КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 – Комп'ютерні науки,
освітня програма «Інформаційна аналітика та впливи»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

"Автоматизовані інформаційні системи аналізу та виявлення шахрайства у платіжних сервісах"

Студента 2-го курсу групи ІАВ-21

Коваля Богдана Сергійовича
(прізвище, ім'я, по батькові)

(підпис студента)

Науковий керівник:

доктор технічних наук, доцент
(науковий ступінь, вчене звання)

Хлевна Юлія Леонідівна
(прізвище, ім'я, по батькові)

(дата) (підпис)

Попередній захист:

(Висновок: «До захисту в Екзаменаційній комісії»)

Завідувач кафедри
технологій управління _____ Морозов В.В. _____
(підпис) (прізвище, ініціали) (дата)

Київ – 2021

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій

Кафедра технологій управління
Освітньо-кваліфікаційний рівень Магістр
Спеціальність 122 - Комп'ютерні науки
Освітня програма Інформаційна аналітика та впливи

ЗАТВЕРДЖУЮ
Завідувач кафедри
професор Морозов В.В.

«____» _____20___року

З А В Д А Н Н Я
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Студент _Коваль Богдан Сергійович_____

Група_ІАВ-21_____

1. Тема кваліфікаційної роботи _Автоматизовані інформаційні системи аналізу та виявлення шахрайства у платіжних сервісах _____
_____

Затверджена наказом по від «_9_»_листопада_2020 р. № _4_.

2. Строк подання студентом готової роботи – "_5_" _травня___ 2021 р.

3. Цільова установка та вихідні дані до роботи _побудова програмної технології виявлення шахрайства в платіжних сервісах із динамічним набором даних (який отримується в режимі реального часу), реалізованої за допомогою мови програмування Python та допоміжних бібліотек і впроваджена із використанням хмарних обчислень та інфраструктури AWS____

4. Зміст роботи _дослідження існуючих математичних інструментів та підходів виявлення____
(перелік питань, що підлягають розробці)

аномалій у великих наборах даних; програмна імплементація відповідного математичного апарату до вирішення задачі класифікації шахрайських транзакцій в платіжних сервісах та її апробація; реалізація технології виявлення шахрайства в платіжних сервісах із використанням сформованих програмних модулів; впровадження технології із використанням хмарних обчислень_____

5. Перелік графічного матеріалу (слайдів) __представлено 32 рисунки, 16 формул, 2 таблиці, 6 додатків та презентація із 25 слайдів, що демонструють формулювання технології виявлення шахрайства в платіжних сервісах, її архітектуру, програмний код, візуалізацію процесу роботи програмного модуля із вхідними даними транзакцій, отримання результатів, тощо. _____

6. Календарний план виконання роботи:

№ з/п	Назва частин роботи	%	Виконання роботи	
			За планом	Фактично
1.	Вибір теми дипломної роботи	2	01.10.2020	01.10.2020
2.	Протокол кафедри ТУ про затвердження тем дипломних робіт та призначення наукових керівників	1	09.11.2020	09.11.2020
3.	Формування переліку нормативних матеріалів, літератури з проблематики дипломної роботи	5	11.01.2021	11.01.2021
4.	Розробка розгорнутого плану кваліфікаційної роботи	3	15.01.2021	14.01.2021
5.	Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін.	4	19.01.2021	18.01.2021
6.	Постановка задачі класифікації та її адаптації до умов предметної галузі (платіжних систем)	3	29.01.2021	28.01.2021
7.	Вивчення наукової літератури по темі та наявних рішень, їх аналіз	8	12.02.2021	12.02.2021
8.	Дослідження математичного апарату та підходів до вирішення задач класифікації, їх застосування до виявлення аномалій у великих наборах даних	10	26.02.2021	26.02.2021
9.	Формування математичних методів та моделей, які будуть основою алгоритмічного ядра технології виявлення шахрайства в платіжних сервісах	12	11.03.2021	11.03.2021
10.	Вивчення та затвердження комплексу програмних інструментів (бібліотек, інфраструктури), за допомогою яких буде реалізовано програмну імплементацію математичних моделей	5	18.03.2021	18.03.2021
11.	Підготовка бази даних транзакцій та інфраструктури до реалізації програмного рішення	4	24.03.2021	23.03.2021
12.	Реалізація програмної технології виявлення шахрайства в платіжних сервісах	12	7.04.2021	7.04.2021
13.	Апробація використаних методів та моделей, їх порівняння	5	12.04.2021	12.04.2021
14.	Створення програмного продукту на базі отриманої технології	9	21.04.2021	21.04.2021
15.	Концепт впровадження програмного продукту виявлення шахрайства в платіжних сервісах за допомогою сучасних технологій та зважаючи на вимоги до навантажених інформаційних систем	8	30.04.2021	30.04.2021
16.	Оформлення кваліфікаційної роботи. Підготовка висновків і пропозицій	5	04.05.2021	04.05.2021
17.	Передача кваліфікаційної роботи науковому керівникові	1	05.05.2021	05.05.2021
18.	Передача кваліфікаційної роботи рецензенту для рецензування	1	07.05.2021	07.05.2021
19.	Попередній захист кваліфікаційної роботи	1	11.05.2021	11.05.2021

20.	Подача готової роботи на кафедру	1	20.05.2021	20.05.2021

Дата видачі завдання «_1_» _жовтня_____ 2020 р.

Керівник роботи _д.т.н., доцент Хлевна Юлія Леонідівна_____
(посада, прізвище, ім’я, по батькові)

_____
(підпис)

Завдання прийняв до виконання студент групи _ІАВ-21_

_Коваль Богдан Сергійович_____
(прізвище, ім’я, по батькові)

_____
(підпис)

ЗМІСТ

АНОТАЦІЯ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра технологій управління
Спеціальність 122 - Комп'ютерні науки,
освітня програма "Інформаційна аналітика та впливи"

Дипломна робота магістра Коваля Богдана Сергійовича.

Тема роботи – «Автоматизовані інформаційні системи аналізу та виявлення шахрайства у платіжних сервісах».

Мета дипломної роботи магістра – розробити програмну технологію виявлення шахрайства в платіжних сервісах на основі математичних моделей та методів виявлення аномалій у великих наборах даних із подальшим впровадженням програмного продукту на її основі.

Об'єкт дослідження – платіжні сервіси, а саме математичне моделювання та програмна імплементація їх операційних процеси та результатів функціонування.

Предмет дослідження – методики вирішення задач класифікації (виявлення аномалій) у великих наборах незбалансованих даних та їх програмна реалізація, інфраструктура програмного продукту із застосуванням хмарних обчислень.

Наукова новизна роботи – реалізовано та апробовано програмну імплементацію технології виявлення шахрайства в платіжних системах, що базується на мові програмування Python, її інструментарію та опирається на математичні методи та моделі виявлення відхилень у великих наборах даних, зокрема на регресійні та дискретні математичні методи, а також штучні нейронні мережі. Визначено ефективність та особливості застосування отриманих моделей в умовах незбалансованого набору даних та запропоновано концепт автоматизованої ІС виявлення шахрайства в платіжних сервісах, що здатний оперувати в умовах високого навантаження та доступності.

У роботі досліджуються існуючий математичний інструментарій та підходи до вирішення задач класифікації, із використанням яких формуються математичні моделі та методи виявлення шахрайських транзакцій у платіжних сервісах. На основі

отриманого математичного апарату побудована програмна реалізація технології виявлення шахрайства із використанням мови програмування Python. Створено концепт впровадження отриманої програмної технології як основи для програмного продукту, що може оперувати в умовах високих навантажень та режимі реального часу.

Дипломна робота складається зі вступу, основної частини, яка включає чотири розділи, висновків, списку використаних джерел та додатків. Всього налічує 108 сторінок та перелік посилань з 88 джерела на 10 сторінках.

Ключові слова: наука про дані, машинне навчання, бінарна класифікація, виявлення аномалій, хмарні обчислення, програмний продукт.

ВСТУП

Розвиток технологій трансформує фінансові ринки. Автономність клієнтів фінансових установ за рахунок розширення спектру інструментів здійснення транзакцій різних рівнів увійшли у повсякденне життя. З позиції фінансових установ наслідком такої автономності спостерігається оперативніше та якісніше здійснення операцій, зниження витрат та ризиків пов'язаних із збереженням та транспортуванням готівки. Клієнти ж отримують зручність і простоту процедур транзакцій у режимі реального часу. Разом з тим, існує динамічна проблема пов'язана із збереженням цілісності та істинності транзакцій, проблема шахрайства. Фінансові установи, у разі шахрайських дій із їхніми клієнтами мають фінансові збитки, зниження лояльності клієнтів та їх втрату. Так, в Україні в 2016 році збитки від шахрайських операцій були близько 330 млн. грн., зараз, у 2020, - близько 1 млрд. грн. - це свідчить, що наявні системи не є ефективними. Також Україна випереджає майже на 50% США за відсотком шахрайських операцій (на 10 тис. транзакцій) - і цей відрив тільки збільшується, що свідчить, що не тільки сфера загалом потребує нових рішень, а й особливо вони необхідні в Україні. Для унеможливлення цього потрібна трансформація підходів та засобів до моніторингу, виявлення та контролю незаконних дій. Звичайно, найкращим методом боротьби із шахрайством є його попередження. Таке попередження можливе за рахунок розробки системи в основі якої є прогнозування шахрайських дій на рівні виникнення підозрілої транзакцій та прогнозування ймовірності її виникнення [18, 79].

Більшість шахрайських операцій не виявляються відразу, а інша частина не виявляється взагалі. Як правило, про шахрайство банк повідомляє сам клієнт, який бачить, що на його рахунку відбувались операції не за його участі. Після повідомлення про шахрайство банк перевіряє достовірність наданих даних і випадку підтвердження, що операція дійсно була шахрайською, повертає кошти клієнту або вирішує питання інакше. Навіть у найкращому для клієнта варіанті

він втратить час й зусилля для повернення коштів. У інших випадках він може не дочекатись своїх коштів взагалі. Для уникнення описаних ситуацій слід виявляти шахрайство ще на етапі проведення транзакції, щоб у раз його наявності заморозити рахунок, вжити відповідні міри й зберегти кошти клієнта.

У сучасному, інформаційному світі дані нагромаджуються у надвеликих обсягах, особливо дані фінансової історії банківського рахунку, адже кожен день здійснюються мільярди операцій з кредитними картами. При проведенні транзакції збирається досить широка її характеристика, починаючи від базових показників, як наприклад час проведення чи сума транзакцій, так й досить специфічних, наприклад час вводу ПІН-коду та територіальне розташування терміналу.

Завдяки досягненням у сфері інформатизації стало можливим аналізувати дані у режимі реального часу, під час їх надходження до системи. Варто відзначити методи машинного навчання для вирішення задач класифікації, які й будуть використовуватись в даній роботі. Сучасні ЕОМ дозволяють проводити мільярди ітерацій в секунду, таким чином застосовуючи досить складні інструменти машинного навчання (які базуються на затратних математичних розрахунках) у режимі реального часу.

В Україні у 2020 році збитки від бізнесових шахрайських операцій становили понад 1 млрд грн. Більше того, Україна майже на 50% випереджає США за відсотком шахрайських операцій (на 10 000 транзакцій), і цей розрив лише зростає, вказуючи на те, що нові рішення потребує не тільки галузь в цілому, але вони особливо потрібні в Україні [79]. Щоб запобігти цьому потрібно трансформувати підходи та засоби для моніторингу, виявлення та контролю незаконних дій в платіжних сервісах [88]. Без сумніву, найкращий спосіб боротися з шахрайством – попередження. Таке попередження можливе за рахунок розробки системи в основі якої є прогнозування шахрайських дій на рівні виникнення підозрілої транзакцій та прогнозування ймовірності її виникнення.

Тому доцільно розвивати науково-практичні підходи попередження шахрайських операцій при здійсненні транзакцій методами машинного навчання.

Метою роботи є розробка програмної технології виявлення шахрайства в платіжних сервісах на основі математичних моделей та методів виявлення відхилень у великих наборах даних із подальшим впровадженням програмного продукту на її основі, який здатний працювати під навантаженням у режимі реального часу. Для реалізації поставленої мети необхідним є вирішення таких завдань:

- визначити математичні інструменти й методи, які використовуються для вирішення задач класифікації, та на їх основі сформувати математичні моделі виявлення шахрайства в платіжних сервісах;
- провести аналіз набору даних транзакцій із встановленням закономірностей класифікації транзакцій й виявлення серед них шахрайських;
- реалізувати програмну імплементацію моделей виявлення шахрайства в платіжних сервісах із використанням машинного навчання та мови програмування Python;
- побудувати агреговану технологію виявлення шахрайства в платіжних системах, що базується на реалізованих моделях;
- створити програмну імплементацію побудованої технології виявлення шахрайства в платіжних сервісах;
- реалізувати програмний продукт повного циклу на базі технології та із виконанням поставлених вимог, який готовий то впровадження на підприємствах (банках, фінансових установ, платіжних шлюзах, тощо)

Об’єктом дослідження виступає платіжний сервіс, його операційні процеси та результати функціонування, а саме рух коштів та характеристика транзакцій у ній.

Предметом дослідження є використання методів машинного навчання та штучних нейронних мереж для аналізу транзакцій та виявлення шахрайства

(аномалій) у платіжних сервісах із великими наборами незбалансованих даних, а також налаштування інфраструктури програмного продукту на їх основі із застосуванням хмарних обчислень.

Методи дослідження. Теоретичною основою дослідження стали загальнонаукові методи пізнання (системність, комплексність, аналіз, синтез, аналогія), використовувались елементи штучного інтелекту, нейронних мереж, машинного навчання, дата-майнінгу та графічного аналізу, хмарні обчислення.

Отримані результати базуються на сучасних наукових дослідженнях. Вони можуть бути застосовані на практиці при аналізі та моделюванні функціонування платіжних систем та виявлення шахрайства з кредитними картами, власне елементами системи. Розроблено концепт впровадження програмного продукту виявлення шахрайства в платіжних системах, що може бути розгорнутий за допомогою хмарних обчислень та бути використаний фінансовими установами як України, так і всього світу. Тема є актуальною як для ринку в цілому, так і для України особливо, адже кіберзлочинністю знаходиться на досить високому рівні завдяки висококваліфікованим кадрам. Більш того, в Україні базується декілька міжнародних платіжних систем. Таким чином, аналіз платіжних систем, транзакцій та, як наслідок, їх класифікації на шахрайські та чесні дозволить не лише значно зекономити час на зусилля клієнтів, а й покращити рейтинги надійності й безпеки українських платіжних систем та банків.

Наукова новизна одержаних результатів

- реалізовано та апробовано програмну імплементацію технології виявлення шахрайства в платіжних системах, що базується на мові програмування Python та її інструментарію та функціонує із точністю 99% на тестових наборах даних, що позитивно виділяє її на фоні інших технологій.

- сформовано та обґрунтовано математичні моделі виявлення шахрайства в платіжних системах, зокрема із застосуванням регресійних та дискретних математичних методів, а також штучних нейронних мереж для

вирішення задачі класифікації аномалій (шахрайства) у функціонуванні платіжних систем. Визначено ефективність та особливості застосування отриманих моделей, що забезпечує високу точність отриманого класифікатора.

- побудовано концепт автоматизованої інформаційної системи виявлення шахрайства в платіжних сервісах, що здатний оперувати в умовах високого навантаження та доступності.

Практична значимість роботи полягає в тому, що її результат, а саме реалізована технологія та створений концепт інформаційної системи виявлення шахрайства в платіжних сервісах, є концептом повноцінного програмного продукту повного циклу, який готовий до впровадження на підприємствах, пов'язаних із фінансовим сервісом та проведенням банківських розрахунків, та інтеграції з клієнтськими сервісах для забезпечення безпеки, прозорості та підтримки розрахунків.

Особистий внесок здобувача. Усі наукові результати, які відображено у кваліфікаційній роботі, отримані автором самостійно. Результати співавторів сумісних публікацій до тексту кваліфікаційної роботи не включено. У надрукованих статтях, опублікованих у співавторстві, магістранту належить наступне: [1] – практична реалізація методології CRISP-DM у задачах виявлення шахрайства у платіжних системах; [2] – визначення принципів роботи із даними; [3] – запропоновано технологію виявлення шахрайства; [4] – описано архітектуру, принципи та моделі функціонування, інфраструктуру автоматизованої системи виявлення шахрайства в платіжних системах; [5] – надано складову бази алгоритмічного ядра технології (фундаментальні математичні моделі), їх оцінку та ефективність; [6] – сформовано математичний інструментарій виявлення аномалій (шахрайства) в платіжних сервісах та реалізовано програмну імплементацію моделей, надано рекомендації стосовно побудови технології виявлення шахрайства в платіжних сервісах.

Апробація результатів роботи. Автор виступав доповідачем на VI Information Technology and Interactions (Satellite) (м. Київ, 2020р.); Conference Proceedings, IT&I 2020 – Information Technology and Interactions. Proceedings of the 7th International Conference "Information Technology and Interactions" (IT&I-2020). Workshops Proceedings (м. Київ, 2020р.); XVII міжнародній науково-практичній конференції «Шевченківська весна – 2019: Економіка» (м. Київ, 2019р.).

Публікації. Основні наукові положення, висновки і результати магістерської кваліфікаційної роботи знайшли відображення у 6 друкованих працях, з них: 1 стаття опублікована у рецензованому фаховому виданні України, 2 статті англійською мовою, що не є перекладом із інших мов у виданні, включеному до наукометричної бази даних Scopus, 3 тез доповідей у матеріалах конференцій. За період навчання у магістратурі опубліковано 4 наукові праці [1-4].

Структура та обсяг роботи. Кваліфікаційна робота складається зі вступу, 4 розділів, висновків, списку літератури з 88 пунктів та 6 додатків. Загальний обсяг кваліфікаційної роботи становить 117 сторінок, із них 108 сторінок основного тексту, який містить 32 рисунки.

РОЗДІЛ 1
ФОРМУЛЮВАННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ ТА ВИЯВЛЕННЯ ШАХРАЙСТВА У ФУНКЦІОНУВАННІ ПЛАТІЖНИХ СЕРВІСІВ

1.1 Визначення задачі класифікації та її інтерпретація для виявлення шахрайства в платіжних сервісах

Задача класифікації — формалізована задача, яка містить множину об'єктів (ситуацій), поділених певним чином на класи. Задана скінченна множина об'єктів, для яких відомо, до яких класів вони відносяться. Ця множина називається вибіркою. До якого класу належать інші об'єкти невідомо. Необхідно побудувати такий алгоритм, який буде здатний класифікувати довільний об'єкт з вихідної множини [52].

Класифікувати об'єкт — означає, вказати номер (чи назву) класу, до якого відноситься даний об'єкт.

Класифікація об'єкта — номер або найменування класу, що видається алгоритмом класифікації в результаті його застосування до даного конкретного об'єкту.

В математичній статистиці задачі класифікації називаються також задачами дискретного аналізу. В машинному навчанні завдання класифікації вирішується, як правило, за допомогою методів штучної нейронної мережі при постановці експерименту у вигляді навчання з учителем.

Нехай X – множина описів об’єктів, Y – множина номерів (чи назв) класів. Існує невідома цільова залежність – відображення (1.1), значення якої відомі лише на елементах скінченної навчальної вибірки (1.2) [45].

$$y^*: X \to Y \tag{1.1}$$

$$X^m = \{(x_1, y_1), \dots, (x_m, y_m)\} \tag{1.2}$$

Потрібно побудувати алгоритм (1.3), здатний класифікувати довільний об’єкт $x \in X$.

$$a: X \to Y \tag{1.3}$$

Загальнішим є імовірнісне формулювання завдання. Припускається, що множина пар «об'єкт, клас» $X \times Y$ є ймовірнісним простором з невідомою ймовірнісною мірою P. Є скінченна навчальна вибірка спостережень (1.2), згенерована згідно з ймовірнісною мірою P. Необхідно побудувати алгоритм (1.3), здатний класифікувати довільний об’єкт $x \in X$ [46].

Характеристикою називається відображення $f: X \to D_f$, де D_f – множина допустимих значень характеристики. Якщо задані характеристики $f_1, \dots, f_n$, то вектор (1.4) називається характеристичним описом об'єкта $x \in X$. Характеристики можна ототожнювати із самими об'єктами. При цьому множину (1.5) називають простором характеристик.

$$x = (f_1(x), \dots, f_n(x)) \tag{1.4}$$

$$X = D_{f_1} \times \dots \times D_{f_n} \tag{1.5}$$

Залежно від множини D_f характеристики поділяються на такі типи:

- Бінарні характеристики: $D_f = \{0, 1\}$;
- Номінальні характеристики: D_f – скінченна множина;
- Порядкові характеристики: D_f – скінченна впорядкована множина;
- Кількісні характеристики: D_f – множина дійсних чисел.

Типи класів:

- Двокласова класифікація. Найпростіший в технічному відношенні випадок, який служить основою для вирішення складніших завдань;
- Багатокласова класифікація. Коли число класів досягає багатьох тисяч (наприклад, при розпізнаванні ієрогліфів або злитого мовлення), завдання класифікації стає істотно важчим;
- Непересічні класи;

- Пересічні класи. Об'єкт може належати одночасно до декількох класів;
- Нечіткі класи. Потрібно визначати ступінь належності об'єкта кожному з класів, звичайно це дійсне число від 0 до 1 [46].

У нашому випадку нас цікавити бінарна характеристика множини з двокласовою специфікацією.

1.2 Визначення платіжної системи й особливості її функціонування

Платіжна система — платіжна організація, члени платіжної системи та сукупність відносин, що виникають між ними при проведенні переказу коштів. Проведення переказу коштів є обов'язковою функцією, що має виконувати платіжна система [54].

Основа віртуальної системи оплати — використання грошових замінників. Традиційні платіжні системи мають оборотні документи: чеки, документальні акредитиви. З появою комп'ютерів і електронних комунікацій з'явилися альтернативні електронні платіжні системи: дебетові картки, кредитні картки, електронні перекази коштів, прямі кредити, прямі дебети, інтернет-банкінг та електронна комерція платіжних систем.

Деякі платіжні системи включають в себе кредитні механізми, але по суті, інший аспект оплати. Платіжні системи використовуються замість торгів грошових коштів у внутрішніх і міжнародних угодах і складаються з основних послуг, що надаються банками та іншими фінансовими інститутами.

Платіжні системи можуть бути фізичні або електронні, які мають свої власні процедури та протоколи. Стандартизація дозволила деяким з цих систем і мереж зростати в глобальному масштабі, але все ще є багато продуктів, що орієнтовані на конкретні країни або системи.

Приклади загальних світових платіжних систем — кредитні картки та мережі банкоматів. Конкретні форми платіжних систем також використовуються для врегулювання фінансових операцій для продуктів на фондових ринках, ринках облігацій, валютних ринках, ринках ф'ючерсів, ринках похідних цінних паперів і опціонних ринках. А також для переказу коштів між фінансовими інститутами, як всередині країни з використанням клірингу і в валових розрахунках (RTGS) системи, так і на міжнародному рівні з використанням мережі SWIFT [88].

Термін електронна оплата може відноситися тільки до електронної комерції – оплати для покупки і продажу товарів або послуг, що пропоновані через Інтернет- або, в широкому сенсі, до будь-якого типу електронного переказу коштів.

Приклади: China UnionPay, MasterCard, PrivatMoney, PayPal, Western Union.

Захист інформації забезпечується суб’єктами переказу грошей шляхом обов’язкового впровадження та використання відповідної системи захисту.

Система захисту інформації складається з:

1) законодавчих актів України та інших нормативно-правових актів, а також внутрішніх нормативних актів суб’єктів переказу, що регулюють порядок доступу та роботи з відповідною інформацією, а також відповідальність за порушення цих правил;

2) заходів охорони приміщень, технічного обладнання відповідної платіжної системи та персоналу суб’єкта переказу;

3) технологічних та програмно-апаратних засобів криптографічного захисту інформації, що обробляється в платіжній системі.

Система захисту інформації повинна забезпечувати:

1) цілісність інформації, що передається в платіжній системі, та компонентів платіжної системи;

2) конфіденційність інформації під час її обробки, передавання та зберігання в платіжній системі;

3) неможливість відмови ініціатора від факту передавання та отримувачем від факту прийняття документа на переказ, документа за операціями із застосуванням засобів ідентифікації, документа на відкликання;

4) забезпечення постійного та безперешкодного доступу до компонентів платіжної системи особам, які мають на це право або повноваження, визначені законодавством України, а також встановлені договором [18].

Розробка заходів охорони, технологічних та програмно-апаратних засобів криптографічного захисту здійснюється платіжною організацією відповідної платіжної системи або іншою установою на її замовлення відповідно до законодавства України та вимог, встановлених Національним банком України.

При побудові політики безпеки платіжної системи використання криптографічного захисту інформації та безпечного розподілу ключів значно посилює безпеку роботи системи [63].

За принципами використання криптографічний захист може бути вбудованим у платіжну систему або бути додатковим механізмом, який може відключатися.

Виділяють дві групи криптографічних алгоритмів:

1. Загальні криптоалгоритми мають повністю відкритий алгоритм, а їх криптостійкість визначається ключами шифрування.

2. Спеціальні криптоалгоритми мають таємний алгоритм шифрування.

Загальні криптоалгоритми розподіляються на дві групи:

1. Симетричні алгоритми – криптографічні алгоритми, для яких шифрування і розшифрування виконуються одним ключем. Відправник і отримувач повідомлення повинні мати один і той самий ключ.

2. Асиметричні алгоритми – криптографічні алгоритми, для яких шифрування і розшифрування виконуються за допомогою різних ключів.

Криптографічні алгоритми використовуються з метою:

- по-перше, шифрування інформації (приховування змісту повідомлень і даних);
- по-друге, забезпечення захисту даних і повідомлень (інформації) від модифікації, викривлення або підробки.

Для асиметричних криптографічних алгоритмів є можливість сформувати додаткову інформацію у вигляді електронного цифрового підпису [7].

1.3 Аналіз викликів виявлення шахрайства у платіжних сервісах

Внесок у розвиток машинного навчання, науки про дані та застосування даних моделей в економіці внесли такі вчені як Бернуллі Я., Пуассон С., Марков А.А., Чебишев П.П., Колмогоров А.Н., Тьюрінг А.

