Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: Система виявлення шахрайських банківських базових операцій на основі машинного навчання

Виконав: студент (ка) IV курсу, групи СБ-42

Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)
Проць Петро Олегович
підпис (прізвище та ініціали)

Керівник Стадник М.А.
підпис (прізвище та ініціали)
Нормоконтроль Дроздова Т.В.
підпис (прізвище та ініціали)

Завідувач кафедри Загородна Н.В.
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Процю Петру Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи Система виявлення шахрайських банківських базових операцій на основі машинного навчання

Керівник роботи Стадник Марія Андріївна, к. т. н..,
доцент кафедри КБ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 02 » 05 2025 року № 4/7-361

2. Термін подання студентом завершеної роботи 27.06.2025

3. Вихідні дані до роботи Дані про транзакції, що містять банківське шахрайство

4. Зміст роботи (перелік питань, які потрібно розробити)
Визначити теоретичні основи виявлення шахрайських транзакцій.
Проаналізувати та порівняти існуючі методи виявлення шахрайства
Розробити систему виявлення шахрайських операцій методами машинного навчання
Безпека життєдіяльності, основи охорони праці
Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Титулка. Тема, мета, задачі. Шахрайство у банківських операціях. Методи виявлення шахрайства у фінансових транзакціях. Проблеми та виклики при розпізнаванні шахрайства за допомогою ML. Методи машинного навчання. Навчання «із вчителем». Навчання «без вчителя». Методи глибинного навчання. Архітектура Системи. Набір даних. Підготовка даних. Очищення даних. Інженерія ознак. Нормалізація даних. Розбиття даних та збереження ознак. Модуль виявлення аномалій. Архітектура нейронної мережі. Результати роботи
Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Мариненко С.Ю., к. т. н., доц. кафедри МТ		

7. Дата видачі завдання 29.01.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	29.01 – 30.01	Виконано
2.	Підбір джерел для аналізу brute-force атаки та їх наслідки	31.01 – 05.02	Виконано
3.	Опрацювання джерел в галузі дослідження	06.02 – 20.02	Виконано
4.	Опрацювання теоретичних основ виявлення шахрайства у банківських операціях	22.02 – 12.03	Виконано
5.	Аналіз та порівняння наявних методів виявлення шахрайства	15.03 - 25.03	Виконано
6.	Розробка системи виявлення шахрайських банківських операцій	25.02 – 10.04	Виконано
7.	Оформлення розділу «Теоретичні Основи Виявлення Шахрайських Транзакцій У Банківських Системах»	10.02 – 05.03	Виконано
8.	Оформлення розділу «Аналіз Та Порівняння Існуючих Підходів»	26.03 – 11.04	Виконано
9.	Оформлення розділу «Розробка Системи Виявлення Шахрайських Банківських Операцій Засобами Python Та Tensorflow»	12.04 - 20.05	Виконано
10.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	26.05 – 01.06	Виконано
11.	Оформлення кваліфікаційної роботи	02.06 – 07.06	Виконано
12.	Нормоконтроль	17.06 – 18.06	Виконано
13.	Перевірка на плагіат	17.06 – 18.06	Виконано
14.	Попередній захист кваліфікаційної роботи	16.06 – 17.06	Виконано
15.	Захист кваліфікаційної роботи	27.06.2025	

Студент _____ Проць П.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Стадник М.А.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Система виявлення шахрайських банківських базових операцій на основі машинного навчання // Кваліфікаційна робота ОР «Бакалавр» // Проць Петро Олегович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБ-42 // Тернопіль, 2025 // С. 65, рис. – 21, табл. – - , кресл. – - , додат. – 4 .

Ключові слова: машинне навчання, виявлення шахрайства, виявлення аномалій, нейронна мережа.

У кваліфікаційній роботі розглянуто систему виявлення шахрайських банківських операцій з використанням методів машинного навчання. Проведено детальний аналіз існуючих підходів, зокрема, методів навчання із вчителем та без вчителя, а також глибинних методів машинного навчання. На основі цього аналізу розроблено систему, що інтегрує алгоритми аномального виявлення (Isolation Forest) та глибинного навчання (нейронні мережі з використанням TensorFlow). Здійснено підготовку фінансових даних про транзакції, визначено оптимальні параметри моделі та проведено тестування на практичних прикладах. Запропонована система демонструє високу ефективність виявлення шахрайських транзакцій, що підтверджується отриманими показниками точності, повноти та AUC. Результати роботи можуть бути використані для впровадження у банківських установах для підвищення рівня безпеки транзакцій та зменшення фінансових втрат від шахрайських дій.

ABSTRACT

Fraudulent Banking Transaction Detection System Based on Machine Learning // Thesis of educational level "Bachelor"// Prots Petro // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, group СБ-42 // Ternopil, 2025 // P. 65 fig. – 21, tab. – _–_, drawing – _–_, added. – 4.

Keywords: machine learning, fraud detection, anomaly detection, neural network.

The thesis focuses on developing a machine learning-based system for detecting fraudulent banking operations. A comprehensive analysis of existing approaches was conducted, including supervised, unsupervised, and deep learning techniques. Based on this analysis, a model integrating anomaly detection algorithms (Isolation Forest) and deep learning (neural networks using TensorFlow) was developed. Financial transaction data preparation was performed, optimal model parameters were determined, and practical testing was conducted. The proposed system demonstrated high efficiency in identifying fraudulent transactions, as evidenced by obtained accuracy, recall, and AUC metrics. The results can be implemented in banking institutions to enhance transaction security and reduce financial losses due to fraud.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ML	—	Machine Learningl
AI	—	Artificial Intelligence
DL	—	Deep Learning
LR	—	Logistic Regression
SVG	—	Support Vector Machine
DNN	—	Deep Neural Network
XAI	—	Explainable artificial intelligence
MLP	—	Multilayer perceptron
RNN	—	Recurrent Neural Network
GNN	—	Graph neural network
GAN	—	Generative adversarial network

ВСТУП

Актуальність теми. Основним ризиком для банківської галузі є шахрайство, яке підриває довіру споживачів до фінансових установ і призводить до грошових втрат. За оцінками платіжного сервісу Stripe, шахрайство коштує бізнесу в усьому світі понад 20 мільярдів доларів США. Кіберзлочинці щодня зловживають банківськими операціями, як от зняттям готівки в банкоматах, онлайн-переказами та відкриттям рахунків. Через збільшення обсягів транзакцій, зростаючу інформаційну обізнаність злочинців та наростаючу складність виконання шахрайства традиційні методи виявлення шахрайства (стандартизовані правила, ручний моніторинг) вже не є достатньо ефективними. Як наслідок, вкрай важливо впроваджувати системи виявлення шахрайських банківських операцій на основі машинного навчання, які можуть миттєво ідентифікувати сумнівні транзакції.

Мета і задачі дослідження. Мета роботи – розробити систему виявлення шахрайських банківських операцій із використанням технологій машинного навчання засобами Python та TensorFlow.

Для досягнення мети роботи необхідно виконати наступні завдання:

- Провести аналіз природи банківського шахрайства та існуючих методів його розпізнавання.
- Розглянути сучасні підходи з використанням різних методів машинного навчання.
- Обрати найбільш ефективний підхід в контексті розпізнавання шахрайських операцій.
- Реалізувати практичну систему з використанням Python та бібліотеки TensorFlow.

Об’єкт дослідження. Методи виявлення шахрайських банківських операцій

Предмет дослідження. Написання системи виявлення шахрайських банківських операцій вбудованими методами Python та TensorFlow

Практичне значення одержаних результатів. Практична цінність результатів базується на можливості застосування результату дослідження як частини комплексної системи захисту банківських додатків від несанкціонованого доступу та зміни інформації

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ВИЯВЛЕННЯ ШАХРАЙСЬКИХ ТРАНЗАКЦІЙ У БАНКІВСЬКИХ СИСТЕМАХ

1.1 Шахрайство у банківських операціях

Шахрайство у банківській сфері становить серйозну загрозу втрат для бізнесу та клієнтів банків [1]. Банківське шахрайство у маніпулюванні банківськими операціями включає в себе широкий спектр незаконних дій з метою отримання фінансів чи інформації. Згідно онлайн-статті [2] фінансове шахрайство класифікується на наступні категорії (див. рисунок 1.1.): шахрайство з кредитними картками (credit card fraud), партнерське шахрайство (affiliate fraud), шахрайство з поверненням платежів (chargeback fraud), шахрайство з обліковими записами (account fraud), тестування карток (prepaid card fraud), шахрайство з передплаченими картками [2].

Рисунок 1.1 – Типи шахрайств

У банківських послугах впроваджений захист від різних типів фішингу, крадіжки особистих даних, шахрайства з платіжними картами,

несанкціонованими переказами і багато іншого [3]. Наприклад, шахраї можуть зловживати вкраденими даними карток у магазині, що може не бути виявлено у момент проведення транзакції [1]. За даними PwC, у 2022 році 56% компаній у світі стали жертвами фінансового шахрайства [4], а згідно з опитуванням KPMG, 71% фінансових компаній були жертвами дій, які вважаються внутрішнім чи зовнішнім шахрайством протягом останнього року [1]. Ці статистичні дані свідчать про необхідність застосування комбінацій анти шахрайських інструментів. Банківські транзакції реалізуються в різних формах, що охоплює готівкові операції, перекази між рахунками, платежі за товари, відкриття депозитів чи кредитів тощо та величезну кількість інших. Шахрайські дії сприяють не прямим фінансовим втратам, а й призводять до подальших репутаційних збитків банку щодо клієнтів. Крім цього, регулятори у всіх країнах практично аналогічно вимагають від банків встановлення жорсткої політики щодо ухилення від сплати податків, та ведення діяльності проти інших злочинів управління рахунками. Цим підвищується цікавість фінансових установ до технологічних інновацій у сфері виявлення шахрайства, оскільки кожна успішна атака підтверджує неефективність розробленої стратегії фінтех-підприємства.

1.2 Основи машинного навчання та його застосування в протидії шахрайству

Машинне навчання (ML) – це підгалузь штучного інтелекту, що дозволяє комп’ютерним системам навчатися на основі даних з метою прогнозування або прийняття рішень без явного програмування на кожен випадок. У сфері кібербезпеки, зокрема в банківській безпеці, ML використовується для автоматичного виявлення аномалій, шаблонів шахрайства, шкідливої активності в мережі тощо. AI-системи (Artificial Intelligence) для виявлення шахрайства аналізують великі обсяги транзакційних даних і навчаються розрізняти підозрілі дії та легітимні операції [3]. На основі історичних даних про відомі шахрайські та нормальні транзакції моделі ML можуть виявляти приховані закономірності,

недоступні для ручного аналізу, і таким чином своєчасно сигналізувати про можливе шахрайство. Особливістю ML-моделей є здатність адаптуватися до нових загроз. На відміну від жорстких правил, що потрібно постійно оновлювати, алгоритми машинного навчання поступово підлаштовуються під зміну поведінки зловмисників, «навчаючись» на нових даних [5]. Зокрема, у банківській сфері ML дозволяє виявляти нові схеми шахрайства, які раніше не спостерігалися, шляхом аналізу аномальних відхилень у поведінці користувачів та транзакційних потоках. Важливо, що сучасні ML-системи можуть працювати в реальному часі, обробляючи постійні потоки платежів і миттєво блокуючи підозрілі операції або поміщаючи їх на додаткову перевірку [5]. Це дає банкам змогу попередити фінансовий злочин до того, як він призведе до втрат. У кібербезпеці машинне навчання застосовується не лише для фінансових транзакцій, а й для виявлення атак на інформаційні системи, шкідливого програмного забезпечення, аномальної мережевої активності тощо. Всі ці задачі об’єднує необхідність розподілу великої кількості подій на «нормальні» та «потенційно шкідливі», що є типовою задачею на «класифікацію», в якій машинне навчання чудово показує себе порівняно з іншими алгоритмами. Алгоритми ML, такі як нейронні мережі, дерева рішень, метод опорних векторів, істотно підвищують ефективність кіберзахисту завдяки автоматизації і самонавчання на даних про інциденти. Узагальнену схему роботи ML-алгоритмів для виявлення банківського шахрайства приведено на рисунку 1.2.

Рисунок 1.2 – Схема роботи ML для виявлення шахрайства

Звісно, ML-рішення не є досконалими, вони можуть давати хибні спрацьовування або пропускати загрози, однак загалом дозволяють значно покращити виявлення злочинних дій у порівнянні з традиційними підходами. У банківському секторі спостерігається активна інтеграція AI/ML у внутрішні процеси і клієнтські сервіси з метою підвищення безпеки та якості обслуговування [3]. Банки розглядають ML як необхідний інструмент, що допомагає не лише виявляти шахрайство, але й загалом покращувати управління ризиками і відповідність регулятивним вимогам.

1.3 Методи виявлення шахрайства у фінансових транзакціях

Існує кілька підходів до виявлення шахрайських транзакцій, які еволюціонували з часом. Традиційні методи ґрунтувалися на наборах правил та порогових значень, визначених експертами. Наприклад, система могла блокувати або позначати транзакцію, якщо сума перевищує певний ліміт або якщо операція надходить з незвичної геолокації. Системи засновані на правилах були відносно простими й зрозумілими, однак мали обмежену гнучкість: шахраї навчилися обходити відомі правила, проводячи операції дещо нижче порогів, або маскуючи свою активність [6]. Крім того, зі збільшенням обсягів даних ручне налаштування правил стало громіздким та неефективним, створюючи велику кількість хибних спрацьовувань.

Більш передовий підхід – виявлення аномалій у даних. Такі методи намагаються побудувати модель «нормальної» поведінки клієнтів і транзакцій, а потім виявляють значні відхилення від цієї норми [6]. Аномальними можуть бути операції, що нетипові для певного користувача (наприклад, раптові великі покупки в іншій країні) або ж ті, що «не вписуються» в загальні статистичні закономірності. Методи виявлення аномалій можуть використовувати як прості статистичні показники, так і складні алгоритми (наприклад, кластеризацію, деревоподібні алгоритми). Перевагою цих методів є здатність виявляти нові,

невідомі раніше шаблони шахрайства. Проте недоліком може бути підвищена кількість хибних тривог, адже не кожна аномалія є шахрайством (незвичні, але законні дії клієнта теж можуть відхилятися від норми).

На сьогодні найбільш результативними вважаються методи з використанням машинного навчання. На відміну від статичних правил, ML-алгоритми будують моделі на основі історичних даних, що містять приклади шахрайських і легітимних транзакцій. Навчання «зі вчителем» (supervised learning) дозволяє навчити модель розпізнавати відомі типи шахрайства за наявності розмічених даних (де кожна транзакція наперед позначена як “шахрайська” або “нормальна”). Навчання «без вчителя» (unsupervised) застосовується, коли немає міток даних, тобто модель сама шукає схожі групи і аномалії, що можуть вказувати на шахрайство. Окремо виділяють глибинне навчання (deep learning) – це використання багатошарових нейронних мереж, які можуть автоматично виділяти складні закономірності із наборів даних. Детальніше представлення методів виявлення шахрайських операцій у вигляді дерева наведено на рисунку 1.3.

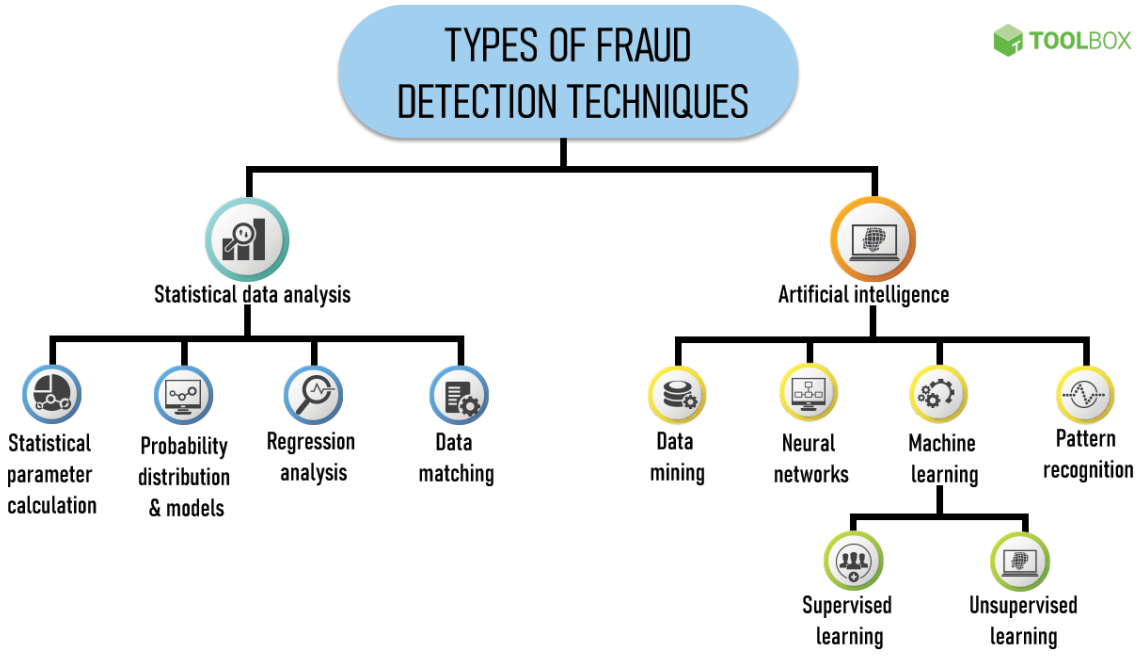


Рисунок 1.3 – Типи методів виявлення шахрайських операцій

На рисунку 1.3 представлена структурована схема різних методів виявлення шахрайства, що розподіляються на два основні типи: статистичний аналіз даних (statistical data analysis) та штучний інтелект (artificial intelligence). *Статистичний аналіз даних* включає: розрахунок статистичних параметрів (statistical parameter calculation), імовірнісні розподіли та моделі (probability distribution & models), регресійний аналіз (regression analysis), зіставлення даних (data matching). *Штучний інтелект* включає такі методи: інтелектуальний аналіз даних (data mining), нейронні мережі (neural networks), машинне навчання (machine learning), яке ділиться на навчання «зі вчителем» та «без вчителя», розпізнавання шаблонів (pattern recognition).

Сучасні системи виявлення шахрайства часто комбінують кілька підходів. Наприклад, може спочатку спрацьовувати просте правило для грубого відсіву підозрілих випадків, далі алгоритм «без учителя» оцінює, наскільки транзакція нетипова, а потім ML-модель «із вчителем» дає фінальний прогноз щодо ймовірності шахрайства. Використання гібридних рішень дозволяє знизити навантаження та підвищити точність: правила швидко відсіюють очевидні випадки, а ML заглиблюється в аналіз решти транзакцій.

1.4 Особливості обробки та аналізу фінансових даних

Дані фінансових транзакцій мають низку характерних особливостей, що впливають на побудову систем виявлення шахрайства. Перш за все, це великий обсяг і висока швидкість надходження даних: банки обробляють тисячі транзакцій за секунду, особливо у періоди пікової активності. Система повинна бути здатною масштабуватися і обробляти потоки даних практично без затримок (on-line).

По-друге, транзакційні дані зазвичай незбалансовані: частка шахрайських операцій дуже мала у порівнянні з легітимними (часто менше 0,1%). Цей дисбаланс ускладнює навчання моделей, оскільки алгоритм, який просто класифікує все як «не шахрайство», матиме високий відсоток загальної точності,

але не виконуватиме своєї функції. Для боротьби з цим застосовують методи балансування – наприклад можна застосувати надсемплинг шахрайського класу або генерацію синтетичних прикладів (SMOTE, тощо) [7].

