 - val_recall: 0.9998
Epoch 2/5
3110/3110 ━━━━━━━━━━━━━━━━━━━━ 18s 6ms/step - accuracy: 0.9960 - loss: 0.0145 - precision: 0.9944 - re
call: 0.9975 - val_accuracy: 0.9989 - val_loss: 0.0074 - val_precision: 0.9981 - val_recall: 0.9996
Epoch 3/5
3110/3110 ━━━━━━━━━━━━━━━━━━━━ 18s 6ms/step - accuracy: 0.9974 - loss: 0.0097 - precision: 0.9964 - re
call: 0.9985 - val_accuracy: 0.9979 - val_loss: 0.0077 - val_precision: 0.9958 - val_recall: 0.9999
Epoch 4/5
3110/3110 ━━━━━━━━━━━━━━━━━━━━ 18s 6ms/step - accuracy: 0.9982 - loss: 0.0071 - precision: 0.9974 - re
call: 0.9990 - val_accuracy: 0.9983 - val_loss: 0.0052 - val_precision: 0.9975 - val_recall: 0.9992
Epoch 5/5
3110/3110 ━━━━━━━━━━━━━━━━━━━━ 17s 6ms/step - accuracy: 0.9984 - loss: 0.0061 - precision: 0.9978 - re
call: 0.9991 - val_accuracy: 0.9985 - val_loss: 0.0072 - val_precision: 0.9975 - val_recall: 0.9996
2666/2666 ━━━━━━━━━━━━━━━━━━━━ 4s 1ms/step - accuracy: 0.9981 - loss: 0.0057 - precision: 0.9971 - rec
all: 0.9992

Model Evaluation:
Accuracy: 1.00
Precision: 1.00
Recall: 1.00
F1: 1.00
```

Рисунок 1 – Процес та результати тренування моделі

На основі отриманих результатів навчання розроблена згорткова нейронна мережа (ЗНМ) продемонструвала винятково високу продуктивність у завданні

виявлення шахрайських транзакцій. Протягом п'яти епох навчання точність як навчання, так і валідації постійно зростала, досягнувши приблизно 99,8–99,9% з дуже низьким значенням втрат (близько 0,005–0,007). Значення точності та повноти як на навчальному, так і на валідаційному наборах також були надзвичайно високими (близько 0,998–0,999), що свідчить про те, що модель успішно ідентифікує майже всі шахрайські транзакції, практично не даючи хибнопозитивних результатів.

Під час остаточної тестової оцінки модель досягла 100% точності, прецизійності, повноти та F1-балу, що підтверджує її хороше узагальнення та високу надійність у розрізненні шахрайських та легітимних операцій. Такі результати показують, що кроки попередньої обробки, обрана архітектура ЗНМ та стратегія оптимізації були ефективними. Однак, незважаючи на ці ідеальні показники, важливо зазначити, що така висока продуктивність може також свідчити про незначне перенавчання через синтетичне балансування та відносно просту структуру даних про транзакції. Тому для розгортання в реальних фінансових системах рекомендується додаткова валідація на реальних, невидимих потоках транзакцій, щоб забезпечити стійкість моделі та стабільну продуктивність у динамічних реальних умовах.

**Перелік посилань**

1. Credit Card Fraud Detection: How To Spot & Avoid Fraud URL: https://www.identityguard.com/news/credit-card-fraud-detection
2. credit card scams to watch out for in 2025 URL: https://lifelock.norton.com/learn/fraud/credit-card-scams
3. Financial Fraud Detection in the AI Era: Best Practices for Compliance and Risk Management – Avahi URL: https://avahi.ai/blog/financial-fraud-detection-in-the-ai-era/
4. Explainable Credit Card Fraud Detection with Image Conversion — d9ffa86ca415aeefcbed3703ed6d602598db.pdf URL: https://pdfs.semanticscholar.org/e437/d9ffa86ca415aeefcbed3703ed6d602598db.pdf

## Додаток Б

## Програмний код та посилання на GIT-репозиторій)

```python
import pandas as pd
import matplotlib
from sklearn.preprocessing import RobustScaler
from imblearn.over_sampling import SMOTE

def load(path="creditcard.csv"):
    data = pd.read_csv(path)

    return data

def preprocess(data):
    new_data = data.copy()
    new_data['Amount'] = RobustScaler().fit_transform(new_data['Amount'].to_numpy().reshape(-1, 1))
    time = new_data['Time']
    new_data['Time'] = (time - time.min()) / (time.max() - time.min())
    smote = SMOTE(random_state=4444)
    X = new_data.drop("Class", axis=1)
    y = new_data["Class"]