Особливості використання моделей прогнозування шахрайських дій представлено у роботах [7 – 11]. Основна суть представлених робіт полягає у моделях класифікації транзакцій на предмет шахрайства різними методами. Цей підхід доречно використовувати для прийомів шахрайства, які спостерігались раніше. У роботі [12] сформовано підхід за яким транзакція вважається шахрайською, якщо вона відрізняється від звичайної поведінки користувача. Це пов'язано з тим, що передбачається, що зловмисники будуть вести себе зовсім по-іншому, ніж власник облікового запису. Підхід до опрацювання ризиків шахрайства з платіжними системами, який передбачає комбінацію представлених підходів визначено у роботах [13, 14]. Виходячи з вищесказаного спочатку потрібно напрацювати і виявити модель поведінки користувача кредитної карти, а потім вже виявляти шахрайство. Для вирішення цієї задачі можуть використовуватись різноманітні методи та алгоритми [15]. Разом з тим, у літературі про шахрайство з кредитними картками не існує жодного потужного

алгоритму, який би був стандартом для всіх фінансових установ [16]. Кожна техніка має свої переваги та недоліки. Крім того, шахрайські схеми динамічні та вимагають постійного переопрацювання їх прогнозування. Тому дослідження моделей виявлення шахрайства, зміни їх параметрів, поєднання алгоритмів для підтримки переваг один одного та покриття їх слабких сторін при виявленні шахрайства з фінансовими платіжними системами з позиції системності має як науковий так і практичний інтерес.

За даними [18] визначено такі шахрайські схеми: соціальна інженерія; перекази з картки на картку; перекази через онлайн-банкінг; перехоплення доступу до мобільного банкінгу; підроблений мобільний банкінг; покупки за допомогою Apple Pay і Google Pay; розкрадання через SMS-банкінг. З'ясовано, що основним видом шахрайства в банківській сфері є шахрайство з платіжними картками. У роботі наведено деякі методи боротьби із шахрайськими операціями та встановлено перевагу використання сучасних технологій, в основі яких моделі та методи виявлення шахрайства. Особливості використання моделей виявлення шахрайства представлені в роботах [19-21]. Основною суттю представлених робіт є класифікація моделей шахрайських операцій різними методами. Доцільно використовувати цей підхід для методів виявлення шахрайства, які спостерігались раніше. Робота [24] формує підхід, згідно з яким транзакція вважається шахрайською, якщо вона відрізняється від звичайної поведінки користувача. Це базується на припущенні, що зловмисники поводитимуться зовсім інакше, ніж власник облікового запису. Підхід до управління ризиками шахрайства з платіжними системами, який передбачає поєднання представлених підходів, визначений у роботах [25-26]. Беручи до уваги вищесказане, доцільно спочатку розробити та визначити модель поведінки користувачів, а потім виявити шахрайство. Для вирішення цієї проблеми можуть бути використані різні методи та алгоритми. Варто врахувати дослідження представлене у роботі [27] – відсутність ефективного та точного алгоритму щодо шахрайства, який би був

стандартом для всіх фінансових транзакцій. Кожна техніка має свої переваги та недоліки. Крім того, підходи до шахрайства є динамічними та вимагають постійної переробки прогнозів, також це пов’язано із тим, що кожен бізнес у якому можливі шахрайські схеми є унікальним та повинен спиратися на власну корпоративну систему. Прикладом застосування уніфікованих моделей методів під умови конкретизованих установ представлено у роботі [28]. З позиції застосування алгоритмів для виявлення шахрайських операцій в конкретних бізнесах доречно застосовувати набір систематизованих способів забезпечення прогнозування, виявлення та контролю за шахрайськими операціями у фінансових системах, які у роботі [2] отримали назву технологій . Моделі, методи та алгоритми машинного навчання лежать в основі такої технології. Недоліком представленої роботи є те, що у ній не представлено адаптацію розробленої моделі у програмний продукт. Разом з тим, на ринку представлено велику кількість автоматизованих систем, які націлені на запобігання шахрайству, які отримали назву антифрод-систем [25]. В літературних даних переважає інформація про комплексні системи виявлення банківського шахрайства. В основі таких систем аналітичні платформи, що дозволяють реалізовувати логіку в окремих сегментах. Зазвичай такі системи використовуються у банківській сфері. Прикладом таких систем є: ARIC White Label від компанії Featurespace, FICO Application Fraud Manager від компанії FICO, FRAUD- Analysis від компанії BSS, IBM Safer Payments від компанії IBM, Fraud and Security Intelligence (SAS FSI) [31, 32].

У роботі [33] представлено принципи роботи систем, які направлені на ідентифікацію інструментів банківського шахрайства. Прикладом таких систем є: Digital Banking Fraud Detection, WebSafe, IBM Trusteer Rapport, ThreatMetrix, Group-IB Secure Bank, тощо.

Вузькоспеціалізованими системами виявлення ознак банківського шахрайства є: FPS.Bio, SmartTracker.FRAUD.

Змішаними системами протидії банківського шахрайства є: RSA Adaptive Authentication and Transaction Monitoring, BI.ZONE Cloud Fraud Prevention.

Використання таких систем пов'язано, зазвичай, з додатковими витратами. Крім того, досвід використання готових програмних продуктів показує, що такі продукти не націлені на концепції виявлення шахрайства локального бізнесу на основі конкретизованих наявних даних за окремою компанією. Впровадження автоматизованої системи виявлення шахрайства охоплює набагато ширший, ніж вибір системи, а також має враховувати спектр завдань від впровадження ідеології між працівниками, формалізації процедур збору та зберігання інформації до змін в організаційній структурі та розподілу ролей в команді. Це у деякій мірі знайшло відображення у роботі [34]. Але у роботі відсутній опис самої автоматизованої системи.

Спеціалізована автоматизована система не має на меті конкуренції із представленими, а, навпаки, доповнить їх функції.

1.4 Прикладне застосування машинного навчання для аналізу й моніторингу функціонування платіжних систем

В наш час компанії, що надають фінансові послуги, знайшли нові можливості завдяки машинного навчання, яке являє собою ряд процесів, що наділяють комп'ютери здатністю робити припущення, засновані на відомих властивостях, витягнутих з навчальних даних. Найголовніше в машинному навчанні – це дані. Комп'ютери аналізують нову інформацію і зіставляють її з уже існуючими даними, щоб відшукати закономірності, подібності та відмінності. При цьому постійно вдосконалюється їх здатність більш точно і ефективно аналізувати дані, класифікувати інформацію і робити припущення, що дає можливість приймати кращі рішення, засновані на даних.

Багато стартапів перевернули з ніг на голову екосистему фінтеха саме завдяки впровадженню машинного навчання в якості однієї зі своїх ключових технологій. Компанії використовують різні алгоритми машинного навчання для вирішення завдань, які можна поділити на кілька категорій. Давайте подивимося на деякі з прикладів використання машинного навчання на благо таких компаній.

1. Кредитний скоринг

Все частіше компанії, що діють у сфері кредитування, використовують машинне навчання для прогнозування кредитоспроможності клієнтів, а також для побудови моделей кредитних ризиків. Серед таких компаній — Kabbage, Inc., фінансує малий бізнес за допомогою платформи кредитування, сервіс віддаленого мікрокредитування LendUp і визнаний лідер галузі фінансових технологій Lending Club. Зокрема, команда Kabbage спеціалізується на розробці алгоритмів машинного навчання нового покоління та аналітики для побудови моделей кредитного ризику та аналізу існуючого портфеля. Серед безлічі алгоритмів машинного навчання для визначення рейтингу кредитоспроможності позичальника використовуються наступні: багатошаровий перцептрон, логістична регресія, метод опорних векторів, а також алгоритм посилення класифікаторів AdaBoost (або Adaptive Boosting) і квантування векторів при навчанні [8].

2. Прийняття рішень

Фінансові обчислення та прийняття рішень можуть здійснюватися за допомогою алгоритмів машинного навчання, які дозволяють комп'ютерам ефективніше і швидше обробляти данні і приймати рішення щодо кредитування, страхування, захисту від шахрайства і т. д. Моделі машинного навчання широко використовуються такими компаніями як Affirm, BillGuard і ZestFinance. Останній вдалося знайти новий підхід до традиційних завдань завдяки машинному навчання і аналізу великих масивів даних. Компанія аналізує тисячі потенційних кредитних змінних – від фінансової інформації до використання технологій, щоб краще оцінити такі фактори як можливості потенційного шахрайства, ризик

невиконання зобов'язань і ймовірність довгострокових відносин з клієнтами. Як результат, підприємство може приймати «правильні» рішення про надання кредитів, що призводить до підвищення доступності кредитів для позичальників і більш високому відсотку їх погашення [8].

3. Витяг інформації

Час поговорити про різновиди інформаційного пошуку, метою якого є автоматичне отримання структурованих даних при обробці неструктурованої або слабоструктурованої інформації. Як правило, це стосується роботи з веб-контентом, тобто статтями, публікаціями в соціальних мережах і різними документами. Наприклад, спеціалізована система пошуку AlphaSense для фінансових компаній використовує алгоритми обробки природної мови і складні алгоритми машинного навчання [7].

Завдяки потужним алгоритмом власної розробки фірма Dataminr здатна зробити моментальний аналіз потоків публікацій Twitter і інших даних із соціальних мереж і веб-джерел і перетворити їх у корисну інформацію, яка може бути застосована на практиці. Компанія націлена на клієнтів зі сфери фінансів і новин, а також державного сектора і корпоративної безпеки. Обробляючи по 500 млн твітів щодня, алгоритми Dataminr здатні відшукати релевантну інформацію про цікавлять клієнтів нових висхідних тренди і останніх гарячих новинах на хвилини і навіть години раніше того моменту, коли вони, власне, стануть такими.

4. Захист від шахрайства

За результатами проведених IBM досліджень, щорічно шахрайство завдає фінансової індустрії збиток, рівний приблизно 80 млрд доларів. Машинне навчання дає більш ефективні методи виявлення шахрайства. Завдяки створеним рішень можна проводити аналіз історії транзакцій для побудови моделі, яка могла б розпізнати шахрайські дії. Крім того, технології машинного навчання також застосовуються фінтех-компаніями для розробки систем біометричної аутентифікації користувачів. Стартап EyeVerify розробив технологію з

застосуванням алгоритмів машинного навчання, що дозволяє використовувати модне «селфи» для забезпечення безпеки своїх фінансових операцій. Їх флагманський продукт Eyeprint ID – програмне забезпечення, яке ідентифікує користувача по малюнку вен на білках очей та інших мікроскопічних особливостей очі [11].

5. Алгоритмічні стратегії для торгівлі

Машинне навчання застосовується для створення високоефективних алгоритмічних стратегій для торгівлі. Основною формою алгоритмічної торгівлі є високочастотний трейдинг, в якому для швидкої торгівлі цінними паперами задіяні спеціальні алгоритми і торгові роботи. Машинне навчання надає потужні інструменти для вивчення закономірностей ринку. Завдяки предиктивному моделюванню, програмуванню та алгоритмам машинного навчання компанія KFL Capital Management Ltd., управляє фондом інвестицій, стала експертом в області прогнозування змін поведінки ринку на підставі фінансових даних. Торгова фірма Binatix впроваджує надсучасні алгоритми мобільного навчання, які допомагають виявити закономірності, що дають перевагу в інвестуванні [11].

1.5 Дослідження та використання машинного навчання для аналізу платіжних систем в Україні

Запобігання фроду реалізується багатьма способами і має комплексний підхід. Більшість систем, що перешкоджають шахрайству, включають у себе низку засобів, які взаємодіють за визначеним алгоритмом та налаштовані в контексті умов конкретної платіжної системи. Зазвичай, такі комплекси є комерційною таємницею фінансових установ, тому вони не мають документального опису, розбору та оцінки ефективності у відкритому доступі [13].

Але деякі з засобів, що входять до цих комплексів, загальновідомі. Так, у своїй роботі «Платіжні системи» автор Вовчак О.Д. проводить дослідження існуючих заходів захисту безпеки платіжних систем. У створення таких заходів він включає розробку підтвердження особи, яка здійснює авторизацію у платіжній системі, та його здійснення за допомогою додаткового запитання, відповідь на яке було вказане при реєстрації облікового запису. Дана робота дозволяє переконатись, що особа, яка здійснює авторизацію, являється власником даного облікового запису. Недоліками цієї роботи є обмежений вибір способів підтвердження авторизації, відсутність захисту від фішингу і відсутність підтвердження транзакції, запит на яку може прийти від окремо взятого облікового запису. У розроблюваній системі запобігання фроду, буде здійснюватися перевірка позитивної моделі поведінки користувача, що захистить від описаних проблем.

У роботі автора Чайковського Я. І. «Платіжні системи» [88] запобігання фроду описано за допомогою криптографічних засобів захисту. Такий спосіб допомагає забезпечити конфіденційність даних, що передаються між користувачем і платіжною системою, та запобігає здійсненню фальшивих транзакцій. Такі транзакції шахраї можуть провести використовуючи систему запитів та відповідей платіжної системи, і при цьому, не використовуючи обліковий запис, що ускладнює пошук злочинців. Недоліками даного способу є відсутність захисту від розповсюдження інформації або надання доступу до облікових записів самими користувачами. Система запобігання фроду на основі характерних ознак підозрілої активності, буде в змозі виявити подібну ситуацію та попередити про неї.

Також існує загальноприйнята технологія, яка не являє собою комплексний підхід до вирішення поставленого завдання, але часто виступає у якості допоміжного засобу в різних засобах захисту інформації. CAPTCHA або автоматизований тест Тюрінга – технологія, яку винайшли ще в 2000 році

оформлена у вигляді спотвореного або важко розбірливого тексту або зображення, розпізнавання якого важко реалізувати на програмному рівні.

Застосовується з метою унеможливлення автоматизації або імітації авторизації користувачів шляхом перебору паролів чи іншими методами. Недоліком є те, що на даний момент виникають і розвиваються технології, основані на машинному навчанні, які здатні розпізнавати CAPTCHA. Додаткове підтвердження або двоетапна автентифікація являє собою більш нову технологію. Даний метод працює лише у випадку наявності у користувачів інших технічних або цифрових засобів, що дозволяють переконатися у тому що саме ця особа виконує певні дії. Недоліками такого захисту є збільшення часу на використання програмних продуктів та необхідність підключення у процес автентифікації інших пристроїв або додатків, захист яких може виявитись ненадійним.

MasterCard Early Detection Fraud Fighting System – це найновіша технологія, яка була запущена весною 2017 року для банків емітентів карток MasterCard. Дана технологія базується на прогнозуванні та зборі статистичних даних. Основною метою цієї системи являється пошук і повідомлення про можливий витік даних користувачів певного банку у бази даних шахраїв, для подальшого сповіщення керівництва банка емітента. Недоліками такої системи є можливість затримки в процесі передачі інформації про можливий фрод до користувачів та неточність визначення шахрайства, що може спричинити витрати часу на реагування на такі ризики.

Частка шахрайських операцій в Україні з платіжними картками знизилася приблизно на чверть за 2018 рік. У 2018 році частка шахрайських операцій щодо всіх операцій за картками становить 0,0077%, тобто це 77 грн на 1 млн грн, витрачених за допомогою картки. Роком раніше ця цифра була близька до 0,01%, що становить 100 грн на 1 млн грн [79].

Водночас середній чек в шахрайстві за рік зріс із 1900 грн до 2100 грн. Але при цьому на 1 млн усіх платіжних операцій припадає лише 25 шахрайських операцій (0,0025%).

За даними кіберполіції, одними з найбільш поширених видів кібершахрайства в сфері електронних платежів і карткових розрахунків стали скімінг, вірусне зараження банкоматів, Black-Box атаки і несанкціоноване списання коштів з допомогою онлайн-банкінгу [79].

Таким чином, наведені алгоритми з виявлення шахрайства в платіжних системах будуть корисні й вітчизняним компаніям. В Україні офіційно зареєстровано 24 вітчизняні платіжні системи, у 10 з них платіжними організаціями є банки й відповідно у решти 14 – небанківські установи. 9 платіжних систем є міжнародними, решта – внутрішньодержавними. Безперечно, кожна з систем бажає поставляти кращий сервіс своїм клієнтам, забезпечувати безпеку їх коштів, зберігати як їх, так і власні кошти.

Беручи до уваги вхідні дані, наявний інструментарій та ідентифіковані проблеми галузі, зосередимось на вирішенні таких завдань:

- визначити математичні інструменти й методи, які використовуються для вирішення задач класифікації, та на їх основі сформувати математичні моделі виявлення шахрайства в платіжних сервісах;
- провести аналіз набору даних транзакцій із встановленням закономірностей класифікації транзакцій й виявлення серед них шахрайських;
- реалізувати програмну імплементацію моделей виявлення шахрайства в платіжних сервісах із використанням машинного навчання та мови програмування Python;
- побудувати агреговану технологію виявлення шахрайства в платіжних системах, що базується на реалізованих моделях;
- створити програмну імплементацію побудованої технології виявлення шахрайства в платіжних сервісах;

- реалізувати програмний продукт повного циклу на базі технології та із виконанням поставлених вимог, який готовий то впровадження на підприємствах (банках, фінансових установ, платіжних шлюзах, тощо).

РОЗДІЛ 2
ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ОСНОВИ АНАЛІЗУ МЕТОДАМИ МАШИННОГО НАВЧАННЯ ТА ЗАДАЧ КЛАСИФІКАЦІЇ

2.1 Регресійні методи вирішення задач класифікації

У даному пункті будуть сформовані й описані основні математичні моделі, які покладені в основу моделей машинного навчання та на основі яких базуються більш складні й комплексні моделі, які ми будемо використовувати для аналізу транзакцій. Будуть розглянуті окремі математичні складові кожної моделі та сформовані моделі у цілому.

Першим й найпростішим методом буде застосування лінійної регресії (не в чистому вигляді, а як база перш за все для логістичної регресі). Лінійна регресія – метод моделювання залежності між скаляром y та векторною змінною X. У випадку, якщо X також є скаляром, регресію називають простою [51]. Загалом лінійна регресія це лінійна функція, загальна модель якої визначається у виді:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k + u, \tag{2.1}$$

де y – залежна змінна, $(x_1, x_2, \dots, x_k)$ – вектор незалежних змінних, $(\beta_0, \beta_1, \dots, \beta_k)$ – вектор параметрів, u – випадкова похибка, розподіл якої в загальному випадку залежить від незалежних змінних, але математичне сподівання якої рівне нулю.

Задача лінійної регресії полягає у оцінці вектора параметрів на основі деяких експериментальних значень y та $(x_1, x_2, \dots, x_k)$.

Лінійна регресія проста в застосуванні та інтуїтивно зрозуміла, але, очевидно, для вирішення поставленої задачі вона є дуже примітивною. Тому розглянемо між складні та комплексні види регресії.

Для вирішення нашого завдання класифікації підходить логістична регресія. Це статистичний регресійний метод, що застосовують у випадку, коли залежна змінна є категорійною, тобто може набувати тільки двох значень (чи, загальніше, скінченної множини значень).

Нехай деяка множина Y може набувати тільки двох значень, які, як правило, позначаються цифрами 0 та 1. Нехай ця величина залежить від деякої множини пояснювальних змінних $x = (1, x_1, x_2, \dots, x_k)$. Залежність Y від $x_1, x_2, \dots, x_k$ можна визначити ввівши додаткову змінну y^* [55]. Тоді:

$$Y = \begin{cases} 0, y^* \leq 0 \\ 1, y^* > 0 \end{cases} \quad (2.2)$$

де $y^* = \theta_0 + \theta_1 x_1 + \cdots + \theta_k x_k + u$.

Логістична регресія (Logistic regression) – метод побудови лінійного класифікатора, що дозволяє оцінювати апостеріорні ймовірності приналежності об'єктів класів.

Логістична регресія - це різновид множинної регресії, загальне призначення якої полягає в аналізі зв'язку між декількома незалежними змінними (також званими регресорами або предикторами) і залежною змінною. Бінарна логістична регресія, як випливає з назви, застосовується в разі, коли залежна змінна є бінарною (тобто може приймати тільки два значення). Іншими словами, за допомогою логістичної регресії можна оцінювати вірогідність того, що подія настане для конкретного випробуваного (хворий/здоровий, повернення кредиту/дефолт і т.д.).

Наступним інструментом є метод опорних векторів [56] – метод аналізу даних для класифікації та регресійного аналізу за допомогою моделей з керованим навчанням з пов'язаними алгоритмами навчання, які називаються опорно-векторними машинами. Для заданого набору тренувальних зразків, кожен із яких відмічено як належний до однієї чи іншої з двох категорій, алгоритм тренування ОВМ будує модель, яка відносить нові зразки до однієї чи іншої

категорії, роблячи це неймовірнісним бінарним лінійним класифікатором. Модель ОВМ є представленням зразків як точок у просторі, відображених таким чином, що зразки з окремих категорій розділено чистою прогалиною, яка є щонайширшою. Нові зразки тоді відображуються до цього ж простору, й робиться передбачення про їхню належність до категорії на основі того, на який бік прогалини вони потрапляють.

На додачу до виконання лінійної класифікації, ОВМ можуть ефективно виконувати нелінійну класифікацію при застосуванні так званого ядрового трюку, неявно відображуючи свої входи до просторів ознак високої вимірності [22].

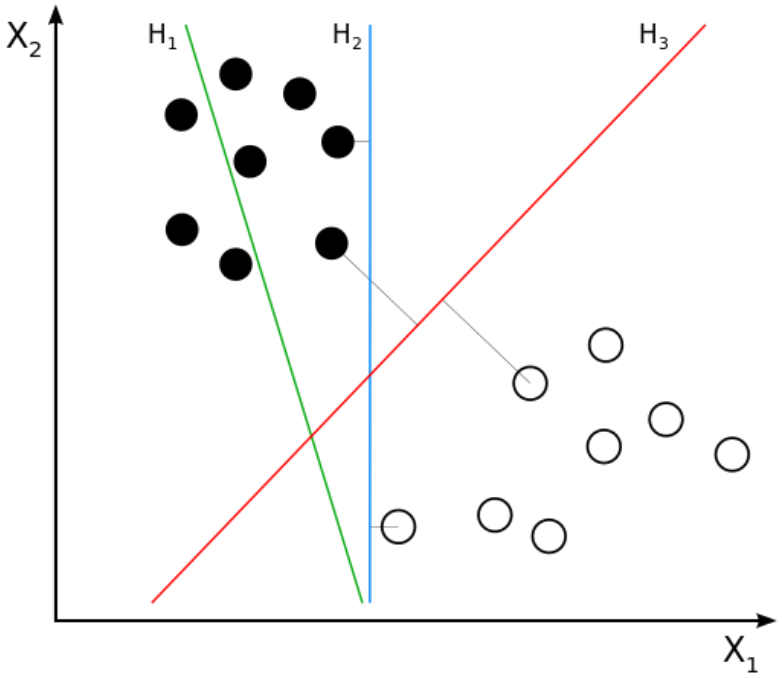


Рисунок 2.1 – Графічне зображення класифікації методом опорних векторів

Формальніше, опорно-вектора машина будує гіперплощину, або набір гіперплощин у просторі високої або нескінченної вимірності, які можна використовувати для класифікації, регресії та інших задач. Інтуїтивно, добре розділення досягається гіперплощиною, яка має найбільшу відстань до

найближчих точок тренувальних даних будь-якого з класів (так зване функційне розділення) [60].

Продемонструємо наочно метод опорних векторів на Рис. 2.1. Як бачимо, H1 не розділяє ці класи. H2 розділяє, але лише з невеликим розділенням. H3 розділяє їх із максимальним розділенням, тому і є класифікатором.

2.2 Дискретні методи вирішення задач класифікації

Наступний метод вирішення задач класифікації використовує дещо інший підхід. Метод k-найближчих сусідів – простий непараметричний класифікаційний метод, де для класифікації об'єктів у рамках простору властивостей використовуються відстані (зазвичай евклідові), пораховані до усіх інших об'єктів. Вибираються об'єкти, до яких відстань найменша, і вони виділяються в окремий клас.

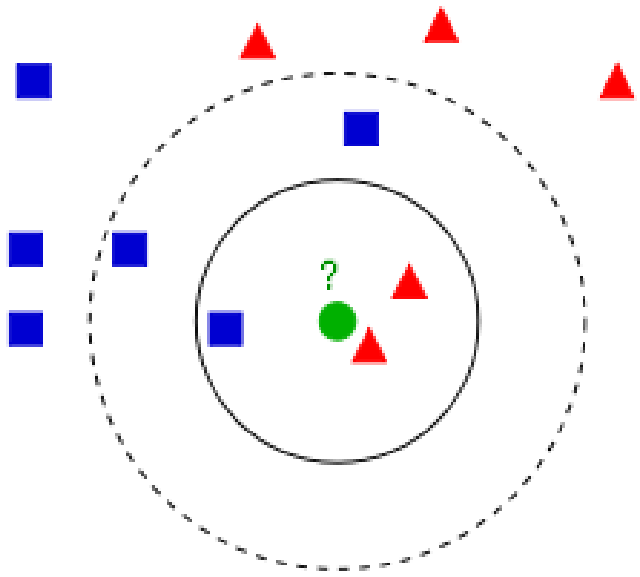


Рисунок 2.2 – Приклад класифікації k найближчих сусідів

Пояснімо більш детально даний метод на наочному прикладі Рис. 2.2. Тестовий зразок (зелене коло) повинен бути класифікований як синій квадрат (клас 1) або як червоний трикутник (клас 2). Якщо k = 3, то класифікується як 2-й

клас, тому що всередині меншого кола 2 трикутника і тільки 1 квадрат. Якщо k = 5, то він буде класифікований як перший клас (3 квадрата проти 2-ох трикутників всередині більшого кола).

Основним принципом методу найближчих сусідів є те, що об'єкт присвоюється тому класу, який є найбільш поширеним серед сусідів даного елемента. Сусіди беруться, виходячи з множини об'єктів, класи яких уже відомі, і, виходячи з ключового для даного методу значення k, вираховується, який клас є найчисленнішим серед них. Кожен об'єкт має кінцеву кількість атрибутів (розмірностей). Передбачається, що існує певний набір об'єктів з уже наявною класифікацією [61].