Підготовка фінансових даних включає очищення і перетворення різноманітної інформації, що надходить із систем банку. Транзакції містять числові характеристики (сума, час, баланс рахунку), категорійні ознаки (тип операції, канал виконання – банкомат, онлайн-банкінг, POS-термінал; код торговця; країна тощо), а також можливо текстові поля (призначення платежу). На етапі *попередньої обробки* дані потрібно відфільтрувати від хибних або неповних записів, заповнити або видалити пропущені значення, трансформувати категорійні поля у числовий формат (наприклад, шляхом кодування категорій) [7]. Фінансові дані можуть містити аномальні викиди (outliers), наприклад, дуже великі суми, що за потреби нормалізують (масштабують) до єдиного діапазону, щоб запобігти домінуванню цих значень при навчанні моделі [8]

Важливим кроком є обробка ознак (feature engineering) – створення інформативних та значущих показників із наявних «сирих» даних. Для ідентифікації шахрайства корисними можуть бути такі ознаки, як частота операцій клієнта за певний період, середній інтервал часу між транзакціями, географічна відстань між точками здійснення послідовних платежів, невідповідність місця операції типу картки (наприклад, транзакція за кордоном для картки, що раніше використовувалась лише в місцевому регіоні) тощо [7]. Аналізуючи історичні дані, можна виявити характеристики, які найбільш сильно відрізняють шахрайські транзакції від легальних, і включити їх як додаткові поля для моделі. Також враховуються зв’язки між операціями: наприклад, чи пов’язані кілька підозрілих транзакцій з одним отримувачем або IP-адресою. Покрокову схему процесу попередньої обробки даних приведено на рисунку 1.4.

Фінансові дані потребують особливої уваги до конфіденційності та безпеки при обробці. У реальних системах може бути необхідним анонімізувати персональні дані (імена, номери рахунків) перед передачею їх у модель. Крім того, великі банки часто мають розподілені дані – транзакції зберігаються в

різних системах, що вимагає інтеграції джерел інформації. Дані можуть надходити із внутрішніх баз банку, а також з зовнішніх джерел (чорні списки шахрайських карток, дані про пристрої тощо) для покращення повноти ознак.

Особливості обробки та аналізу фінансових даних

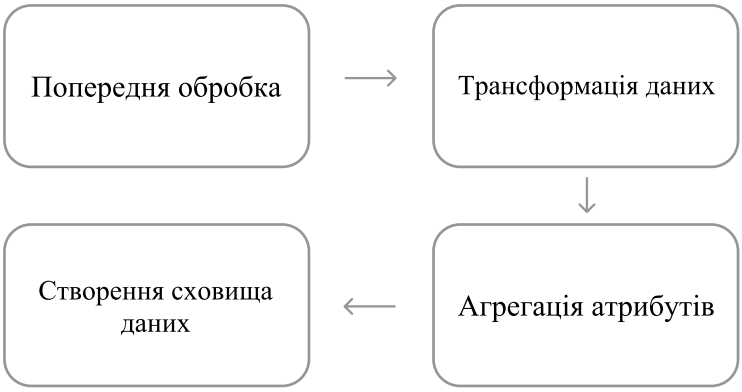


Рисунок 1.4. – Схема попередньої обробки даних

Нарешті, під час аналізу фінансових даних варто враховувати явище еволюції патернів. Те, що було ознакою шахрайства кілька років тому, сьогодні може втратити актуальність, адже шахраї постійно змінюють тактику . Тому дані для навчання мають бути актуальними. Застарілі датасети (наприклад, 5–10-річної давності) можуть містити вже неактуальні сценарії шахрайства. Часто застосовують ковзні вікна даних або періодично донавчають модель на нових даних, щоб вона не втрачала ефективність через “старіння” ознак шахрайства.

1.5 Проблеми та виклики при розпізнаванні шахрайства за допомогою ML

Незважаючи на успіхи машинного навчання, впровадження його у банківське виявлення шахрайства пов’язано з низкою викликів. Один з головних викликів – це якість та розміченість даних. Для алгоритмів «із вчителем» потрібні великі обсяги правильно розмічених транзакцій (де точно відомо, які були шахрайськими). Отримати такі дані складно: необхідна експертиза

аналітиків, судові підтвердження випадків шахрайства тощо. Експертиза потребує часу і ресурсів, тому розмічені набори даних зазвичай обмежені за обсягом. Шахрайство є рідкісною подією, і класи сильно незбалансовані. Це призводить до того, що методи виявлення аномалій і пов'язане глибинне навчання у цій сфері застосовуються відносно рідко – лише ~18% та ~3% дослідників відповідно використовують їх, тоді як близько 57% покладаються на моделі з учителем [1]. Глибинні нейронні мережі вимагають дуже великих обсягів даних для навчання і значної обчислювальної потужності, тому не кожна фінансова установа готова їх застосувати на практиці. До того ж, складність архітектури DNN (deep neural network) ускладнює їх налаштування – результат залежить від вдалої конструкції мережі та вибору гіперпараметрів (кількість шарів, нейронів, швидкість навчання тощо).

Іншою проблемою є помилкові спрацювання та втручання в обслуговування клієнтів. Система, що занадто чутлива, може генерувати велику кількість false positives – тобто позначати цілком законні транзакції як підозрілі [3]. Це негативно впливає на досвід клієнтів: наприклад, якщо картка постійно блокується при спробі оплатити покупки, клієнт буде незадоволений. Тому необхідно шукати баланс між чутливістю і специфічністю моделі. Зменшення «false positives» часто досягається ціною певного збільшення ризику пропустити шахрайство (хибно-негативні помилки), тому банки визначають політику порогу спрацювання, виходячи зі своєї толерантності до ризику [1]. Оптимізація цього балансу – нетривіальне завдання, яке потребує тісної співпраці фахівців з даних та бізнес-аналітиків.

Важливим викликом при опрацюванні фінансових даних є здатність протидіяти новим видам шахрайства (адаптивність моделей). Шахраї постійно вигадують нові методи: наприклад, атаки з використанням соціальної інженерії, коли клієнта обманом змушують здійснити переказ на рахунок злодія, або складні багатокрокові схеми відмивання грошей. ML-модель, навчена на історичних шаблонах, може не одразу розпізнати нову схему. Це частково вирішується впровадженням алгоритмів «без учителя» для виявлення аномалій

(вони допоможуть виявити невідому раніше активність) та постійним донавчанням моделей на нових даних. Деякі дослідники пропонують також використовувати напівконтрольовані підходи (semi-supervised), коли модель спершу вчиться на обмеженому розміченому наборі, а потім донавчається на невідомих даних, використовуючи власні прогнози. Такі підходи дозволяють врахувати як мітки експертів, так і великий масив необроблених транзакцій .

Ще один аспект – інтеграція ML-системи у банківську інфраструктуру. Часто банк має розгалужену IT-систему, де транзакції обробляються в режимі реального часу на основних рахункових системах. Підключення до цього потоку ML-модуля для оцінки ризику повинно відбуватися з мінімальною затримкою і безпечно. Питання сумісності та масштабованості тут ключові: алгоритми мають масштабуватися на мільйони транзакцій на добу, а платформа повинна легко бути інтегрованою та працювати із різними архітектурами коду, не створюючи додаткових незручностей при впровадженні та підтримці. Крім того, враховується конфіденційність: моделі працюють з чутливими персональними та фінансовими даними, тому важливо дотримуватися вимог стандартів безпеки (шифрування даних, контроль доступу) і законодавства про захист даних. Використання клієнтських даних для навчання має відповідати регуляторним вимогам (наприклад, GDPR в ЄС) – іноді потрібна деперсоналізація даних або отримання згоди користувачів.

До того ж, прозорість та пояснюваність алгоритмів стають дедалі більш значущими. Фінансові установи та регулятори хочуть розуміти, чому система відхилила ту чи іншу транзакцію. Але складні моделі (як-от глибинні нейронні мережі чи ансамблі дерев з сотнями параметрів) діють як «чорні скриньки». Це породжує проблему довіри: співробітники служби безпеки можуть вагатися, покладатися на рішення моделі, якщо не можуть його пояснити клієнту чи керівництву. Тому сучасні дослідження приділяють увагу методам пояснюваного AI (XAI) – наприклад, візуалізація важливості ознак, генерація інтерпретацій моделей – щоб надати обґрунтування для кожного спрацювання системи . Підвищення прозорості не лише допомагає валідувати модель, а й

виявити можливі упередження (bias) або помилки в її роботі. В результаті основні виклики розпізнавання банківського шахрайства можна представити наступною схемою (див. рисунок 1.5):

Рисунок 1.5 – Виклики розпізнавання банківського шахрайства

Таким чином, для успішного застосування ML у виявленні шахрайства необхідно подолати ряд викликів: зібрати та підготувати якісні дані, обрати правильну модель і налаштувати її для балансування помилок, інтегрувати рішення у інфраструктуру банку та забезпечити його надійну і безпечну роботу. У роботі основний фокус буде зосереджено саме на виборі та налаштуванні моделі, а також частково розглянуто процес попередньої обробки вхідних даних для використання їх у моделі.

РОЗДІЛ 2 АНАЛІЗ ТА ПОРІВНЯННЯ ІСНУЮЧИХ ПІДХОДІВ

2.1 Огляд застосування ML у задачах виявлення шахрайства

Методи машинного навчання широко досліджуються та впроваджуються для протидії фінансовому шахрайству. Аналіз наукових публікацій показує, що більшість дослідників зосереджуються на підходах із вчителем: зокрема, у 56,7% досліджень використовується подібне навчання для виявлення фінансового шахрайства. Це пояснюється високою ефективністю таких методів за наявності розмічених даних – вони дозволяють досягати високої точності розпізнавання відомих шаблонів шахрайства. Значно менша частка робіт (близько 18%) присвячена методам без учителя, а глибинне навчання прямо використовувалося лише в ~3% випадків [1]. Тим не менш, у практиці великих фінансових компаній спостерігається тренд до впровадження саме ML-рішень. Наприклад, платіжні платформи (Stripe, PayPal тощо) застосовують алгоритми, що навчаються на масивних наборах транзакцій, щоб виявляти шахрайство і миттєво блокувати підозрілі платежі, використовуючи досвід мільйонів користувачів глобально [1].

Ключова перевага ML у боротьбі з шахрайством – *здатність виявляти нові та складні закономірності*, які неможливо повністю передбачити вручну складеними правилами [6]. Машинне навчання, по суті, автоматизує процес «навчання на помилках»: модель аналізує історію шахрайських інцидентів і поступово вдосконалює свої прогнози для майбутніх транзакцій. На практиці часто використовують ансамблі кількох моделей або кілька рівнів алгоритмів, щоби досягти кращих результатів.

У сучасних банківських рішеннях ML-алгоритми інтегруються у *системи моніторингу транзакцій*. Це може бути окрема служба (модуль), що приймає інформацію про кожну транзакцію (суму, час, рахунок відправника і отримувача, тип операції тощо) і повертає оцінку ризику або ж пропоноване рішення (пропустити, відхилити, позначити для перевірки). Наприклад, система Stripe Radar оцінює понад 1000 різних ознак транзакції за допомогою ML, щоб за 100

мілісекунд винести рішення щодо блокування оплати [9]. Прикладом цього є архітектура Stripe Radar Wide & Deep Shield [10], схема роботи якої наведена на рисунку 2.1.

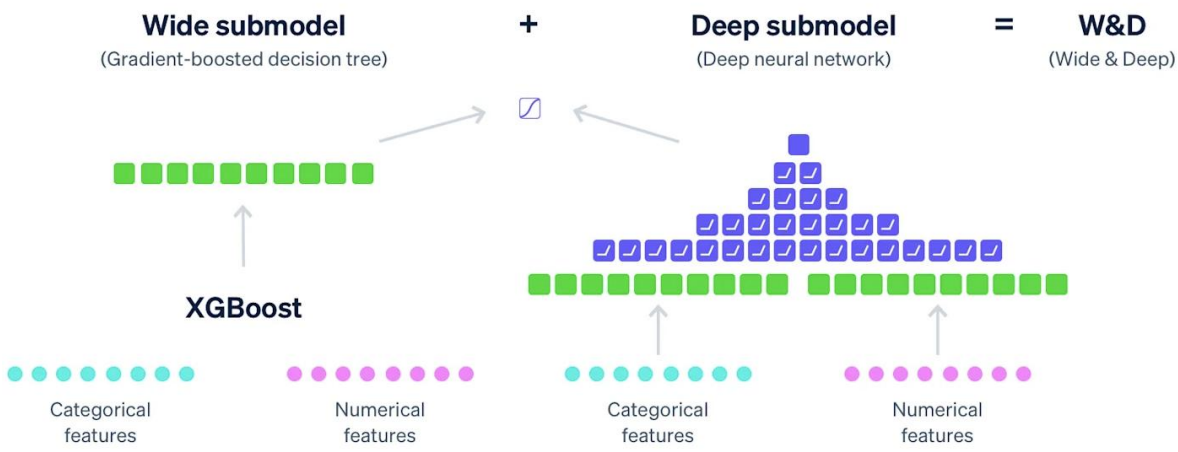


Рисунок 2.1 – Модель Wide & Deep

Схема роботи моделі Wide & Deep поєднує дві підмоделі: підмодель Wide, яка використовує алгоритм XGBoost (градієнтний бустинг на деревах рішень та Deep-підмодель, що є заснованою на архітектурі глибокої нейронної мережі. Wide-підмодель є навченою ефективно опрацьовувати категорійні та числові ознаки, дозволяючи захопити прості, але критичні закономірності. Deep-підмодель використовує ті ж самі категорійні та числові ознаки, але дана підмодель є здатною знаходити складні та нелінійні взаємозв'язки між ознаками завдяки своїй багатошаровій архітектурі. Поєднання цих двох підходів у вигляді моделі Wide & Deep забезпечує ефективне виявлення простих та складних взаємозв’язків у даних та цим підвищує загальну точність та надійність прогнозів

Банки JPMorgan Chase, Westpac використовують алгоритми виявлення аномалій для виявлення відхилень від типової поведінки клієнтів, тоді як Capital One застосовує ML для розпізнавання підроблених чеків та фальсифікацій у документах [6]. Такі приклади свідчать, що ML успішно працює “на передовій” (Front Lines) захисту фінансових послуг.

Загалом, огляд свідчить, що машинне навчання стало невід'ємним компонентом сучасних систем запобігання шахрайству в банківській галузі. Подальші підрозділи цього розділу розглянуть основні категорії алгоритмів машинного навчання – «із вчителем», «без вчителя», глибинні – та їх особливості у застосуванні до задач виявлення шахрайських транзакцій, а також порівняють їх ефективність.

2.2 Методи навчання «із вчителем» для виявлення шахрайства

Навчання "із вчителем" (*supervised learning*) – це підхід, при якому модель навчається на мітках: у даному випадку на прикладах транзакцій з позначками «шахрайська» чи «ні». Метою є побудувати класифікатор, який на основі ознак нової транзакції передбачить її клас. Цей підхід є найпоширенішим у задачах фінансового шахрайства [1], оскільки за наявності якісного розміченого датасету він демонструє високу точність і надійність. Далі буде детальніше описано кожен з методів навчання «із вчителем».

2.2.1 Логістична регресія

Логістична регресія (Logistic Regression) – статистичний метод, що моделює ймовірність шахрайства на основі лінійної комбінації ознак. Перевагами цього методу є простота, висока швидкість роботи та інтерпретованість (можна оцінити внесок кожної ознаки в ризик) [8]. На рисунку 2.2. додатково описано принцип роботи методу.

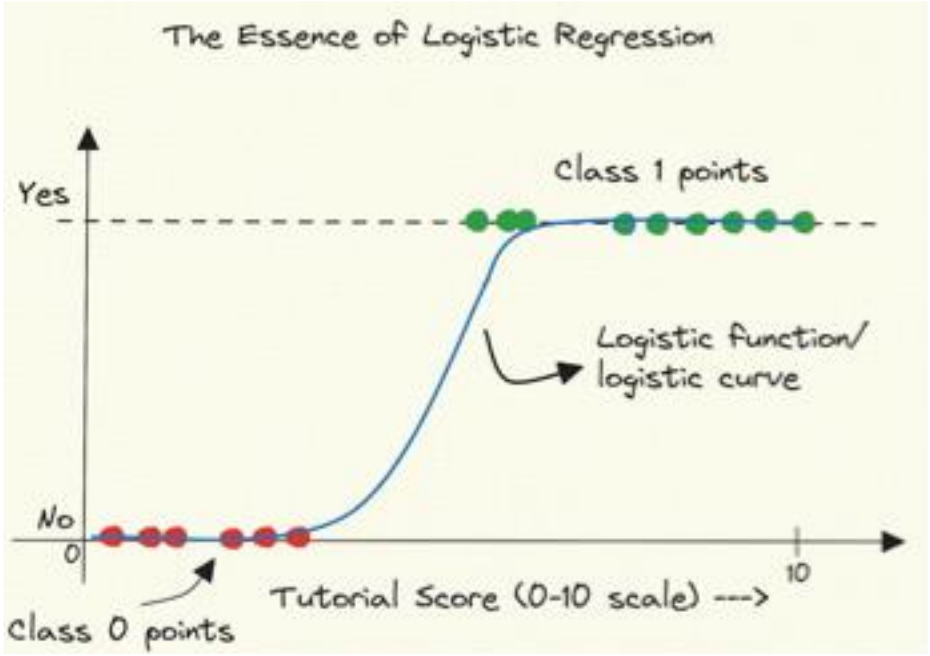


Рисунок 2.2 – Метод логістичної регресії

По горизонтальній осі відкладається вхідне зважене значення певної ознаки (у рисунку зображено оцінку в діапазоні від 0 до 10), а на вертикальній осі знаходиться ймовірність належності значення до одного з двох класів ("Yes" або "No"). Логістична функція виглядає як S-подібна крива, яка трансформує числове значення ознаки в інтервал від 0 до 1. Точки позначені червоним кольором (підписані як «class 0 points») знаходяться в нижній частині графіка і належать до негативного класу ("No"), а зелені точки («class 1 points») знаходяться у верхній частині графіка і вказують на позитивний клас ("Yes"). Ця крива дозволяє визначити межу між класами та прогнозувати ймовірність належності до одного з класів для нових спостережень.

Для цього методу існує велика кількість «активаційних функцій», що вирізняються крутизною нахилу, та пологістю на вході/виході. Основними функціями, що найчастіше застосовуються у логістичній регресії є наступні: Лінійна, Сигмоїд, ReLu та Гіперболічний тангенс. Важливою їх особливістю є можливість диференціювання з невеликими обчислювальними ресурсами задля навчання моделі

2.2.2 Дерева рішень

Дерева рішень (*Decision Trees*) – деревоподібні моделі, які навчаються розбивати простір ознак на області з переважно шахрайськими чи нешахрайськими прикладами. Дерева легко пояснити (шлях від кореня до листка є набором правил). Для прикладу на рисунку 2.3 представлено дерево рішень для визначення шахрайства.

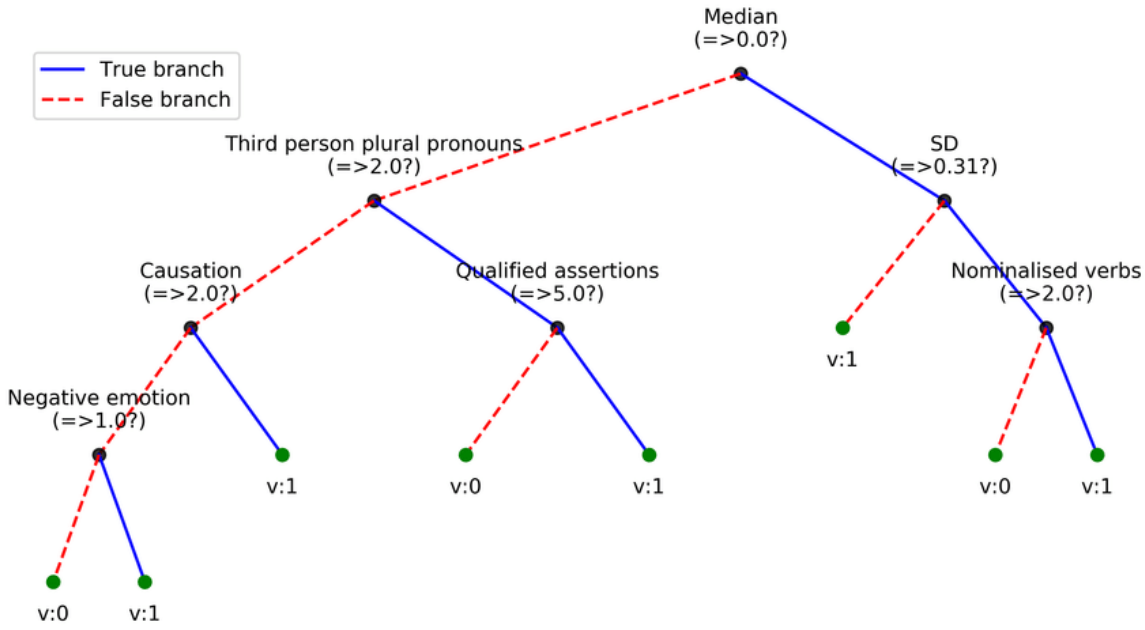


Рисунок 2.3 – Приклад дерева рішень для визначення шахрайства

Слід також уточнити, що дерева рішень будуються автоматично шляхом покрокового навчання за наростаючої точності, що і вирізняє їх від простого набору правил, заданих ручним чином. Проте одне дерево може перенавчитися на даних, тому частіше використовують ансамблі дерев.