    X_res, y_res = smote.fit_resample(X, y)

    new_data = pd.DataFrame(X_res, columns=X.columns)
    new_data["Class"] = y_res

    return new_data
```

Рисунок Б.1 – Код модуля Dataset

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, MaxPooling1D, Flatten, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import Precision, Recall
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from tensorflow.keras.callbacks import EarlyStopping
import matplotlib.pyplot as plt
from tensorflow.keras.models import load_model
from decimal import Decimal

def build(input_shape = (30, 1)):
    model = Sequential([
        Conv1D(32, kernel_size=3, activation='relu', input_shape=input_shape),
        MaxPooling1D(pool_size=2),
        Conv1D(64, kernel_size=3, activation='relu'),
        MaxPooling1D(pool_size=2),
        Flatten(),
        Dense(64, activation='relu'),
        Dropout(0.3),
        Dense(1, activation='sigmoid')
    ])

    model.compile(
        optimizer=Adam(learning_rate=0.001),
        loss='binary_crossentropy',
        metrics=['accuracy', Precision(name="precision"), Recall(name="recall")]
    )

    return model
```

```python
def train(model, data):
    X = data.drop("Class", axis=1).values
    X = X.reshape((X.shape[0], X.shape[1], 1))
    y = data['Class'].values

    # Split the data
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.25, random_state=4444
    )

    # Early stopping
    early_stopping = EarlyStopping(
        monitor='val_loss',
        patience=5,
        restore_best_weights=True
    )

    # Train the model
    history = model.fit(
        X_train, y_train,
        epochs=1,
        batch_size=256,
        validation_data=(X_test, y_test),
        callbacks=[early_stopping],
        verbose=1
    )

    metric_plots = [
        ('accuracy', 'Accuracy'),
        ('precision', 'Precision'),
        ('recall', 'Recall')
    ]

    plot_index = 1
    for metric_key, metric_label in metric_plots:
        if metric_key in history.history:
            plt.subplot(1, 3, plot_index)
            plt.plot(history.history[metric_key], label=f'Train {metric_label}')
            if f'val_{metric_key}' in history.history:
                plt.plot(history.history[f'val_{metric_key}'], label=f'Val {metric_label}')
            plt.title(metric_label)
            plt.xlabel('Epoch')
            plt.ylabel('Value')
            plt.legend()
            plt.grid(True)
            plot_index += 1

    plt.show()

    return model, history
```

```python
def save(model, fileName = "CNN_model.h5"):
    model.save(fileName)

def load(fileName = "CNN_model.h5"):
    return load_model(fileName)
```

```python
def evaluate(model, data):
    X = data.drop("Class", axis=1).values
    X = X.reshape((X.shape[0], X.shape[1], 1))
    y = data['Class'].values

    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.25, random_state=4444
    )

    evaluation = model.evaluate(X_test, y_test)

    # Expecting metrics: loss, accuracy, precision, recall
    loss, accuracy, precision, recall = evaluation[:4]
    f1 = 2 * (precision * recall) / (precision + recall)
    print(f"\nModel Evaluation:\nAccuracy: {accuracy}\nPrecision: {precision}\nRecall: {recall}\nF1: {f1}")

    # Confusion matrix
    y_pred_prob = model.predict(X_test)
    y_pred = (y_pred_prob > 0.5).astype(int).flatten()

    cm = confusion_matrix(y_test, y_pred)
    disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['Non-Fraud', 'Fraud'])
    disp.plot(cmap='Blues', values_format='d')
    plt.title("Confusion Matrix")
    plt.grid(False)
    plt.show()

def predict(model, data, x):
    X = data.drop("Class", axis=1).values
    y = data['Class'].values

    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.25, random_state=4444
    )

    print(model.predict(X[x].reshape(1, X.shape[1], 1)))
    print(f"actual: {y[x]}")
```

Рисунок Б.2 – Код модуля Model

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from tensorflow.keras.models import load_model
import pandas as pd
from sklearn.preprocessing import RobustScaler
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split

import Dataload, Model

def CNN_metrics(model):

    data = Dataload.load()
    preprocessed_data = Dataload.preprocess(data)

    X = preprocessed_data.drop("Class", axis=1).values
    X = X.reshape((X.shape[0], X.shape[1], 1))
    y = preprocessed_data['Class'].values

    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.25, random_state=4444
    )
```