Отже, метод найближчого сусіда – найпростіший метод класифікації. Процес навчання в ньому полягає в запам’ятовуванні всіх об’єктів навчальної вибірки. Перевагами методу є простота реалізації та наочна інтерпретація результатів. До недоліків можна віднести те, що метод нестійкий до шуму, потребує зберігання усієї вибірку у пам’яті (що призводить до її неефективного використання), метод не дозволяє задавати параметри налаштування та метод потребує багато часу для обчислення на великих наборах даних.

Наступним методом вирішення задач класифікації є дерева ухвалення рішень, які використовується в галузі статистики та аналізу даних для прогнозних моделей. Структура дерева містить такі елементи: «листя» і «гілки». На ребрах («гілках») дерева ухвалення рішення записані атрибути, від яких залежить цільова функція, в «листі» записані значення цільової функції, а в інших вузлах — атрибути, за якими розрізняються випадки. Щоб класифікувати новий випадок, треба спуститися по дереву до листа і видати відповідне значення. Подібні дерева рішень широко використовуються в інтелектуальному аналізі даних. Мета полягає в тому, щоб створити модель, яка прогнозує значення цільової змінної на основі декількох змінних на вході [57].

Кожен лист являє собою значення цільової змінної, зміненої в ході руху від кореня по листа. Кожен внутрішній вузол відповідає одній з вхідних змінних. Дерево може бути також «вивчено» поділом вихідних наборів змінних на підмножини, що засновані на тестуванні значень атрибутів. Це процес, який повторюється на кожному з отриманих підмножин. Рекурсія завершується тоді, коли підмножина в вузлі має ті ж значення цільової змінної, таким чином, воно не додає цінності для пророкувань. Процес, що йде «згори донизу», індукція дерев рішень (TDIDT), є прикладом поглинаючого «жадібного» алгоритму, і на сьогодні є найбільш поширеною стратегією дерев рішень для даних, але це не єдина можлива стратегія. В інтелектуальному аналізі даних, дерева рішень можуть бути використані як математичні та обчислювальні методи, щоб допомогти описати, класифікувати і узагальнити набір даних, які можуть бути записані таким чином (2.3):

$$(x, Y) = (x_1, x_2, \dots, x_k, Y) \tag{2.3}$$

Залежна змінна Y є цільовою змінною, яку необхідно проаналізувати, класифікувати й узагальнити. Вектор x складається з вхідних змінних $x_1, x_2, \dots, x_k$ тощо, які використовуються для виконання цього завдання [57].

Дерева рішень, використовувані в Data Mining, бувають двох основних типів:

- Аналіз дерева класифікації, коли прогнозований результат є класом, до якого належать дані;
- Регресійний аналіз дерева, коли прогнозований результат можна розглядати як дійсне число (наприклад, ціна на будинок, або тривалість перебування пацієнта в лікарні).

У контексті поточної задачі нас цікавить перший тип дерев рішень для вирішення питань класифікації.

Поширеною технікою машинного навчання, яка опирається на дерева рішень, є градієнтний бустінг – техніка, яка використовується для вирішення задач класифікації та регресії, результатом якої є модель прогнозування у формі ансамбля (збірки) декількох простих моделей, зазвичай саме дерев рішень. Градієнтний бустінг – один с найкращих способів, націлених на побудову композиції. Ми будемо будувати композицію наступного вигляду (2.4):

$$a_N(x) = \sum_{n=1}^{N} b_n(x) \quad (2.4)$$

де a_N – композиція з N базових алгоритмів, b_n – базовий алгоритм.

Ми не усереднюємо, а сумуємо алгоритми, оскільки кожен наступний коректує помилки попереднього. Також будемо вважати, що ми маємо певну функцію втрат *L (y, z)*, яка вимірює значення помилки для одного об'єкта, прикладом функції може бути звичайна функція методу найменших квадратів, яка була описана вище.

Побудову моделі ми починаємо з ініціалізації, будуємо перший базовий алгоритм $b_0(x)$, який не повинен бути надто важким, це може бути найпростіша функція виду $b_0(x) = 0$ (де на виході ми будемо отримувати константу), або середня відповідь по всій навчальній вибірці [44]:

$$b_0(x) = \frac{1}{l}\sum_{i=1}^{l} y_i \quad (2.5)$$

Будемо дійти методом індукції й покладемо, що ми вже побудували *N-1* алгоритмів (для *N=1* це буде означати, що ми побудували лише початковий алгоритм $b_0(x)$). Наше завдання – зрозуміти, яким повинен бути наступний навчальний алгоритм $b_n(x)$. Задача буде виглядати так:

$$\sum_{i=1}^{l} L(y_i, a_{N-1}(x_i) + b_i(x_i)) \to min \quad (2.6)$$

Ми сумуємо втрати на всій навчальній вибірці: суми вже побудованої структури $a_{N-1}(x_i)$ й нового алгоритм у $b_i(x_i)$, й будемо намагатися вибрати останній таким чином, щоб мінімізувати помилку композиції.

Для початку спростимо собі задачу й спробуємо дати відповідь на питання які значення наш новий алгоритм повинен приймати на об'єктах навчальної вибірки:

$$\sum_{i=1}^{l} L(y_i, a_{N-1}(x_i) + s_i) \to min \quad (2.7)$$

s_i – зсув прогнозу на *i*-му об'єкті.

Отже, ми отримаємо наступну задачу оптимізації. Нам потрібно знайти такий вектор $s = (s_1, s_2, \dots, s_l)$, який буде мінімізувати дану функцію

$$F(s) = \sum_{i=1}^{l} L(y_i, a_{N-1}(x_i) + s_i) \to min \quad (2.8)$$

Вектор, який якнайбільше зменшує функцію, це антиградієнт, оскільки він направлений в сторону найшвидшого зменшення функції. Отже:

$$s = -\nabla F = (-L'_z(y_1, a_{N-1}(x_1)), \dots, -L'_z(y_l, a_{N-1}(x_l))) \quad (2.9)$$

де $-L'_z(y_l, a_{N-1}(x_l))$ – зсув по *l*-му об'єкту.

Отже, ми вже зрозуміли, як саме необхідно сзунути прогнози вже побудованої композиції, щоб зменшити значення функції втрат. Ми будемо налаштовувати наступний алгоритм, $b_N(x)$ так, щоб він був якомога ближче до зсувів s_i і близкість будемо вимірювати за допомогою середньоквадратичного відхилення. Функціонал даної задачі буде виглядати так:

$$b_N(x) = argmin \frac{1}{l} \sum_{i=1}^{l} (b(x_i) - s_i)^2 \quad (2.10)$$

До того ж, вся інформація про фунцію витрат L міститься в зсувах, градієнті.

На кінцевому етапі, після знайдення алгоритму $b_N(x)$, ми додаємо його до композиції [60].

Баєсів підхід до класифікації заснований на теоремі, яка стверджує, що коли щільність розподілу кожного з класів відома, то шуканий алгоритм можна виписати в явному аналітичному вигляді. Більше того, даний алгоритм оптимальний, тобто володіє мінімальною вірогідністю помилок. На практиці щільності розподілу класів, як правило, не відомі, тож їх доводиться оцінювати по

навчальній вибірці. В результаті баєсів алгоритм перестає бути оптимальним, адже відновити щільність по вибірці можливо тільки з деякою похибкою. Чим коротше вибірка, тим вище шанси підігнати розподіл під конкретні дані і зіткнутися з ефектом перенавчання. І хоча даний алгоритм є одним з найстаріших, аналіз показує, що він втримує стійкі позиції і сьогодні. Цей підхід лежить в основі й багатьох інших достатньо успішних баєсових методів класифікації. До їх числа відносяться: лінійний дискримінант Фішера, метод парзенового вікна, метод радіальних базисних функцій (RBF) та інші [50].

2.3 Штучні нейронні мережі для вирішення задач класифікації

Штучні нейронні мережі також можуть використовуватись й для вирішення задач класифікації. Штучна нейронна мережа — це мережа простих елементів, званих нейронами, які отримують вхід, змінюють свій внутрішній стан (збудження) відповідно до цього входу, і виробляють вихід, залежний від входу та збудження. Мережа утворюється з'єднанням виходів певних нейронів зі входами інших нейронів з утворенням орієнтованого зваженого графу. Ваги, як і функції, що обчислюють збудження, можуть змінюватися процесом, званим навчанням, який керується правилом навчання [87].

Складові штучної нейронної мережі [87]:

1. Нейрони

Нейрон з міткою j, що отримує вхід $p_j(t)$ від нейронів-попередників, складається з наступних складових:

- Збудження $a_j(t)$, що залежить від дискретного параметра часу
- Порогу θ_j, що залишається незмінним, якщо його не змінить функція навчання

- Функції збудження f, яка обчислює нове збудження в заданий час $t + 1$ з $a_j(t), \theta_j$ та мережевого входу $p_j(t)$, даючи в результаті відношення $a_j(t+1) = f\left(a_j(t), p_j(t), \theta_j\right)$.
- Функції виходу f_{out}, яка обчислює вихід з активації: $o_j(t) = f_{out}(a_j(t))$.

Функція виходу часто є просто тотожною функцією. Нейрон входу не має попередників, а слугує інтерфейсом входу для всієї мережі. Аналогічно, нейрон виходу не має наступників, і відтак слугує інтерфейсом виходу для всієї мережі.

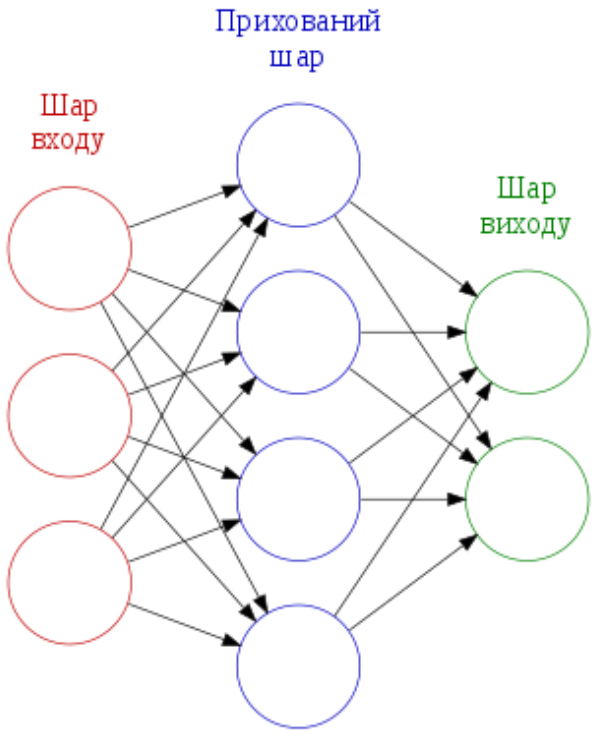


Рисунок 2.3 – Узагальнена модель штучної нейронної мережі

2. З’єднання та ваги

Мережа складається зі з'єднань, кожне з яких передає вихід нейрону i до входу нейрону j. В цьому сенсі i є попередником j, а j є наступником i. Кожному з'єднанню призначено вагу w_{ij}.

3. Функції поширення

Функція поширення обчислює вхід $p_j(t)$ до нейрону j з виходів $o_i(t)$ нейронів-попередників і зазвичай має вигляд:

$$p_j(t) = \sum_i o_i(t) w_{ij} \quad (2.11)$$

4. Правило навчання

Правило навчання – це правило або алгоритм, який змінює параметри нейронної мережі, щоби заданий вхід до мережі видавав придатний вихід. Цей процес навчання зазвичай полягає в зміні ваг та порогів змінних мережі [82].

Існує три основні парадигми навчання, кожна з яких відповідає певній навчальній задачі. Ними є кероване навчання, спонтанне навчання та навчання з підкріпленням. Нас цікавить перша парадигма, адже вона використовується для вирішення задач класифікації [59].

Кероване навчання використовує набір прикладів пар $(x, y), x \in X, y \in Y$, і має на меті пошук функції $f: X \to Y$ в дозволеному класі функцій, яка відповідає цим прикладам. Іншими словами, ми хочемо вивести відображення, на яке натякають ці дані; функцію витрат пов'язано з невідповідністю між нашим відображенням та даними, і вона неявно містить апріорне знання про предметну область. Задачами, що вписуються до парадигми керованого навчання, є розпізнавання образів (відоме також як класифікація) та регресія (відома також як наближення функцій). Парадигма керованого навчання є застосовною також і до послідовних даних (наприклад, до розпізнавання писання вручну, мовлення та жестів). Його можна розглядати як навчання з «учителем» у вигляді функції, яка забезпечує постійний зворотний зв'язок стосовно якості отриманих досі розв'язків.

Автокодувальник або автоасоціатор (англ. autoencoder) – це штучна нейронна мережа, що використовується для навчання ефективних кодувань. Метою автокодувальника є навчитися представленню (кодуванню) набору даних,

зазвичай задля зниження розмірності. Нещодавно концепція автокодувальника стала застосовуватися ширше для навчання породжувальних моделей даних [53].

Архітектурно найпростішою формою автокодувальника є не-рекурентна нейронна мережа прямого поширення, що є дуже подібною до багатошарового перцептрону (БШП) із вхідним шаром, вихідним шаром та одним або декількома прихованими шарами, що з'єднують їх. Відмінності між автокодувальниками та БШП, однак, полягають в тому, що в автокодувальнику вихідний шар має таку ж кількість вузлів, як і вхідний шар, і в тому, що замість тренування передбаченню цільового значення Y для заданих входів X, автокодувальники тренують відбудові їхніх власних входів X. Таким чином, автокодувальники є моделями спонтанного навчання.

Алгоритм тренування автокодувальника може бути узагальнено як:

Для кожного входу x:

1) Виконати прохід прямого поширення для обчислення активацій на всіх прихованих шарах, а потім і на вихідному шарі, щоби отримати вихід x';

2) Виміряти відхилення x' від входу x (зазвичай застосовуючи квадратичну похибку);

3) Зворотно поширити цю похибку мережею та виконати уточнення вагових коефіцієнтів.

Автокодувальники часто тренують із застосуванням зворотного поширення (таких як метод спряжених градієнтів, найшвидший спуск тощо). І хоча вони часто є досить дієвими, існують фундаментальні проблеми із застосуванням зворотного поширення до тренування мереж із багатьма прихованими шарами. Щойно похибки зворотно поширюються до перших кількох шарів, вони стають дуже маленькими та незначними. Це означає, що мережа майже завжди навчатиметься відбудови усереднення всіх тренувальних даних. І хоча більш передові методи зворотного поширення (такі, як метод спряжених градієнтів) до певної міри можуть розв'язувати цю проблему, вони все

одно призводять до дуже повільного процесу навчання, та слабких розв'язків. Цій проблемі можна зарадити застосуванням початкових вагових коефіцієнтів, що є наближенням кінцевого розв'язку [53].

2.4 Представлення даних та знань в інформаційних системах

В ІС використовуються такі основні формалізми для подання знань:

1) продукційні правила;
2) семантичні мережі;
3) фрейми [60, 73] .

Правило продукції "якщо А, то В" ("якщо є подія А, тоді є подія В") володіє незаперечними перевагами перед іншими формалізмами, оскільки легко сприймається користувачами, зокрема, фінансовими підприємствами при вирішенні задачі класифікації шахрайства.

База знань продукційної ІС складається з безлічі правил продукції (бази правил) і кінцевого набору фактів (бази фактів).

Продукційні системи мають переваги і недоліки. До недоліків відносяться:

1) труднощі складання продукційного правила , відповідного елементу знання (потрібно, щоб розглянута область вже була достатньо вивчена, і разом з тим, рівень деталізації не повинен бути надмірно докладним);
2) труднощі запису правила з використанням єдиного формату запису ЯКЩО - ТО;
3) труднощі, пов'язана із забороною взаємного прямого виклику одного правила з іншого [83].

До переваг відносяться:

1) модульність (можливість неупорядкованого поповнення або видалення знань);
2) модифікованість (додавання нових знань не зачіпає наявних знань);

3) інтерпретованість (зрозумілість людині і машині);

4) здатність до пояснення зроблених висновків (на підставі яких ознак отримано даний висновок);

5) ефективність (висока достовірність висновків).

Використання правила продукції в ймовірнісній формі, у вигляді формули Байєса, вельми зручно і перспективно, оскільки дозволяє практично за однією і тією ж схемою обчислень здійснювати висновки як в прямому (" ознака > шахрайство"), так і в зворотному ("шахрайство > ознака") напрямах . Причиною тому - принципова особливість формули Байєса, яка встановлює зв'язок не між "причиною" і "наслідком", а між двома довільними подіями [60].

Перелік причин, що ускладнюють практичне використання Байєсівської стратегії в платіжних системах:

1) наявність статистичної залежності між ознаками;

2) необхідність знання апріорних ймовірностей P (Yj) шахрайства Yj (принаймні, такі знання дуже бажані);

3) "дефекти" даних, обумовлені неоднорідністю і неповнотою даних;

4) існування зовнішніх і внутрішніх "заважаючих" факторів.

Системи нечіткого виводу - це такі системи, в яких умови і укладення окремих правил формулюються у формі нечітких висловлювань щодо значень тих чи інших лінгвістичних змінних.

Основними етапами нечіткого висновку є:

1) формування бази правил;

2) фазифікація вхідних змінних;

3) агрегування підумови в нечітких правилах продукції;

4) активізація або композиція підвисновків;

5) акумулювання висновків в нечітких правилах продукції;

6) дефаззифікація вихідних змінних [74].

Отже, було досліджені й сформульовано методи машинного навчання для розв'язання практичних завдань, а саме моделювання й аналізу платіжних систем й транзакцій у них та виявлення шахрайства. Була розглянута й описана теоретична база, на яку спираються всі методи, а саме були визначені такі поняття як лінійна регресія, логістична регресія, дерева рішень, метод опорних векторів, градієнтний бустінг, поняття нейронної мережі та інші. Були розглянуті й описані теоретично інші, більш складні поняття й підходи машинного навчання до моделювання функціонування платіжних систем. У наступному розділі буде розглянуто застосування й реалізація сформульованих математичних моделей та методів із використанням машинного навчання та їх програмна імплементація.

РОЗДІЛ 3
АНАЛІЗ ТРАНЗАКЦІЙ ПЛАТІЖНОЇ СИСТЕМИ Й ВИЯВЛЕННЯ ШАХРАЙСТВА

3.1 Підготовка бази даних транзакцій

Для дослідження проблеми виявлення шахрайських транзакцій та знайдення рішення була отримана база даних платіжної системи з транзакціями по рахунках. База даних відображає транзакції, що були здійснені протягом 2 днів, загалом містить 284 807 транзакцій, 492 з яких є шахрайством (0.172%). База даних складається лише з числових даних. З питань конфіденційності поля бази даних є анонімізованими. Через це, немає можливості вказати опис тієї чи іншої особливості, за яке відповідає поле, та дати більш чіткий опис даних з економічної точки зору [49].

По-перше, ми повинні провести базовий, поверхневий аналіз бази транзакцій, щоб зрозуміти, з якими даними нам доведеться працювати.

Усі 28 параметрів (V1, V2, ..., V28) були отримані за допомогою перетворення методом головних компонент. Єдині 2 поля, що не були трансформовані, є "час" та "кількість". Значення "час" демонструє кількість секунд, що пройшла між даною транзакцією та першою транзакцією. Поле "кількість" демонструє суму коштів, яка пройшла транзакцію.

Перш за все, підключаємо необхідні бібліотеки та завантажуємо набір даних (Додаток В.1). Виведемо зразок запису транзакції (Додаток В.2) для демонстрації (Рис. 3.1).

Як вже зазначалось, усі 28 полів (усі не були виведені на екран) представляють собою суто числову характеристику.

	Time	V1	V2	V3	V4	V5	V6	V7
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461

Рисунок 3.1 – Представлення бази даних транзакцій

Перевіримо записи на відсутність даних (Додаток В.3) та бачимо, що дані по усім транзакціям присутні, порожніх клітинок немає.

Варто зазначити, що даний набір даних є дуже незбалансованим (Додаток Г.1), оскільки цільовий клас – шахрайські транзакції – становить лише 0.17% від усіх транзакцій (Рис. 3.2). Якщо використовувати їх для побудови моделей, ми ймовірно отримаємо безліч хибних класифікацій через перенавчання моделі. Отримана модель буде припускати, що скоріш за все транзакція є звичайною, тому що майже весь набір даних складається з таких транзакцій.

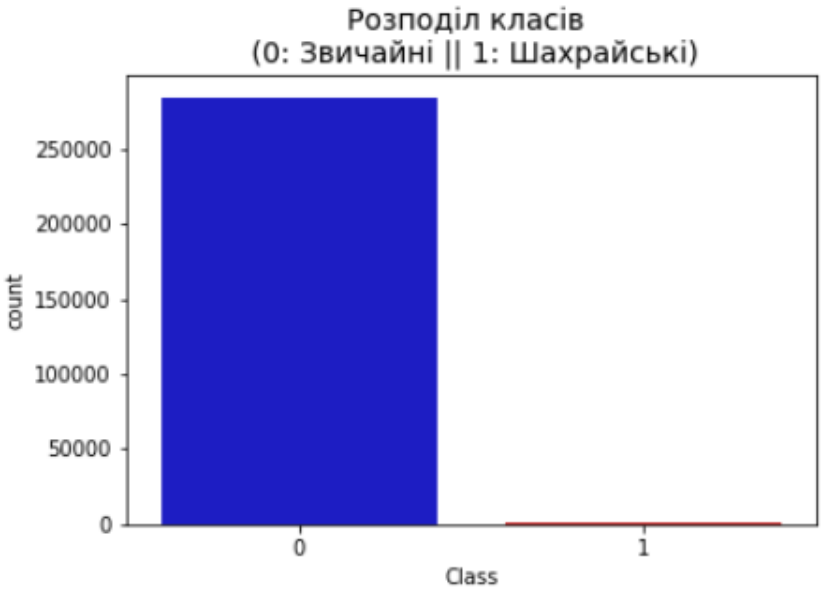


Рисунок 3.2 – Розподіл початкової бази транзакцій за класами

Як ми вже зрозуміли, на графіку звичайні транзакції значно кількісно переважують шахрайські (Додаток Г.2). Для більш чіткого розуміння даних продемонструємо розподіли записів (Додаток Г.3) за часом та сумою (Рис. 3.3).

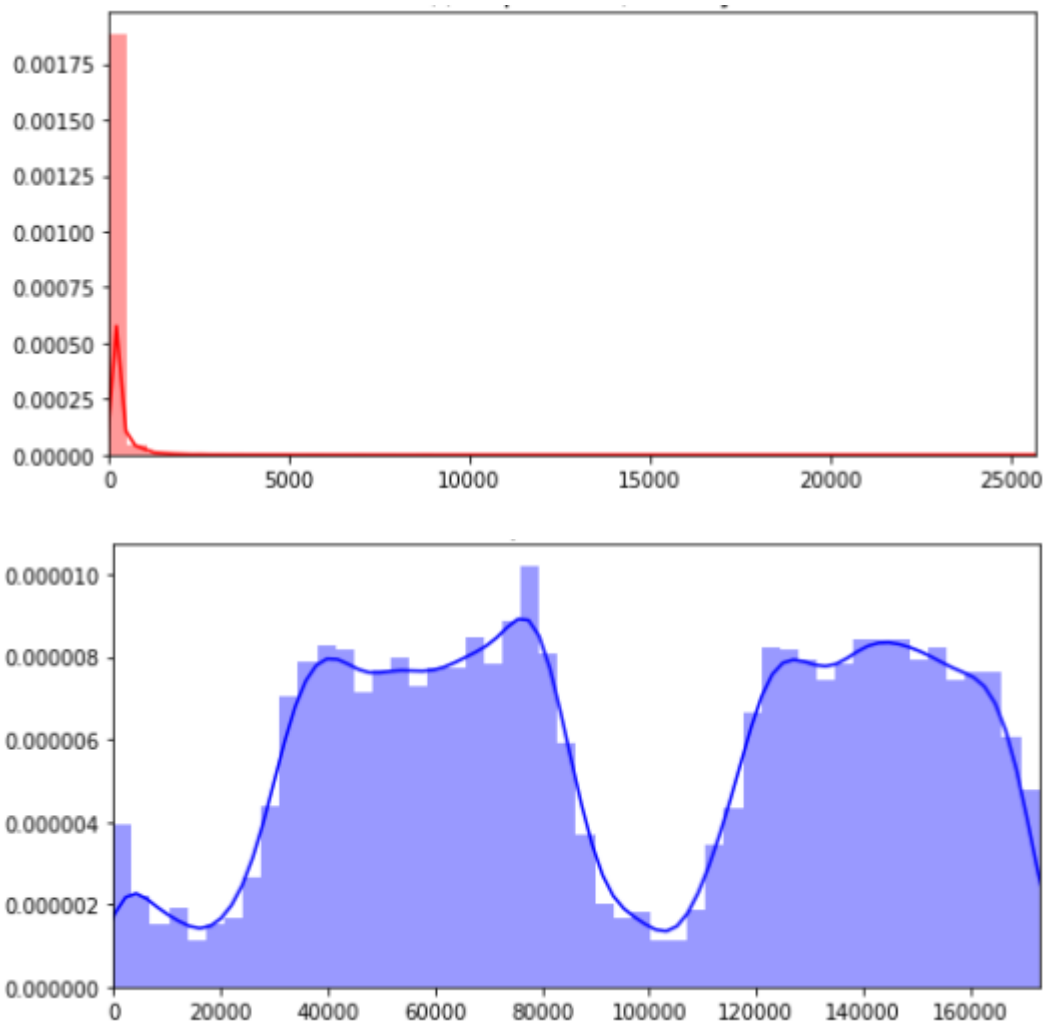


Рисунок 3.3 – Розподіл транзакцій за сумою (зверху) за часом (знизу)

Можемо спостерігати циклічність у надходженні транзакцій, адже у нічний час покупці є менш активними. Оскільки усі поля, крім "час" та "сума" були попередньо трансформовані та масштабовані, поля "час" та "сума" вибиваються із загального масштабу й можуть приймати значно більші значення за решту полів. Для подальших розрахунків зведемо усі поля до єдиного масштабу. Також варто

відзначити, що більша частина транзакцій – з невеликими сумами (до 100 у.о.), середня сума транзакції складає 80 у.о. У той же час, саме транзакції з невеликими сумами, як вияснилось, найчастіше виявляються шахрайськими.

3.2 Аналіз транзакцій та визначення стратегії виявлення шахрайства

В даній частині роботи, ми спершу масштабуємо записи колонок "час" та "сума", щоб звести усі записи до єдиного масштабу. Як пам'ятаємо, усі інші поля були вже попередньо масштабовані.