Випадковий ліс (*Random Forest*) – ансамбль багатьох дерев рішень, де кожне дерево будується на дещо різних вибірках і підмножинах ознак, а потім «голосує» за клас. Random Forest показує високу точність і стійкість до оверфітингу, добре працює на незбалансованих даних.

2.2.3 Метод опорних векторів

Метод опорних векторів (SVM) – алгоритм, який шукає оптимальну гіперплощину для розділення класів у просторі ознак. Він добре працює для складних багаторозмірних даних, але погано масштабується на дуже великі вибірки. Даний метод може також бути застосованим для вирішення задачі класифікації - методу найближчих сусідів, де клас нової транзакції визначається класом k найбільш схожих транзакцій з історичних даних. На рисунку 2.4 представлено принцип роботи даного методу, що розділяє дані на групи, застосовуючи опорні вектори

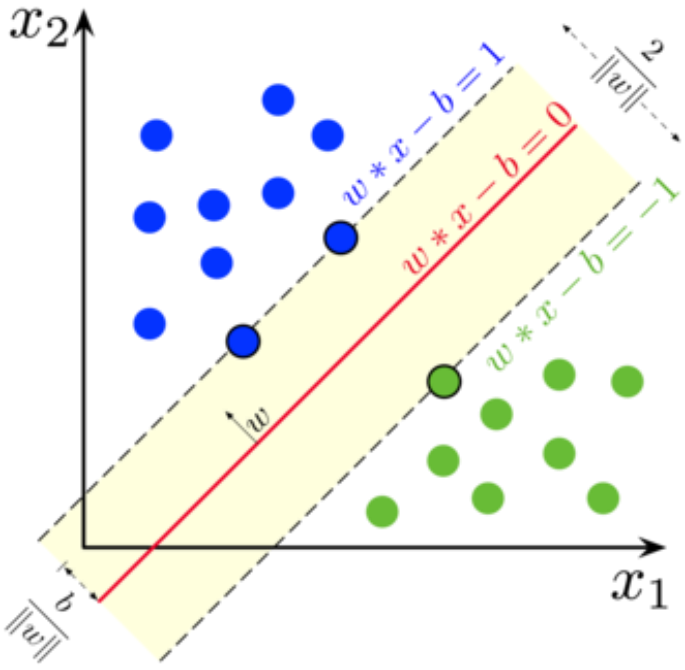


Рисунок 2.4 – Принцип роботи SVM

Центральна лінія, відмічена червоним кольором, є гіперплощиною, що відділяє два набори точок, позначені синім та зеленім кольорами, що належать до різних класів. Опорні вектори – це точки, що знаходяться найближче до розділювальної площини, а ширина смуги між двома класами максимізується для забезпечення найкращого розподілу.

Для виявлення шахрайства метод SVM використовується рідко через високі вимоги до пам’яті та часу на великих даних і чутливість до шуму

2.3 Методи навчання «без вчителя» для виявлення аномалій

Навчання без учителя (unsupervised learning) не потребує вручну розмічених даних. У контексті визначення шахрайства дана задача зазвичай вирішується як виявлення аномалій – пошук транзакцій, які значно вибиваються із загальної картини нормальної поведінки. Ідея полягає в тому, що шахрайські операції рідкісні та «не схожі» на типові легальні транзакції, тож їх можна виявити як статистичні аномалії.

2.3.1 Метод кластеризації

Кластеризація – це метод навчання «без вчителя», що полягає у групуванні схожих даних у кластери на основі їх характеристик. Кожен кластер містить транзакції, які мають схожі ознаки: наприклад, суми операцій, частоту платежів, або географічну прив’язку. Транзакції, які істотно відрізняються від будь-якого сформованого кластера, ідентифікуються як аномальні. Це відбувається тому, що вони не відповідають жодній з визначених груп типової поведінки, що часто є індикатором шахрайства [1].

Задача кластеризації ефективно вирішується методом k-середніх. Він працює шляхом вибору певної кількості центрів кластерів і послідовної оптимізації їх розташування. Початково алгоритм обирає випадкові точки як початкові центри кластерів. Далі він призначає кожну транзакцію до найближчого центра і перераховує центри як середні значення всіх транзакцій у кожному кластері. Цей процес повторюється кілька разів, доки кластерна структура не стабілізується. Цей процес візуально зображено на рисунку 2.5:

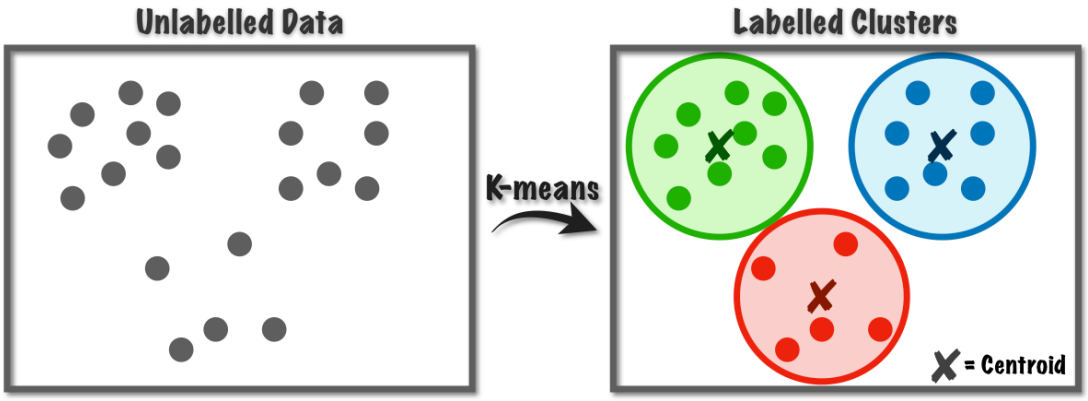


Рисунок 2.5 – Метод k-середніх

Аномалії виявляються як транзакції, що знаходяться далеко від визначених центрів кластерів, адже вони суттєво відрізняються за своїми характеристиками від типових даних у кластері [11].

2.3.2 Щільнісна кластеризація

Метод щільнісної кластеризації, зокрема алгоритм DBSCAN (Density-Based Spatial Clustering of Applications with Noise), ґрунтується на оцінці щільності точок у просторі ознак. Він автоматично формує кластери з даних, що густо розташовані разом. Для наглядності, порівняння методів k-means та DBSCAN наведено на рисунку 2.6.

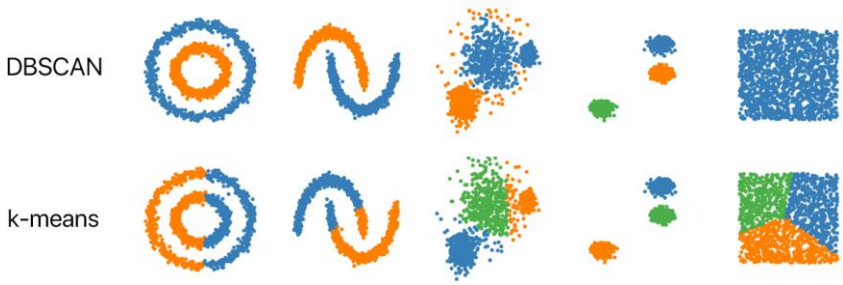


Рисунок 2.6 – Порівняння ефективності методів k-means та DBSCAN

Точки, що розташовані в менш щільних ділянках або взагалі не знаходяться поблизу жодної групи, позначаються як шум. Саме такі транзакції часто є потенційними шахрайствами, оскільки їхні характеристики істотно відрізняються від звичайної поведінки клієнтів. Цей метод дозволяє ефективно виявляти аномалії без попереднього визначення кількості кластерів, на відміну від методу k-середніх [12].

2.3.3 Автоенкодер

Автоенкодер (autoencoder) – це нейронна мережа, що навчається стискати вхідні дані у проміжне компактне представлення, після чого намагається максимально точно відновити вихідні дані з цього стислого представлення. Цей процес проілюстровано на рисунку 2.7.

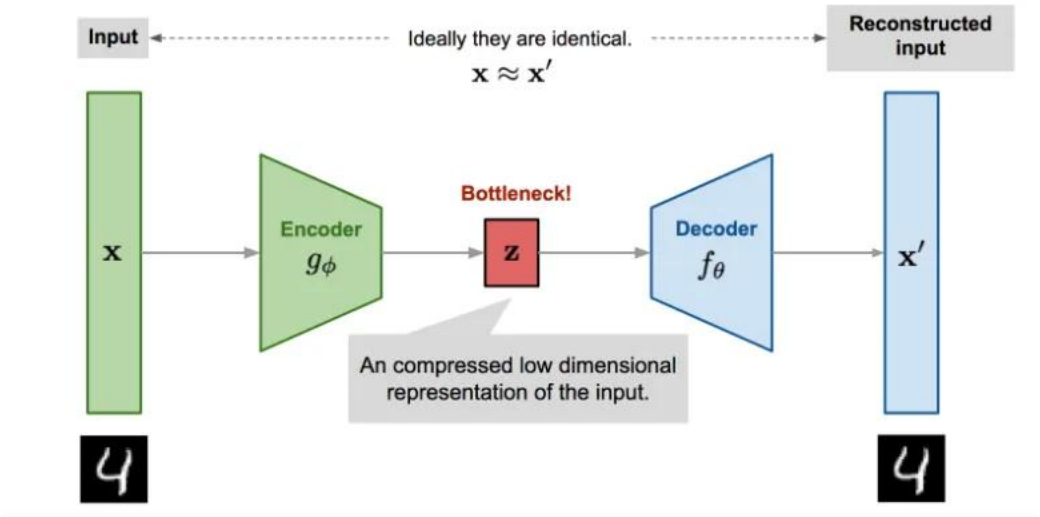


Рисунок 2.7 – Застосування автокодера для виявлення аномалій

Під час тренування автоенкодер навчається виключно на нормальних транзакціях. При застосуванні моделі для нових транзакцій обчислюється помилка реконструкції – різниця між початковими і відновленими даними. Якщо нова транзакція значно відрізняється від тих, на яких навчалась мережа, вона матиме велику помилку реконструкції. Таким чином, автоенкодер ефективно

ідентифікує аномальні транзакції як ті, що неможливо якісно відновити, що свідчить про відхилення від норми і, ймовірно, про шахрайство [13].

2.3.4 Ізоляційний ліс

Метод лісу ізоляції (Isolation Forest) заснований на випадковому поділі даних з метою швидкої ізоляції аномалій. Він створює ансамбль випадкових дерев, де транзакції розподіляються між гілками дерев за допомогою випадково вибраних ознак та порогових значень. Через свою нетипову природу, аномальні транзакції швидко ізолюються у верхніх частинах дерев, тоді як нормальні транзакції потребують більше кроків для їхньої ізоляції. Для кожної транзакції визначається міра аномальності за допомогою середньої кількості розподілів, необхідних для її ізоляції. Таким чином, транзакції, що легко ізолюються, класифікуються як аномальні [14].

2.3.5 Однокласовий SVM

One-Class SVM – це варіація методу опорних векторів, призначена для роботи з єдиним класом даних, який представляє нормальну поведінку. Цей алгоритм будує гіперплощину, яка окреслює межі нормальних транзакцій у просторі ознак. Всі дані, які лежать поза цією гіперплощиною, розглядаються як аномальні. Принцип роботи полягає в тому, що гіперплощина формується таким чином, щоб максимально компактно охоплювати нормальні дані, залишаючи за межами точки, які статистично та структурно відрізняються від цієї нормальної поведінки. Завдяки можливості використання нелінійних функцій ядра, однокласовий SVM ефективно ідентифікує складні закономірності та дозволяє знаходити транзакції, які навіть незначно відрізняються від типового шаблону, сигналізуючи про потенційне шахрайство [15].

2.3.6 Характеристика методів навчання «без учителя»

Перевагами методів навчання без учителя є здатність виявити нові невідомі типи шахрайства, які не представлені в ромічених навчальних вибірках. Вони особливо корисні на початковому етапі, коли бракує розмічених даних, або як додатковий рівень захисту поверх моделей «із вчителем». Наприклад, якщо з'являється зовсім новий вид шахрайства, supervised-модель його, ймовірно, проігнорує (бо не навчена на ньому), а визначення аномалій може видати сигнал «ця транзакція підозріла, бо дуже незвична», що дасть змогу проведення детального дослідження згодом.

Недоліками цих методів є можлива велика кількість хибних спрацьовувань. Аномалією може бути рідкісна, але легальна операція (наприклад, клієнт вперше в житті купує автомобіль – сума велика і нетипова, але це не шахрайство). Налаштувати чутливість моделі «без вчителя» складно, потрібні пороги, які часто визначають емпірично. Крім того, деякі шахрайські транзакції можуть не виглядати аномальними, якщо шахраї діють обережно і маскують свої операції під нормальні.

На практиці алгоритми «без вчителя» іноді використовуються у зв'язці із тими, що використовують розмічені дані. Наприклад, підозрілі з точки зору аномалій транзакції потім передаються для класифікації моделлю «із вчителем» (двоступеневий підхід). Або результати виявлення аномалій служать новими параметрами для моделі із вчителем (збільшують її поінформованість).

В літературі відзначається, що методи без учителя та глибинне навчання в контексті банківського шахрайства досліджені значно менше, ніж класичні методи з учителем . Це пояснюється складністю їх застосування та інтерпретації, а також тим, що банки все ж часто мають хоча б невеликі вибірки розмічених шахрайств і надають перевагу будувати рішення на їх основі. Однак зі зростанням нових загроз інтерес до виявлення аномалій зростає.

2.4 Глибинне навчання

Глибинне навчання (Deep Learning) – це підклас методів машинного навчання, що використовує багатошарові нейронні мережі для побудови моделей високої складності. У задачі виявлення шахрайства глибинне навчання може застосовуватися як у режимі «із вчителем» (глибокі нейронні мережі-класифікатори), так і у режимі «без вчителя» (автокодери, генеративні моделі для синтезу даних тощо).

Далі буде описано приклади глибинних моделей, застосовуваних у фінансовому секторі. *Багатошарові перцептрони* (MLP) – мережі із кількома прихованими шарами, які можуть апроксимувати складні залежності між вхідними ознаками і результатом шляхом виявлення високорівневих шаблонів. Виявлено, що нейронні мережі ефективні для моделювання нелінійних співвідношень та комбінованих ознак, що важко врахувати вручну [7]. Наприклад, мережа може врахувати одночасно і географію, і час, і суму транзакції, виявивши комплексний шаблон шахрайства. *Рекурентні нейронні мережі* (RNN, зокрема LSTM) – застосовуються, коли важливо врахувати послідовність операцій у часі. На рисунку 2.8 зображено схему роботи комірок LSTM:

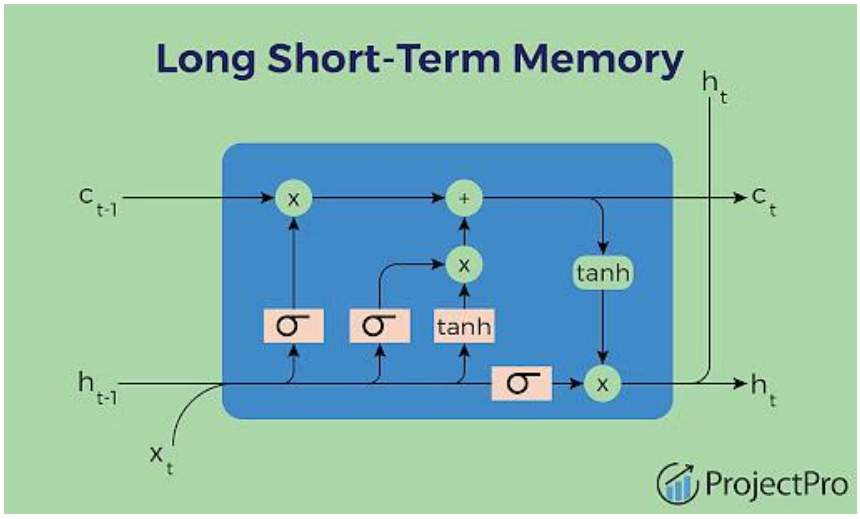


Рисунок 2.8 – Схема роботи комірки LSTM

З рисунку видно, що за «пам'ять» відповідає окрема функція активації, що також має ваги на вході, що дозволяє «забувати» та «пам'ятати» певну послідовність подій. Наприклад, LSTM-мережа може аналізувати потік транзакцій клієнта як часовий ряд і помітити, що послідовність дій «баланс переведений у готівку, потім великий переказ за кордон» є нетиповою і ризиковою. *Графічні* нейронні мережі (GNN) – перспективний напрям, де транзакції та рахунки розглядаються як граф (вузли – акаунти/картки, ребра – операції). Приклад роботи даного методу зображено на рисунку 2.9.

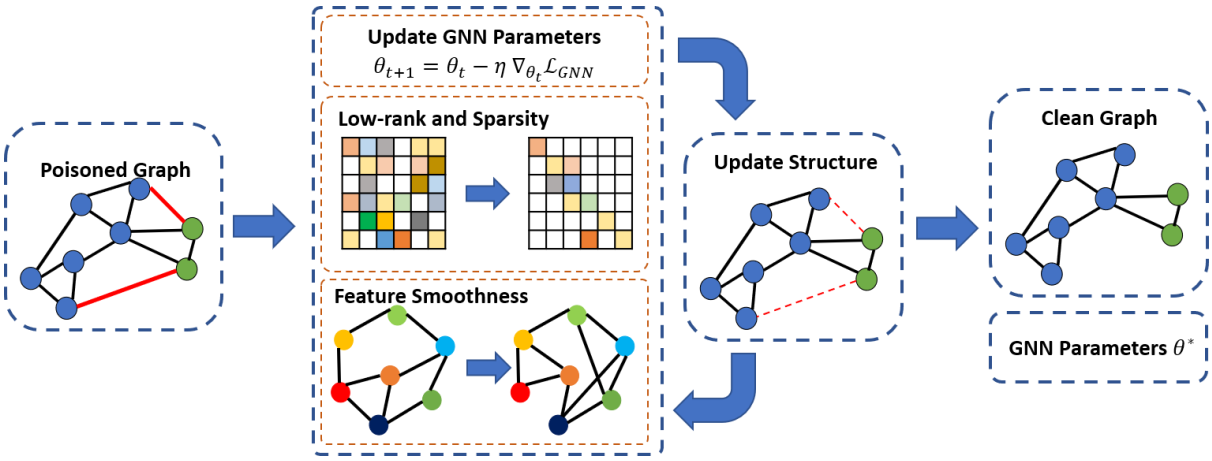


Рисунок 2.9 – Схема роботи GNN

GNN можуть навчатися на такому графі для виявлення аномальних зв'язків або субграфів, характерних для шахрайських схем (наприклад, «каруселі» платежів між пов'язаними рахунками) [16]. *Автокодери* та *генеративні моделі* – як згадувалося у попередніх підрозділах, автоенкодери можуть виконувати роль детектора аномалій [17]. Також *генеративно-змагальні мережі* (GAN) застосовуються для синтезу додаткових прикладів шахрайських транзакцій з метою балансування вибірки або для моделювання поведінки зловмисника, що теж допомагає в навчанні стійкіших моделей [18].