```
evaluation = model.evaluate(X_test, y_test)
loss, accuracy, precision, recall = evaluation[:4]
f1 = 2 * (precision * recall) / (precision + recall)

names = ["Accuracy", "Precision", "Recall", "F1-score"]
values = [accuracy, precision, recall, f1]

plt.figure(figsize=(6,4))
bars = plt.bar(names, values, color='skyblue')
plt.title("Created Method Evaluation Metrics")
plt.ylim(0, 1)
plt.grid(axis='y')

# Put values *inside* bars, vertically centered
for bar in bars:
    height = bar.get_height()
    plt.text(
        bar.get_x() + bar.get_width()/2.0,
        height / 2,
        f'{height:.4f}',
        ha='center',
        va='center',
        fontsize=10,
        color='black'
    )

plt.show()
```

```
def FDIC_metrics():

    fdic = {
        "Accuracy": 0.9995,
        "Precision": 0.9146,
        "Recall": 0.8035,
        "F1-score": 0.8549
    }

    plt.figure(figsize=(6,4))
    bars = plt.bar(fdic.keys(), fdic.values(), color='salmon')
    plt.title("FDIC Model Metrics")
    plt.ylim(0, 1)
    plt.grid(axis='y')

    # Values inside bars
    for bar in bars:
        height = bar.get_height()
        plt.text(
            bar.get_x() + bar.get_width()/2.0,
            height / 2,
            f'{height:.4f}',
            ha='center',
            va='center',
            fontsize=10,
            color='black'
        )

    plt.show()
```

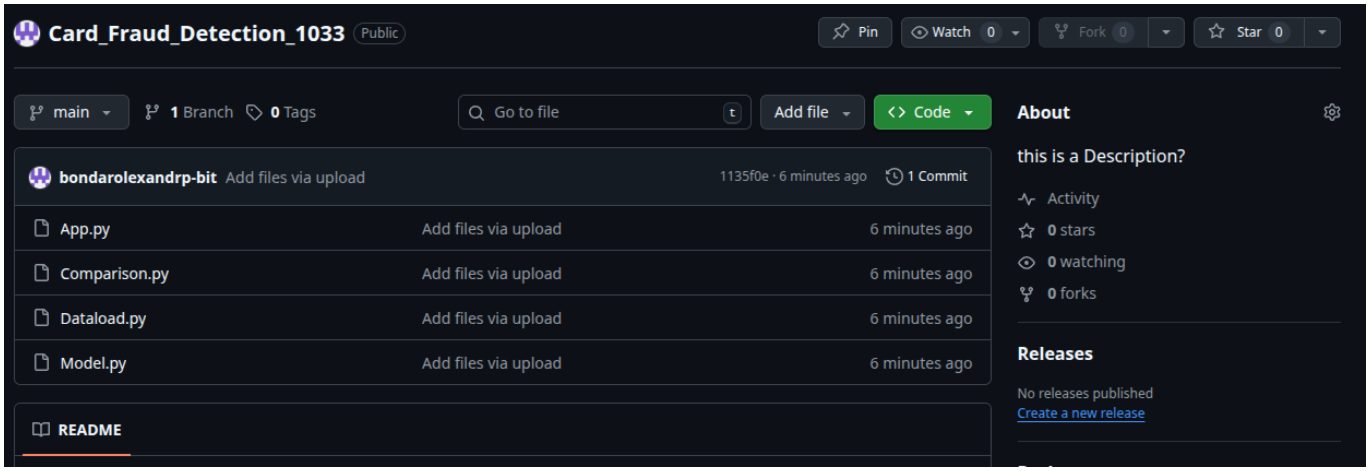
```python
def both_metrics(model):
    data = Dataload.load()
    preprocessed_data = Dataload.preprocess(data)

    X = preprocessed_data.drop("Class", axis=1).values
    X = X.reshape((X.shape[0], X.shape[1], 1))
    y = preprocessed_data['Class'].values

    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.25, random_state=4444
    )

    evaluation = model.evaluate(X_test, y_test)
    loss, accuracy, precision, recall = evaluation[:4]
    f1 = 2 * (precision * recall) / (precision + recall)

    cnn_vals = [accuracy, precision, recall, f1]
    fdic_vals = [0.9995, 0.9146, 0.8035, 0.8549]
    labels = ["Accuracy", "Precision", "Recall", "F1-score"]

    for i, label in enumerate(labels):
        plt.figure(figsize=(4,3))
        bars = plt.bar(
            ["Created Method", "FDIC"],
            [cnn_vals[i], fdic_vals[i]],
            color=['skyblue', 'salmon']
        )

        plt.ylim(0, 1)
        plt.title(label)
        plt.ylabel(label)
        plt.grid(axis='y')