Також ми повинні створити збалансований піднабір даних з однаковою частотою шахрайських та звичайних транзакцій, що допоможе майбутнім алгоритмам демонструвати більш точні результати.

Що буде піднабором даних? У нашому випадку це буде набір даних з відношенням 50/50 звичайних та шахрайських операцій. Тобто, кількість шахрайських й звичайних операцій буде однаковою.

Навіщо створювати піднабір даних? Ми з’ясували, що початковий набір даних є дуже незбалансованим. Його використання може створити наступні проблеми:

- Перенавчання. Так як майже усі записи відносяться до звичайних, наша модель буде емпірично відносити майже кожну транзакцію як не шахрайську.
- Хибна кореляція. Хоча ми й не знаєм за що власне відповідають поля "V", безперечно буде корисно розуміти, як кожне з них впливає на цільову функцію. Знову ж таки, маючи незбалансований набір даних кореляційна матриця буде нечіткою й зміщеною у сторону звичайних транзакцій.

Перед тим, як застосовувати випадкову супердискретизацію до тренувального набору даних, ми перш за все повинні розділити початковий набір

даних на тренувальний й тестовий (Додаток Д.1). Застосовувати техніки балансування даних (супердискретизації чи субдискретизації) варто лише до тренувального набору даних для отримання моделі, тестувати ж модель необхідно на вхідному, оригінальному наборі даних.

На наступному етапі роботи ми застосуємо техніку випадкової супердискретизації (Додаток Д.2), яка полягає у видаленні тих записів з набору даних, яких більше. Таким чином, ми досягаємо відношення 50/50 шляхом виключення звичайних транзакцій (Рис. 3.4).

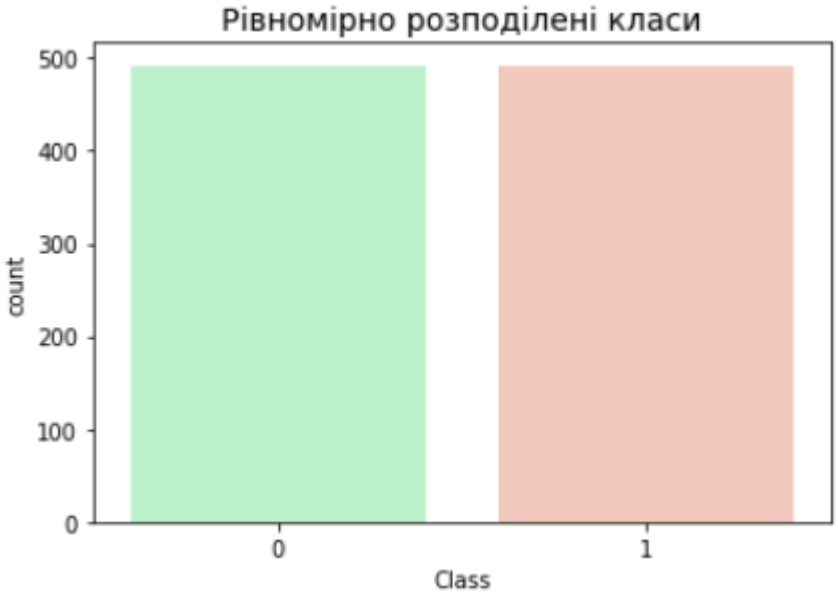


Рисунок 3.4 – Гістограма розподілення класів після супердискретизації

Кореляційні матриці є основою для розуміння даних. Для нас цікаво зрозуміти, які аргументи значно впливають на класифікацію транзакції.

Особливо показовим є співставлення матриці для збалансованого (Рис. 3.5) й незбалансованого (Рис. 3.6) наборів даних. З матриць видно, наскільки важливо використовувати правильно підготовлений набір даних.

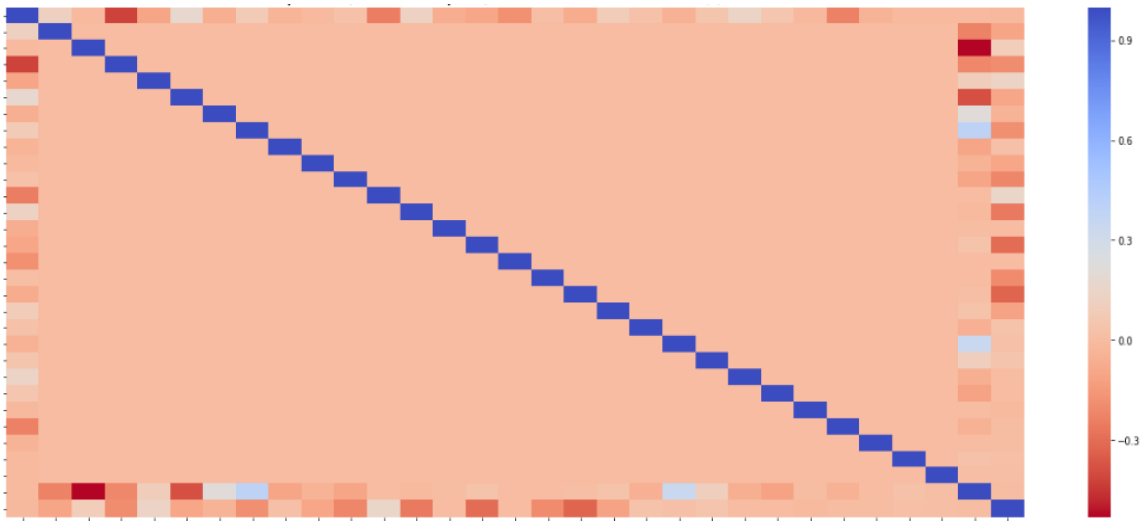


Рисунок 3.5 – Кореляційна матриця незбалансованих даних

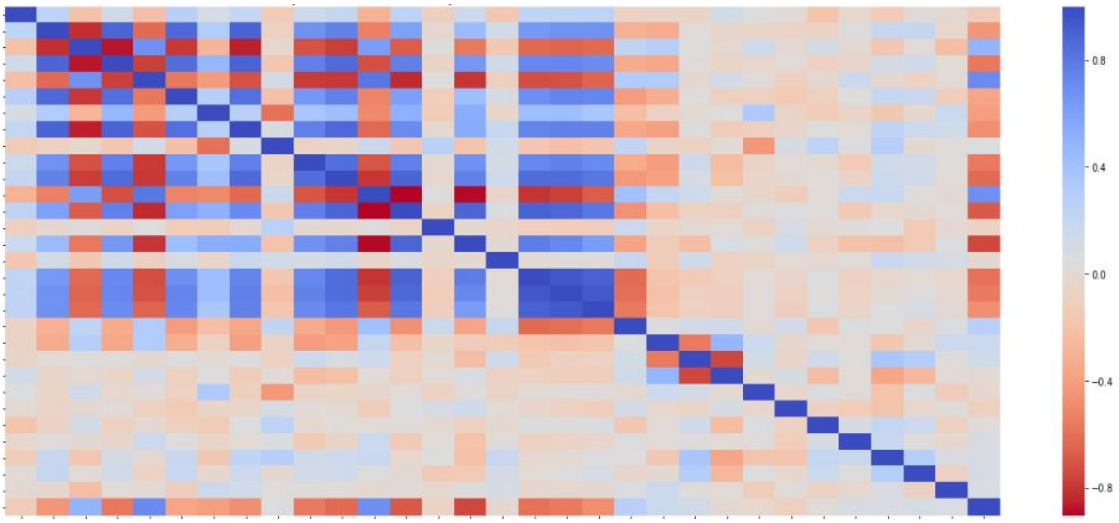


Рисунок 3.6 – Кореляційна матриця збалансованих даних

Аналіз кореляційної матриці:

- негативний зв'язок: V10, V12, V14, V17 (Рис. 3.7). Чим менше значення даних змінних, тим ймовірніше транзакція буде шахрайською.
- позитивний зв'язок: V2, V4, V11, V19 (Рис. 3.8). Чим більше є значення змінної, тим ймовірніше операція є шахрайською.

Як бачимо з Рис. 3.7, значення полів V17, V14, V12, V10 для шахрайських операцій приймають значно нижчі значення ніж для звичайних. Більш як 75%

значень в усіх випадках шахрайських транзакцій мають менше значення. З Рис. 3.8 спостерігаємо протилежну ситуацію для полів V11 та V4. Значення полів для шахрайських операцій сягають більших значень ніж у випадку зі звичайними.

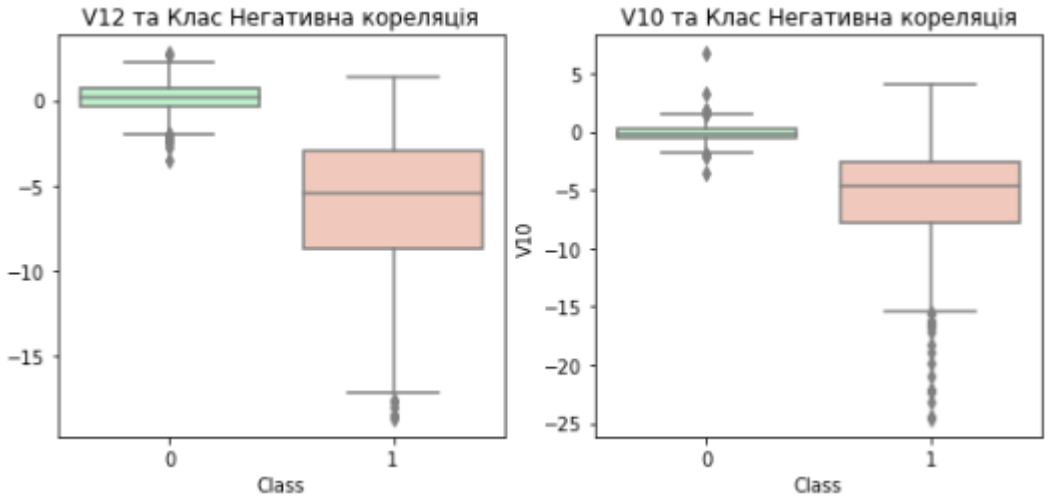


Рисунок 3.7 – Значення полів V17 та V14 з негативною кореляцією в залежності від класу

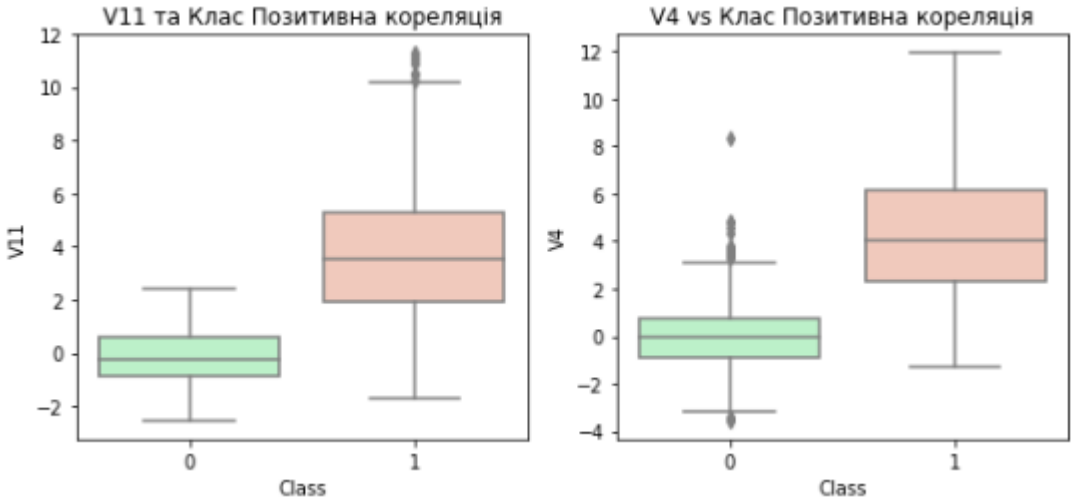


Рисунок 3.8 – Значення полів V11 та V4 з позитивною кореляцією в залежності від класу

Також варто відзначити, що перш за все нас цікавлять «центрові» значення полів, тобто ті значення, які знаходяться в межах від 1го квартиля до 3го квартиля. Саме вони будуть найвагоміше впливати на побудову й навчання моделей.

Дані були проаналізовані й готові до навчання. У наступних секціях ми застосуємо як прості (базові) методи побудови класифікаторів, так і комплексні, які представляють собою композиції простих. Ці 2 групи методів будуть розглянуть в окремих підрозділах, оскільки мають значні відмінності у побудові та застосуванні (інтуїтивно, базові моделі більш прості у застосуванні та прості для навчання).

Отже, для класифікації шахрайських транзакцій ми будемо використовувати такі методи:

I. Прості регресійні та дискретні моделі:
    1. Логістична регресія;
    2. Метод k-найближчих сусідів;
    3. Метод опорних векторів;
    4. Дерева прийняття рішень;
II. Комплексні моделі (ансамблі):
    1. Градієнтний бустінг, а саме 3 імплементації:
        1) GBM (Gradient Boosting Machine);
        2) xgboost;
        3) LightGBM;
    2. Нейронні мережі:
        1) базова нейронна мережа, навчена на супердискретизованих даних;
        2) базова нейронна мережа, навчена на субдискретизованих даних;

3) нейронна мережа з використанням автоасоціаторів (autoencoders) у комбінації базовим лінійним класифікатором.

Усі класифікатори й принципи їх роботи були описані у попередніх розділах. Ініціалізуємо моделі (Додаток Ж.1) Зазначимо, що усі моделі класифікаторів не є частиною Python, а потребують додатково завантаження й встановлення.

3.3 Застосування базових регресійних та дискретних моделей для виявлення шахрайства

У даній секції ми застосуємо 4 базові підходи до побудови моделей класифікаторів й вирішимо який з них найкраще виявляє шахрайство. Зазначимо, що ми будемо застосовувати й інші підходи (комплексні ансамблі, які будуть тими чи іншими комбінаціями простих моделей), проте дані методи будуть розглянуті в наступному розділі окремо через значну відмінність у побудові та застосуванні, порівняно з базовими дискретними та регресійними методами, розглянутими в даному розділі.

Перед початком ми повинні розділити наші дані на тренувальні й тестові.

Для кожного класифікатора побудуємо модель й знайдемо її точність (Додаток Ж.2). Спочатку передаємо моделі набір даних, на якому модель буде навчатися. Після проходження всіх етапів навчання оцінюємо моделі за допомогою функції кросс-валідації, яка буде повертати перехресно перевірену середньоквадратичну похибку (cross-validation rmse error), так що ми зможемо оцінити наші моделі й обрати найкращу.

Продемонструємо отримані результати, що побудовані на основі збалансованих даних (Табл. 3.1). Точність перевірялась на тренувальній вибірці, а оцінка кросс-валідації - на тестовій.

Таблиця 3.1 – Порівняння точності й оцінки кросс-валідації моделей

	Логістична регресія	Метод опорних векторів	Метод k-найближчих сусідів	Дерева ухвалення рішень
Точність	94.0%	92.0%	92.0%	90.0%
Оцінка кросс-валідації	93.52%	93.78%	93.14%	91.84%

Як бачимо з результатів, логістична регресія показала найкращу точність з оцінкою в 94%. Це є тренувальний результат, який був отриманий з оцінки того, як точно модель визначає шахрайство саме на тренувальній вибірці. Для більш точного результату перевіримо отримані моделі на тестовій вибірці (Додаток Ж.3). Пам’ятаємо, що це все ще балансована вибірка, тому результат буде неточним.

Проаналізуємо й порівняємо графіки навчальних кривих для усіх 4 моделей.

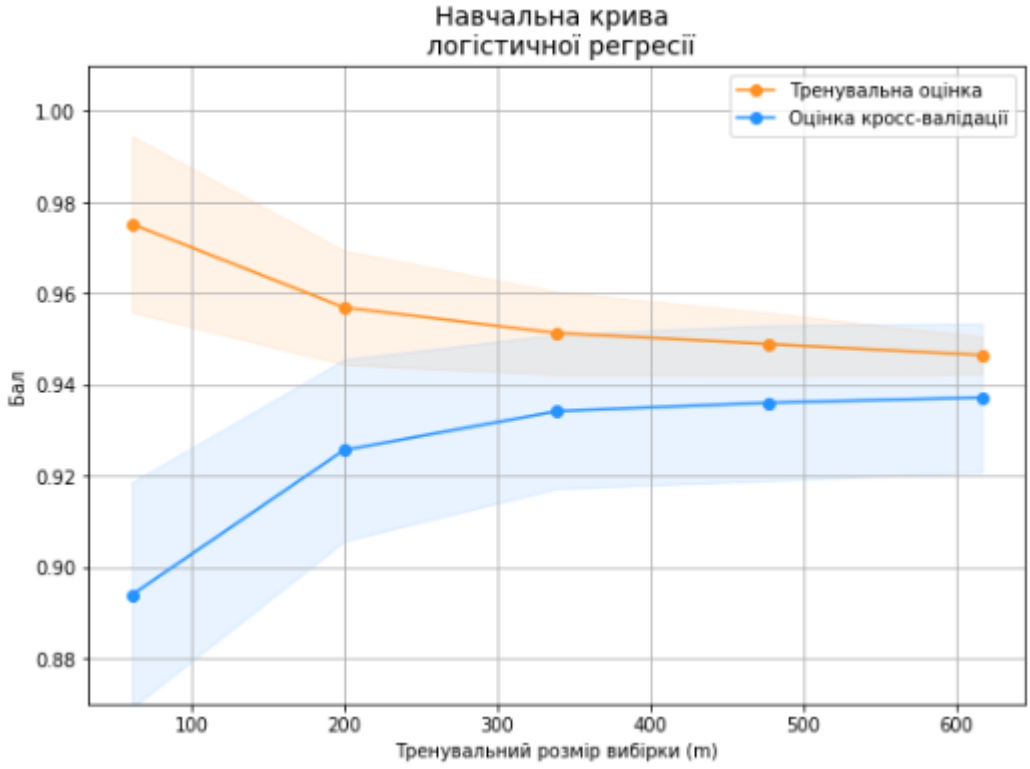


Рисунок 3.9 – Навчальна крива логістичної регресії

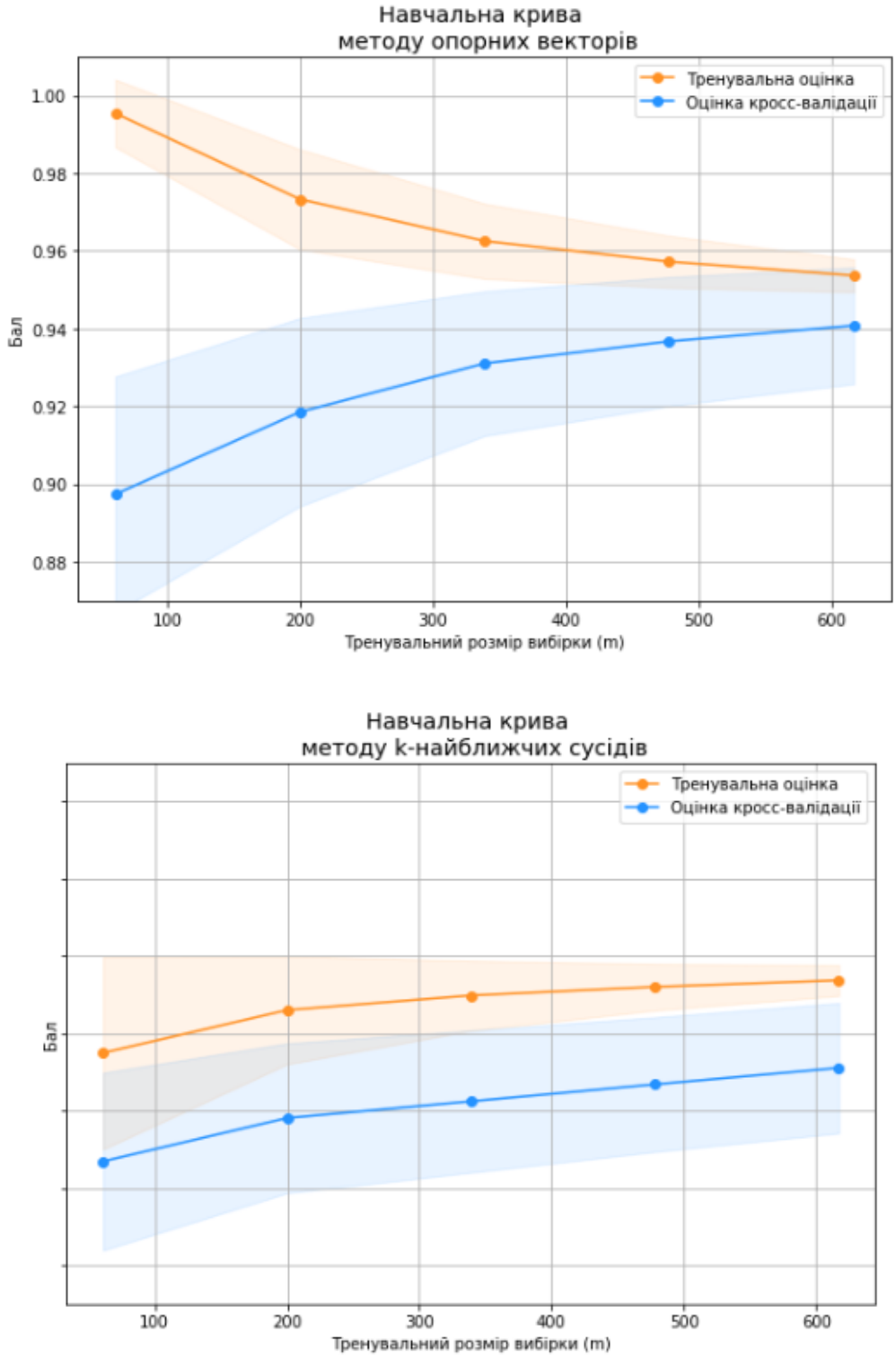


Рисунок 3.10 – Навчальна крива методу опорних векторів (зверху) та методу k-найближчих сусідів

Логістична регресія пройшла близько 600 ітерацій навчання. Саме на даному етапі оцінка крос-валідації (тестова оцінка) досягла свого максимального значення (Рис. 3.9). Надалі поліпшення точності не відбувалось, а тому навчання було припинене.

Ситуація з навчанням методу k-найближчих сусідів та методу опорних векторів близька до ситуації с логістичною регресією (Рис. 3.10). Ці 3 моделі досягали максимуму оцінки своєї кросс-валідації приблизно на 600 ітерації.

З Рис. 3.10 також видно, що загалом результат поліпшувався до 600 ітерації у випадку з методом k-найближчих сусідів та методом опорних векторів.

Дерева рішень досягли свого найкращого результату значно раніше, на 200 ітерації, проте й найкраща точність у даного метода найменша (Рис. 3.11).

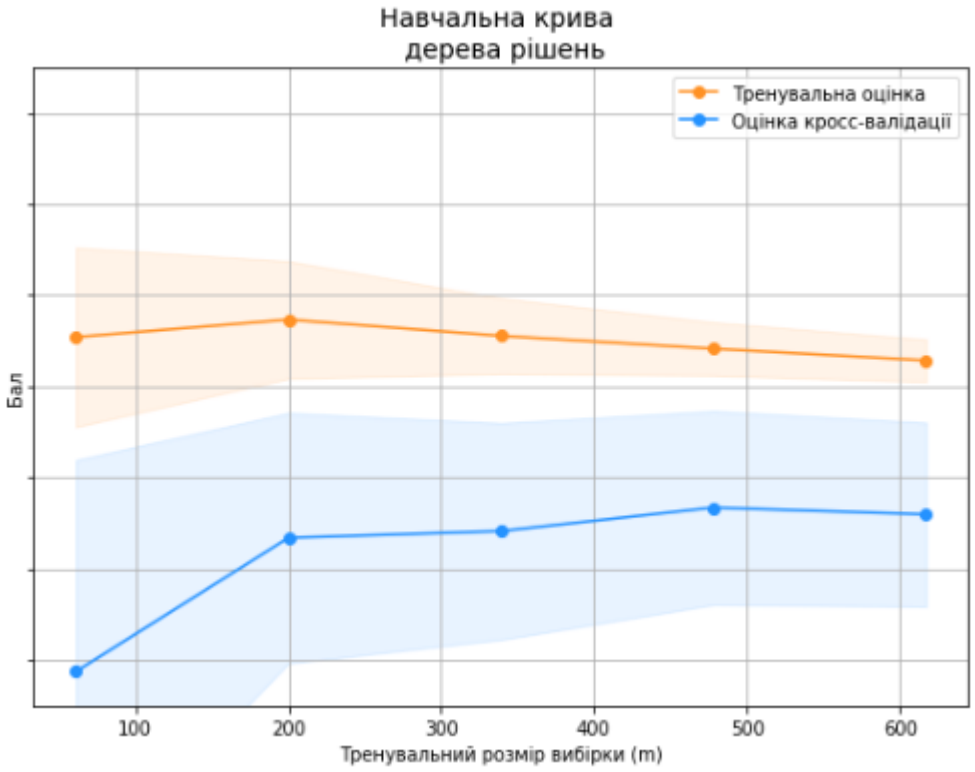


Рисунок 3.11 – Навчальна крива дерева рішень.

Як бачимо з отриманих результатів, найкраще себе продемонстрував метод логістичної регресії з результатом 94% на тренувальній вибірці та 93.52% на

тестовій вибірці (найкращий результат оцінювався як максимальне середнє арифметичне між даними 2 показниками). Метод k-найближчих сусідів та метод опорних векторів також продемонстрували досить точний результат, а метод опорних векторів показав на тестовій вибірці навіть кращий результат, ніж логістична регресія – 93.78%.