Головною перевагою deep learning є висока гнучкість і потенційно краща результативність при достатніх даних. Глибинні мережі можуть самі виокремити важливі ознаки (їхні глибокі шари будують абстрактні представлення даних). У деяких дослідженнях відзначається, що поєднання базових алгоритмів

машинного навчання із глибинними моделями покращує показники (наприклад, ансамбль Random Forest + глибинна мережа давав приріст точності) . Відомо, що глибинні моделі домінують в таких сферах, як розпізнавання зображень, або голосу, тобто у даних, у яких є дуже складні патерни. У фінансовому шахрайстві вони поки що не витіснили повністю класичні методи, але теж застосовуються.

Обмеження deep learning:

- *Вимоги до даних*: потрібні дуже великі обсяги транзакцій для навчання, особливо якщо мережа глибока. Для багатьох банків критично бракує достатньо прикладів саме шахрайств, тому вони або комбінують дані з різних джерел, або генерують синтетичні. Відомо, що deep learning моделі дуже "голодні" до даних і можуть переобучитися на малих вибірках .

- *Обчислювальні ресурси*: тренування DNN – витратний процес (потрібні GPU тощо). Реальний час виконання теж може бути питанням, але зазвичай inference (прогнозування) навіть глибокої мережі досить швидке, інша справа – затрати пам'яті і CPU.

- *Інтерпретація*: як згадувалося, глибокі моделі є по-суті "чорним ящиком", оскільки складно пояснити їх рішення. У банківській сфері це мінус з точки зору довіри і регуляторних вимог.

- *Налаштування*: багато гіперпараметрів і архітектурних рішень (кількість шарів, нейронів, функції активації, оптимізатори). Не існує універсальної схеми, часто потрібен досвід і спроби/помилки, щоб підібрати оптимальну мережу під конкретні дані.

Наразі у фінансовому секторі deep learning найчастіше використовується великими гравцями з масивами даних – наприклад, світові платіжні системи. В академічній літературі є роботи, що демонструють переваги глибинних мереж, але також є застереження, що їх складність має виправдовуватися значним виграшем у якості [19]. Багато банків знаходять, що добре налаштований Random Forest або XGBoost дає майже такий самий результат, як нейромережа, але простіший в розгортанні та поясненні [1]. Однак перспективи за глибинним навчанням – з зростанням обчислювальної потужності та розвитком технік

transfer learning (перенесення знань з одних даних на інші) глибинні моделі можуть зайняти провідне місце і тут.

2.5 Порівняння ефективності підходів

Для оцінки різних підходів до виявлення шахрайства зазвичай розглядають такі критерії: accuracy, precision, recall, F1, AUC. Саме завдяки цим критеріям можна зробити висновок наскільки добре метод знаходить шахрайські транзакції і при цьому не помилково блокує легальні. Здатність виявляти нові схеми має на увазі чи може метод спрацювати на невідомий раніше тип атаки. Швидкість та масштабованість – чи підходить метод для роботи в режимі реального часу на великих потоках даних. Критерії щодо простоти впровадження та підтримки включає складність навчання, потребу в мітках, вимоги до обчислювальних ресурсів, інтерпретацію.

Методи “із вчителем” (дерева, ансамблі, регресія) демонструють високу точність на відомих шаблонах шахрайства і, як правило, забезпечують краще співвідношення precision/recall, ніж підходи «без учителя», якщо є хороший навчальний набір. У порівняльних експериментах на загальнодоступних даних (наприклад, датасет кредитних карток з Kaggle) алгоритми XGBoost і Random Forest часто посідають перші місця за F1-мірою, випереджаючи прості моделі на зразок дерева чи логістичної регресії . Приміром, в одному дослідженні XGBoost досяг найвищого показника F1 ~0.88, тоді як окреме дерево рішень мало F1 ~0.81, а метод опорних векторів через дисбаланс даних практично не зміг виявити шахрайств (F1 ~0) . Цей приклад ілюструє, що складні ансамблеві моделі перевершують прості моделі за точністю в умовах фінансових даних з нерівномірним класовим розподілом.

Методи «без учителя» поступаються за абсолютною точністю, тобто вони можуть виявити лише частину шахрайств і часто мають нижчий precision (більше хибних тривог). Однак їх перевага – ширше охоплення невідомих випадків. У ситуації, коли шахрайська схема нова, виявлення аномалій може дати принаймні

повідомлення, тоді як модель, тренована виключно на розмічених даних, не розпізнає. Тому з точки зору ефективності в довгостроковій перспективі найкраще комбінувати підходи «із вчителем» та «без вчителя»: перші забезпечують високу точність на відомих шаблонах, другі додають можливість спіймати “чорних лебедів”.

Глибинне навчання за достатніх умов може перевершити класичні алгоритми. Але на практиці у багатьох випадках вдало налаштований градієнтний бустинг (наприклад, CatBoost) забезпечив не гіршу точність, ніж глибинна нейромережа, при значно меншому часі розробки і обчислень . У роботі , де порівнювали LightGBM, XGBoost, CatBoost, нейронну мережу та їх ансамбль, було відзначено, що CatBoost показав найвищу точність, а різні методи масштабування та балансування даних помітно вплинули на поліпшення результатів [19]. Нейромережа не давала суттєвого виграшу, а додавання її до ансамблю лише трохи покращило ефективність. Цей висновок відповідає поширеній думці: якщо є достатньо інженерії ознак і потужний ансамбль дерев, то виграш від DL може бути мінімальним. Але у випадках, коли шаблони дуже складні або дані мають іншу природу (наприклад, текстові поля, або треба аналізувати послідовності операцій) – тоді deep learning стає незамінним.

З точки зору продуктивності: алгоритми на основі дерев (RF, XGBoost) досить швидкі в прогнозуванні і можуть бути оптимізовані. Нейромережі теж можуть працювати швидко на відповідному апаратному забезпеченні. А ось методи «без учителя», як-от DBSCAN чи k-NN, можуть бути повільними на великих наборах, хоча сучасні реалізації та попереднє агрегування даних вирішують це питання.

Підсумовуючи проведене в рамках порівняння моделей методи “із вчителем” забезпечують найкращу точність і використовуються як основа більшості систем, алгоритми виявлення аномалій доповнюють їх, додаючи виявлення нових аномалій, а глибинне навчання – перспективний шлях підвищення точності на складних даних, хоча й вимагає більше ресурсів. Найефективнішими вважаються гібридні рішення: наприклад, на першому етапі – швидке виявлення

аномалій, на другому – потужна supervised-модель для підтвердження шахрайства. Кінцева мета – максимізувати recall (вловити якомога більше шахрайств) при збереженні високого precision (мінімум хибних блокувань).

Для кращого розуміння ефективності кожного з методів у роботі було проведено дослідження точності роботи ряду найпопулярніших алгоритмів машинного навчання: дерев рішень, k-NN, логістичної регресії, методу опорних векторів, випадкових лісів, методу XGBoost та багатошарового перцептрону. Нижче наведено графік порівняння метрики F1 цих моделей (за даними з відкритого набору транзакцій) (див. рисунок 2.10):

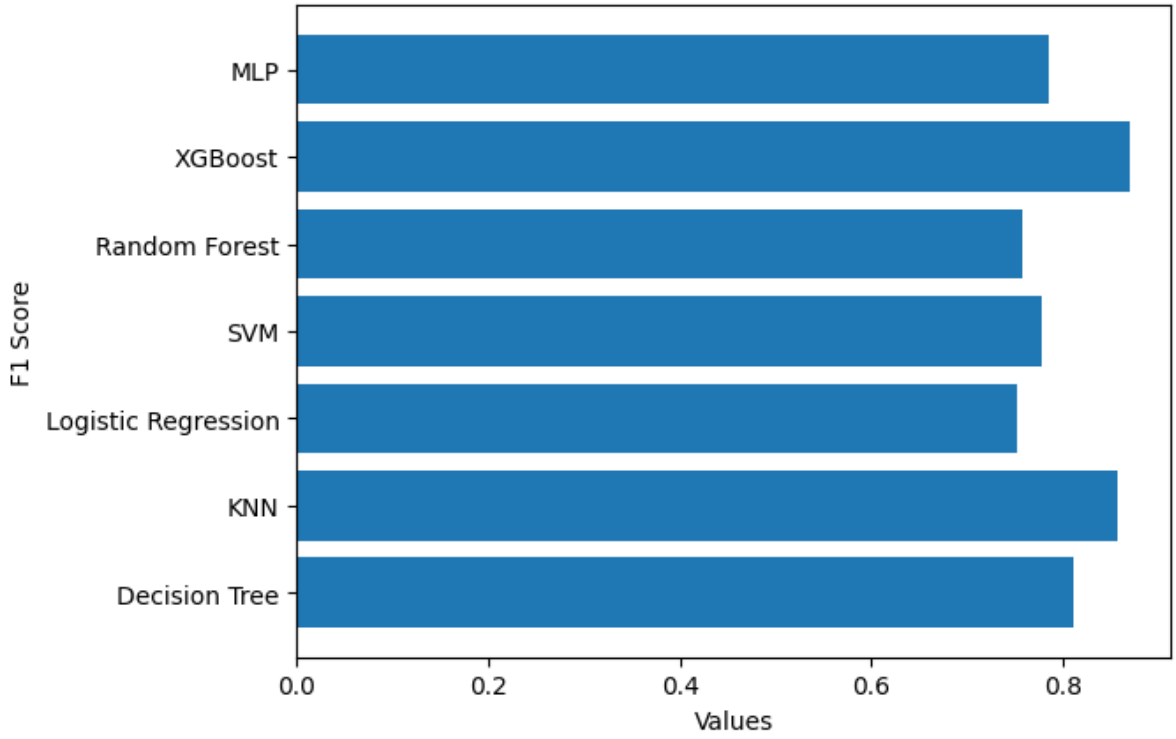


Рисунок 2.10 – Порівняння ефективності роботи кількох моделей

Враховуючи високий рівень ефективності, що представлено на рисунку вище, а також особливості структури вхідних даних (велика кількість вхідних параметрів, табличність параметрів та наявність задачі бінарної класифікації) було обрано двох-етапну гібридну архітектуру, що використовує Ізоляційний ліс на першому рівні та MLP як головну модель для практичної реалізації у 3 розділі.

РОЗДІЛ 3 РОЗРОБКА СИСТЕМИ ВИЯВЛЕННЯ ШАХРАЙСЬКИХ БАНКІВСЬКИХ ОПЕРАЦІЙ ЗАСОБАМИ PYTHON ТА TENSORFLOW

3.1 Опис обраного підходу

Для реалізації системи виявлення шахрайських банківських операцій у даній роботі обрано підхід, що поєднує модель глибинного навчання та попереднє виявлення аномалій. Зокрема, результат роботи моделі виявлення аномалій застосовується як додаткова ознака для моделі глибинного навчання, що у свою чергу призводить до підвищеної точності роботи. Такий вибір обґрунтовується тим, що нейронні мережі можуть виявляти складні нелінійні залежності між ознаками і виявляти приховані шаблони, притаманні шахрайським транзакціям, що важко дається більш простим системам, однак вони не здатні виявити атаки що не були відомими раніше. Система ж виявлення аномалій здатна виявити непритаманну поведінку, властиву подібному роду атак. Тобто глибинне навчання забезпечує сильний показник розпізнавання відомих шаблонів, а anomaly detection додає чутливості до потенційно нових типів шахрайства (стійкість до «атаки нульового дня»)

Перед класифікацією нейронною мережею кожна транзакція також оцінюється простішим алгоритмом anomaly detection (із застосуванням Isolation Forest). Якщо транзакція відзначається як явно аномальна (наприклад, потрапляє в топ-1% найбільш нестандартних за метрикою моделі), система може негайно помістити її на перевірку, навіть без очікування результату основної моделі. Однак у більшості випадків рішення приймається основною нейронною мережею, що є наступним рівнем розпізнавання шахрайства.

При цьому нейронна мережа навчатиметься в режимі з учителем на розмічених даних: їй подаватимуться приклади транзакцій з міткою «шахрайська» або «нормальна». Мета – мінімізувати помилки класифікації.

Вибір поєднання підходів пояснюється прагненням максимізувати повноту виявлення при збереженні високої точності. Очікується, що така система зможе виявити більшу частину шахрайських операцій і при цьому обмежити частку помилкових блокувань до прийнятного рівня.

Варто зазначити, що у процесі проектування було розглянуто альтернативні архітектури: наприклад, використання ансамблю класичних моделей (Random Forest + XGBoost) або спрощеної логістичної регресії для інтерпретованості. Однак ці системи гірше працюють із збільшенням кількості ознак та нелінійності їх взаємозв'язків, а логістична регресія виявилась би недостатньо гнучкою, щоб уловлювати всі закономірності. Тому з урахуванням вимог до реалізації і прагнення високої якості було обрано саме глибинну нейронну мережу як ядро системи.

3.2 Архітектура та компоненти системи

Розроблена система має модульну архітектуру, що інтегрується у потік обробки банківських транзакцій. Основні компоненти та їх взаємодія такі:

- Джерела даних,
- Підсистема збору і агрегування даних,
- Модуль попередньої обробки,
- Модуль виявлення аномалій,
- ML-модель (нейронна мережа).

Далі буде проведено детальніший опис кожного компоненту.

Щодо джерела даних, то система отримує вхідні дані з транзакційного потоку банку в режимі реального часу. Це можуть бути повідомлення від систем обробки карток, систем онлайн-банкінгу, SWIFT/міжбанківських систем тощо. В рамках даної роботи було обрано набір відкритих даних із системи Kaggle (детальніше див. підпункт 3.3.1) Кожна транзакція надходить з атрибутами: ідентифікатор, час, номер картки, сума, тип операції (визначений кодом),

геолокація транзакції, назва точки обслуговування, геолокація точки обслуговування (якщо доступна), та інші менш важливі для роботи атрибути.

До підсистеми збору і агрегування даних відноситься перший модуль, який відповідає за об’єднання різнорідних даних про транзакцію в єдиний профіль для моделі. Він може звертатися до внутрішніх баз, щоб витягнути додаткову інформацію (наприклад, історичні агрегати по рахунку: скільки транзакцій за останню годину, середній баланс, чи були раніше випадки шахрайства по цьому клієнту тощо). Також можуть підтягуватися зовнішні дані – наприклад, чи входить отримувач у чорний список, чи підозріла IP-адреса тощо. На жаль, реалізацію даного модуля немає можливості реалізувати, адже вхідний набір даних вже повний та не містить додаткових джерел.

Одержавши повний набір ознак транзакції, модуль попередньої обробки здійснює їх перетворення та нормалізацію. Тут реалізовані всі кроки, описані у пункті 3.3: очищення (хоча для онлайнових даних очищення мінімальне, бо дані мають бути валідні за визначенням), інженерія ознак, масштабування, тощо. Цей же модуль формує додаткові ознаки: розраховує відстань між геолокацією поточної транзакції і місцем знаходження продавця, інтервал часу від останньої активності, незвичність торговця для клієнта тощо. В разі потреби тут же можуть застосовуватися техніки балансування – але в режимі реального часу балансування не здійснюється (воно виконується на етапі навчання моделі, див. підпункт 3.3). Після цього транзакція представлена у вигляді вектора числових ознак заданої довжини, готового для подачі в моделі.

Модуль виявлення аномалій є окремим компонентом, що отримує на вхід підготовлений вектор ознак транзакції і обчислює міру її «аномальності». Використовується попередньо навчений алгоритм Isolation Forest, який був навчений на вибірці нормальних транзакцій і знає, як виглядає «норма». У контексті роботи це відфільтрований набір даних за негативною міткою шахрайства. На виході модуль видає числовий показник від 0 (норма) до 1 (сильно підозріла аномалія) для кожної операції.

DL-модель (нейронна мережа): Головний модулем є це TensorFlow ML-модель (нейронна мережа). Вона отримує на вхід вектор ознак транзакції та значення anomaly score, що просто є додатковою ознакою із float-значенням у межах від 0 до 1. Модель обчислює ймовірність шахрайства P(fraud) для даної транзакції. Архітектура моделі буде більщ детально описаною у підрозділі 3.4, загалом це багатошаровий перцептрон з декількома прихованими шарами.

Описану у цьому пункті архітектуру можна поділити на дві частини: офлайнова (batch) частина – підготовка даних, навчання моделі (розглядається в підрозділах 3.3 і 3.4), та онлайнова частина – «бойова» експлуатація, описана вище. На офлайн-етапі система навчила модель ML, налаштувала пороги; на онлайн-рівні виконує інференс моделі на потоках даних. Такий поділ відповідає стандартній практиці, тобто ML-pipeline тренує модель на історичних даних, потім розгортає її у production-середовищі для роботи у реальному часі [7].

Схематично процес можна уявити так: потік транзакцій → обробка даних → виявлення аномалій → нейронна мережа → рішення (пропустити/блокувати) з оповіщенням. На рисунку 3.1 наведено схему роботи даного процесу. Цей цикл виконується для кожної, тому важливо, щоб усі модулі були.

Рисунок 3.1 – Схема роботи процесу виявлення шахрайських операцій

В ході проектування архітектури особлива увага приділена надійності. У робочій моделі також мають бути передбачені механізми оновлення моделі на ходу: нову версію нейромережі можна завантажити на сервер без зупинки системи і перевести трафік на неї (Blue-Green deployment), що важливо для

періодичного покращення алгоритмів. Даний механізм також допомагає оцінювати та порівнювати ефективність роботи алгоритмів та підходів в реальному часі (A/B testing). У даній роботі основна увага була зосереджена саме на виявленні шахрайства, тож не включає в себе всю інфраструктуру рішення.

Отже, архітектура системи забезпечує повний цикл обробки – від отримання сирих даних до прийняття рішення – та гнучкість у налаштуванні рівня контролю.

3.3 Підготовка даних для навчання моделі

Ефективність ML-моделі значною мірою залежить від якості підготовки даних. На етапі навчання були виконані *такі кроки*: формування навчальної вибірки, очищення даних, інженерія ознак (feature engineering). Нижче наведено більш деталізований опис кожного із кроків.

3.3.1 Формування навчальної вибірки

Із відкритого каталогу даних Kaggle було обрано набір із назвою «Credit Card Transactions Fraud Detection», що включає всі доступні випадки підтвердженого шахрайства за останні роки, а також випадкову вибірку нормальних транзакцій [20]. У додатку А відображено інформацію про даний набір даних, а саме представлено перелік наявних ознак із їх відповідними типами даних. Всього зібрано 1852394 транзакцій, з них шахрайських приблизно 1%, що відображено на рисунку 3.2.

```
total.value_counts('is_fraud')
```
✓ [6] 30ms

2 rows | 2 rows × 1 cols

is_fraud	count
0	1842743
1	9651

Рисунок 3.2 – Кількість транзакцій відносно міток шахрайства

Задля того, аби низька кількість шахрайських транзакцій у наборі не впливала на точність моделі, було програмно «доповнено» набір даними із позитивним значенням мітки. Реалізацію даної дії представлено на рисунку 3.3.

```
df_majority = total[(total['is_fraud']==0)]
df_minority = total[(total['is_fraud']==1)]

df_majority.shape,df_minority.shape
```

[13]

```
((1842743, 33), (9651, 33))
```

```
df_minority_upsampled = resample(df_minority,
                                 replace=True,    # sample with replacement
                                 n_samples= 1842743, # to match majority class
                                 random_state=42)  # reproducible results
df_minority_upsampled.shape
```

[14]

```
(1842743, 33)
```

```
total_upsampled = pd.concat([df_minority_upsampled, df_majority])
```

Рисунок 3.3 – Доповнення даних із позитивним значенням мітки шахрайства, для балансування датасету

Для збереження актуальності, даний набір містить дані останніх 1-2 років (занадто старі випадки могли бути не релевантні щодо сучасних ознак, як згадувалось у пункті 1.4).