        # Values inside bars
        for bar in bars:
            height = bar.get_height()
            plt.text(
                bar.get_x() + bar.get_width()/2.0,
                height / 2,
                f'{height:.4f}',
                ha='center',
                va='center',
                fontsize=10,
                color='black'
            )

        plt.show()
```

Рисунок Б.3 – Код модуля Comparison

```python
import Comparison, Model, Dataload

def main():
    data = Dataload.load
    preprocessed_data = Dataload.preprocess(data)

    model = Model.build()
    Model.train(model, preprocessed_data)
    Model.evaluate(model, preprocessed_data)
    Model.save(model, "CNN_model_2nd.h5")
    model = Model.load("CNN_model_2nd.h5")

    Comparison.CNN_metrics(model)
    Comparison.FDIC_metrics()
    Comparison.both_metrics(model)

if __name__ == "__main__":
    main()
~
```

Рисунок Б.4 – Код модуля App

Вихідний код, використаний у дослідженні, доступний у репозиторії GitHub: https://github.com/bondarolexandrp-bit/Card_Fraud_Detection_1033.



**Додаток В**

**Презентація**

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

# МЕТОД ВИЯВЛЕННЯ ШАХРАЙСЬКИХ ТРАНЗАКЦІЙ У ФІНАНСОВИХ ОПЕРАЦІЯХ ІЗ ВИКОРИСТАННЯМ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

**Виконав:**
Студент 2 курсу, групи КНм-24-1
Олександр БОНДАР
**Кирівник:**
к. т. н., доцент кафедри КН
Олександр ПАСІЧНИК

## АКТУАЛЬНІСТЬ ТЕМИ

Сучасний етап розвитку технологій значною мірою зумовлений стрімким прогресом у сфері цифрових технологій, інформаційних систем та програмних рішень. Вони відіграють вирішальну роль у повсякденному житті та у різних галузях, включаючи фінанси та електронну комерцію. У зв'язку з цим, виявлення виявлення шахрайських транзакцій у фінансових операціях є важливою задачею для підвищення безпеки платежів та запобігання економічним втратам.

У цьому контексті актуальним є застосування методів інтелектуального аналізу даних та штучного інтелекту для автоматизованого, високоточого та швидкого виявлення шахрайської діяльності. Згорткові нейронні мережі, поширені у задачах класифікації та розпізнавання патернів, демонструють високий потенціал у моделюванні складних залежностей у транзакційних даних, що робить їх доцільним інструментом для вирішення задачі детекції шахрайства.

## МЕТА РОБОТИ, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

**Мета і задачі роботи.** Метою кваліфікаційної роботи магістра є підвищення точності виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж.

**Об'єкт дослідження.** Процес виявлення виявлення шахрайських транзакцій у фінансових операціях.

**Предмет дослідження.** Нейромережеві методи та технології для виявлення виявлення шахрайських транзакцій у фінансових операціях.

## ЗАДАЧІ ДОСЛІДЖЕННЯ

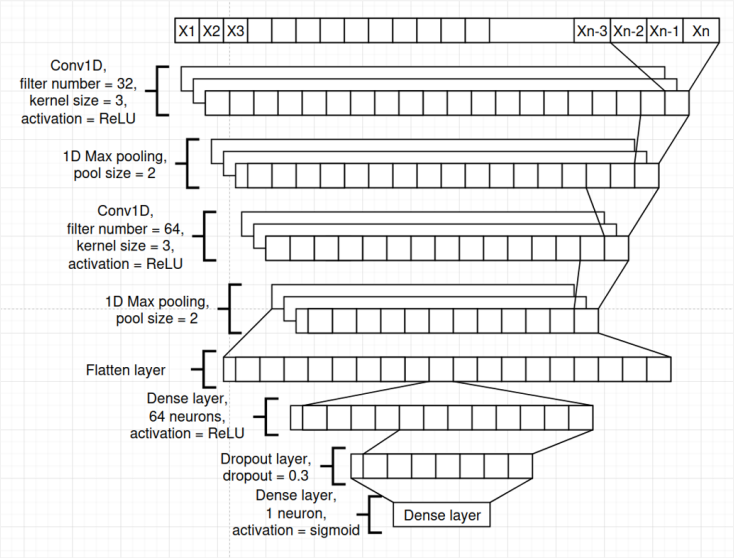
- провести аналіз методів виявлення шахрайських транзакцій у фінансових операціях;
- провести аналіз можливостей, переваг та недоліків згорткових нейронних мереж для виявлення шахрайських транзакцій у фінансових операціях;
- спроєктувати метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж;
- виконати програмну реалізацію методу;
- виконати дослідження точності виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж.