Для більш наочної демонстрації результатів виведемо таблицю помилок для виявлення шахрайства методом логістичної регресії (Рис. 3.12). У лівому верхньому та правому нижньому квадратах (жовтий колір) правильні результати, у інших квадратах (чорний колір) неправильні:

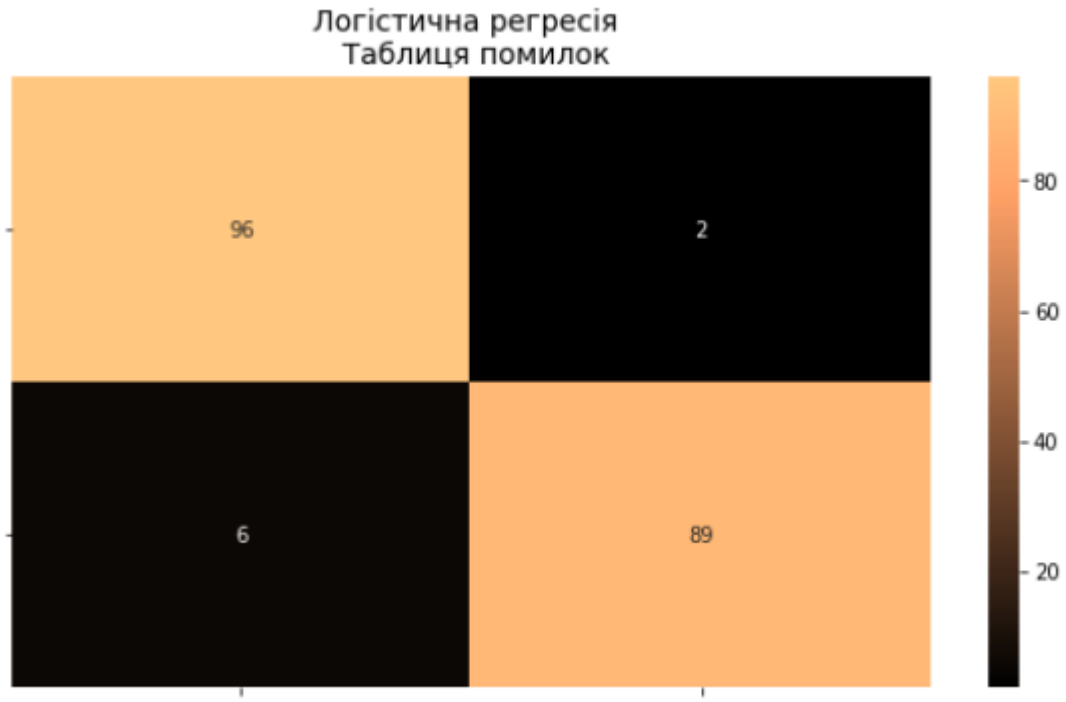


Рисунок 3.12 – Таблиця помилок логістичної регресії

Як бачимо з Рис. 3.12, даний метод правильно виявив 96 + 89 = 185 транзакцій. Інші 8 транзакцій потрапили у невідповідні групи, тобто були прогнозовані неправильно.

3.4 Застосування градієнтного бустінгу та нейронних мереж для виявлення шахрайських транзакцій

У даному підрозділі будуть розглянуті ансамблі (комплексні моделі) для вирішення задачі класифікації шахрайських транзакцій.

При побудові класифікатора, що повинен оперувати незбалансованими даними та якісно виявляти аномалії в наборах даних, має зміст застосувати техніку градієнтного бустінгу. Для цього використаємо та порівняємо 3 найпопулярніші імплементації: GBM, xgboost та LightGBM. Градієнтний бустінг буде проводитись на тих же самих даних, на яких оперували попередні моделі.

Зазначимо, що реалізація GBM максимально близька до математичного опису градієнтного бустінгу. Таким чином, GBM використовує математичні залежності градієнтного бустінгу у чистому вигляді, без додаткових налаштувань, фіч та пристосувань. GBM реалізовано у пакеті sklearn, проте даний метод загалом демонструє дещо гірші результати якраз через застосування прямолінійного підходу, без пристосування до наданих даних. Також GBM значно повільніший у навчанні ніж LightGBM та xgboost, які є сучасними імплементаціями техніки градієнтного бустінгу.

Для демонстрації результату побудуємо ROC-криві – графік, що дозволяє оцінити якість бінарної класифікації, відображає співвідношення між часткою об'єктів від загальної кількості носіїв ознаки, вірно класифікованих до загальної кількості об'єктів, що не несуть ознаки, помилково класифікованих, як такі, що мають ознаку. Також відома як крива похибок. Аналіз класифікацій із застосуванням ROC-кривих називається ROC-аналізом. Кількісну інтерпретацію ROC дає показник AUC – площа, обмежена ROC-кривою і віссю частки помилкових позитивних класифікацій. Чим вище показник AUC, тим якісніше діє класифікатор, при цьому значення 0,5 демонструє непридатність обраного методу класифікації (відповідає звичайному вгадуванню).

Як бачимо з Рис. 3.13, найгірше себе показав GBM. Причиною цього є те, що даний метод використовує відносно застарілі алгоритми, які є зацикленими та погано оптимізованими. Xgboost та LightGBM показали загалом схожі, й досить гарні результати. Проте LightGBM обходить за рахунок швидкості навчання та необхідних ресурсів, xgboost модель є більш «важкою».

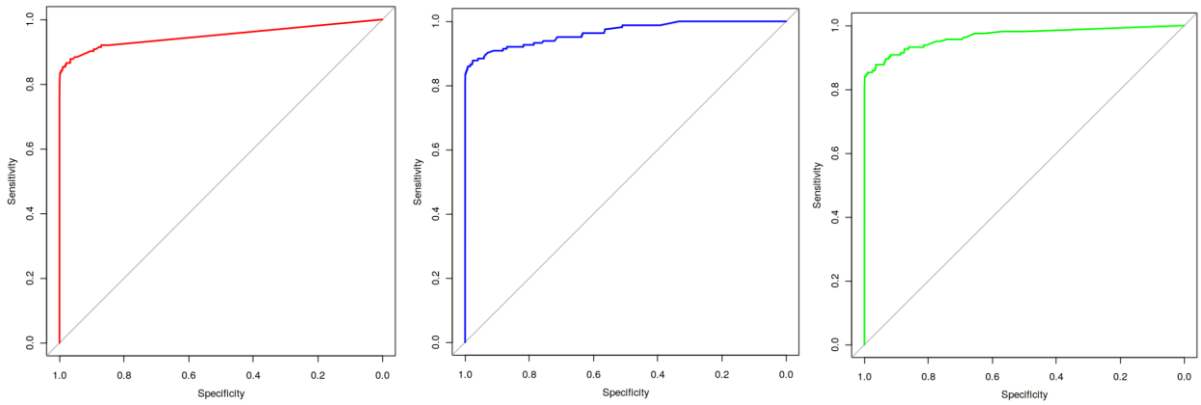


Рисунок 3.13 – ROC-криві GBM (зліва), xgboost (по центру), LightGBM (праворуч)

При застосуванні техніки градієнтного бустінгу було отримано наступні результати (надані оцінки, отримані на тестовому наборі даних):

1. LightGBM – 96.67%
2. xgboost – 96.54%
3. GBM – 94.98%

Отже, можемо зробити наступні висновки, базуючись на отриманих результатах технік градієнтного бустінгу (Табл. 3.2).

Перейдемо до створення класифікаторів шахрайських транзакцій методами нейронних мереж. Як зазначалось, ми побудуємо 3 моделі: проста нейронна мережа навчена на субдискретизованих даних; проста нейронна мережа навчена на супердискретизованих даних; нейронна мережа з використанням автоасоціаторів (autoencoders).

Таблиця 3.2 – Порівняння технік градієнтного бустінгу

	Переваги	Недоліки
GBM	• Класичний прямолінійний підхід до градієнтного бустінгу	• Відсутній ранній вихід з процесу навчання; • Повільне навчання; • Низька точність.
xgboost	• Висока точність та надійність;	• Повільний відносно lightGMB;
LightGBM	• Висока ефективність навчання; • Низьке використання оперативної пам’яті; • Висока точність • Підтримка паралельного навчання • Працює з великими базами даних	• Відсутність документації

Структура базової нейронної мережі: проста модель, яка складається з одного вхідного шара, одного прихованого шара на 32 вузла, та один вихідний шар, який може приймати одне з двох можливих значень: 0 чи 1.

Ми проведемо два навчання нейронної мережі: перше за допомогою супердискретизації, а інше за допомогою субдискретизації. Іншими словами, у першому випадку ми звузимо наші дані до пропорції 50/50, тобто випадковим

чином викинемо значну частину звичайних транзакцій. У іншому випадку ми розширемо наші дані, додавши нові записи шахрайських даних, які будуть отримані на основі вже наявних шахрайських даних випадковим чином.

Для навчання нейронної мережі було проведено 20 ітерацій на відповідному наборі даних. Після навчання нейронної мережі ми перевіряємо її на оригінальному наборі даних і порівнюємо результати між самими нейронними мережами так і найкращими класифікаторами.

Як бачимо з Рис. 3.14, нейронна мереж на супердискретизованих даних значну частину звичайних транзакцій (вісь Y) помістила у клас шахрайських, проте лише 1 шахрайська транзакція пройшла. Загалом оцінка нейронної мережі склала 93.1%.

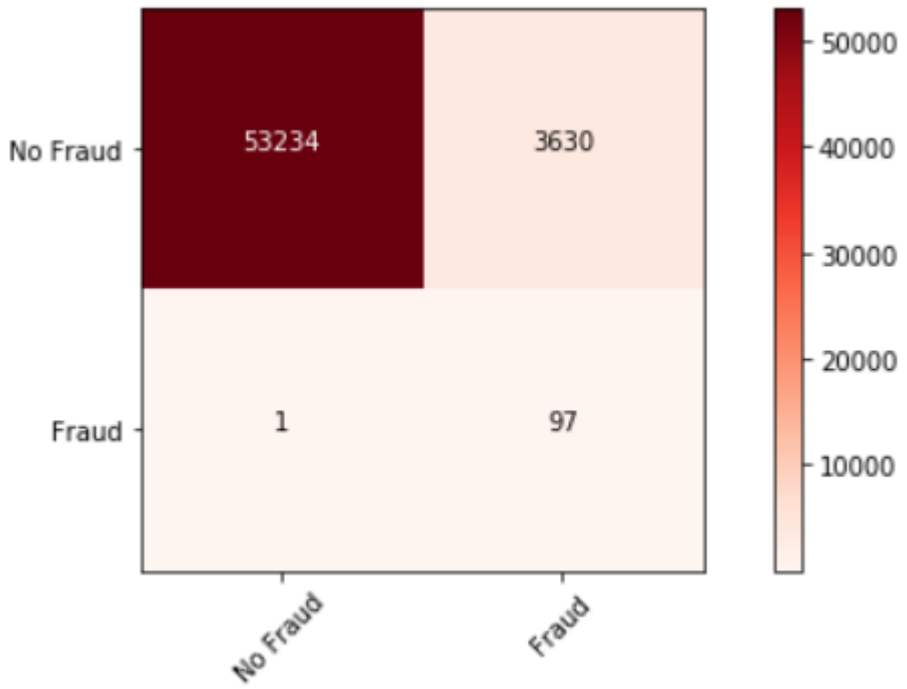


Рисунок 3.14 – Матриця помилок для навчання за допомогою супердискретизації

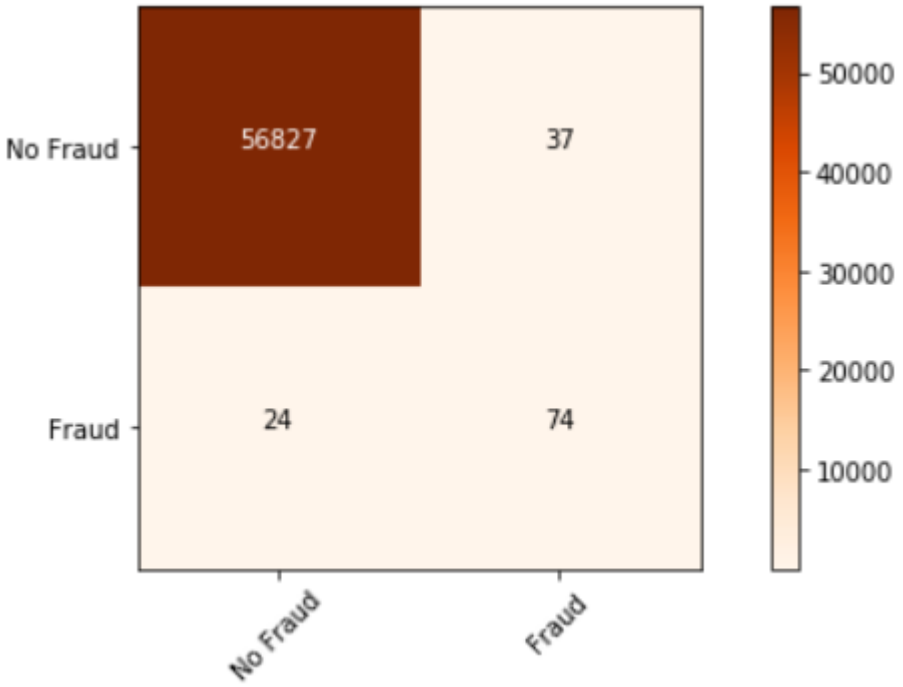


Рисунок 3.15 – Матриця помилок для навчання за допомогою субдискретизації

Субдискретизація (розширення даних) показала кращі результати як серед нейронних мереж, так і серед усіх моделей загалом, продемонструвавши 99.9% правильних класифікацій (Рис. 3.15). Проте варто відзначити, що уже 24 шахрайські транзакції пройшли, а тому відсоток зупинення саме шахрайських транзакцій нижчий.

Також, існує техніка, що дозволяє навчати нейронні мережі на відносно незбалансованих даних, використовуючи гібридне навчання. Наш підхід базується на використанні нейронної мережі для трансформації та перетворення вхідних даних за допомогою автоасоціаторів (autoencoders, після застосування якого ми навчаємо простий лінійний класифікатор.

Для побудови моделі ми будемо використовувати всі шахрайські транзакції та 3000 звичайних транзакцій, обраних випадковим чином, для спрощення.

Давайте візуалізуємо характер шахрайських та не-шахрайських транзакцій за допомогою T-SNE. T-SNE - це метод декомпозиції набору даних, який зменшує

розмірність даних і надає тільки верхні n-компонентів з максимальною інформацією.

Кожна точка представляє транзакцію. Операції без шахрайства представлені у вигляді зеленого кольору, а трансакції з шахрайством - як червоні (Рис. 3.16). Дві осі являють собою компоненти, витягнуті трансформаціями T-SNE.

З наведеного графіка (Рис. 3.16) можна спостерігати, що існує багато транзакцій, які не є шахрайськими, і які дуже близькі до шахрайських операцій, тому їх важко точно класифікувати. У цьому нам стануть у нагоді автоасоціатори. Згадаємо, що автоасоціатор (або автокодер) - це штучна нейронна мережа, що використовується для навчання ефективних кодувань. Нашою метою є представлення (кодування) набору даних задля зниження розмірності.

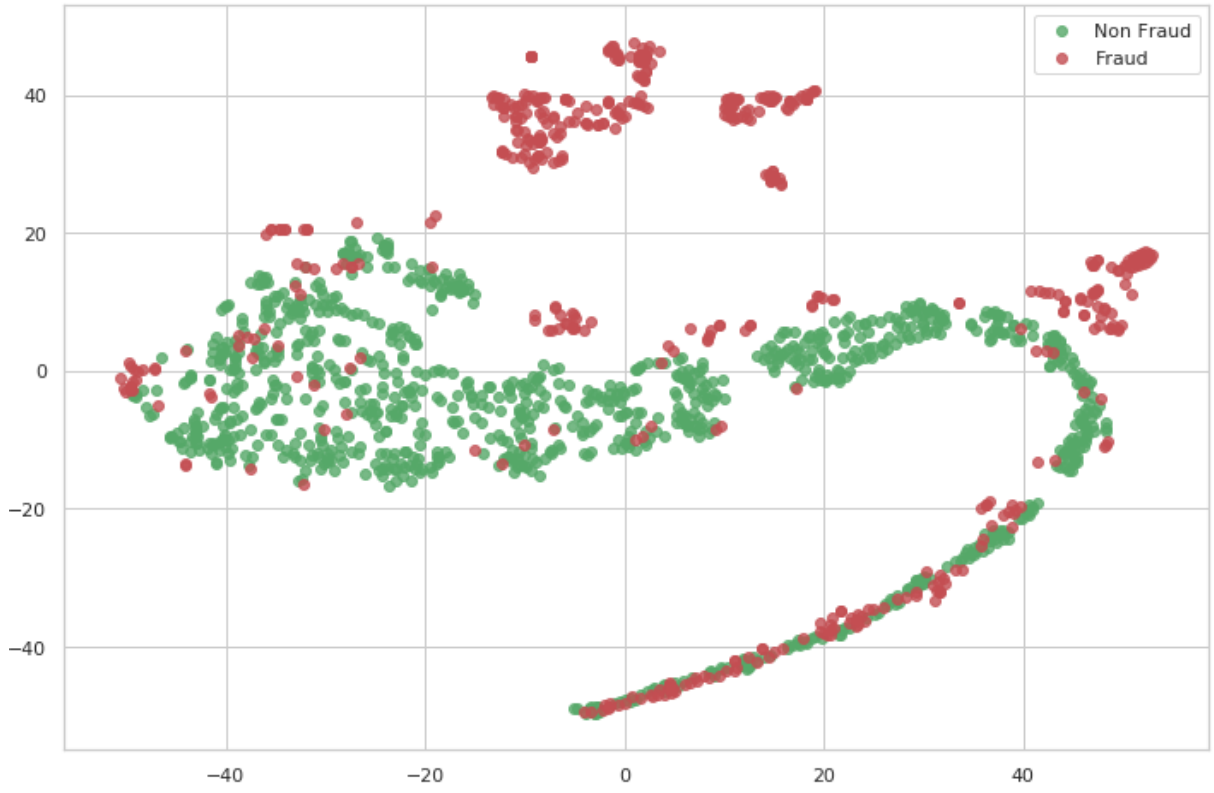


Рисунок 3.16 – T-SNE представлення початкових даних.

Спочатку ми створимо автоасоціатор, якому надамо лише звичайні транзакції, без шахрайства. Модель спершу буде намагатися вивчити випадки, які не стосуються шахрайства. Ця ж модель потім буде використана для формування уявлень про випадки шахрайства, і ми очікуємо, що вони відрізнятимуться від тих, які не є шахрайськими.

Краса цього підходу полягає в тому, що нам не потрібно надто багато зразків даних для вивчення позитивних варіантів (шахрайських). Також ми будемо використовувати 2000 значень випадків без шахрайства для навчання автоасоціатору. Крім того, нам не потрібно запускати цю модель на великій кількості ітерацій. Пояснення: Вибір невеликих зразків даних з оригінального набору даних базується на припущенні, що характеристики одного класу (без шахрайства) будуть відрізнятися від характеристик інших (шахрайство). Щоб відрізнити ці характеристики, нам потрібно показати автокодерам тільки один клас даних. Це пояснюється тим, що автокодер намагатиметься вивчити лише один клас і автоматично розмежувати інший клас.

Ми можемо спостерігати, що зараз шахрайські операції та транзакції без шахрайства є досить видимими і лінійно відокремленими (Рис. 3.17). Тепер нам не потрібна будь-яка складна модель для класифікації цього, навіть прості моделі можна використовувати для прогнозування.

Застосуємо базовий лінійний класифікатор. Отримана точність на трансформованому наборі даних – 98.28%.

Отже, можна відзначити, що комплексні моделі справляється із задачами класифікації значно краще, ніж прості моделі (що було досить очікувано). Найкращий абсолютний результат із блокування шахрайських транзакцій продемонструвала нейронна мережа на супердискретизованих даних - лише 1 шахрайська транзакція пройшла, проте було заблокована відносно велика кількість звичайних транзакцій. Загальна оцінка встановилась на значенні 93.1% правильних класифікацій. Градієнтний бустінг показав найкращий результат у

96.67% на реалізації моделі за допомогою LightGBM. До переваг градієнтного бустінгу варто віднести швидкість навчання у порівнянні з нейронними мережами. Автоасоціатор продемонстрував можливості трансформації та підлаштування даних для простих моделей. Від так, завдяки автоасоціатору звичайний лінійний класифікатор набрав 98.28% правильних класифікацій. А найкраще себе продемонстрував себе класифікатор на основі нейронної мережі, навченої на субдискретизованих даних - 99.9%. Хоча варто відзначити, що цього результату модель досягла не завдяки ефективному блокуванню шахрайських транзакцій, а більш лояльному ставленню до них - пропустивши 24 з 98 шахрайських транзакцій. Усі оцінки отримані на основі тестового набору даних.

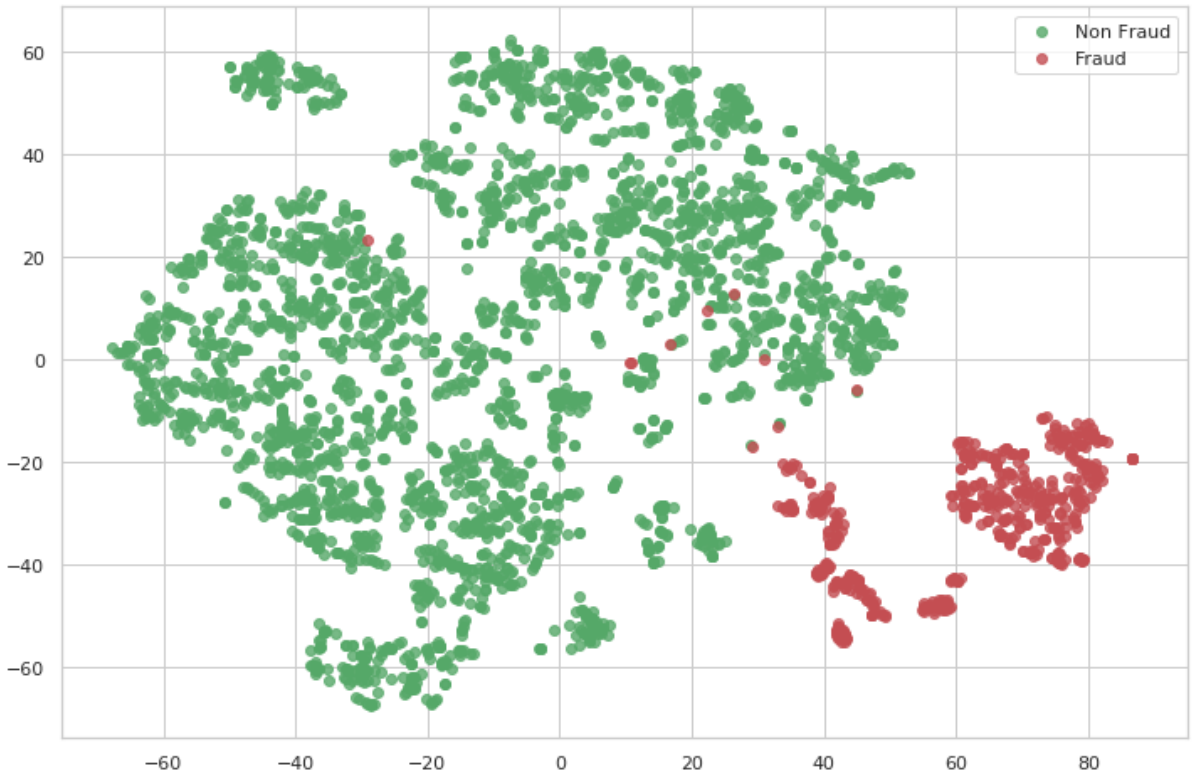


Рисунок 3.17 – T-SNE представлення даних, які були опрацьовані автоасоціаторами

У наступному розділі буде розглянуто побудову автоматизованої інформаційної системи (програмного продукту), яка, базуючись на отриманій технології виявленні шахрайства в платіжних системах та використовуючи її у якості алгоритмічного ядра, здатна опрацьовувати транзакції та виявляти аномалії у режимі реального часу та під високим навантаженням.

РОЗДІЛ 4
РОЗРОБКА ТЕХНОЛОГІЇ ВИЯВЛЕННЯ ШАХРАЙСТВА В ПЛАТІЖНИХ СИСТЕМАХ ТА КОНЦЕПЦІЯ СТВОРЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ НА ЇЇ ОСНОВІ

4.1 Алгоритм побудови технології класифікації шахрайства та виявлення аномалій у системах

За літературними даними [7 - 16] встановлено, що в основі виявлення шахрайства платіжних системах - задачі класифікації. Тому в основі побудови технології виявлення шахрайства в платіжних сервісах запропоновано закласти розробку ефективної та оптимізованої моделі класифікації шахрайства у платіжних системах з позиції усіх етапів дослідження. Для реалізації повного циклу технології виявлення шахрайства в платіжних сервісах обрано мову програмування Python3 за простоту в синтаксисі, широку підтримку спільноти програмістів та величезну кількість доступної документації.

4.1.1 Програмне середовище та вхідні дані транзакцій

Технологія базуватиметься на мові програмування Python та її екосистемі. Такий вибір обгрунтований тим, що Python дозволяє швидко писати код та тестувати гіпозети, може бути запущений майже на будь-якому пристрої. Крім того, найбільшою перевагою є наявність безлічі бібліотек та фреймворків, які суттєво зменшують час розробки.

Встановлено, що для реалізації рішення завдання доцільно обрати такі бібліотеки:

- pandas - обробка та аналіз даних, надає структури даних та операції для маніпуляції з таблиця та часовими рядами, бібліотека оптимізована для роботи з великими обсягами даних;
- numpy - додає підтримку великих, багатовимірних масивів та матриць, разом із об’ємною колекцією високорівневих математичних функцій для операцій з ними;
- matplotlib - пропонує об’єктно-орієнтований програмний інтерфейс для вставки різного роду візуалізацій;
- scikit-learn - впроваджує різні алгоритми для вирішення задач класифікації, регресійного аналізу, кластеризації, включаючи метод опорних векторів, К-найближчих сусідів, random forests та інші;
- xgboost - фреймворк для роботи з алгоритмом екстремального градієнтного підсилення.

Використання означених бібліотек сформує готове до роботи налаштування, яке буде служити свого роду фундаментом для рішення [67-73].

Рішення буде реалізоване, базуючись на наборі даних транзакцій анонімної платіжної системи. Варто відзначити, що технологія реалізована гнучким, масштабованим чином, а тому дане рішення може бути адаптоване до будь-якого набору даних, що має схожу структуру (записи транзакцій). Опишемо нашу структуру даних для більш чіткого розуміння.