3.3.2 Очищення даних

Зібраний набір зразу містив у собі лише валідні повні дані без викидів. Підтвердимо це перевіривши дані на isnull, результат роботи відображено на рисунку 3.4.

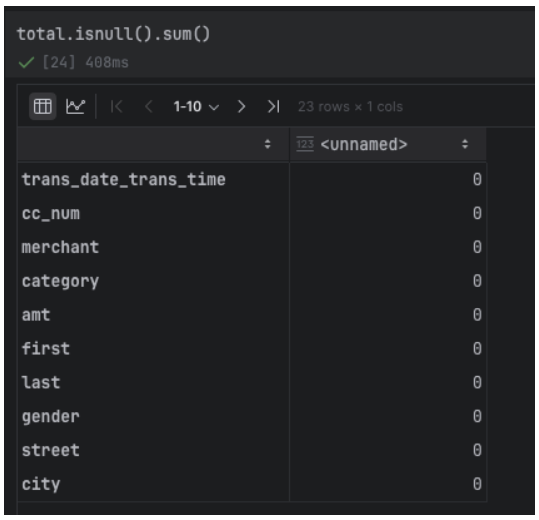


Рисунок 3.4 – Перевірка даних на повноту

Поля, що мали надто багато пропущених значень, було або вилучено, або заповнено нейтральними значеннями («unknown» категорія для текстових, 0 для числових, якщо доречно). Випадки явних помилок (негативна сума тощо) відфільтровано.

Після цього набір було додатково позбавлено полів, що явно не несуть кореляції із кінцевою міткою шахрайства, задля збільшення точності моделі (графіки зв’язку ознак та кінцевої мітки шахрайства приведено у додатку Б). Виконання цього кроку наведено на лістингу 3.1:

Лістинг 3.1 – Видалення полів, що не корелюють з міткою шахрайства

```
total.drop(['cc_num','trans_date_trans_time','first','last','street','city','zip','unix_time','trans_num','dob','trans_date','job'],axis=1,inplace=True)
```

Технічно цей крок знаходиться в самому кінці процесу обробки даних, що не зашкодить інженерії ознак (див. підпункт 3.3.3). Тож набір містив достовірні дані, готові до подальшої обробки та аналітики.

3.3.3 Інженерія ознак

Відносно даного датасету було проведено аналіз полів, типів їх даних задля визначення стратегії подальшої роботи з ними. У додатку А наведено перелік вхідних ознак з їх відповідними типами даних. Для кожної транзакції обчислено ряд додаткових ознак, корисних для моделювання:

- Ознаки часу: до цих ознак входили такі, як вік клієнта (що може бути додатково складеним до показника вікової групи), місяць транзакції та рік її вчинення (оскільки взято дані лише останніх років, то доступні значення цієї ознаки лише 2024 та 2025).
- Географічні: відстань між місцем цієї операції і місцем знаходження продавця (було вирішено розділити відстань окремо на різницю за широтою та за довготою); штат здійснення, оскільки всі дані були взяті із США.
- Історичні мітки: чи був клієнт раніше жертвою шахрайства

До категорійних ознак (тип операції, категорія продавця, регіон, тип пристрою) застосовано метод обробки One-Hot Encoding – де кожна категорія була перетворена на бінарний стовпець (або кілька стовпців для спільних категорій), що відображає належність до даної ознаки. Це значно збільшило розмірність вектора ознак, але нейронна мережа може з цим працювати (можливим наслідком для мережі в такому підході може бути overfit, але емпіричним шляхом було з’ясовано, що в даному дослідженні це не мало

негативного ефекту). У даному наборі цими ознаками були «category» та «gender».

Ознаки, описані вище, були додані до базових полів транзакції. Лістинг коду, що відповідає за ці перетворення наведено у додатку В.

Таким чином, фінальний набір ознак значно розширився і містив інформацію не тільки про саму транзакцію, а й про контекст навколо неї. Загальна кількість числових ознак становила 27.

3.3.4 Перетворення та масштабування

Числові ознаки (суми, кількості) часто мали різні масштаби і розподіли. Для них застосовано нормалізацію: зокрема, у більшості було застосовано StandartScaler із бібліотеки scikit-learn (віднімання середнього і ділення на стандартне відхилення). Всі ознаки приведено до приблизно однакового діапазону, щоб жодна не домінувала у навчанні моделі через свій масштаб. На лістингу 3.2 наведено код нормалізації:

Лістинг 3.2 – Масштабування за допомогою StandartScaler

```
sc = StandardScaler()
X = sc.fit_transform(X)
```

Після виконання цих кроків дані були готові для навчання моделі. У підсумку кожна транзакція представлена вектором з *27* ознак.

Наступним кроком дані було розбито на тренувальну і тестову множини у співвідношенні 80/20 відповідно, при цьому шахрайські приклади порівну розподілено між ними, аби модель можна було адекватно оцінити на незнайомих випадках. Практична реалізація розбиття датасету спирається на функціонал бібліотеки scikit та її код наведено у лістингу 3.3:

Лістинг 3.3 – Розбиття датасету на навчальну та тестову вибірку

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
```

Дані було додатково перетворено у формат, зручний для TensorFlow за допомогою коду, відображеного на лістингу 3.4:

Лістинг 3.4 – Збереження ознак у форматі TensorFlow

```
X_train_vectorized = np.vectorize(lambda val: tf.convert_to_tensor(val))(X_train_mlp)
X_test_vectorized = np.vectorize(lambda val: tf.convert_to_tensor(val))(X_test_mlp)
```

В цілому, етап підготовки даних відповідав стандартному data science workflow і забезпечив надійний фундамент для їх обробки моделлю.

3.4 Реалізація ML-алгоритму засобами TensorFlow та Python

3.4.1 Опис процесу розробки

Розробка моделі здійснювалась мовою Python з використанням бібліотеки TensorFlow (версія 2.x) для побудови та навчання нейронної мережі. Основні кроки реалізації моделі наступні:

- Побудова архітектури нейронної мережі,
- Підготовка даних для моделі,
- Налаштування процесу навчання,
- Валідація результатів навчання та робота з гіперпараметрами,
- Реалізація модуля anomaly detection,
- Інтеграція з системою.

Кожен з цих кроків потребує більш детального опису, що зроблено у наступних підпунктах (3.4.2-3.4.7)

3.4.2 Реалізація модуля anomaly detection

Паралельно був навчений алгоритм Isolation Forest на тих самих навчальних даних (використовуючи лише негативний клас як "норму"). Це зроблено за допомогою бібліотеки scikit-learn. Ліс містив 100 дерев, глибина обмежена так, щоб кожен лист містив кілька прикладів.

Після навчання моделі кожній транзакції тренувального та тестового наборів присвоєно додаткову float-ознаку для подальшого опрацювання нейронною мережею. Реалізацію цього модуля та створення ознаки наведено у лістингу 3.5.

Лістинг 3.5 – Реалізація модуля виявлення аномалій

```
iso_forest = IsolationForest()
iso_forest.fit(X_train)

iso_forest_ytrain = np.vectorize(lambda val: val > -
1)(iso_forest.predict(X_train))[:, np.newaxis]
iso_forest_yhat = np.vectorize(lambda val: val > -
1)(iso_forest.predict(X_test))[:, np.newaxis]

X_train_mlp = np.hstack([X_train, iso_forest_ytrain])
X_test_mlp = np.hstack([X_test, iso_forest_yhat])
```

Виявилось, що багато із прикладів мали високий показник anomaly_score, та при цьому були помилково класифіковані мережею як «легітимні» – отже, модуль визначення аномалій справді доповнює нейронну мережу

3.4.3 Побудова архітектури нейронної мережі

Зважаючи на табличний характер даних (набір числових та категорійних ознак) і задачі бінарної класифікації, було вирішено використовувати багатошаровий перцептрон (Multi-Layer Perceptron). Архітектура мережі

підібрана експериментально: Вхідний шар: розмір = 27. Прихований шар 1: Dense (повнозв'язаний) на 55 нейронів, активація ReLu. Прихований шар 2: Dense на 26 нейронів, Sigmoid. Прихований шар 3: Dense на 12 нейронів, Sigmoid. Вихідний шар: Dense на 1 нейрон з активацією sigmoid (щоб видавати ймовірність шахрайства від 0 до 1).

На рисунку 3.5 відображено архітектуру нейронної мережі із відповідною кількістю нейронів кожного шару та фукцією активації:

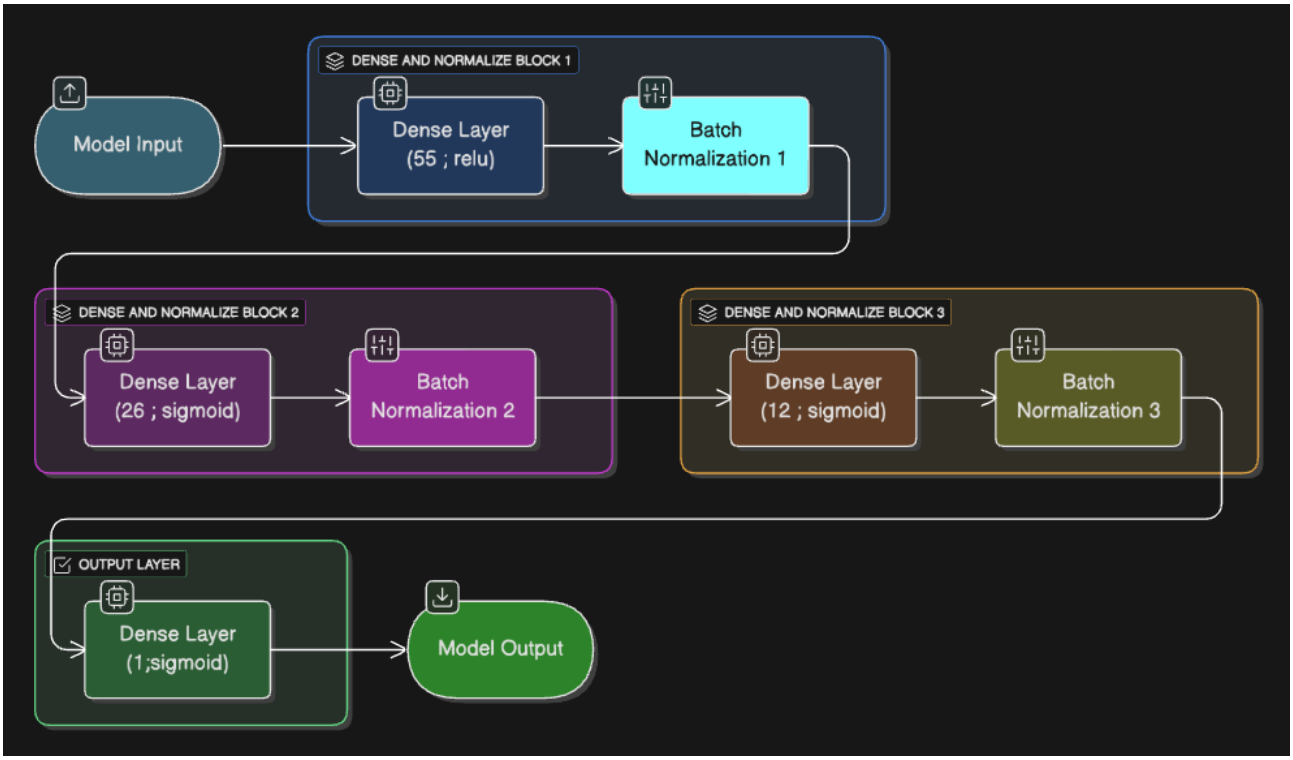


Рисунок 3.5 – Архітектура нейронної мережі

Практичну реалізацію цієї архітектури виконано за допомогою бібліотеки TensorFlow. Застосований код для побудови моделі приведено у лістингу 3.6:

Лістинг 3.6 – Реалізація архітектури багатошарового перцептрону

```
mlp = tf.keras.Sequential([
  tf.keras.Input(shape=X_train_vectorized[0].shape),
  tf.keras.layers.Dense(55, activation='relu'),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Dense(26, activation='sigmoid'),
```

Продовження лістингу 3.6

```
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Dense(12, activation='sigmoid'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```

Ця 3-рівнева архітектура є достатньо глибокою, щоб вловити складні взаємозв'язки, але при цьому і не занадто, щоб ризикувати перенавчанням і складністю навчання. Кількість нейронів у шарах вибиралась шляхом перевірки на validation-наборі: більші мережі (256→128→64) не дали суттєвого приросту якості. Обрана конфігурація виявилась збалансованою.

Для боротьби з перенавчанням у модель додано шар Batch Normalization: після кожного Dense-шару, що стабілізує розподіл активацій і пришвидшує навчання.

3.4.4 Налаштування процесу навчання

Для навчання обрано оптимізатор Adamax (модифікований адаптивний метод стохастичного градієнтного спуску) з початковим кроком навчання 0.01. Як функцію втрат встановлено binary crossentropy (крос-ентропія для бінарної класифікації.

Навчання проводилося протягом 20 епох. Кожна епоха запускалася із використанням навчання «блоками», де зміна ваг моделі реалізовувалась за результатами обробки 32 рядків даних одночасно. У лістингу 3.7 наведено фрагмент коду, що налаштовує даний процес:

Лістинг 3.7 – Налаштування процесу навчання мережі

```
mlp.compile(
    optimizer=tf.keras.optimizers.Adamax(.01),
    loss=tf.keras.losses.BinaryCrossentropy(),
```

Продовження лістингу 3.7

```
      metrics=[tf.keras.metrics.Precision(),
tf.keras.metrics.Recall(),
      tf.keras.metrics.AUC()]
    )
    mlp.fit(X_train_vectorized, y_train, epochs=20)
```

Крім основної метрики втрат, відстежувалися такі як precision, recall та AUC. Це можливо через API keras Metrics. Особливо важливою метрикою в контексті визначення шахрайства був recall, щоб він був бизьким до 100%.

3.4.5 Перевірка результатів навчання та робота з гіпер-параметрами

Після завершення тренування було зафіксовано такі показники на тестувальному наборі: точність ~99.5%, precision ~0.80, recall ~0.85, AUC ~0.95. Висока загальна точність очікувана через домінування негативного класу, більш інформативні – precision та recall для позитивного класу: 80% і 85% відповідно означають, що з усіх транзакцій, які модель позначає шахрайськими, 80% справді такими є, і вона виявляє 85% від усіх шахрайських випадків. Це досить хороший баланс. AUC 0.95 свідчить про відмінну роздільну здатність моделі між класами.

3.4.7 Інтеграція з системою

У робочому середовищі модель використовується через наявний метод predict, що на вході отримує тензор ознак, готових після попередньої обробки. Обчислений sigmoid-вихід порівнюється з встановленим порогом значення. Поріг, до речі, визначений виходячи з бажаного precision/recall на тренувальному наборі даних. В даному випадку оптимальним виявився поріг ~0.5 (класичний), що є інтуїтивним і доволі точно відображає реальне середовище. Є можливість зміщення даного порогового значення відповідно до конкретних вимог

(наприклад, якщо банк хоче більш консервативний метод роботи, можна підняти поріг для меншої кількості false-positives).

3.5 Тестування та оцінка ефективності системи

Після побудови і навчання моделі було проведено всебічне тестування системи на тестових даних і в умовах, наближених до реальних. Мета – перевірити, чи відповідає ефективність системи очікуванням, визначити її сильні та слабкі сторони, і впевнитися у надійності роботи.

3.5.1 Оцінка на тестовому наборі даних

Як зазначалося, на незалежному тестовому наборі даних модель досягла наступних середніх показників: precision (точність позитивного прогнозу) ≈ 0.8, recall (повнота виявлення) ≈ 0.85.

Це означає, коло 80% транзакцій, позначених як шахрайські, дійсно є шахрайством, а 20% виявляються хибними. Система також виявляє близько 85% усіх шахрайських операцій серед тестових даних. Решту 15% пропускає (вони не були ідентифіковані як шахрайство). AUC (ROC) ≈ 0.94. Крива помилок дуже високо над випадковою лінією, що вказує на чудову диференціацією між класами (мережа «розуміє» різницю між позитивними та негативними результатами).

Ці цифри демонструють досить високий рівень ефективності для задач, де прецеденти шахрайства дуже рідкісні. Для наглядності: якщо, скажімо, на 100 000 транзакцій припадає 100 шахрайських, дана система знайде в середньому 85 операцій, та зробить 100* (1/0.8 - 1) близько 20 хибних спрацьовувань. Тобто зі 100 тис. операцій коло 85 шахрайських схем буде визначено та 20 нормальних операцій будуть помилково позначені підозрілими, що не сильно вплине на досвід користування системою (зазвичай у подібних випадках робиться додаткова перевірка особи, що убезпечує операції). В реальності ці значення

можуть дещо змінюватися залежно від умов, але порядок дає уявлення про баланс.

3.5.2 Порівняння зі стандартними методами

Було проведено порівняння з деякими базовими підходами на тому ж тестовому наборі. Результати представлені у вигляді матриць неточностей у додатку Г

Дерева рішень, виявили 90% операцій при precision 0.7, *логістична регресія* показала precision 0.60 та recall 0.70, *random forest* (100 дерев): precision 0.75 і recall 0.8, *isolation forest* сам по собі показав precision лише 0.10, хоча recall 0.90. Тобто виявлення аномалій як самостійний критерій непридатний для використання, адже велика кількість нормальних, але унікальних сценаріїв буде блокуватися, проте, як і планувалось, даний метод корисний для підсилення результативності (високий recall) у поєднанні з точнішою основною моделлю.

3.5.3 Огляд результатів

Загалом, результати тестування показують, що розроблена система може значно покращити виявлення шахрайства у банківських операціях. Вона автоматично знаходить більшість підозрілих транзакцій та робить це з прийнятним рівнем помилкових спрацьовувань. Є місце додатковим технікам для покращення – зокрема, навчання на більшому наборі даних, оновлення ознак, можливо впровадження окремих моделей для специфічних банківських операцій (наприклад, окрема модель для кредитних заявок). Але навіть в поточному вигляді система перевершує традиційні методи і відповідає сучасним практикам «AI-driven fraud detection».

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Порядок дій ІТ-установи у разі ядерного удару

У разі виникнення загрози ядерного удару або радіаційної аварії ІТ-установа повинна діяти згідно з чітким алгоритмом, який гарантує максимальний захист життя і здоров'я персоналу, а також збереження інформаційних ресурсів та основної інфраструктури. Дані рекомендації ґрунтуються на профільних підручниках і офіційних джерелах з безпеки життєдіяльності та охорони праці.

Перш за все, при отриманні сигналу повітряної тривоги або офіційного сповіщення про загрозу ядерного удару, керівництво ІТ-установи повинно негайно організувати евакуацію всіх працівників у найближчі укриття. Працівники мають залишити робочі місця без зволікання, не намагаючись врятувати особисті речі чи техніку, якщо це може призвести до затримки. Укриття повинно бути визначене заздалегідь: це може бути підвал, спеціально обладнане сховище або внутрішня кімната без вікон, розташована якомога далі від зовнішніх стін і даху. Перед початком роботи в установі всі співробітники повинні бути ознайомлені з маршрутами евакуації та розташуванням укриттів, а також пройти відповідний інструктаж.

Якщо ситуація дозволяє, відповідальні особи мають виконати базові заходи для захисту критично важливих інформаційних систем. Необхідно оперативно вимкнути електроживлення неключового обладнання, щоб уникнути пошкодження від електромагнітного імпульсу. Ключові сервери та мережеве обладнання слід перевести на резервне живлення, підключивши їх до UPS чи генераторів. Якщо це передбачено інструкціями з безпеки, слід ініціювати резервне копіювання даних на віддалені або хмарні сховища, щоб уникнути втрати інформації. Серверні приміщення потрібно максимально ізолювати – ущільнити двері, вентиляційні отвори, вжити заходів для мінімізації проникнення пилу та радіоактивних часток.