# НАУКОВА НОВИЗНА ОДЕРЖАНИХ РЕЗУЛЬТАТІВ

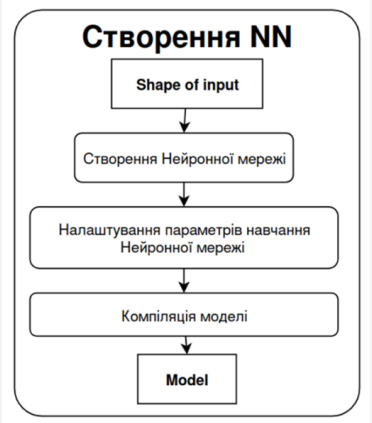
Удосконалено метод виявлення шахрайських транзакцій у фінансових операціях, який відрізняється від існуючих використанням згорткової нейронної мережі з попередньою нормалізацією та масштабуванням ключових класифікаційних параметрів та балансування даних, що дозволяє підвищити точність класифікації фінансових транзакцій.

# ЗАГАЛЬНА СХЕМА МЕТОДУ РОБОТИ ВИЯВЛЕННЯ ШАХРАЙСЬКИХ ТРАНЗАКЦІЙ

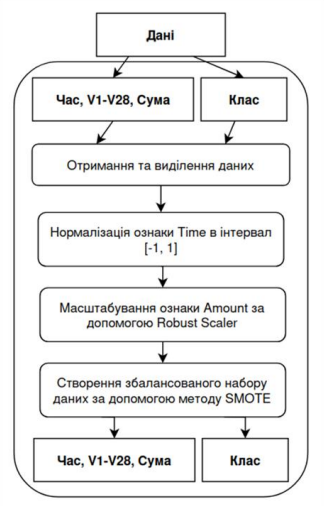
# АРХІТЕКТУРА НЕЙРОННОЇ МЕРЕЖІ



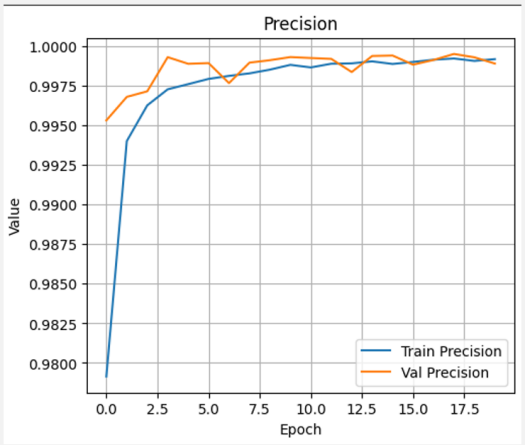

# ЕТАПИ НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ

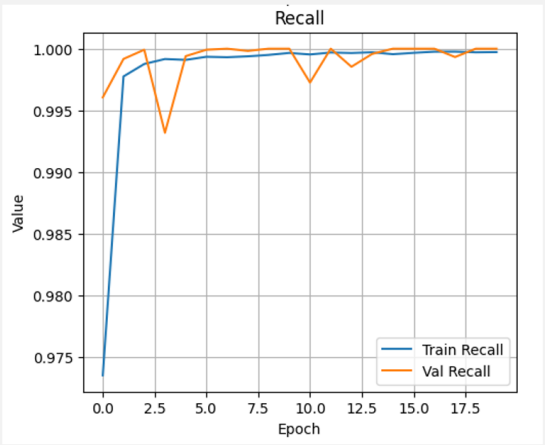




# ЕТАП ПОПЕРЕДНЬОЇ ОБРОБКИ ДАНИХ ВИКОРИСТАНИХ ПРИ НАВЧАННІ

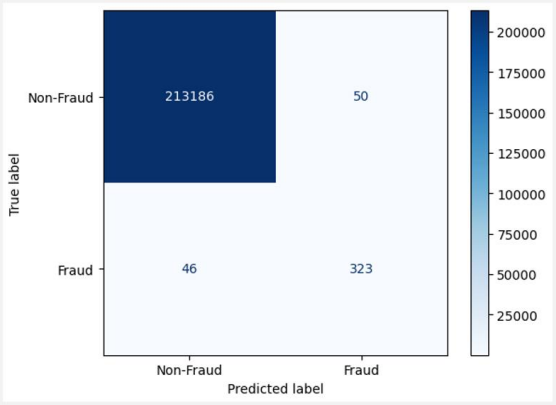



# ДИНАМІКА МЕТРИК PRECISION ТА RECALL ПРОТЯГОМ НАВЧАННЯ



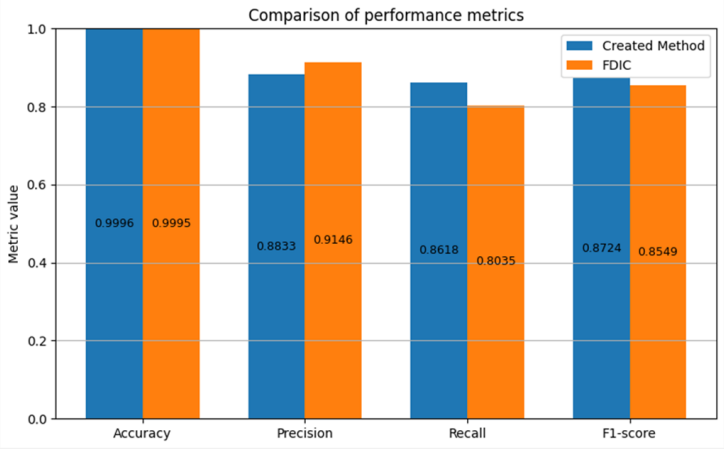



## МАТРИЦЯ ПОМИЛОК ОЦІНЕНА НА НАВЧАЛЬНІЙ ТА ВАЛІДАЦІЙНІЙ ПІДМНОЖИНАХ.





## ПОРІВНЯННЯ МЕТРИК ДЛЯ СТВОРЕНОГО МЕТОДУ ТА МЕТОДУ FDIC



## ВИСНОВКИ

В результаті виконання роботи було виконано такі задачі:

- проведено аналіз методів виявлення шахрайських транзакцій у фінансових операціях;
- проведено аналіз можливостей, переваг та недоліків згорткових нейронних мереж для виявлено шахрайських транзакцій у фінансових операціях;
- спроєктовано метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж;
- виконано програмну реалізацію методу;
- виконано дослідження точності виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж.

# ДЯКУЮ ЗА УВАГУ

Wed Dec 17 10:15:58 EET 2025, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

# Anti-Plagiarism (UA) v-15.284 Educational

**The maximum coincidence with one document 4.0%**

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 12%**

| | Document | | Sum coincidence on the DB | |
|---|---|---|---|---|
| ID: 253439<br>Title: КВАЛІФІКАЦІЙНА РОБОТА на тему Метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж<br>Added in a DB: 2025-12-17<br>Authors: Олександр БОНДАР<br>Heads: Олександр ПАСІЧНИК<br>Consultants:<br>Opponents: | Symbols | Lexemes | Symbols | Lexemes |
| | 103744 | 746 | 5996 (6%) | 62 (8%) |

Plagiarism sources

| ID | Description | Plagiarism presence in the document | |
|---|---|---|---|
| | | Symbols | Lexemes |

**Протокол аналізу звіту подібності науковим керівником**

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Олександр БОНДАР

**Співавтор:**

**Назва:** КВАЛІФІКАЦІЙНА РОБОТА на тему Метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж

**Науковий керівник:** Олександр ПАСІЧНИК, к.т.н., доцент

**Підрозділ:** Кафедра комп'ютерних наук

**Коефіцієнт подібності 1:**9%

**Коефіцієнт подібності 2:**3.5%

**Мікропробіли:** 0

**Заміна букв:** 3

**Інтервали:** 0

**Білі знаки:** 1

**Дата створення звіту:** 2025-12-17 09:59:02.0

**Після аналізу Звіту подібності констатую наступне:**

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення** (маніпуляції), як **передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2025-12-17

*Дата*

експерт Петровський С.С

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж

Автор: Бондар Олександр Петрович

Освітня програма: Комп'ютерні науки