Опорною конструкцією для аналізу є об'єкт з прямокутними даними тобто, фактично, є таблицею. Загалом таблиця складається з 10 стовпців (атрибутів) та 6 362 620 записів - тобто наш датасет є матрицею розмірності 6362620 на 10. Також у наборі даних немає порожніх клітинок - усі атрибути усіх записів заповнені. Кожен рядок - запис окремої транзакцій. Стовпці - атрибути транзакції, до яких входять:

- step - відображає одиницю часу в дійсності. Транзакції, що перебувають у датасеті, були здійснені протягом 743 годин. Тобто, "1" - перша година спостереження, "743" - 743-тя година спостереження;
- type - тип транзакції. Можливі типи: CASH-IN (поповнення рахунку готівкою), CASH-OUT (зняття готівки), DEBIT (зарахування коштів на рахунок), PAYMENT (оплата товарів чи послуг), TRANSFER (переказ коштів);
- amount - сума транзакції;
- nameOrig - клієнт, що ініціалізував транзакцію;
- oldBalanceOrig - початковий баланс клієнта перед транзакцією;
- newBalanceOrig - баланс клієнта після транзакції;
- nameDest - ID (індифікаційний номер) отримувача транзакції;
- oldBalanceDest - початковий баланс отримувача;
- newBalanceDest - баланс отримувача після транзакції;
- isFraud - ідентифікатор чи є транзакція шахрайською (1) чи ні (0)

Отже, аналіз буде реалізовано на наборі даних банківських операцій, що були здійснені фізичними особами самотужки - наприклад, за допомогою мобільного банкінгу, картки або терміналу - тобто такі операції, що не були проведені за допомогою банку чи іншого наглядового органу. Тому цей набір даних особливо корисний у сучасності - адже рівень оцифрованості грошей підвищується, а частка операцій, здійснених готівкою чи в касі банку, - скорочується. Тому даний набір дає змогу виявити сучасні патерни та закономірності у банківських транзакціях.

4.1.2 Дослідницький аналіз та підготовка даних

Перед тим, як приступати до створення концепту технології та побудови моделі, зупинимось на ручному дослідженні даних. Це важливий крок у побудові будь-якої моделі, адже це дозволити виділити важливі фактори, зосередитись на них та не зважати на так званий "білий шум". Підготовка даних базується на:

1. Визначенні, який тип транзакцій найчастіше є шахрайським (та відповідно адаптуємо датафрейм) (Додаток А.1)

Як бачимо, усі шахрайські транзакції є оба переказом коштів між рахунками (TRANSFER) або зняттям готівки (CASH_OUT). У подальших кроках доцільно позбутись записів усіх інших типів транзакцій - вони будуть лише заважати моделі, не даючи їй змоги ідентифікувати шахрайські транзакції.

Для наглядності та оптимізації бінарно закодуємо тип транзакції: "0" відповідатиме типу TRANSFER, "1" - типу CASH_OUT. (Додаток А.2)

2. Обгрунтуванні аномалій при переказі коштів.

Логічно, що при проведенні переказу (amount != 0) початковий рахунок отримувача (oldBalanceDest) не може бути порожнім у той самий час, як і кінцевий рахунок (newBalanceDest) - вони повинні відрізнятися на суму транзакції (Додаток А.3).

Встановлено, що майже кожна 2-га аномальна транзакція є шахрайською (що не дивно), у той час як серед звичайних транзакцій таких - близько 0.62%. З одного боку, це - гарна новина, адже отримано чіткий індикатор того, що операція - шахрайська. Проте, для побудови моделі такий розклад скоріше шкідливий - наявність чітко вираженого індикатора агресивно класифікує набір даних, а тому шахрайським операціям, які під таку класифікацію не підпадають буде легко загубитися. Тому доцільно позначити відповідні баланси в таких записах як "-1" замість "0" - це дозволить моделі бути більш розумною, адже таким чином ми наявно виділимо такі записи від решти (Додаток А.4).

3. Аналізі аномальних транзакції з боку відправника.

З точки зору відправника (oldBalanceOrig - newBalanceOrig) такі випадки також трапляються, тому аналогічно відокремимо і їх, проте позначимо такі транзакції по-іншому, аби не плутатит із аномальними транзакціями відправника (Додаток А.5).

4. Кількісному виявленні певної аномалії в транзакції

Від так, логічно, що різниця між початковим балансом та кінцевим балансом повинна відрізнятись на суму транзакції (зі знаком мінус для відправника і знаком плюс для отримувача). Введемо 2 нові колонки та підрахуємо помилку в балансі отримувача (errorBalanceDest) та відправника (errorBalanceOrig) - на скільки відрізняється очікуване сальдо рахунків від фактичного (Додаток А.6).

4.1.3 Візуалізація даних транзакцій

Найкращих спосіб підтвердити, що набір даних містить достатньо інформації для побудови моделі, яка може здійснювати правильні передбачення, - це візуалізувати відмінності між шахрайськими транзакціями та звичайними.

1. На графіку (Рис. 4.1) продемонструємо, як звичайні та шахрайські транзакції мають різні проекції (відбитки) їх дисперсії протягом часу спостереження. З графіку зрозуміло, що шахрайські транзакції розподілені більш рівномірно у часі ніж звичайні (шахрайські слабо корелюються із часом, можуть відбуватись завгодно коли, у той час як звичайні транзакції відбуваються приблизно в однакових проміжках часу).

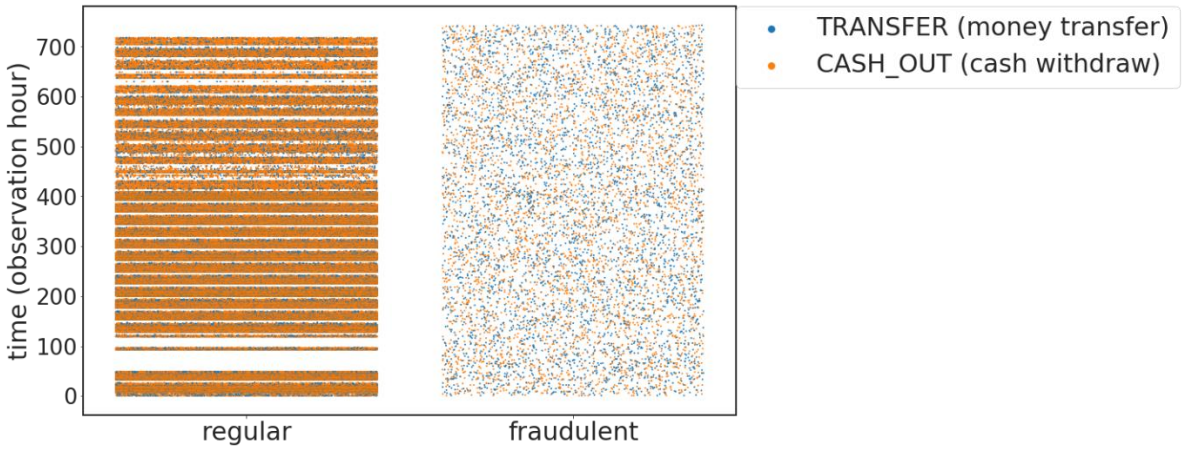


Рисунок 4.1 – Смугастий розподіл у звичайних операціях проти однорідного у шахрайських (протягом часу)

Крім того, визначено, що кількість операцій зняття готівки (CASH_OUT) переважають кількість переказів (TRANSFER) у звичайних транзакціях, у той час як у шахрайських - їх кількість приблизно однакова. Це дозволяє зробити наступне припущення - переказ коштів підробляється частіше.

2. Графіки (Рис. 4.2 and Рис. 4.3) знизу показують, що хоч наявність шахрайства у транзакції може бути виявлена базуючись на початковому атрибуті "amount", новий введений атрибут "errorBalanceDest" (відмінність у очікуваному балансі та фактичному) виступає кращим класифікатором. Від так, з Рис. 4.2 видно, що звичайні транзакції мають значно більшу кількість переказів (трансферів), ніж шахрайські, які до того ж не обмежені у сумі транзакції. У свою чергу, шахрайські транзакції збалансовані по типу (переказ коштів чи зняття готівки) та до того ж не перевищують ліміт.

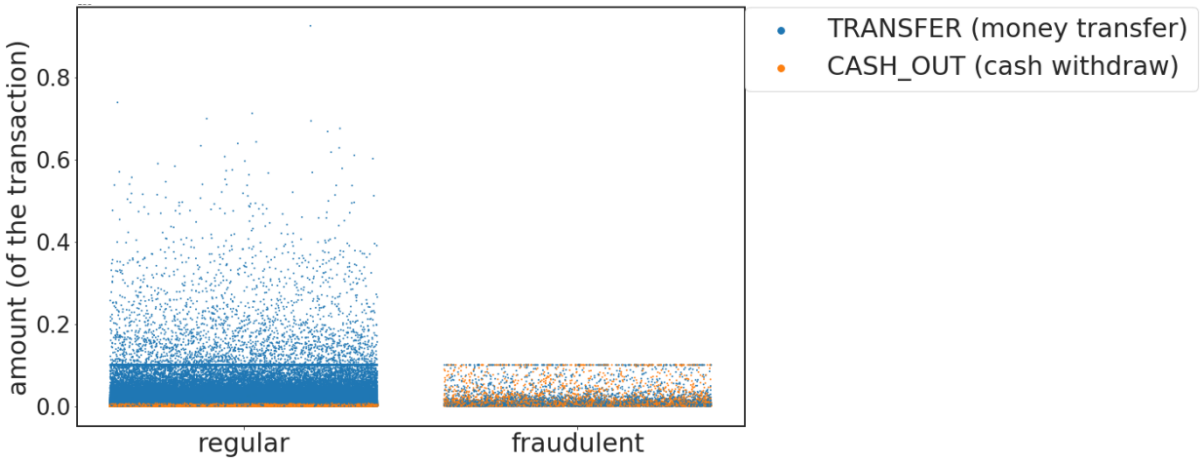


Рисунок 4.2 – Розподіл суми транзакції серед звичайних та шахрайських транзакцій

Розподіл транзакцій базуючись на атрибуті "errorBalanceDest" майже чітко проводить відмінність між звичайними та шахрайськими транзакціями (Рис. 4.3) - як бачимо, у останніх зазвичай ця помилка негативна. Тобто у звичайних транзакціях, сума oldBalanceDest + amount зазвичай більша ніж newBalanceDest, що є допустимим (наприклад, певна частина суми була витрачена на комісію), і

також, у шахрайських транзакціях ця сума зазвичай є меншою за новий баланс, що підтверджує аномалію та дозволяє ідентифікувати такі транзакції як шахрайські.

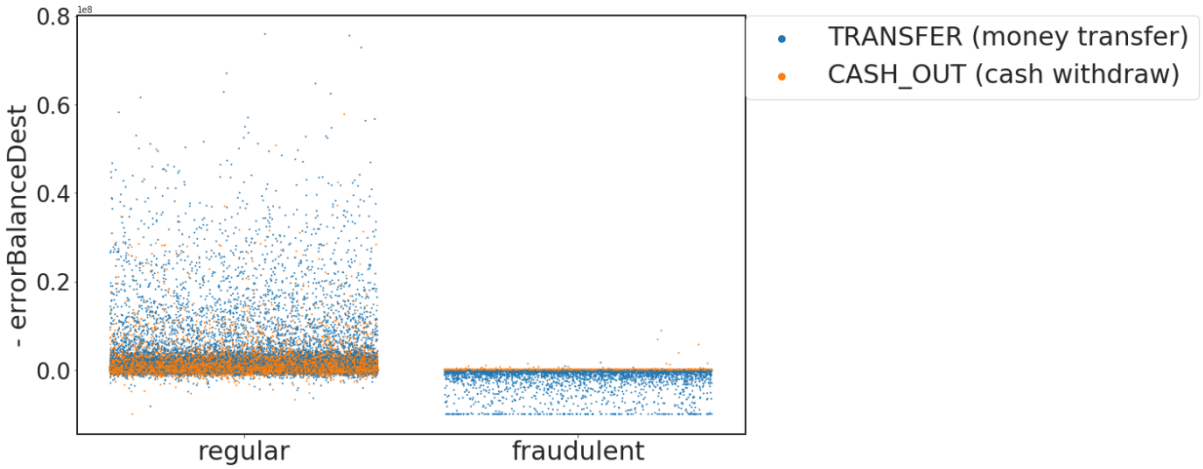


Рисунок 4.3 – Протилежна полярність розподілів помилки балансів у звичайних та шахрайських транзакціях

3. 3D графік (Рис. 4.4) найкраще демонструє відмінності у розподілі звичайних та шахрайських транзакцій.

Визначено, що початковий атрибут "step" (транзакції за часом) слабо відображає відношення транзакції до того чи іншого класу. Також варто відзначити смугастий характер розподілу даних з плином часу.

4. Для фінальної візуальзації закономірностей у наборі даних застосуємо простий, проте надійний спосіб візуалізації - теплову карту, яка продемонструє кореляцію між атрибутами (Рис. 4.5).

Позбудимось атрибутів, які є нерелевантними (нейтрального кольору) та не впливають на результат (Додаток А.7)

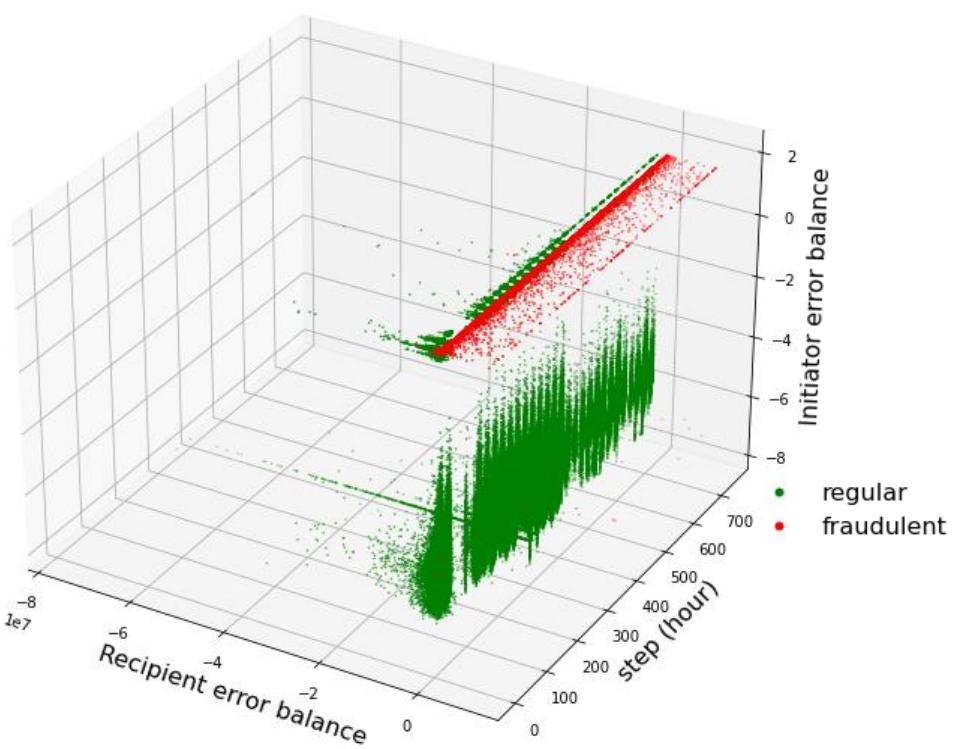


Рисунок 4.4 – Відмінність у розподілі шахрайських та звичайних транзакцій базуючись на помилках балансів

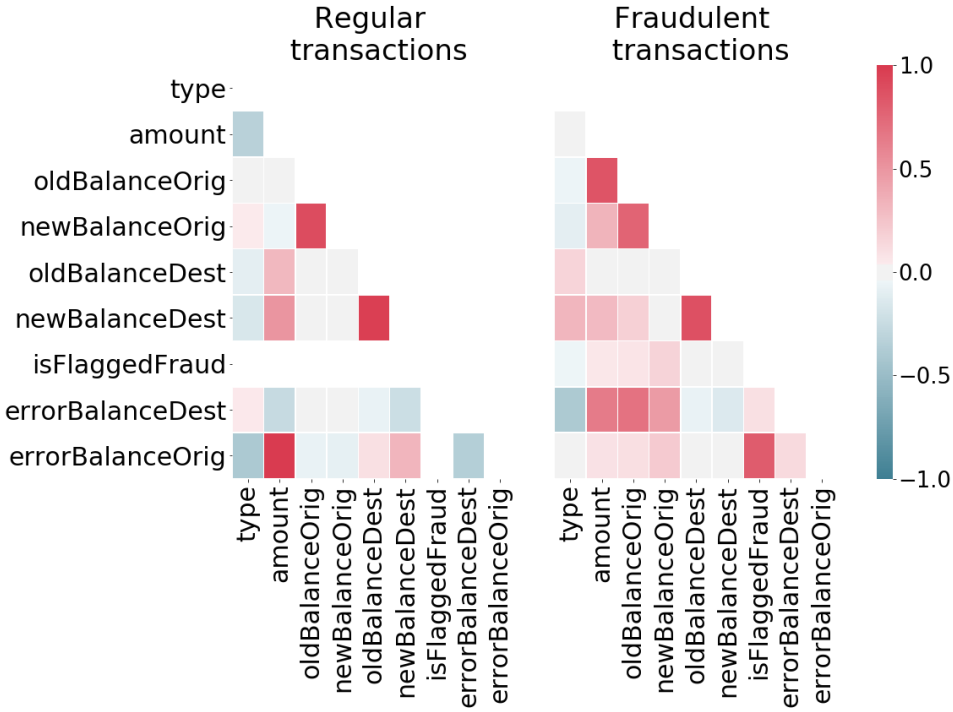


Рисунок 4.5 – Теплові мапи кореляції атрибутів у звичайних та шахрайських транзакціях

Отримаємо чистий, підготовлений до побудови моделі датасет. Оскільки ми попередньо визначили критично важливі атрибути моделі та логічно виокремили їх на програмному рівні, переходимо до етапу реалізації. Виведемо отримані атрибути, на основі яких будемо будувати модель. (Додаток А.8)

4.2 Створення концепту та реалізація моделі виявлення шахрайства в незбалансованому наборі даних

Виявивши та сформовавши атрибути, які дозволяють чітко відокремити шахрайські транзакції від звичайних доцільно врахувати при створенні ТПВШ-дані є незбалансованими, а саме перекошеними у бік звичайних, тобто звичайних транзакцій значно більше, ніж шахрайських, що може результувати й у перекошеність моделі. Частка шахрайських транзакцій становить лише 0.3%.

Також важливо визначити, яку метрику використовувати для оцінювання на моделі та, власне, за допомогою якого алгоритму реалізувати модель. Тобто необхідно не тільки створити саму модель, а й могти якісно її аналізувати, щоб отримати реальний результат.

Вибір метрики: так як дані дуже нездалансовані, для оцінювання моделі будемо використовувати площу під кривою точності відгуку (AUPRC), а не звичну площу під кривою робочої характеристики отримувача (AUROC), тому що перша більш чутлива до відмінностей між алгоритмами та налаштуванням їх параметрів.

Вибір алгоритму: Перший підхід до роботи з незбалансованими даними - збалансувати їх, відкинувши більшість класів перед застосуванням алгоритму (так називаєма субдискретизація, або андер-семплінг). Недоліком субдискретизації є те, що модель, навчена таким чином, не буде добре працювати з реальними перекошеними тестовими даними, оскільки майже вся інформація була відкинута.

Кращим підходом може бути розширення записів, що відносяться до класу меншості, наприклад, за допомогою техніки синтетичного розширення (додавання) вибірки меншості (SMOTE) (міститься в бібліотеці “imblearn”). Було створено 5 різних моделей, орієнтовуючись на виявлення аномалій та навчання зі вчителем: логістична регресія, К-найближчих сусідів, метод опорних векторів (SVM), класифікатор Байєса [12, 17]. Та найкращий результат досягається при застосуванні алгоритму, заснованому на ансамблях дерев рішень, що ефективно працює на незбалансованих даних. Такі алгоритми не тільки дозволяють побудувати модель, яка може впоратися з потенційно відсутніми значеннями в наших даних, але вони ще й мають найменший час процесінгу даних, що дозволятиме швидше аналізувати результат, тобто потенційна інформаційна система працювати швидше. Серед цих алгоритмів (що базуються на ансамблях дерев рішень) є 2 найбільш ефективних - Random Forest та XGBoost, та останній алгоритм градієнтного бустінгу все ж показує кращий результат. Крім того, XGBoost дозволяє зважувати позитивний клас (наявність шахрайства) більш ефективно у порівнянні з негативним класом (відсутність шахрайства) - що дозволяє більш продуктивно опрацьовувати незбалансовані дані.

Розділимо початковий набір даних на тренувальний та тестовий у пропорції 80:20 (Додаток А.9).

Ініціалізуємо класифікатор виявлення шахрайства, реалізованого на базі градієнтного бустінгу (XGBoost) та знаходимо його оцінку за допомогою згаданого раніше методу AUPRC (Додаток А.10).

Побудований алгоритм за методом AURPC має оцінку 0.9986, що говорить про дуже високу ефективність класифікатора.

Вичислимо власне точність моделі - відсоток транзакцій, які були правильно класифіковані. Тобто, до правильно класифікованих відносимо такі, що: 1) класифіковані шахрайськими та насправді є шахрайськими, або 2) класифіковані як звичайні та насправді є звичайними. Відповідно, інші: 1) ті, що

шахрайські, проте класифіковані як звичайні, або 2) звичайні, та були класифіковані як шахрайські відходять у категорію визначених неправильно. Точність моделі буде визначатись як відношення суми правильних класифікацій до загальної кількості записів транзакцій. Також побудуємо матрицю невідповідностей для наочного представлення (Рис. 4.6).

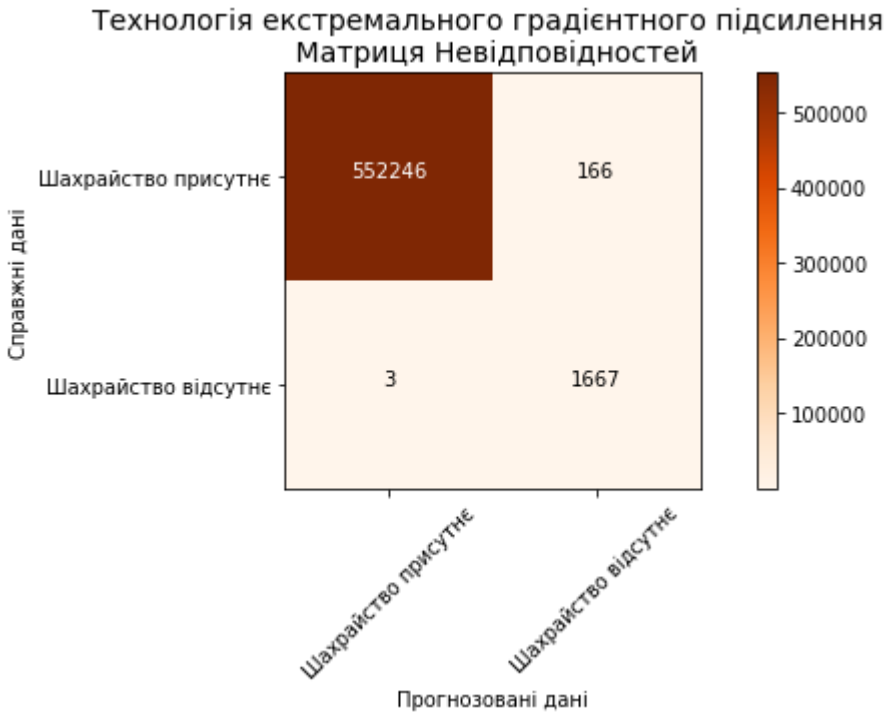


Рисунок 4.6 – Матриця невідповідностей реалізованої технології екстремального градієнтного підсилення

Отже, точність моделі, реалізованої за допомогою технології екстремального градієтного підсилення склала 99.97%.

Які ж атрибути сигналізують про шахрайську операцію? Продемонструємо ефект кожного атрибуту на кінцевий прогноз у графіку нижче. Як бачимо, штучно вирахувана змінна "errorBalanceOrig" (помилка балансу відправника) найбільше свідчить про шахрайство у транзакції, порівняно із іншими атрибутами (Рис. 4.7).

Для наочного представлення, щоб реалізована модель не була "чорною скринькою" - візуалізуємо прийняття рішень моделлю. Завдяки тому, що рішення базується на деревах рішень, візуалізація є простою та зрозумілою. Від так, отримана модель класифікує транзакції керуючись наступними рішеннями (Додаток Б).

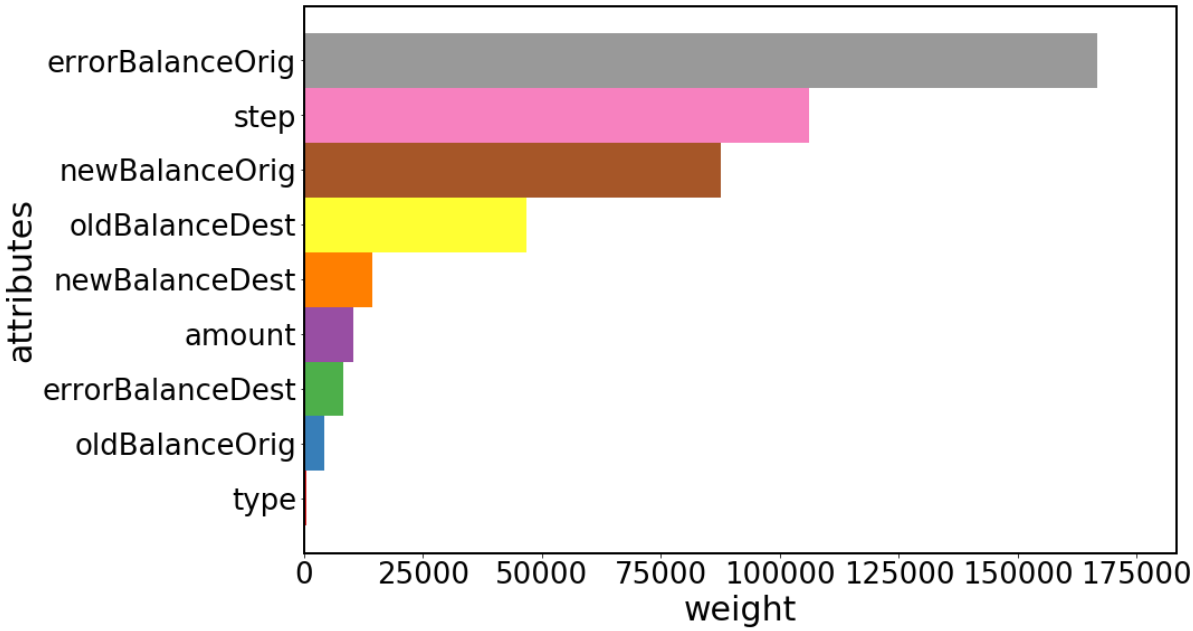


Рисунок 4.7 – Порядок важливостей атрибутів у реалізованій моделі

4.3 Розробка концепції створення автоматизованої системи виявлення шахрайства в платіжних системах

4.3.1 Постановка технічного завдання та вихідні дані

Цифрова трансформація суспільства сприяє автономності здійснення операцій клієнтами. Візити до установ для здійснення фінансових операцій здійснюються все рідше, а все більшого попиту набувають платіжні системи та сервіси.