Після прибуття в укриття слід одразу зачинити всі двері, вікна, вентиляційні отвори, вимкнути системи кондиціонування та обігрівачі. Якщо є можливість, треба перейти в кімнату без вікон. В укритті важливо дотримуватися дистанції від інших людей, щоб уникнути перехресного забруднення, і не торкатися предметів, які можуть бути забруднені радіоактивним пилом. Категорично забороняється вживати їжу чи воду, які могли бути відкритими під час вибуху або потрапити під вплив зовнішнього середовища. За першої ж можливості слід провести дезактивацію: акуратно зняти верхній шар одягу, намагаючись не торкатися шкіри забрудненими частинами, і помістити його у герметичний пакет. Всі відкриті ділянки шкіри потрібно обмити великою кількістю води з милом, уникаючи сильного тертя, особливо у місцях подряпин чи ран. Якщо води немає, можна використати вологі серветки або чисту тканину. Поверхні, до яких дотикалися, потрібно також протерти вологою ганчіркою.

Під час перебування в укритті необхідно постійно слідкувати за офіційними повідомленнями. Оскільки ймовірне відключення електропостачання та зв’язку, слід заздалегідь мати автономний радіоприймач із батарейками, який може стати єдиним джерелом інформації. Важливо не залишати укриття щонайменше 24 години або до отримання чітких інструкцій від органів влади чи рятувальних служб. Якщо установа має запас води й їжі, їх слід розподіляти раціонально, пам’ятаючи, що вихід назовні може бути небезпечним протягом кількох діб.

Після офіційного дозволу залишити укриття, відповідальні особи повинні організувати перевірку стану приміщень, провести додаткову дезактивацію поверхонь і обладнання. Необхідно оцінити можливість відновлення роботи: підключити резервні джерела живлення, перевірити цілісність даних та працездатність мережевого обладнання. Якщо основна локація зазнала значного забруднення або руйнувань, слід організувати евакуацію персоналу та критично важливого обладнання у безпечні райони та продовжити роботу через резервні офіси або хмарні сервіси.

Паралельно з технічними заходами, особливу увагу потрібно приділяти психологічному стану працівників. Регулярне проведення тренінгів та навчань з

дій у надзвичайних ситуаціях, надання інформації про алгоритм дій, забезпечення аптечками, засобами індивідуального захисту, водою, їжею та радіоприймачами значно підвищує рівень готовності колективу до екстремальних подій. Керівництво ІТ-установи має не тільки забезпечити матеріальні ресурси для захисту, а й формувати культуру безпеки та взаємної підтримки.

Дотримання цього порядку дій дозволяє максимально знизити ризики для життя персоналу, зберегти інформаційні активи та забезпечити відновлення діяльності після подолання наслідків ядерного удару чи радіаційної аварії.

4.2 Психофізіологічне розвантаження для працівників

Інтенсивна праця, особливо пов’язана з великим психоемоційним напруженням, стресами та монотонними операціями, призводить до втоми працівників і зниження їх працездатності. Накопичення нервово-психічного перенапруження негативно впливає не лише на здоров’я, але й на безпеку: на тлі втоми зростає ризик помилок і виробничого травматизму. Тому важливою складовою охорони праці є психофізіологічне розвантаження – комплекс заходів, спрямованих на відновлення нормального функціонального стану нервової та м’язової систем працівника, зменшення рівня стомлення, напруження і стресу. Законодавство України зобов’язує роботодавця забезпечувати своїм працівникам безпечні та здорові умови праці, в тому числі раціональний режим праці та відпочинку для запобігання перевтомі (ст. 153 Кодексу законів про працю, ст. 13 Закону «Про охорону праці»). Працівникам мають надаватися регламентовані перерви протягом робочої зміни, а також створюватися можливості для відпочинку. Згідно з державними санітарними нормами, на підприємствах слід передбачати спеціальні приміщення для відпочинку в робочий час та психологічного розвантаження персоналу . Наприклад, для офісів та інших приміщень, де виконуються роботи з персональними комп’ютерами, обов’язковим є обладнання *кімнати*

психологічного розвантаження. Така кімната – це спеціально оформлене приміщення, призначене для короткочасного відпочинку і зняття психоемоційного напруження. Проєктні нормативи рекомендують облаштовувати в ній зручні місця для сидіння або лежання, приглушене освітлення, спокійну кольорову гаму інтер'єру, аудіосистему для відтворення релаксуючої музики. Бажано також встановити пристрої для приготування та роздачі тонізуючих напоїв (наприклад, автомат з питною водою, чаєм, кавою) і забезпечити місце для виконання нескладних фізичних вправ або гімнастики (варто уточнити, що закон, наведений вище, було скасовано, відповідно до з цілями дерегуляції господарської діяльності, та все ж рекомендації можуть бути застосованими та нестимуть користь і зараз). Все це дає змогу працівникам під час перерви переключитися з робочих проблем, розслабитися і відновити працездатність.

Ефективним методом зняття нервово-м'язового напруження є проведення сеансів психофізіологічного розвантаження із використанням елементів аутогенного тренування. Аутогенне тренування ґрунтується на свідомому застосуванні прийомів психічної саморегуляції, поєднаних із дихальними вправами та м'язовою релаксацією (розслабленням), часто у супроводі словесного самонавіювання. Сеанси психофізіологічного розвантаження бажано проводити в спеціально обладнаній кімнаті релаксації, яка ізольована від виробничого шуму та подразників і має заспокійливий інтер'єр (наприклад, стіни пофарбовані у приємні пастельні тони – зеленкуватоблакитний, сірий, бежевий тощо, що сприяють заспокоєнню нервової системи). Оптимальна тривалість одного сеансу – 10–15 хвилин. Багаторічна практика виробила умовно три фази сеансу психологічного розвантаження: - Фаза абстрагування від обстановки. На початку сеансу працівники займають зручну позу (сидячи або напівлежачи у кріслі-реклайнері), відволікаються від виробничих питань. У приміщенні лунає тиха мелодійна музика (можливі звуки природи – шум лісу, спів птахів, прибій моря), що допомагає зняти залишкове збудження робочого дня. Протягом 2–3 хвилин учасники адаптуються до спокійної атмосфери і психологічно

налаштовуються на відпочинок. Фаза заспокоєння і релаксації. Ця основна частина сеансу відповідає фазі відновлювального гальмування в корі головного мозку, коли поступово домінують процеси гальмування, що забезпечують відпочинок. Вмикається проєктор або екран, на якому повільно змінюються заспокійливі візуальні образи – наприклад, фотографії природних пейзажів: квітучий луг, лісова галявина, гладка поверхня озера тощо. Через навушники або динаміки продовжує грати спокійна, тиха музика. На фоні музики інструктор або запрограмований аудіозапис повільним, тихим голосом промовляє формули самонавіювання (аутотренінгу), які сприяють зниженню м'язового тонусу і заспокоєнню. Наприклад, повторюються 2–3 рази фрази: "Я повністю розслаблений, спокійний. Моє дихання рівне, глибоке і легке. Все тіло приємно тепле і важке, лоб прохолодний" тощо. Під час цієї фази може використовуватися м'яке світлотерапевтичне тло – застосовують розсіяне світло зеленого спектру, яке поступово приглушується майже до повної темряви наприкінці фази релаксації . Тривалість фази глибокого розслаблення складає 5–7 хвилин. - Фаза активізації. Це заключний етап сеансу, метою якого є поступове повернення працівника до активного стану, підвищення бадьорості та мотивації до подальшої роботи. Напочатку фази підтримується майже повна темрява і тиша, щоб забезпечити максимальний відпочинок. Згодом плавно змінюються стимули: на екрані з'являється візуальний образ, що активує, – наприклад, червона пляма, яка поступово збільшується і робиться яскравішою, або сходження сонця. Одночасно гучність і темп музики наростають; обирається бадьора, енергійна мелодія. В світлотерапії застосовують зміну спектру на тепліший (може вмикатися підсвічування червоним або оранжевим світлом). На цьому фоні промовляються мобілізуючі формули аутотренінгу, які повторюються тричі у все гучнішому темпі. Перед вимовлянням кожної формули учасники роблять глибокий вдих і повільний видих. Приклади формул: "Я відпочив, я сповнений сил!", "Я бадьорий, свіжий, енергійний, маю чудовий настрій!", "Я готовий активно діяти, працювати продуктивно". Наприкінці сеансу світло в кімнаті поступово вмикається, музика затихає. Учасники

виконують кілька легких рухових вправ (потягування, повороти голови, глибокі вдихи) і повертаються до звичайного ритму діяльності.

Проведення таких сеансів 1–2 рази за робочу зміну (наприклад, в обідню перерву або після 3–4 годин роботи) сприяє суттєвому відновленню нервової системи. Після сеансів психофізіологічного розвантаження у працівників зменшується відчуття втоми, з'являється бадьорість, поліпшується настрій, суб'єктивно підвищується мотивація до роботи. В дослідженнях відзначено, що регулярне застосування програм психологічного розвантаження позитивно впливає на продуктивність праці та психічне благополуччя персоналу підприємств. Таким чином, роботодавцю доцільно впроваджувати заходи для зниження стресу і напруження серед працівників, особливо в умовах інтенсивної, монотонної або емоційно важкої роботи.

Окрім спеціальних сеансів релаксації, існують і індивідуальні заходи психофізіологічного розвантаження, які кожен працівник може практикувати самостійно протягом робочого дня. До них належать: регламентовані мікроперерви через кожні 1–2 години роботи (бажано з невеликою розминкою або гімнастикою для очей і тіла), дихальні вправи для зняття напруги, короткі вправи виробничої гімнастики (нахили, потягування, обертання плечима тощо) на робочому місці. Позитивний ефект дає фізична активність під час обідньої перерви – прогулянка на свіжому повітрі, легка зарядка. Важливо також формувати здоровий психологічний клімат у колективі: доброзичливі стосунки, взаємопідтримка і відсутність травмуючих конфліктів помітно знижують рівень стресу. Психологи радять працівникам для профілактики емоційного вигорання дотримуватися простих правил: планувати час для відпочинку, робити регулярні короткі паузи під час роботи, приділяти увагу сім'ї та друзям (відчувати підтримку оточення), знаходити час для улюблених занять і хобі поза роботою. Важливо розмежовувати робочий і особистий час – не "приносити роботу додому" без крайньої необхідності. Крім того, варто періодично переглядати власні трудові цілі та досягнення, знаходити позитивний сенс своєї роботи –

усвідомлення корисності своєї діяльності підвищує стійкість до стресів і мотивує долати тимчасові труднощі.

Отже, психофізіологічне розвантаження працівників є необхідною складовою системи збереження здоров'я і працездатності персоналу. На сучасних підприємствах впроваджуються різні засоби і методи для зняття нервово-емоційного напруження: від спеціально обладнаних кімнат відпочинку до тренінгів з управління стресом. Використання таких заходів у комплексі з раціональною організацією праці (оптимізацією робочого навантаження, чергуванням роботи і відпочинку) та дотриманням гігієнічних норм робочого середовища дозволяє мінімізувати шкідливий вплив виробничих стресорів. Як результат – підвищується рівень безпеки праці, продуктивність та загальна якість життя працівників, що є головною метою системи управління охороною праці.

ВИСНОВКИ

У роботі було виконано дослідження та розроблено систему для виявлення шахрайських банківських операцій з використанням методів машинного навчання. Проведений аналіз підтвердив актуальність використання технологій машинного навчання для протидії банківському шахрайству, що є значною загрозою для фінансових установ та їхніх клієнтів.

У першому розділі розглянуто теоретичні основи шахрайських операцій у банківських системах. Доведено, що шахрайство в фінансовій сфері характеризується високою складністю та адаптивністю, що вимагає постійного вдосконалення методів його виявлення. Розглянуто сучасні підходи, які використовуються у банківському секторі для боротьби з шахрайством, включаючи статистичні, ймовірнісні та регресійні методи.

Другий розділ присвячено аналізу та порівнянню різних методів машинного навчання, зокрема навчання «із вчителем» та «без вчителя», а також глибокого навчання. Виявлено, що методи «із вчителем» (Random Forest, XGBoost) мають високу ефективність у виявленні відомих типів шахрайства, тоді як методи навчання без вчителя (Isolation Forest, автоенкодери) є незамінними для ідентифікації невідомих та нових схем шахрайства.

На основі проведеного порівняння було обрано гібридний підхід, який поєднує наглядувані та безнаглядні методи. Такий підхід дозволяє максимально ефективно використовувати переваги обох типів навчання.

У третьому розділі розроблено і реалізовано практичну модель виявлення шахрайства з використанням Python та TensorFlow. В ході підготовки даних було здійснено їх очищення, нормалізацію та трансформацію. Розроблена модель нейронної мережі продемонструвала високу точність (precision) та повноту виявлення (recall), що підтверджено тестовими результатами. Практичні експерименти показали, що використання нейронних мереж для аналізу даних про транзакції значно зменшує кількість помилкових спрацьовувань і забезпечує ефективне виявлення шахрайських операцій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Sharma, S. (2024). AI and fraud detection in banking. Nature Humanities and Social Sciences Communications.
2. Practical Business Skills. (б. д.). Security and fraud. https://www.practicalbusinessskills.org/en/getting-started/taking-your-business-digital/security-and-fraud.html
3. IBM. (б. д.). AI and fraud detection in banking. https://www.ibm.com/think/topics/ai-fraud-detection-in-banking
4. PwC Colombia. (б. д.). Encuesta de crimen y fraude econ√≥mico. https://www.pwc.com/co/es/publicaciones/encuesta-crimen-fraude-economico.html
5. Tookitaki. (б. д.). Fraud detection using machine learning in banking. https://www.tookitaki.com/compliance-hub/fraud-detection-using-machine-learning-in-banking-1
6. Brankas. (б. д.). Fraud detection algorithms in banking. https://blog.brankas.com/fraud-detection-algorithms-in-banking
7. International Journal of Research Publication and Reviews. (2023). Fraud detection in banking. https://ijrpr.com/uploads/V6ISSUE4/IJRPR41586.pdf
8. Online Scientific Research. (б. д.). Fraud detection in banking: A machine learning approach using credit card transaction data. https://www.onlinescientificresearch.com/articles/fraud-detection-in-banking-a-machine-learning-approach-using-credit-card-transaction-data.html
9. Agarwal, P. (2024, Квітень). Here's how Stripe detects frauds with a 99% accuracy [LinkedIn post].
10. Stripe. (б. д.). How we built it: Stripe Radar. https://stripe.com/blog/how-we-built-it-stripe-radar
11. scikit-learn developers. (б. д.). Clustering. https://scikit-learn.org/stable/modules/clustering.html#k-means
12. scikit-learn developers. (б. д.). sklearn.cluster.DBSCAN. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

13. Towards Data Science. (2019, 10 жовтня). Anomaly detection with autoencoder. https://towardsdatascience.com/anomaly-detection-with-autoencoder-b4cdce4866a6

14. scikit-learn developers. (б. д.). sklearn.ensemble.IsolationForest. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html

15. scikit-learn developers. (б. д.). sklearn.svm.OneClassSVM. https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html

16. IBM. (б. д.). Graph neural network. https://www.ibm.com/think/topics/graph-neural-network

17. Teuwens, R. (б. д.). Anomaly detection with auto-encoders [Code notebook]. Kaggle.

18. Zhou, C., & Paffenroth, R. (2019). Anomaly detection with robust deep autoencoders. arXiv. https://arxiv.org/pdf/1906.11632

19. Detthamrong, U., Chansanam, W., Boongoen, T., & Iam-On, N. (2024). Enhancing Fraud Detection in Banking using Advanced Machine Learning Techniques. Econ Journals. https://doi.org/10.32479/ijefi.16613

20. Kartik2112. (б. д.). Fraud detection [Dataset]. Kaggle. https://www.kaggle.com/datasets/kartik2112/fraud-detection

21. Kovalchuk, O., Karpinski, M., Banakh, S., Kasianchuk, M., Shevchuk, R., & Zagorodna, N. (2023). Prediction machine learning models on propensity convicts to criminal recidivism. Information, 14(3), art. no. 161, 1-15.

22. Karpinski, M., Korchenko, A., Vikulov, P., Kochan, R., Balyk, A., & Kozak, R. (2017, September). The etalon models of linguistic variables for sniffing-attack detection. In 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems : Technology and Applications (IDAACS) (Vol. 1, pp. 258-264). IEEE.

23. Augmented Reality Enhanced Learning Tools Development for Cybersecurity Major Zagorodna N., Skorenkyy Y., Kunanets N., Baran I., Stadnyk M (2022) CEUR Workshop Proceedings, 3309 , pp. 25-32.

24. Boltov, Y., Skarga-Bandurova, I., & Derkach, M. (2023, September). A Comparative Analysis of Deep Learning-Based Object Detectors for Embedded Systems. In 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS) (Vol. 1, pp. 1156-1160). IEEE.

25. Kulchytskyi, T., Rezvorovych, K., Povalena, M., Dutchak, S., & Kramar, R. (2024). LEGAL REGULATION OF CYBERSECURITY IN THE CONTEXT OF THE DIGITAL TRANSFORMATION OF UKRAINIAN SOCIETY. Lex Humana (ISSN 2175-0947), 16(1), 443-460.

26. Андрейчук, Н. І., Кіт, Ю. В., Шибанов, С. В., & Шерстньова, О. В. (2012). Охорона праці. Occupational safety (276 с.). Видавництво Львівської політехніки.

27. Атаманчук, П. С., Мендерецький, В. В., Панчук, О. П., & Чорна, О. Г. (2011). Безпека життєдіяльності [Навчальний посібник] (276 с.). Київ: Центр учбової літератури.

28. Сокуренко, В. В. (ред.). (2021). Безпека життєдіяльності та охорона праці [Підручник] (308 с.; ISBN 978-966-610-248-8). Харків: Харківський національний університет внутрішніх справ.

29. Бедрій, Я. І. (2018). Основи охорони праці. [Підручник] (240 с.). Тернопіль: Навчальна книга – Богдан.

30. Гогіташвілі, Г. Г., & Лапін, В. М. (2018). Основи охорони праці. [Підручник] (121 с.). Київ: Знання.

31. Управління Держпраці у Тернопільській області. (б. д.). Робота в офісі: основні санітарно-гігієнічні вимоги.

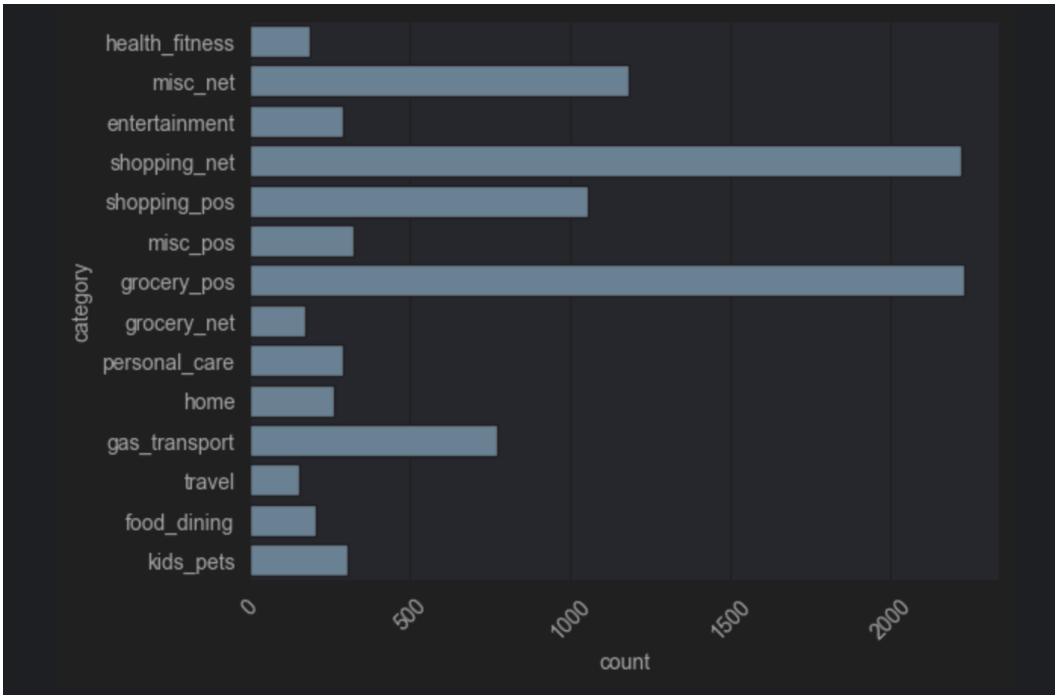
32. Міністерство охорони здоров’я України. (1999). Державні санітарні норми мікроклімату виробничих приміщень (ДСН 3.3.6.042-99).