Рівень вищої освіти: Другий (магістерський)

Спеціальність: 122 - Комп'ютерні науки

Науковий керівник: к.т.н., доцент Пасічник Олександр Анатолійович

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату ) з урахуванням результатів перевірки роботи спеціалізованими програмними засобами комісія зробила такий висновок:

| № | Висновок | Позначка про відповідність |
|---|---|---|
| 1 | Ознаки академічного плагіату | |
| 1.1 | Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту. | ***відповідає*** |
| 1.2 | Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються деталі та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. | |
| 1.3 | Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 1.4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |
| 2 | Інші види порушень академічної доброчесності | ***відсутні*** |

Підтвердження:

*Запозичення, виявлені в роботі Олександра БОНДАРЯ, не є плагіатом, оскільки запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти, які не мають авторства і містять поширені конструкції та загальновідомі терміни, скорочення. Рівень подібності не перевищує допустимої межі. Таким чином, робота є законною та приймається до захисту:*

*Обсяг запозичень, визначений системами виявлення збігів/ідентичності:*

*- за системою Anti-Plagiarism: 4,0 %;*

*- за системою StrikePlagiarism КП1:9,0%; КП2: 3,5 %.*

Завідувач кафедри КН — Олександр БАРМАК

Гарант освітньої програми — Руслан БАГРІЙ

Керівник кваліфікаційної роботи — Олександр ПАСІЧНИК

**ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**
**МОН УКРАЇНИ**
**кафедра комп'ютерних наук**

## ВІДГУК НАУКОВОГО КЕРІВНИКА

**на кваліфікаційну роботу магістра**

*гр. КНм-24-1 Бондаря Олександра* за темою: *Метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж*

**1. Актуальність теми**

*Значні економічні втрати та зниження рівня безпеки платіжних систем, спричинені шахрайськими транзакціями у фінансових операціях, зумовлюють актуальність завдання їх своєчасного виявлення. У зв'язку з цим постає об'єктивна необхідність упровадження ефективних рішень для протидії фінансовому шахрайству. Причиною формування таких підходів є сучасний етап розвитку технологій, що характеризується інтенсивним прогресом цифрових технологій, інформаційних систем і програмних засобів, які створюють підґрунтя для застосування результативних методів аналізу фінансових операцій.*

**2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт**

*Кваліфікаційна робота магістра КНм-24-1 Олександра Бондаря за ступенем обґрунтованості наукових положень, новизни, а також обсягом, структурою та змістом викладеного матеріалу відповідає вимогам щодо наукових робіт. У роботі використані методи, що повністю відповідає предметній області спеціальності 122 Комп'ютерні науки*

**3. Професійні та особистісні якості магістранта**

*В період виконання кваліфікаційної роботи магістра Олександр Бондар виявив себе достатньо кваліфікованим фахівцем здатним на задовільному рівні виконувати поставлені завдання. Володіє необхідними професійними навичками та загальними компетентостями.*

**4. Ступінь самостійності під час виконання кваліфікаційної роботи**

*Кваліфікаційна робота виконана студентом особисто. Визначення мети та постановка задач виконувалося спільно з науковим керівником.*