Сприянням цього є зростання числа онлайн-платежів, сервісів інтеграції таких платежів у фінансових установах, інтернет-магазинах, логістичних

компаніях, страхових полісах, торгових майданчиках, тощо. Це пов’язано із зручністю використання, високою швидкість здійснення фінансових операцій, простотою, прозорістю, контролем за платежами. Але, крім переваг сучасних платіжних систем та сервісів, існує ризик – не збереження цілісності транзакцій. Бізнес та споживачі товарів та послуг все частіше стикається із такими ризиками. Такі ризики можна ідентифікувати як ризик шахрайства. Наслідком є те, що споживачі не отримують замовлені товари чи послуги, втрачають кошти із своїх рахунків. З позиції бізнесу це відображається фінансовими втратами, зменшенням лояльність клієнтів, що веде до цілковитої втрати клієнтів.

В Україні у 2020 році збитки від бізнесових шахрайських операцій становили понад 1 млрд грн. Більше того, Україна майже на 50% випереджає США за відсотком шахрайських операцій (на 10 000 транзакцій), і цей розрив лише зростає, вказуючи на те, що нові рішення потребує не тільки галузь в цілому, але вони особливо потрібні в Україні. Щоб запобігти цьому потрібно трансформувати підходи та засоби для моніторингу, виявлення та контролю незаконних дій в платіжних сервісах. Без сумніву, найкращий спосіб боротися з шахрайством – запобігти. Тому доцільно розвивати науково-практичні підходи попередження шахрайських операцій при здійсненні транзакцій [79].

Проблема, яку намагається вирішити дана робота, полягає в тому, що хоч вже і є побудовані моделі та алгоритми машинного навчання, які дозволяють досить точно вирішувати проблему виявлення шахрайства в платіжних системах, проте, в сучасному світі та за сучасних умов (враховуючи великі потоки даних, високе навантаження, вимоги до оптимізації та перформансу системи) недостатньо лише побудувати модель як, але й потрібно правильно реалізувати її в прикладному середовищі, щоб отриманий програмний продукт задовольняв усі вимоги сучасного світу. Тобто проблема виявлення шахрайства в платіжних системах полягає не тільки в побудові самого алгоритмічного ядра, яке, базуючись на вхідних даних, може виокремлювати шахрайство, але й у побудові

надійної, витривалої автоматизованої системи, яка в режимі реального часу, за умови високого навантаження, здатна керувати потоками даних та ефективно оперувати алгоритмічним ядром (реалізованими раніше моделями) системи. Така система має бути направлена на конкретизовані умови окремого бізнесу та представляє як науковий, так і практичний інтерес.

Створення автоматизованої системи, яка в режимі реального часу здатна аналізувати вхідних потік даних (транзакцій) і класифікувати їх у 2 класи: звичайну чи шахрайську. Запропонувати принципи, які будуть враховувати корпоративні особливості компаній при виявленні шахрайських транзакцій.

У відповідності до поставленої мети сформовано наступні задачі дослідження:

- розробити життєвий цикл побудованого програмного продукту;

- визначити технічні проблеми при розробці автоматизованої системи та запропонувати їх рішення;

- оцінити впровадження розробленої автоматизованої системи у бізнес.

Результат роботи стане наявність прототипу автоматизованої системи, в якому будуть зазначені його архітектура, програмна інфраструктура, інтерфейси, тощо.

Визначення та налаштування інфраструктури, необхідної для побудови автоматизованої системи виявлення шахрайства в платіжних системах. Поширення та зростання автоматизованих систем, перехід від стихійної автоматизації до планового розвитку корпоративних IT-систем, застосування проєктних і бізнес-орієнтованих технологій створили передумови для виявлення шахрайських транзакцій.

4.3.2 Побудова та розгортання автоматизованої системи виявлення шахрайства

Встановлено, що розробка автоматизованої технології для запобігання шахрайських операцій супроводжується труднощами, які пов’язані із:

- вибором моделі;

- розгортанням моделі у вигляді автоматизованої технології;

- впровадженням автоматизованої технології у практику використання бізнесових установ.

Доцільно реалізовувати систему, яка вбудовується в загальну автоматизовану структуру компанії. Тому запропоновано побудову хмарного веб-сервісу, в який буде повністю виділений бізнес-процес аналізу даних і інтерпретації результату: чи є дана транзакція підозрілої чи ні. Хмарні технології допомагають бізнесу істотно пришвидшити реагування на такі виклики як масштабність та доступність та суттєво оптимізувати витрати. Фахівці з розробки й експлуатації створюють зручне для взаємодії середовище між розробниками та бізнесом для підвищення ефективності, скорочення циклу розробки та швидшого виходу продукту на ринок. Це поєднання дає змогу досягати значних змін у підході до розробки хмарних технологій, швидше реагувати на потреби підприємства, послідовно інтегрувати набуті знання, с суттєво скоротити витрати на низку процесів: від тестування до розгортання [35].

До переваг варто віднести:

- підвищена масштабованість: забезпечення швидких темпів росту завдяки підвищенню ефективності використання ресурсів та можливостей центрів обробки даних;
- покращена продуктивність: надання гнучкої інфраструктури для пришвидшення виходу на ринок завдяки усуненню ізольованих процесів;
- швидка оптимізація витрат завдяки скороченню або ліквідації втрат ресурсів інфраструктури

- скорочення часу виходу на ринок: міграція в хмару дозволяє скоротити час, що витрачається на створення IT-інфраструктури.

Наступним кроком став вибір платформи. Встановлено, що раціонально реалізувати рішення розгортання системи на базі платформи Amazon Web Services (AWS) – це найповніша та широко прийнята у світі хмарна платформа, яка пропонує понад 200 повнофункціональних послуг із центрів обробки даних у всьому світі. Мільйони клієнтів – включаючи динамічні стартапи, найбільші підприємства та провідні державні установи – використовують AWS, щоб знизити витрати, стати гнучкішими та швидше впроваджувати інновації [30].

Платформу AWS обрано з позиції оцінки її переваг, а саме:

- простоти використання – платформа AWS дозволяє швидко і безпечно розміщувати на хостингу як існуючі, так і нові додатки на основі моделі SaaS. Для роботи з платформою хостингу додатків AWS можна використовувати консоль управління AWS або API веб-сервісів з докладною документацією;
- гнучкості – AWS дозволяє вибрати операційну систему, мову програмування, платформу інтернет-додатків, бази даних та інші необхідні сервіси. Тобто дана платформа забезпечує отримання віртуального середовища для завантаження програмного забезпечення, необхідного для виявлення шахрайства. Це спрощує процес міграції необхідних додатків і зберігає можливість для створення нових рішень;
- економічності – оплата здійснюється тільки за обчислювальну потужність, обсяг сховища і інші використовувані ресурси без довгострокових контрактів або попередніх зобов'язань;
- надійності. Дана платформа є віртуальною основою багатомільярдного інтернет-бізнесу Amazon.com, якість якої підтверджується практикою використання;
- масштабованість та висока продуктивність – такі інструменти AWS як Auto Scaling і Elastic Load Balancing забезпечують масштабування. Завдяки

великій інфраструктурі Amazon є доступ до обчислювальних ресурсів і ресурсів сховищ саме тоді, коли вони будуть потрібні.

- безпека – AWS використовує комплексний підхід до безпеки і зміцненню інфраструктури, включаючи фізичні, операційні та програмні засоби.

Імплементація автоматизованої системи виявлення шахрайства за допомогою хмарних технологій.

Концепт імплементації автоматизованої системи виявлення шахрайства (Payment Transactions Fraud Detection – PTFD) у платіжних системах за допомогою платформи AWS буде наступний. Рішенням є розгортання моделі машинного навчання (ML) та приклад набору даних транзакцій, щоб навчити модель розпізнавати моделі шахрайства [37]. Реалізовані раніше моделі можуть навчатися самостійно, що дозволятиме адаптувати їх до нових, невідомих патернів та закономірностей. Це рішення доречно використовувати для автоматизації виявлення потенційно шахрайської діяльності та відправлення цієї діяльності для перевірки.

Система PTFD дозволяє запускати автоматизовану обробку транзакцій на прикладі набору даних або на власному наборі даних. На діаграмі (Рис. 4.8) представлена архітектуру розгортання.

Це рішення включає шаблон AWS CloudFormation, який розгортає приклад набору транзакцій з кредитними картками, що міститься в кошику Amazon Simple Storage Service (може бути замінений на будь-який набір даних транзакцій), та екземпляр Amazon SageMaker, який навчає контрольовану та некеровану модель навчання на наборі даних та розгортає дві кінцеві точки. На основі прикладних даних генерується безперервний потік запитів на класифікацію транзакцій. Згенеровані запити запускають функцію AWS Lambda, яка обробляє транзакції з прикладу набору даних і викликає дві кінцеві точки Amazon SageMaker. Кінцеві точки призначають оцінку аномалії та передбачають, чи є ці транзакції шахрайськими на основі навчених моделей ML. Потік доставки даних Amazon

Kinesis Data завантажує оброблені транзакції в інший сегмент Amazon S3 для зберігання. Після завантаження транзакцій в Amazon S3 можна використовувати інструменти та служби аналітики, включаючи Amazon QuickSight, для візуалізації, звітування, спеціальних запитів та більш детального аналізу [39].

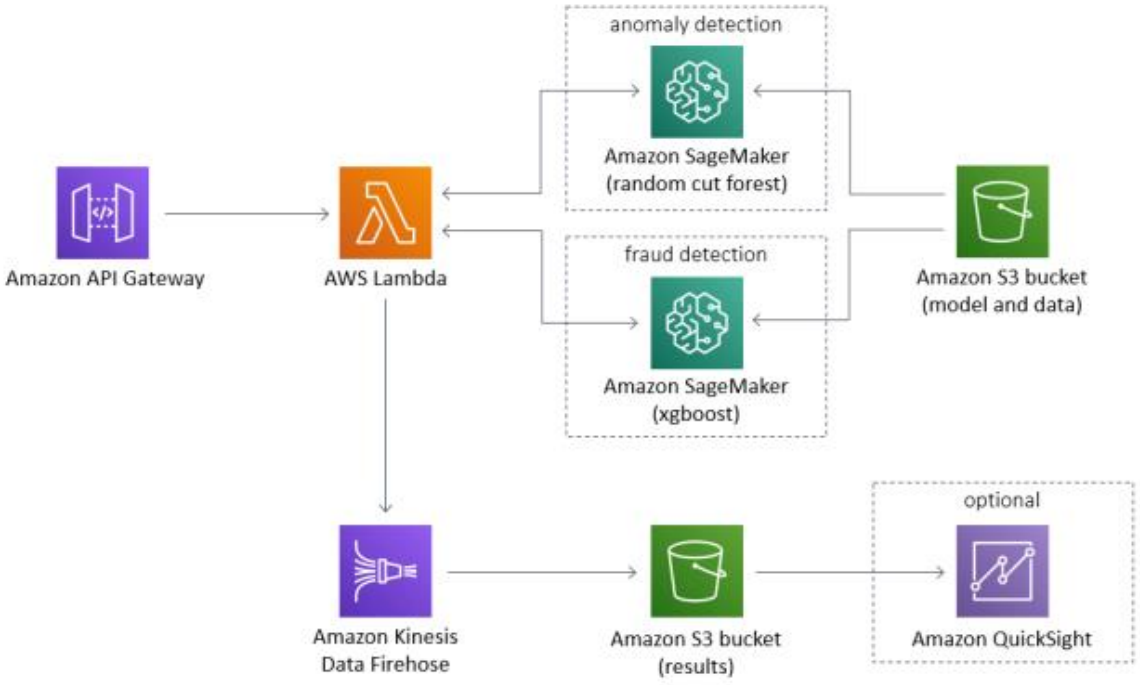


Рисунок 4.8 – Базова архітектура автоматизованої системи виявлення шахрайства в платіжних системах (PTFD) на основі [38]

Amazon Fraud Detector може виконувати прогнози шахрайства з низькою затримкою, що дозволяє системі виявлення шахрайства динамічно коригувати взаємодію з клієнтами у моделях виявлення ризику шахрайства в режимі реального часу. Але припустимо, якщо потрібно генерувати прогнози шахрайства для серії подій після факту самого шахрайства; можливо, не потрібна реакція з низькою затримкою і потрібно оцінювати події за погодинним або щоденним графіком. Як це зробити за допомогою Amazon Fraud Detector? Один із підходів полягає у використанні повідомлення про подію Amazon S3 для запуску функції

лямбда, яка обробляє CSV-файл подій, що зберігаються в Amazon S3, коли файл завантажується у вхідний сегмент S3. Функція запускає кожну подію через Amazon Fraud Detector, щоб генерувати прогнози за допомогою детектора (модель і правила ML) та завантажує результати прогнозування у вихідний сегмент S3. Наступна схема (Рис. 4.9) ілюструє цю архітектуру.

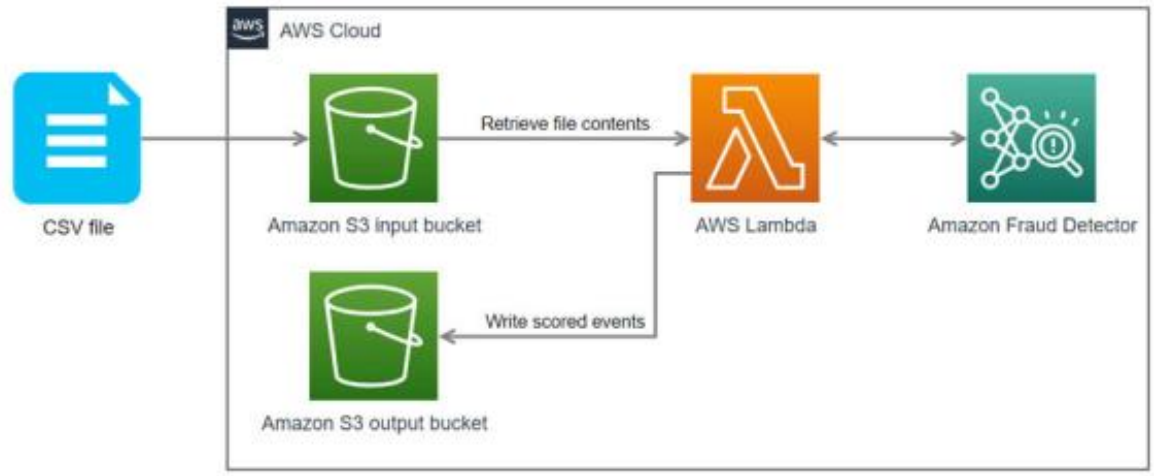


Рисунок 4.9 – Архітектура автоматизованої системи виявлення шахрайства із запуском класифікатора із довільним інтервалом [38]

Також можна підсилити систему за допомогою використання Amazon A2I – це сервіс машинного навчання, який спрощує створення робочих процесів з використанням моделей машинного навчання, необхідних для перевірки людиною. Amazon A2I надає можливість рецензування розробників, усуваючи недиференційовану важку роботу, пов'язану зі створенням систем рецензування, виконуваними людьми, або управлінням великою кількістю рецензентів. Рішення високого рівня резюмується в наступній архітектурі [40].

Робочий процес складається з наступних етапів (Рис. 4.10):

1. Клієнтська програма відправляє інформацію в кінцеву точку Amazon Fraud Detector.

2. Amazon Fraud Detector прогнозує оцінку ризику (в діапазоні від 0 до 1,000) для вхідних даних за допомогою моделі машинного навчання, навченої з використанням історичних даних. Оцінка 0 означає, що ризик шахрайства відсутній, а оцінка 1,000 вказує, що ризик шахрайства є максимальним.

3. Якщо оцінка ризику для конкретного прогнозу падає нижче заздалегідь визначеного порога, подальші дії не робляться.

4. Якщо оцінка ризику перевищує заздалегідь визначений поріг (наприклад, 900 балів), цикл Amazon A2I запускається автоматично і відправляє прогнози для перевірки людиною в Amazon A2I. Приватним персоналом можуть бути співробітники компанії. Вони відкривають інтерфейс Amazon A2I, розглядають справу і виносять рішення (схвалюють, відхиляють або відправляють його для подальшої перевірки).

5. Результат схвалення або відхилення приватної робочої сили зберігається в Amazon Simple Storage Service (Amazon S3). З Amazon S3 його можна безпосередньо відправити в клієнтську програму.

Для налаштування рішення необхідно застосувати наступні кроки:

1. Навчання і розгортання моделі в Amazon Fraud Detector з використанням історичних даних.

2. Налаштування циклу Amazon A2I для персоналу за допомогою Amazon Fraud Detector.

3. Використання моделі для прогнозування оцінки ризику для заданих нових вхідних даних.

4. Налаштування робочого процесу і циклів Amazon A2I.

Також, для повноти процесу, розглянемо рішення виявлення шахрайства (аномалій) за допомогою сервісів Amazon DynamoDB Streams та Amazon SageMaker, яке буде доповнювати попередню реалізацію та працювати послідовно з нею [35].

Це рішення має наступні переваги:

- Ефективніше використання наявні ресурси для виявлення аномалій. Наприклад, якщо використовувати потоки Amazon DynamoDB для аварійного відновлення (DR) або для інших цілей, можна використовувати дані в цьому потоці для виявлення аномалій. Крім того, резервне сховище, як правило, має низьку завантаженість. Дані з низьким рівнем обізнаності можна використовувати для навчальних даних.
- Автоматичне перекваліфікування моделі з новими даними на регулярній основі, без втручання користувача.
- Спрощення використання вбудованого алгоритму Amazon SageMaker Random Cut Forest. Amazon SageMaker пропонує гнучкі розподілені варіанти навчання, які пристосовуються до конкретних робочих процесів у безпечному та масштабованому середовищі.

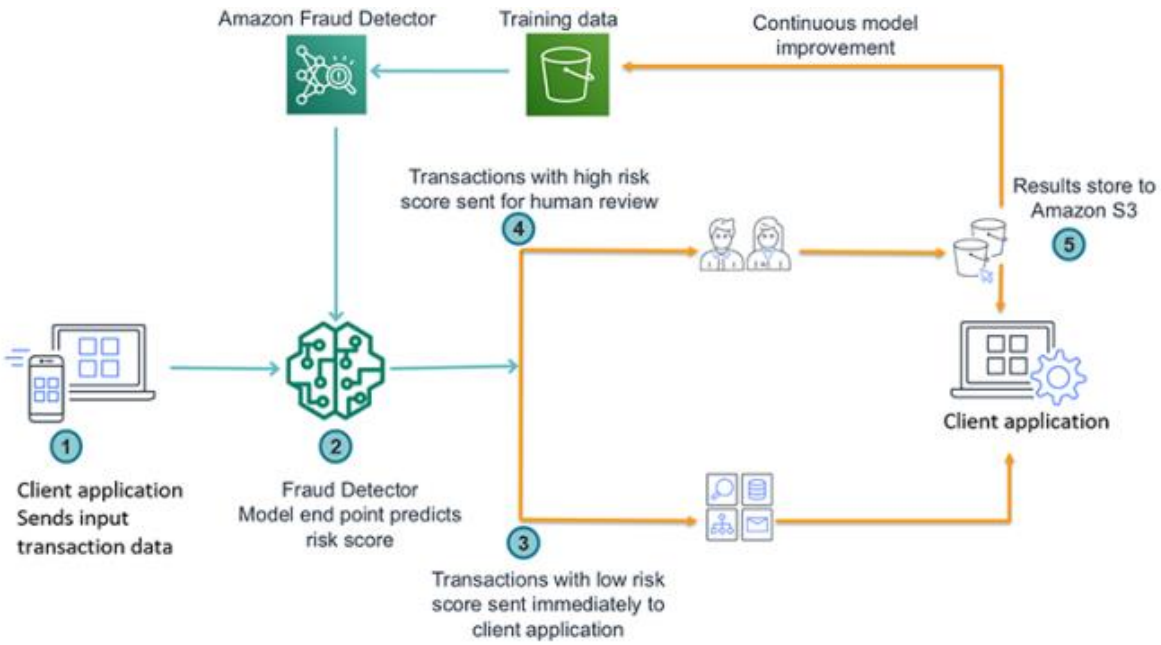


Рисунок 4.10 – Архітектура автоматизованої системи із виявлення шахрайства у режимі реального часу й інтеграції із клієнтськими застосунками [32]

Етапи, за якими дані проходять через архітектуру (Рис. 4.11), такі:

1. Джерело DynamoDB фіксує зміни та зберігає їх у потоці DynamoDB.
2. Завдання AWS Glue – регулярно отримувати дані з цільової таблиці DynamoDB та виконувати навчальну роботу за допомогою Amazon SageMaker для створення або оновлення артефактів моделі на Amazon S3.
3. Також AWS Glue використовує оновлену модель на кінцевій точці Amazon SageMaker для виявлення аномалій в режимі реального часу на основі класифікатора Random Forest.
4. Функція AWS Lambda аналізує дані з потоку DynamoDB і викликає кінцеву точку Amazon SageMaker, щоб отримати висновки.
5. Lambda попереджає користувацькі програми при виявленні аномалій.

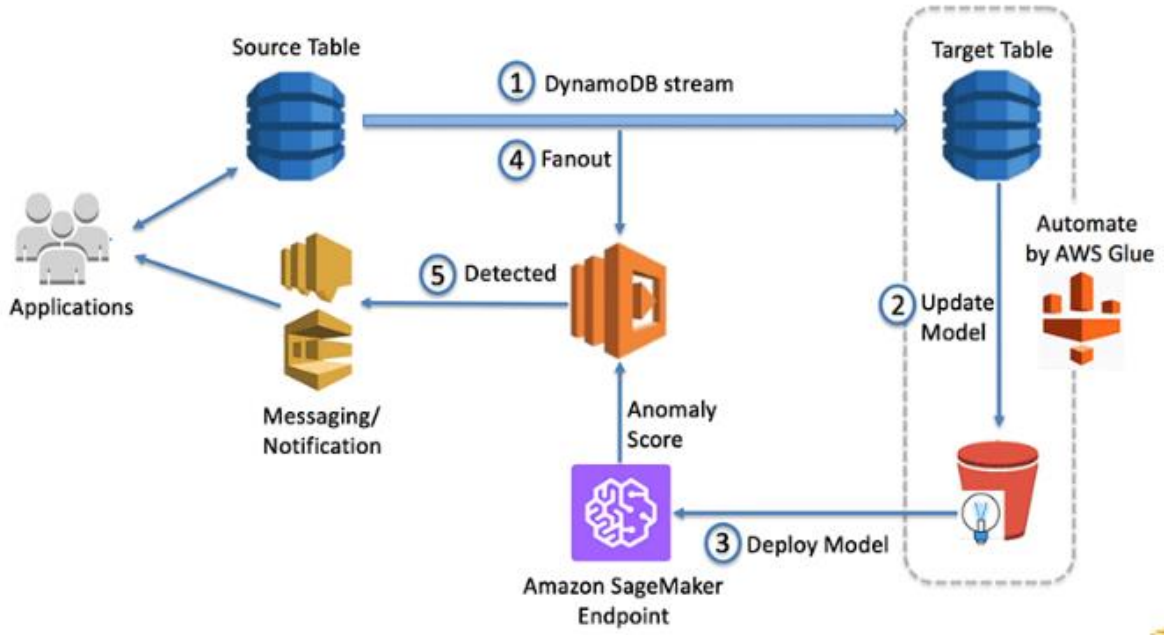


Рисунок 4.11 – Архітектура виявлення аномалій у потоках даних транзакцій за допомогою реалізованої автоматизованої системи [38]

Система виявлення шахрайства виглядатиме в кінцевому вигляді, після об’єднання усіх попередніх підсистем (Рис. 4.12).

Отже, на вхід до системи, яка розробляється буде передаватись 2 артефакти:

- реалізована алгоритмічна модель виявлення шахрайства (її код), яка буде статична протягом усього життєвого циклу окремо взятої імплементації.
- вхідний набір даних транзакцій, який буде динамічним та буде змінюватись та адаптуватись відповідно до поточної ситуації.