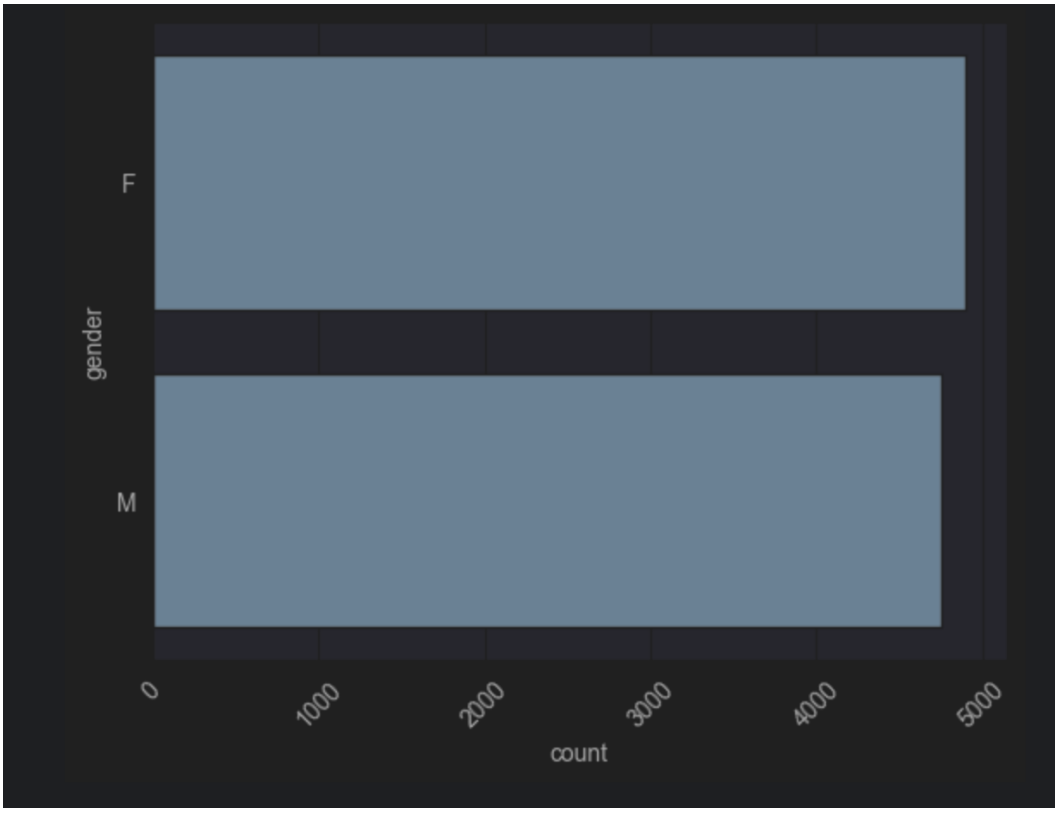
33. Верховна Рада України. (2025). Кодекс законів про працю України (станом на 08.04.2025).

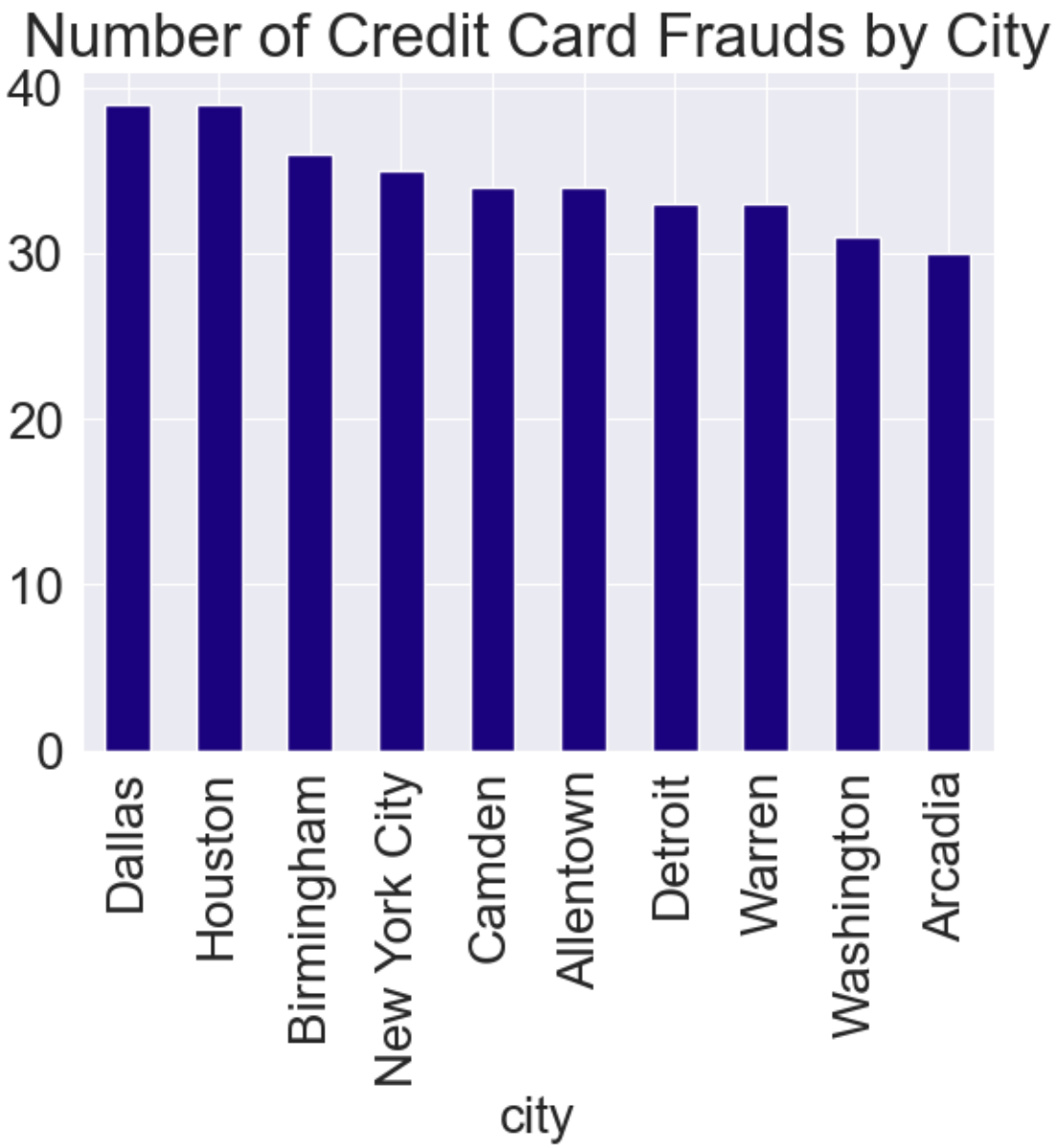
Додаток А Структура вхідних даних датасету

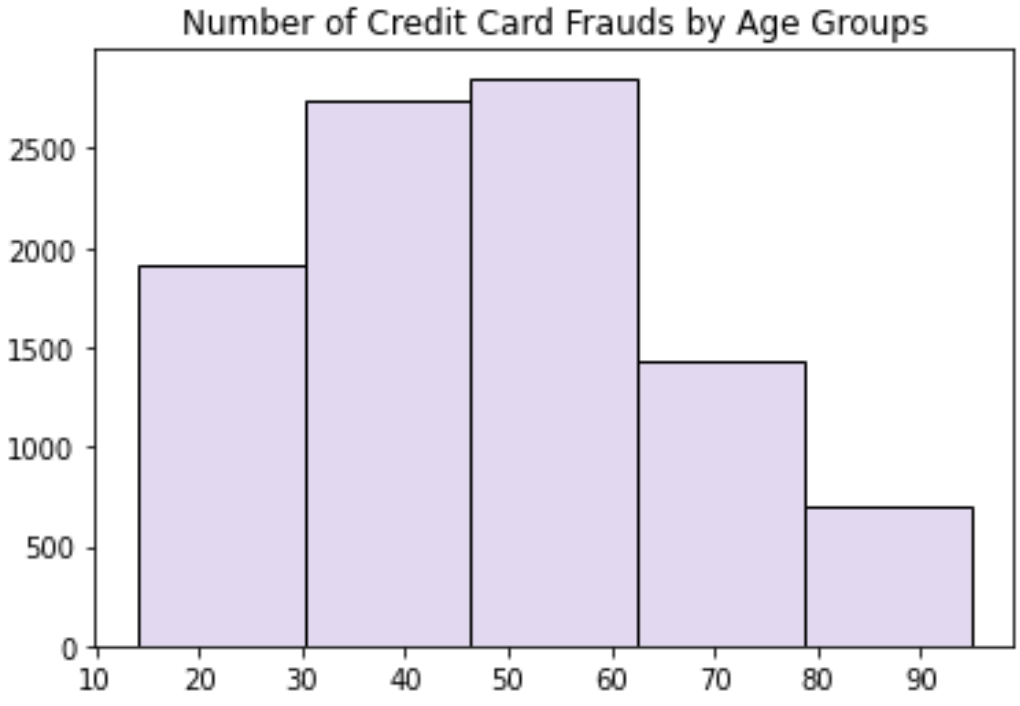
```
Data columns (total 23 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   Unnamed: 0             555719 non-null  int64
 1   trans_date_trans_time  555719 non-null  object
 2   cc_num                 555719 non-null  int64
 3   merchant               555719 non-null  object
 4   category               555719 non-null  object
 5   amt                    555719 non-null  float64
 6   first                  555719 non-null  object
 7   last                   555719 non-null  object
 8   gender                 555719 non-null  object
 9   street                 555719 non-null  object
 10  city                   555719 non-null  object
 11  state                  555719 non-null  object
 12  zip                    555719 non-null  int64
 13  lat                    555719 non-null  float64
 14  long                   555719 non-null  float64
 15  city_pop               555719 non-null  int64
 16  job                    555719 non-null  object
 17  dob                    555719 non-null  object
 18  trans_num              555719 non-null  object
 19  unix_time              555719 non-null  int64
 20  merch_lat              555719 non-null  float64
 21  merch_long             555719 non-null  float64
 22  is_fraud               555719 non-null  int64
dtypes: float64(5), int64(6), object(12)
```

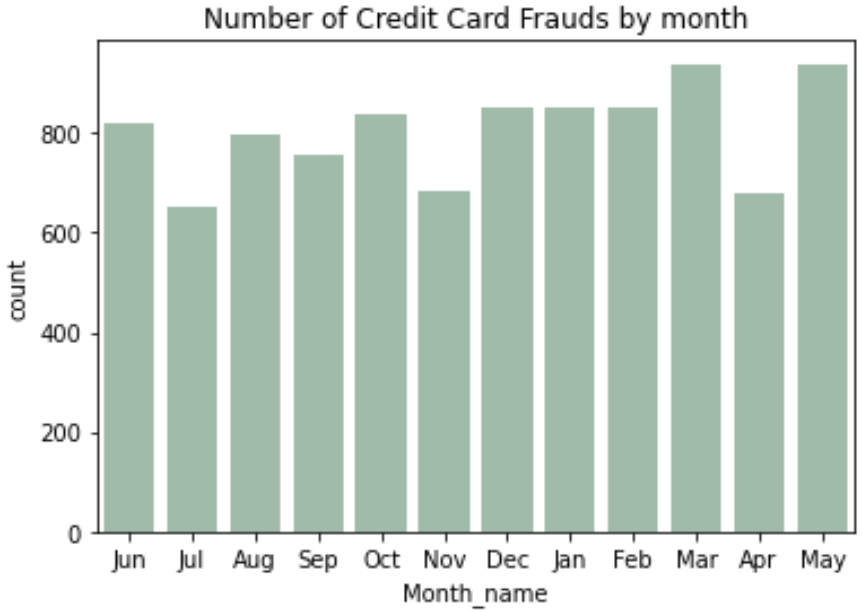
Додаток Б Кореляція різних полів даних із мітками шахрайства із набору

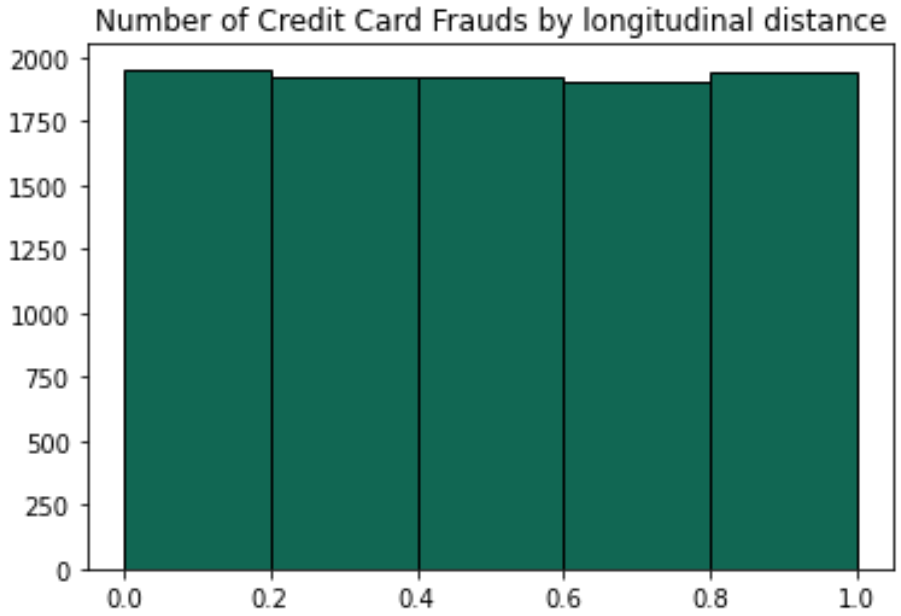


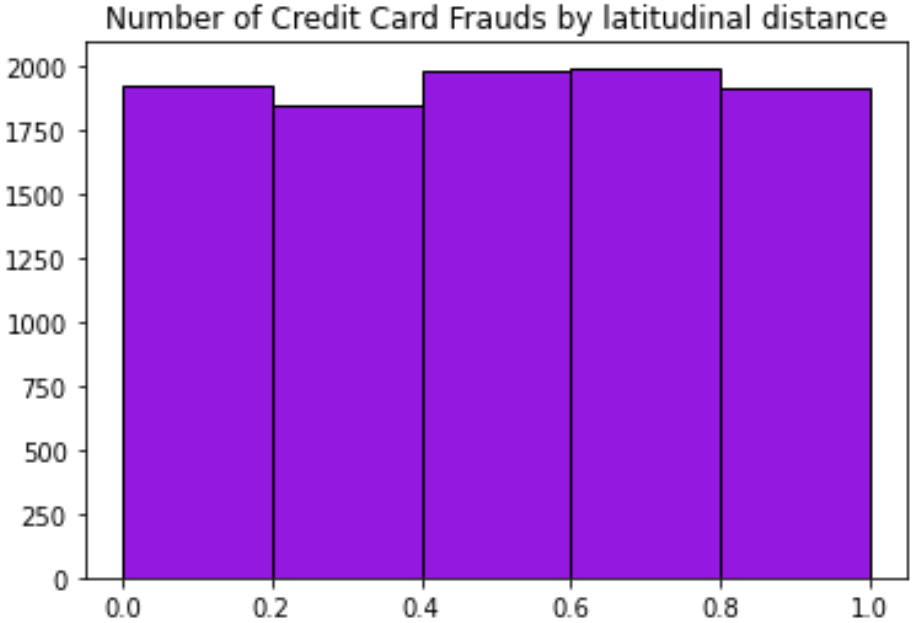



Number of Credit Card Frauds by City
40
30
20
10
0
Dallas
Houston
Birmingham
New York City
Camden
Allentown
Detroit
Warren
Washington
Arcadia
city


Number of Credit Card Frauds by Age Groups
2500
2000
1500
1000
500
0
10
20
30
40
50
60
70
80
90


Number of Credit Card Frauds by month
count
0
200
400
600
800
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Month_name

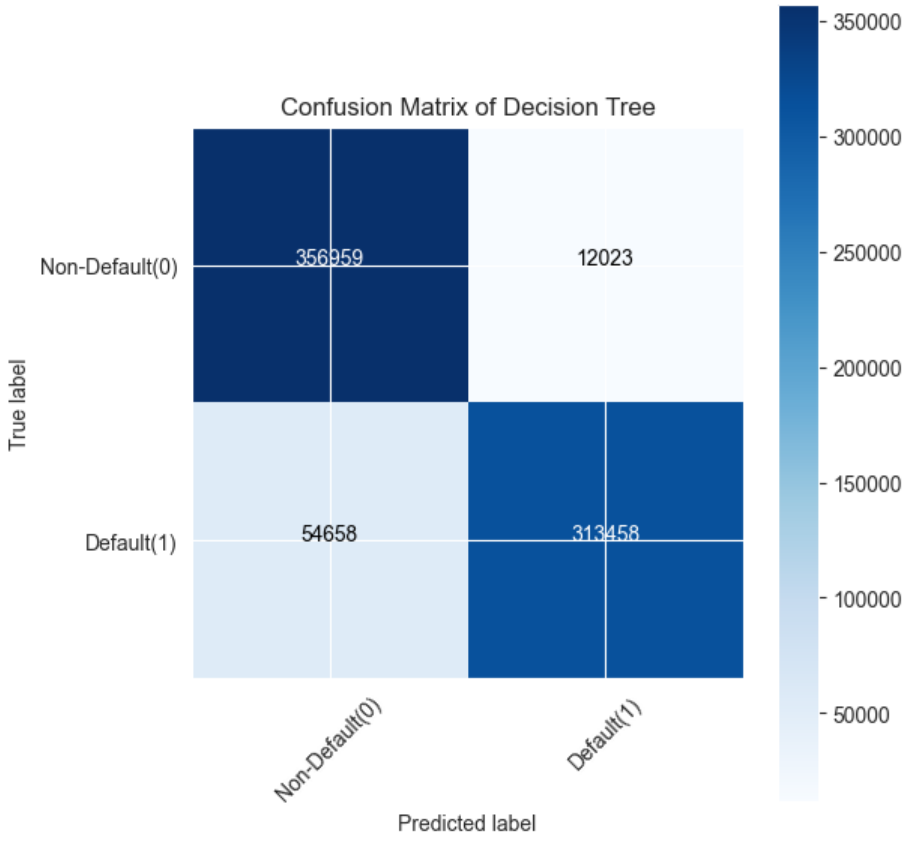

Number of Credit Card Frauds by longitudinal distance
0
250
500
750
1000
1250
1500
1750
2000
0.0
0.2
0.4
0.6
0.8
1.0