**5. Наукова новизна та оригінальність запропонованих підходів**

*Удосконалено метод виявлення шахрайських транзакцій у фінансових операціях, який відрізняється від існуючих застосуванням згорткової нейронної мережі з*

*попередньою нормалізацією та масштабуванням ключових класифікаційних параметрів та балансування даних, що дозволяє підвищити точність класифікації фінансових транзакцій. Результати роботи доповідалися на XVI Всеукраїнській науково-практичній конференції "Актуальні проблеми комп'ютерних наук (АПКН – 2025)", м. Хмельницький, ХНУ, 14-15 листопада 2024 р.*

**6. Ступінь оволодіння методами дослідження**

*Продемонстровано високий рівень володіння методами дослідження, які були використанні у роботі.*

**7. Повнота та якість розкриття теми роботи**

*Тема роботи розкрита якісно на високому рівні, задачі дослідження виконані в повному обсязі.*

**8. Логічність, послідовність, аргументованість, літературна грамотність викладу матеріалу**

*Позитивними рисами кваліфікаційної роботи є системність та послідовність викладення матеріалу. Продемонстрована здатність збирати і аналізувати дані, для забезпечення якості прийняття рішень. У кваліфікаційній роботі магістра формалізовані та систематизовані вимоги до розробленої комп'ютерної системи. Робота відповідає всім граматичним нормам та демонструє зрозумілий, виважений стиль подання інформації.*

**9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин**

*Результати роботи можуть бути використані у банківській сфері та у торгівлі виявлення шахрайських транзакцій у фінансових операціях.*

**10. Висновок про можливість допуску кваліфікаційної роботи до захисту, на яку оцінку заслуговує робота**

*Кваліфікаційна робота магістра Олександра Бондаря виконана повністю у відповідності із представленими вимогами та є завершеною науковою працею. Вона містить рішення наукової задачі, яка по суті полягає у реалізації методу виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж. З огляду на вище сказане, робота рекомендується до захисту та заслуговує на оцінку «добре».*

Науковий керівник _____ к.т.н., доц.. Олександр ПАСІЧНИК

**ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**МОН УКРАЇНИ**

**кафедра комп'ютерних наук**

# ВІДГУК ОПОНЕНТА

**на кваліфікаційну роботу магістра**

*гр. КНм-24-1 Бондаря Олександра* за темою*: Метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж*

**1. Актуальність обраної теми**

*Своєчасний світ визнає об'єктивну необхідність впровадження результативних рішень для подолання проблеми виявлення шахрайських транзакцій у фінансових операціях. Сучасний етап розвитку технологій значною мірою зумовлений стрімким прогресом у сфері цифрових технологій, інформаційних систем та програмних рішень. Вони відіграють вирішальну роль, зокрема, фінанси та електронна комерція. У зв'язку з цим, виявлення шахрайських транзакцій у фінансових операціях є важливою задачею для підвищення безпеки платежів та запобігання економічним втратам.*

**2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт**

*Тема кваліфікаційної роботи у повній мірі відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи магістра згідно Стандарту освіти.*