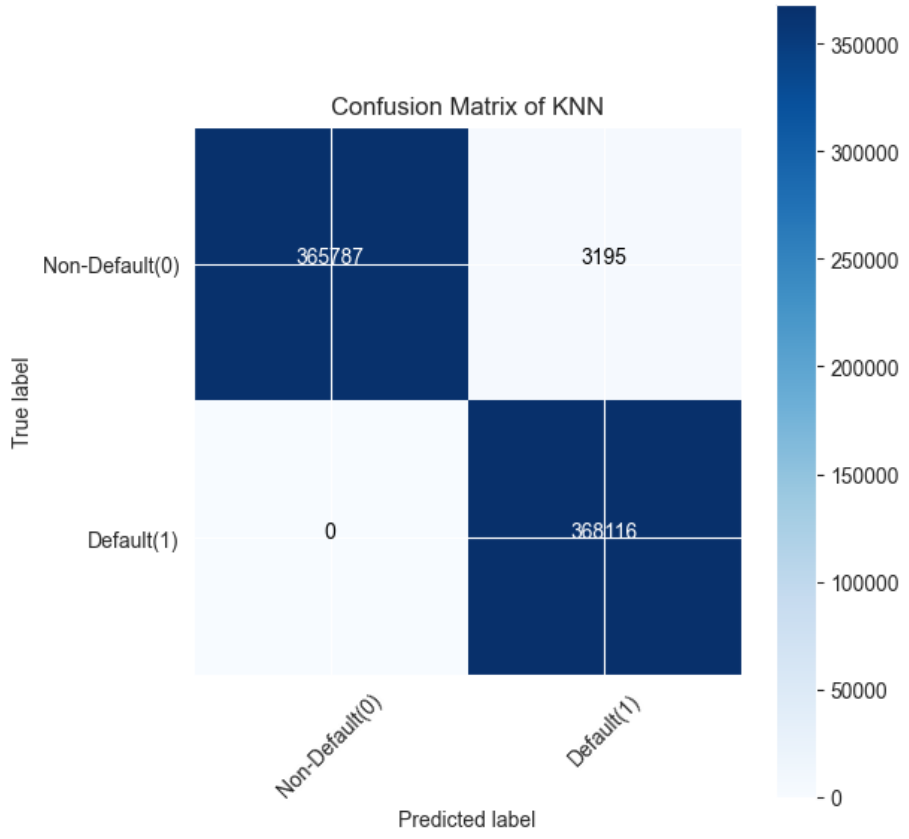

Number of Credit Card Frauds by latitudinal distance
0
250
500
750
1000
1250
1500
1750
2000
0.0
0.2
0.4
0.6
0.8
1.0

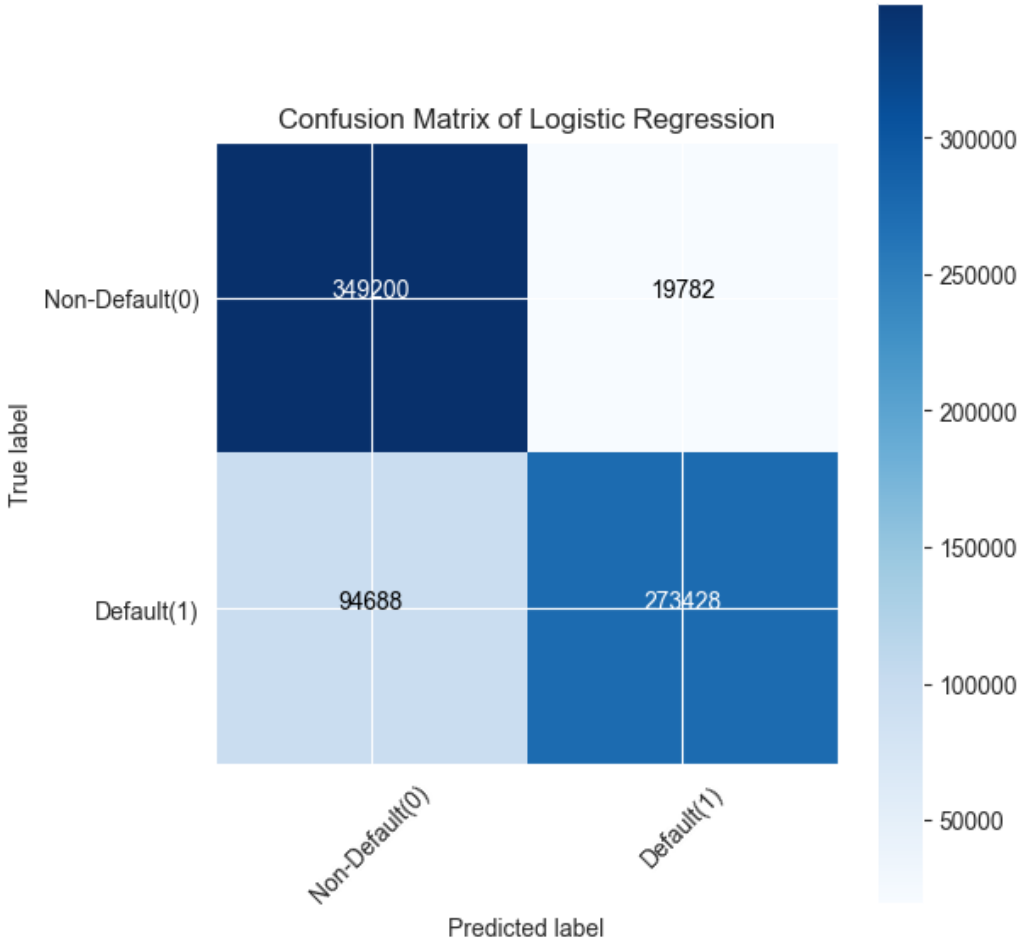
```
    # Time Features

    total['trans_date_trans_time'] =
pd.to_datetime(total['trans_date_trans_time'])
    total['trans_date'] =
total['trans_date_trans_time'].dt.strftime('%Y-%m-%d')
    total['trans_date'] = pd.to_datetime(total['trans_date'])

    total['dob'] = pd.to_datetime(total['dob'])

    total["age"] = total["trans_date"] - total["dob"]
    total["age"] = round(total["age"].dt.days /
365.25).astype('int32')

    total['trans_month'] =
pd.DatetimeIndex(total['trans_date']).month
    total['trans_year'] =
pd.DatetimeIndex(total['trans_date']).year

    # Geo features

    total['latitudinal_distance'] = abs(round(total['merch_lat'] -
total['lat'], 3))
    total['longitudinal_distance'] = abs(round(total['merch_long']
- total['long'], 3))

    # Categorization

    total['merchant'] = total['merchant'].map(lambda merchant:
merchant.replace('fraud_', ''))
    merchants_unique = total['merchant'].unique()
    total['merchant'] = total['merchant'].map(
        lambda merchant: np.where(merchants_unique ==
merchant)[0][0]
```

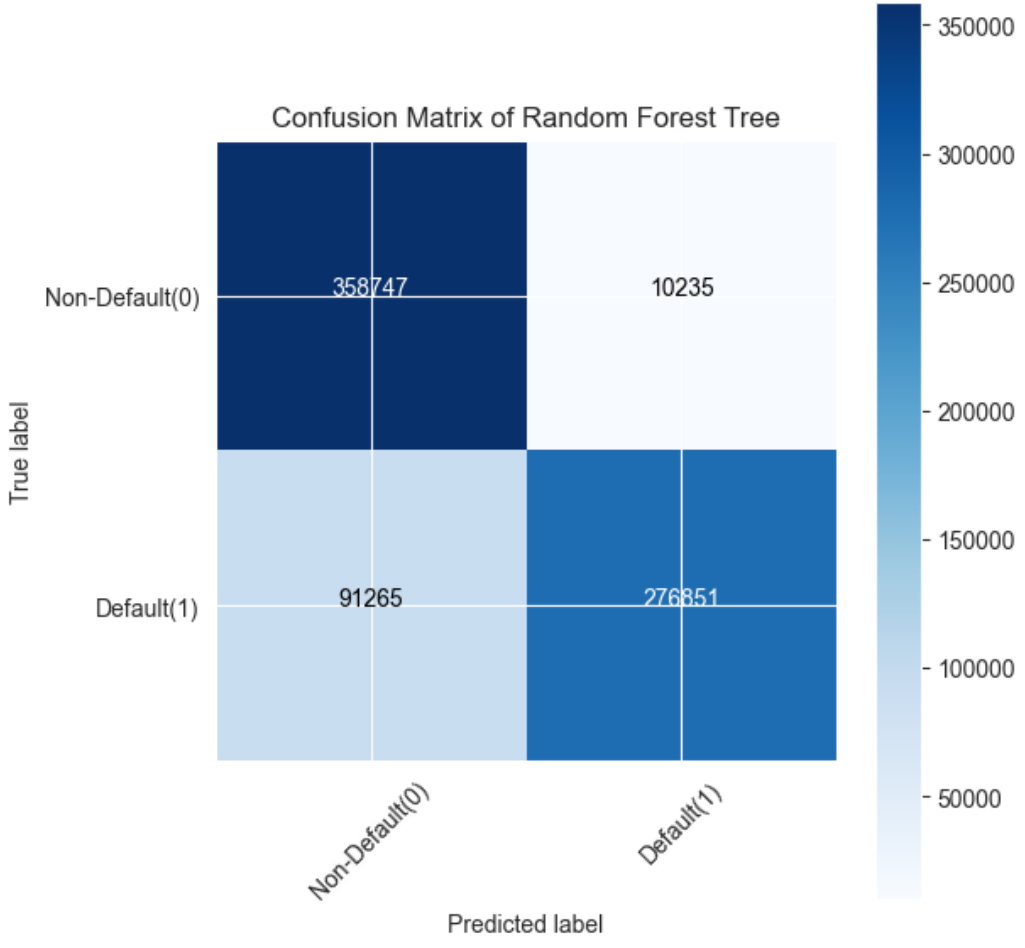
```
    ).astype('int32')

    states_unique = total['state'].unique()
    total['state'] = total['state'].apply(
        lambda state: np.where(states_unique == state)[0][0]
    ).astype('int16')

    # One-hot encoding

    total = pd.get_dummies(total, columns=['category', 'gender'],
drop_first=True)
```

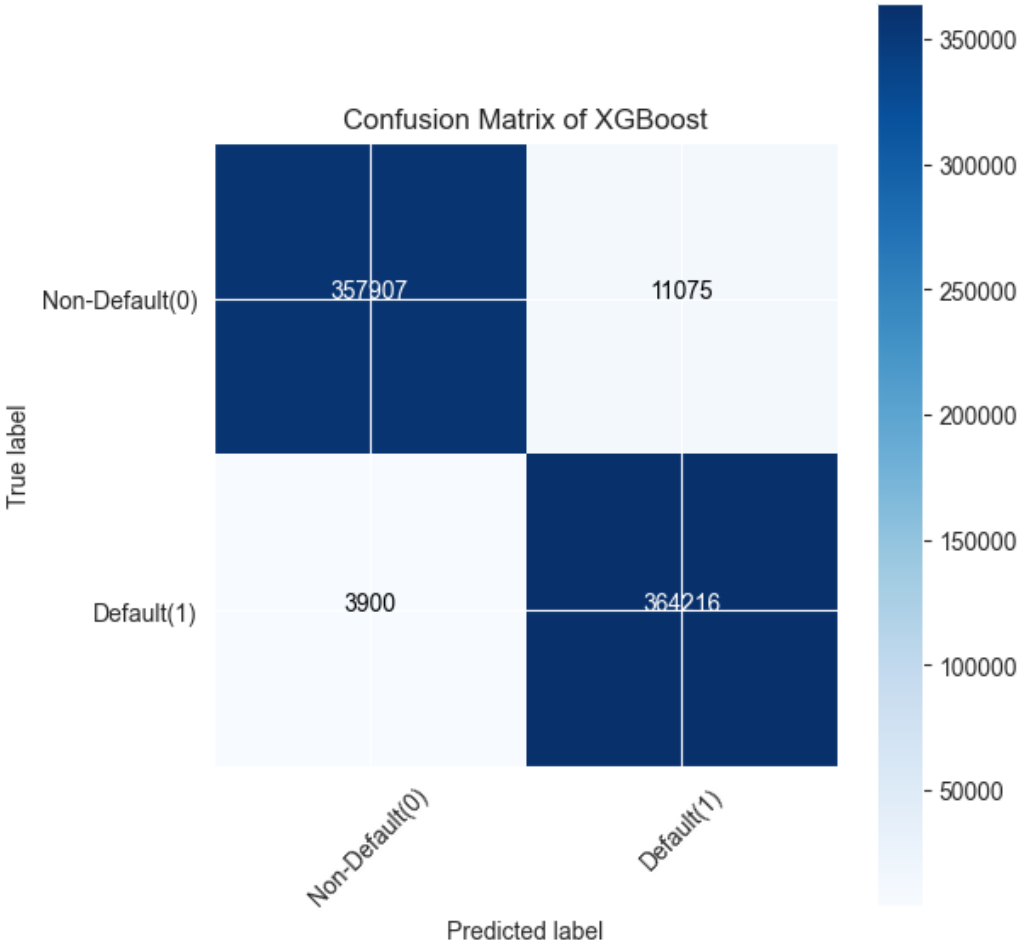
Додаток Г Порівняння матриць неточностей для виявлення шахрайства

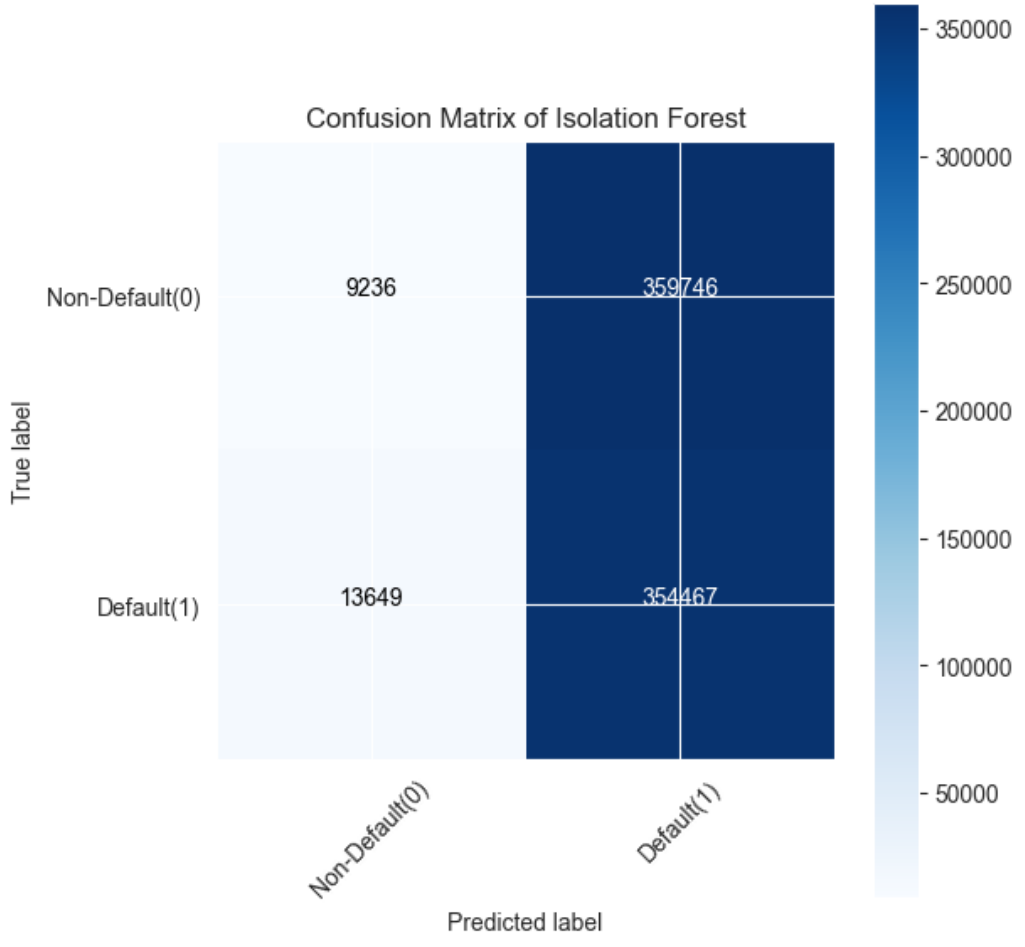


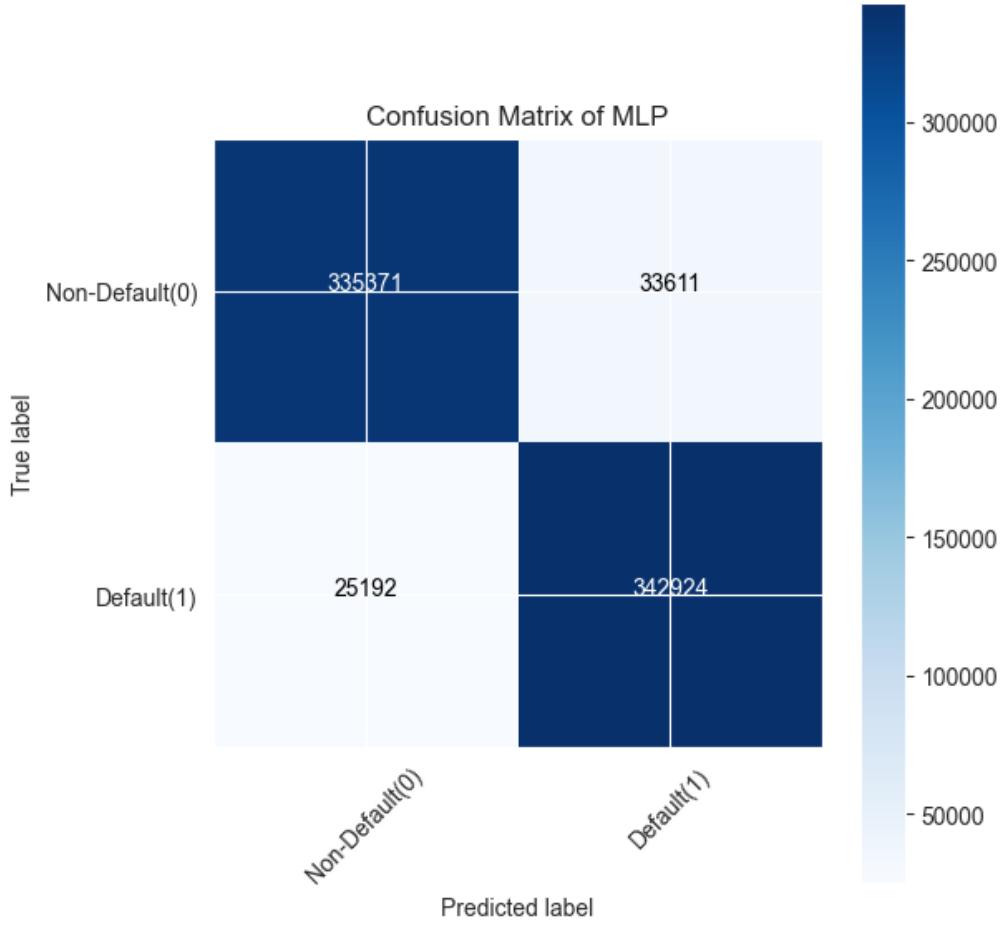



Confusion Matrix of Logistic Regression
Non-Default(0)
Default(1)
True label
349200
19782
94688
273428
Non-Default(0)
Default(1)
Predicted label
300000
250000
200000
150000
100000
50000


Confusion Matrix of Random Forest Tree
Non-Default(0)
Default(1)
True label
358747
10235
91265
276851
Non-Default(0)
Default(1)
Predicted label
350000
300000
250000
200000
150000
100000
50000


Confusion Matrix of XGBoost
Non-Default(0)
Default(1)
True label
357907
11075
3900
364216
Non-Default(0)
Default(1)
Predicted label
350000
300000
250000
200000
150000
100000
50000


Confusion Matrix of Isolation Forest
Non-Default(0)
Default(1)
True label
9236
359746
13649
354467
Non-Default(0)
Default(1)
Predicted label
350000
300000
250000
200000
150000
100000
50000


Confusion Matrix of MLP
Non-Default(0)
Default(1)
True label
335371
33611
25192
342924
Non-Default(0)
Default(1)
Predicted label
300000
250000
200000
150000
100000
50000