**3. Повнота розкриття мети та завдань дослідження**

*В роботі проведено аналіз методів виявлення шахрайських транзакцій у фінансових операціях, можливостей, переваг та недоліків згорткових нейронних мереж для виявлення шахрайських транзакцій у фінансових операціях; спроєктовано метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж; виконано програмну реалізацію методу виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж;виконано дослідження точності виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж.*

**4. Наявність наукової новизни**

*Удосконалено метод виявлення шахрайських транзакцій у фінансових операціях, який відрізняється від існуючих застосуванням згорткової нейронної мережі з попередньою нормалізацією та масштабуванням ключових класифікаційних параметрів та балансування даних, що дозволяє підвищити точність класифікації фінансових транзакцій. Результати роботи доповідалися на XVI Всеукраїнській науково-практичній конференції "Актуальні проблеми комп'ютерних наук (АПКН – 2025)", м. Хмельницький, ХНУ, 14-15 листопада 2025 р.*

**5. Зміст кожного розділу роботи**

*В розділі 1 проведено аналіз та огляд виявлення шахрайських транзакцій, наведено характеристику задачі та проаналізовано існуючі публікації та наукові підходи її вирішення,*

*визначено мету та задачі дослідження. В розділі 2 спроектовано метод шахрайських транзакцій у фінансоих операціях із використанням згорткових нейронних мереж та критерії його оцінювання. В розділі 3 реалізовано програмну систему спроєктованого методу. В розділі 4 виконано експериментальну перевірку методу виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж.*

**6. Ступінь розкриття теми роботи**

*Кваліфікаційна робота магістра присвячена виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж. В рамках роботи було проаналізовано предметну область, програмні системи для вирішення аналогічних завдань, створено відповідний метод та його програмна реалізація із проведенням необхідного тестування та досліджень. Тема роботи розкрита якісно на високому рівні, задачі дослідження виконані в повному обсязі.*

**7. Якість оформлення кваліфікаційної роботи**

*Кваліфікаційна робота магістра відповідає всім вимогам до оформлення таких робіт. Стиль подання інформації є фаховим та зрозумілим. Робота не містить стилістичних відхилень та відповідає всім нормам граматики. Робота виконана логічно, послідовно та аргументовано. Матеріал викладено якісно із дотриманням вимог до професійного літературного стилю.*

**8. Недоліки кваліфікаційної роботи**

*В роботі відсутні дані щодо можливого масштабування отриманого рішення на інші сфери та напрямки життєдіяльності людини.*

**9. Загальний висновок (допускається чи не допускається до захисту), якої оцінки заслуговує кваліфікаційна робота.**

*Беручи до уваги новизну, актуальність, важливість отриманих результатів, їх достовірність та обґрунтованість, вважаю, що кваліфікаційна робота магістра Олександра Бондаря «Метод виявлення шахрайських транзакцій у фінансових операціях із використанням згорткових нейронних мереж» є оригінальним та завершеним науковим дослідженням. Кваліфікаційна робота магістра Олександра Бондаря рекомендується до захисту, рекомендована оцінка «добре».*

Опонент _____ Лисенко С.М.