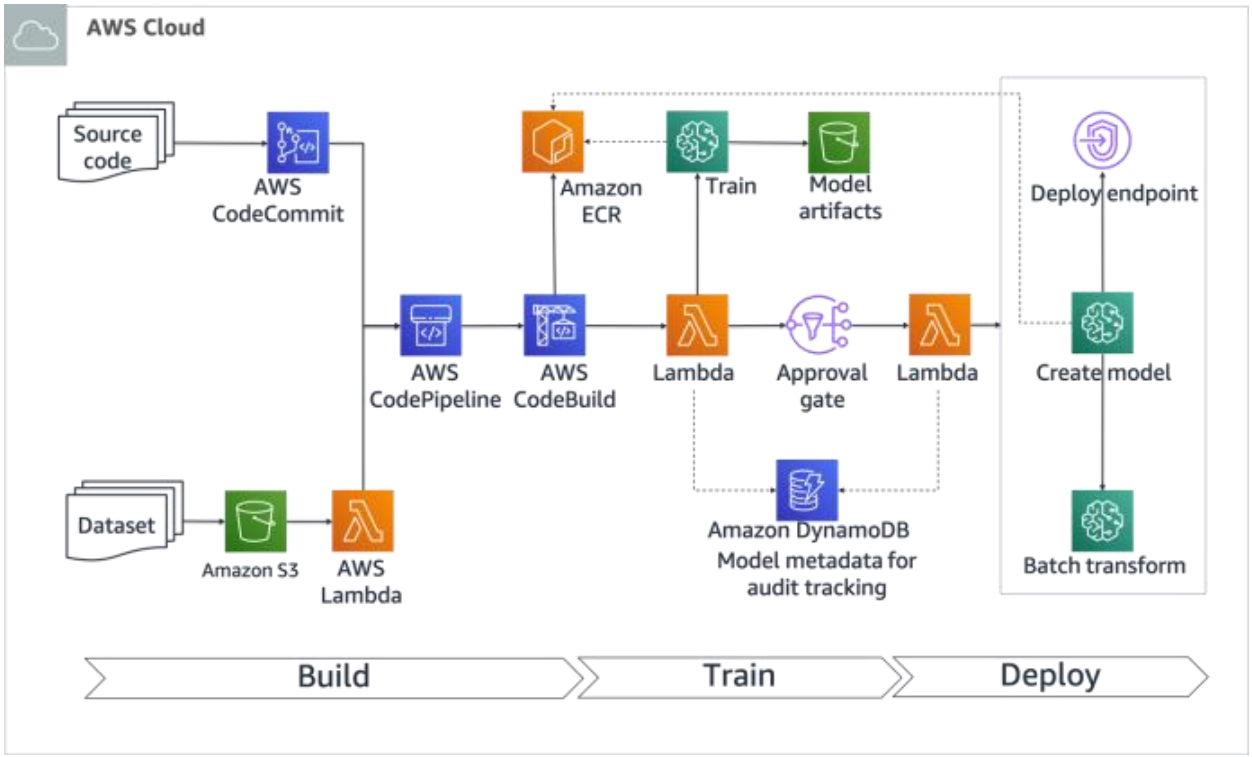


Рисунок 4.12 – Архітектура автоматизованої системи виявлення шахрайства повного циклу [38]

Загалом, ініціалізація та функціонування системи буде реалізовано у 3 кроки: Побудова, Тренування та Розміщення. Після того, як усі ці 3 кроки виконано успішно, буде створено API точку доступу, за допомогою якого можна буде комунікувати із реалізованою автоматизованою системою.

У результаті отримаємо програмний продукт, який вирішує завдання:

1. Запобігання та виявлення шахрайства в платіжних системах, адже реалізовано повністю кероване рішення на основі машинного навчання, що

використовується для виявлення шахраїв. Воно включає в себе всі інструменти, необхідні для створення, розгортання та управління моделями виявлення шахрайства. Всього за кілька клацань миші можна отримати дані аналітики, які підвищать ефективність моделі за рахунок використання бізнес-правил. Ці правила дозволяють контролювати роботу моделі, а також розгортати результати її роботи у вигляді призначених для прогнозування зручних API-інтерфейсів.

2. Виявлення шахрайства за лічені хвилини. Система автоматизує складні етапи створення моделей машинного навчання для виявлення шахрайства. Будь-які дії, від перевірки даних до розгортання моделі, можна виконувати без знань в сфері машинного навчання і навичок програмування. Завдяки цьому можна отримати засновані на технології машинного навчання моделі виявлення шахрайства, придатні до впровадження за лічені хвилини, а не місяці.

3. Відповідності будь-яких потреб бізнесу, адже використовує єдиний в своєму роді продукт машинного навчання, який містить шаблони шахрайських дій, отримані на підставі аналізу наданих даних. Це дозволяє створювати унікальні налаштування системи виявлення шахрайства, оптимізовані з урахуванням конкретних бізнес-сценаріїв. Такий підхід забезпечує дуже високу точність розпізнавання, знижуючи кількість помилкових спрацьовувань.

4.3.3 Впровадження автоматизованої технології у практику використання бізнесових установ

Впровадження автоматизованої системи доречно розглядати як проєкт. При реалізації проєкту впровадження автоматизованої системи виявлення шахрайства у платіжних сервісах рекомендовано оперувати такими принципами:

- функціонування автоматизованої системи виявлення шахрайства має бути документально задекларовано у бізнесовій організації;
- системна оцінка ризиків від шахрайства та визначення ефективності автоматизованої технології;

- автоматизована система має вдосконалювати алгоритми виявлення шахрайства;
- розслідування шахрайських операцій.

Витрати на проєкт впровадження автоматизованої системи виявлення шахрайства можна оцінити через витрати на організаційні, технічні та методологічні показники:

$$x_m^- = S_m^d + S_m^r + S_m^k, \quad (4.1)$$

де x_m^- – оцінка сумарних витрат на проєкт впровадження автоматизованої системи виявлення шахрайства;

S_m^d – організаційні витрати на проєкт впровадження автоматизованої системи виявлення шахрайства;

S_m^r – технічні витрати на створення автоматизованої системи виявлення шахрайства та проєкту її впровадження;

S_m^k – методологічні витрати на на проєкт впровадження автоматизованої системи виявлення шахрайства.

Організаційні витрати пов’язані із формування організаційної структури проєкту. Впровадження розробленої автоматизованої системи доречне при оцінці технологічної зрілості бізнесу. В якості такої оцінки запропоновано використати дослідження [36]. Така оцінка потрібна, оскільки для системності процесу виявлення шахрайство потрібні ряд організаційних заходів. Такі заходи пропонується реалізовувати у розрізі стратегії автоматизованої безпеки бізнесу (розробка та впровадження документів стратегічного бачення і пріоритетів при формуванні політики безпеки бізнесу, корпоративних стандартів безпеки, контроль виконання стратегії політики безпеки бізнесу, стратегії просування автоматизованого продукту, тощо).

Технічні витрати – витрати, які пов’язані із виробничими витратами на розробку програмного продукту, а також з моніторингом та контролем функціонування інформаційної системи виявлення шахрайства.

Методологічні витрати пов’язані із розробкою методології розробки автоматизованої системи та методології проєкту впровадження.

Підсумовуючи вищесказане, можна стверджувати, що в основі впровадження є розробка конкретизованої методології впровадження системи виявлення шахрайства у платіжних сервісах у фінансових установах [43].

Отже, було сформовано концепт автоматизованої інформаційної системи виявлення шахрайства в платіжних системах, що базується на розробленій технології виявлення шахрайства. Описано прототип гнучкої системи, що може бути розміщена та інтегрована у інфраструктуру фінансових підприємств за допомогою хмарних обчислень, а саме платформи Amazon Web Services.

ВИСНОВКИ

Сфера застосувань машинного навчання постійно розширюється. Всеохоплююча інформатизація призводить до накопичення величезних об'ємів даних в науці, виробництві, бізнесі, транспорті, охороні здоров'я. Виникаючі при цьому задачі прогнозування, управління та прийняття рішень часто зводяться до навчання за прецедентів. Раніше, коли таких даних не було, ці завдання або взагалі не ставилися, або вирішувалися зовсім іншими методами.

У даній роботі були досліджені й реалізовані методи машинного навчання для розв’язання практичних завдань, а саме моделювання й аналізу платіжних систем й транзакцій у них та виявлення шахрайства. Була розглянута й описана теоретична база, на яку спираються всі методи, наприклад були визначені такі поняття як лінійна регресія, логістична регресія, дерева рішень, метод опорних векторів, градієнтний бустінг, поняття нейронної мережі, автоасоціаторів та інші. Були розглянуті на практиці й описані теоретично інші, більш складні поняття й підходи машинного навчання до моделювання функціонування платіжних систем.

Було розглянуто 10 різних моделей для класифікації шахрайських транзакцій. В цілому, кожна модель продемонстрували високі показники й може застосовуватись як допоміжний механізм для протидії фроду в платіжних системах та інших фінансових організаціях. Найкращі результати показала нейронна мережа на субдискретизованих даних, майже 99.9% правильних класифікацій. Серед недоліків даної системи є найбільша кількість допущення фроду. Хоча модель неправильно класифікувала лише 0.1% транзакцій, але більшість з них відносяться до тих, що насправді є шахрайство, хоча модель класифікувала їх як чесні. Базові моделі на звужених даних: логістична регресія, метод k-найближчих сусідів, метод опорних векторів, дерева рішень та нейронна мережа на звужених даних продемонстрували точність на рівні 93-94%, пропускаючи при цьому меншу кількість шахрайських транзакцій. Варто відзначити нейронну мережу на звужених даних, яка пропустила надзвичайно

малу кількість шахрайських операцій (лише 1 на 50 000), таким чином майже перешкодивши шахрайству. Але з іншого боку багато звичайних транзакцій були класифіковані моделлю як шахрайські, що може погіршити обслуговування клієнтів.

Окремо згадаємо техніку градієнтного бустінгу та застосування автоасоціаторів. Ці потужні інструменти можуть бути золотою серединою між жорсткістю та точністю. Від так, вони демонстрували результати в межах 96-98% й при цьому не пропускали так багато шахрайських транзакцій, як нейронна мережа на субдискретизованих даних. Саме техніка градієнтного бустінгу у поєднанні із застосуванням автоасоціаторів може рекомендуватися до застосування через свою адаптивність до будь-яких даних (пропорції позитивних та негативних випадків) та баланс точності та надійності.

Варто також відзначити, що навчання й побудова моделей відбувалась на незбалансованому наборі даних. Отже, у даній роботі також було досліджено як аналізувати й досліджувати нестійкі випадки (записи) та виявляти аномалії в даних. Це є надзвичайно корисно, адже як правило дані аномалії (рідкі відхилення від норми) й несуть найбільшу цінність.

Таким чином, реалізація тієї чи іншої моделі залежить суто від побажань клієнтів й стратегії платіжної організації. Якщо увага в першу чергу приділяється безпеці, варто використовувати нейронну мережу на звужених даних. Якщо основний є точність класифікації, варто використовувати нейронну мережу на розширених даних. Інші 4 прості моделі: логістична регресія, дерева рішень, метод опорних векторів та метод k-найближчих сусідів можуть слугувати певним майданчиком для дослідження ефективності того чи іншого базового підходу. А ось техніка градієнтного бустінгу у поєднанні з автоасоціатором може слугувати абсолютно робочим варіантом у якості компромісу між точністю та безпекою. Також варто відзначити високу ресурсну вартість навчання нейронної мережі,

тобто її навчання потребує більшої кількісті обчислювальних ресурсів, ніж побудова інших моделей.

Реалізовано технологію виявлення шахрайства у платіжних системах. В основі такої технології отримано модель з точністю у 99.97% та AURPC показник 99.86% - що є не просто високим результатом, технологія може бути рекомендована до застосування у бізнесі та банківській справі, адже з-поміж 554082 тестових транзакцій лише 3 транзакції, що були класифіковані як справжні, виявилися шахрайськими, 166 насправді звичайних транзакцій були визначені як шахрайські. Перевагою моделі є саме те, що майже жодна шахрайська транзакція не була пропущена. Адже блокування звичайних транзакції хоча й погіршує досвід користування системою, але є більш допустимим, ніж втрата коштів. Побудова технології виявлення шахрайства в платіжних сервісах у 4 етапи:

1. дослідницький аналіз;
2. візуалізація даних із подальшою адаптацією набору даних;
3. створення технології за допомогою існуючих алгоритмів класифікації;
4. візуалізація отриманої моделі та результатів.

Варто зазначити, що етапи 1 та 2 (підготовка та дослідження даних) були не менш критичними, аніж створення самої моделі - вони дозволили краще розуміти дані, їх прикладне значення, та виокремлювати те, що дійсно важливо.

Подальший розвиток технології - це реалізація самої інформаційної системи, яка, базуючись на створеній моделі, могла б аналізувати банківські транзакції в режимі реального часу та впроваджувати відповідні дії - блокувати такі транзакції, що були класифіковані як шахрайські. Дана система мала б комерційну цінність, могла б бути інтегрована в програмне оснащення банків, платіжних систем, інших фінансових установ. Викликом тут буде налаштування взаємодії системи не з готовим датасет, а динамічно отриманими даними. Така система має витримувати високі навантаження, максимально швидко надавати

результат, бути захищеною та високо доступною, архітектура проекту повинна бути гнучкою, щоб пристосовуватись до змін.

На базі технології реалізовано автоматизовану систему виявлення шахрайства в платіжних системах: описано її архітектуру, принципи та моделі функціонування, задано інфраструктуру. У роботі продемонстровано виявлення онлайн-шахрайства за допомогою Amazon Fraud Detector і налаштування робочих процесів перевірки шахрайства людьми за допомогою настроюваного типу завдання Amazon A2I для перевірки і підтвердження прогнозів з високим ризиком. Також представлено приклад того, як створити систему виявлення аномалій на потоках Amazon DynamoDB за допомогою Amazon SageMaker, AWS Glue та AWS Lambda. Крім того, є можливість адаптувати цей приклад до конкретного випадку використання, оскільки AWS Glue дуже гнучкий на основі сценарію користувача і дозволяє додавати нові джерела даних. До цієї архітектури можуть застосовуватися інші типи джерел даних та потоки, оскільки функція AWS Lambda також працює з багатьма іншими потоковими службами AWS.

Запропоновано процес впровадження автоматизованої системи у бізнес розглядати як проєкт. Встановлено, що для раціонального впровадження проєкту потрібно розробити конкретизовану методології впровадження системи виявлення шахрайства у платіжних сервісах у фінансових установах, що є перспективою подальших досліджень.

Встановлено, що не існує ефективного алгоритму, який би був стандартом для всіх фінансових установ при виявленні, запобіганні шахрайства. Це пов’язано із тим, що підходи до шахрайства є динамічними та вимагають постійної переробки прогнозів. Визначено перспективи розвитку науково-практичних підходів попередження шахрайських операцій при здійсненні транзакцій. Встановлено, що машинне навчання є доречним у рішенні задач виявлення шахрайства у платіжних системах. Але виявлення шахрайства в платіжних системах полягає не тільки в побудові самого алгоритмічного ядра, але й у

побудові надійної автоматизованої системи, яка в режимі реального часу, за умови високого навантаження, здатна керувати потоками даних та ефективно оперувати алгоритмічним ядром системи. У роботі описано архітектуру, принципи та моделі функціонування, інфраструктуру автоматизованої системи виявлення шахрайства в платіжних системах. Визначено доцільність застосування хмарного веб-сервісу. Обґрунтовано розгортання моделі у вигляді автоматизованої технології базі платформи Amazon Web Services. Основою автоматизованої системи виявлення онлайн-шахрайства є Amazon Fraud Detector і налаштування робочих процесів перевірки шахрайства в платіжних системах за допомогою настроюваного типу завдання Amazon A2I для перевірки і підтвердження прогнозів з високим ризиком. Наведено приклад створення системи виявлення аномалій на потоках Amazon DynamoDB за допомогою Amazon SageMaker, AWS Glue та AWS Lambda. Автоматизована система враховує динамічність набору даних, оскільки функція AWS Lambda також працює з багатьма іншими потоковими службами AWS. Виокремлено основні три завдання, які вирішує програмний продукт: запобігання та виявлення шахрайства в платіжних системах, виявлення шахрайства за лічені хвилини, інтеграція програмного продукту у бізнес, де використовуються платіжні системи та сервіси (наприклад, сервіси інтеграції платежів у фінансових установах, інтернет-магазинах, логістичних компаніях, страхових полісах, торгових майданчиках, тощо). Визначено, що впровадження автоматизованої системи доречно розглядати як проєкт. Запропоновано принципи впровадження проєкту. Встановлено, що для раціонального впровадження проєкту потрібно розробити конкретизовану методології впровадження програмного продукту виявлення шахрайства у платіжних системах бізнесових установ.

Підсумовуючи, було вирішено наступні завдання:

- визначено математичні інструменти й методи, які використовуються для вирішення задач класифікації, та на їх основі сформовано математичні моделі виявлення шахрайства в платіжних сервісах;

- реалізовано програмну імплементацію моделей виявлення шахрайства в платіжних сервісах із використанням машинного навчання та мови програмування Python;

- побудовано агреговану технологію виявлення шахрайства в платіжних системах, що базується на реалізованих моделях;

- створено програмну імплементацію побудованої технології виявлення шахрайства в платіжних сервісах;

- реалізовано програмний продукт повного циклу на базі технології та із виконанням поставлених вимог, який готовий то впровадження на підприємствах (банках, фінансових установ, платіжних шлюзах, тощо).

Таким чином, у результаті роботи створено технологію виявлення шахрайства в платіжних системах та продемонстровано концепт автоматизованої інформаційної системи на її основі, позитивними відмінностями від інших систем якої є не тільки висока точність класифікатора (більше 99% правильних рішень), а й можливістю працювати в режимі реального часу та під високим навантаженням, що в сумі якісно виділяє використовувані підходи та саму побудовану інформаційну систему на фоні рішень, що розв’язують схожі задачі.

СПИСОК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Koval B., Khlevnyi A. Shabatskaya S. Development of a fraud detection system in payment services using CRISP-DM methodology. // Information Technology and Interactions (Satellite): Conference Proceedings, December 04, 2020, Kyiv, Ukraine / Taras Shevchenko National University of Kyiv and [etc]; Vitaliy Snytyuk (Editor). Kyiv: Stylos, 2020.– P. 143 – 145.

2. Koval B., Khlevna I., Fraud detection technology in payment systems. // Information Technology and Interactions (Satellite): Conference Proceedings, December 04, 2020, Kyiv, Ukraine / Taras Shevchenko National University of Kyiv and [etc]; Vitaliy Snytyuk (Editor). Kyiv: Stylos, 2020.– P.150 – 153.

3. Koval B., Khlevna I**.** Fraud detection technology in payment systems. // IT&I 2020 – Information Technology and Interactions. Proceedings of the 7th International Conference "Information Technology and Interactions" (IT&I-2020). Workshops Proceedings. Kyiv, Ukraine, December 02-03, 2020. CEUR Workshop Proceedings, – P. 85 – 95. (Scopus)

4. Iuliia L.Khlevna, Bohdan S. Koval. Development of the automated fraud detection system concept in payment systems. Applied Aspects of Information Technology. Vol. 4 № 1 (9). P. 37 – 46. DOI: 10.15276/aait.01.2021.3. https://aait.opu.ua/?fetch=articles&with=info&id=70

5. Koval B., Shpyrko V. Fraud detection models and payment transactions analysis using machine learning / SHS Web Conf. Volume 65, 2019. The 8th International Conference on Monitoring, Modeling & Management of Emergent Economy (M3E2 2019)

6. Koval B. Fraud detection models and payment transactions analysis using machine learning // XVII International scientific conference “Shevchenkivska Vesna 2019: Economics”: Conference Proceedings, March 27, 2019, Kyiv, Ukraine

7. Fraud Detection Techniques: Data and Technique Oriented Perspective / S. Sorournejad, Z. Zojaji, R.E. Atani, Amir Hassan Monadjemi / Cornel University Library, 2016. Mode of access: https://arxiv.org/ftp/arxiv/papers/1611/1611.06439.pdf .

8. Lebichot, B., Le Borgne, Y.-A.: Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection. In: Oneto, L., Navarin, N., Sperduti, A., Anguita, D. (eds.) Recent Advances in Big Data and Deep Learning, pp. 78–88. Springer, New York (2019)

9. Caelen, O., Smirnov, E.N.: Improving Card Fraud Detection Through Suspicious Pattern Discovery. In: Benferhat, S., Tabia, K., Ali, M. (eds.) Advances in Artificial Intelligence: From Theory to Practice, pp. 181–190. Springer, New York (2017)

10. Pozzolo, A.D., Caelen, O., Bontempi, G., Johnson, R.A.: Calibrating Probability with Undersampling for Unbalanced Classification. Paper presented at the 2015 IEEE Symposium Series on Computational Intelligence, Cape Town, South Africa, 7-10 December 2015

11. Lebichot B., Le Borgne YA., He-Guelton L., Oblé F., Bontempi G. (2020) Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection. In: Oneto L., Navarin N., Sperduti A., Anguita D. (eds) Recent Advances in Big Data and Deep Learning. INNSBDDL 2019. Proceedings of the International Neural Networks Society, vol 1. Springer, Cham. https://doi-org-443.webvpn.jnu.edu.cn/10.1007/978-3-030-16841-4_8

12. Sorournejad, S. A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective / S. Sorournejad, Z. Zojaji, R.E. Atani, Amir Hassan Monadjemi / Cornel University Library, 2016. Mode of access: https://arxiv.org/ftp/arxiv/papers/1611/1611.06439.pdf

13. Кузнєцова Н.В. Аналіз та прогнозування ризиків шахрайства з кредитними картками. Informatics and Mathematical Methods in Simulation Vol. 8 (2018), No. 1, pp. 16-25

14. Kuznietsova, N.V. Scoring Technology for Risk Assessment of Fraud in Banking / Selected Papers of the XVI International Scientific and Practical Conference "Information Technologies and Security" (ITS 2016). — 2016. — Pp. 54-61 .

15. Linda Delamaire, Hussein Abdou, John Pointon, “Credit card fraud and detection techniques: a review”, Banks and Bank Systems, Volume 4, Issue 2, 2009.

16. MasoumehZareapoor, Seeja.K.R, M.Afshar.Alam, ”Analysis of Credit Card Fraud Detection Techniques: based on Certain Design Criteria”, International Journal of Computer Applications (0975 – 8887) Volume 52– No.3, 2012.

17. Beninel, F. Transfer Learning Using Logistic Regression in Credit Scoring / F. Beninel, W. Bouaguel, G. Belmufti / Cornel University Library, 2012. Mode of access: https://arxiv.org/pdf/1212.6167.pdf

18. Дубина М.В., Садчикова І.В., Середюк І.О. Концептуальні підходи до підвищення рівня безпечності банківського платіжного середовища України. URL: https://www.business-inform.net/export_pdf/business-inform-2020-3_0-pages-349_359.pdf

19. Fraud Detection Techniques: Data and Technique Oriented Perspective / S. Sorournejad, Z. Zojaji, R.E. Atani, Amir Hassan Monadjemi / Cornel University Library, 2016. Mode of access: https://arxiv.org/ftp/arxiv/papers/1611/1611.06439.pdf .

20. Lebichot, B., Le Borgne, Y.-A.: Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection. In: Oneto, L., Navarin, N., Sperduti, A., Anguita, D. (eds.) Recent Advances in Big Data and Deep Learning, (2019)pp. 78–88. Springer, New York

21. Caelen, O., Smirnov, E.N.: Improving Card Fraud Detection Through Suspicious Pattern Discovery. In: Benferhat, S., Tabia, K., Ali, M. (eds.) Advances in Artificial Intelligence: From Theory to Practice, *Springer*, New York (2017) pp. 181–190.

22. Pozzolo, A.D., Caelen, O., Bontempi, G., Johnson, R.A.: Calibrating Probability with Undersampling for Unbalanced Classification. Paper presented at the

2015 IEEE Symposium Series on Computational Intelligence, Cape Town, South Africa, 7-10 December 2015

23. Lebichot B., Le Borgne YA., He-Guelton L., Oblé F., Bontempi G. (2020) Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection. In: Oneto L., Navarin N., Sperduti A., Anguita D. (eds) Recent Advances in Big Data and Deep Learning. INNSBDDL 2019. Proceedings of the International Neural Networks Society, vol 1. Springer, Cham. https://doi-org-443.webvpn.jnu.edu.cn/10.1007/978-3-030-16841-4_8

24. Sorournejad, S. A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective / S. Sorournejad, Z. Zojaji, R.E. Atani, Amir Hassan Monadjemi / Cornel University Library, 2016. Mode of access: https://arxiv.org/ftp/arxiv/papers/1611/1611.06439.pdf

25. Kuznietsova, N.V. Analysis and forecasting the risks of credit card fraud. Informatics and Mathematical Methods in Simulation Vol. 8 (2018), No. 1, pp. 16-25

26. Kuznietsova, N.V. Scoring Technology for Risk Assessment of Fraud in Banking / Selected Papers of the XVI International Scientific and Practical Conference "Information Technologies and Security" (ITS 2016). — 2016. — Pp. 54-61 .

27. Linda Delamaire, Hussein Abdou, John Pointon, “Credit card fraud and detection techniques: a review”, Banks and Bank Systems, Volume 4, Issue 2, 2009.

28. Teslia I. Development concept and method of formation of specific project management methodologies [Текст]/ Teslia I., Yehorchenkov O., Khlevna I., Khlevnyi A. //«Східно-Європейський журнал передових технологій». – №5/3(95) – 2018. – С.6 – 16.

29. Yufeng Kou, Chang-Tien Lu, Sirirat Sinvongwattana, Yo-Ping Huang. Survey of fraud detection techniques. Proceedings of the 2004 IEEE, International Conference on Networking, Sensing & Control. Taipei, Taiwan, March 21-23, 2004

30. M. U. Sapozhnikova, A. V. Nikonov, A. M. Vulfin, M. M. Gayanova, K. V. Mironov and D. V. Kurennov, "Anti-fraud system on the basis of data mining

technologies," *2017 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Bilbao, Spain, 2017, pp. 243-248, doi: 10.1109/ISSPIT.2017.8388649.

31. E. A. Lopez-Rojas and S. Axelsson "A review of computer simulation for fraud detection research in financial datasets" 2016 Future Technologies Conference (FTC) pp. 932-935 2016.

32. W. Fang, X. Li, P. Zhou, J. Yan, D. Jiang and T. Zhou, "Deep Learning Anti-Fraud Model for Internet Loan: Where We Are Going," in *IEEE Access*, vol. 9, pp. 9777-9784, 2021, doi: 10.1109/ACCESS.2021.3051079.

33. Wang Hongbin. (2015) Research and Application of web Log Mining Technology Based on Distributed Computing Platform [D]. Jinan: Shandong University.

34. Halbouni, S.S., Obeid, N. and Garbou, A. "Corporate governance and information technology in fraud prevention and detection: Evidence from the UAE", Managerial Auditing Journal, (2016), Vol. 31 No. 6/7, pp. 589-628. https://doi.org/10.1108/MAJ-02-2015-1163

35. Cloud computing, SoftServe, 2021. Mode of access: https://www.softserveinc.com/uk-ua/services/cloud-devops

36. Overview of Amazon Web Services, 2021. Mode of access: https://d1.awsstatic.com/whitepapers/aws-overview.pdf

37. Cloud Computing Solutions Architect: A Hands-On Approach: A Competency-based Textbook for Universities and a Guide for AWS Cloud Certification and Beyond by Arshdeep Bahga, Vijay Madisetti

38. AWS Documentation, 2021. Mode of access:https://docs.aws.amazon.com/index.html?nc2=h_mo)

39. AWS Certified Cloud Practitioner Study Guide: CLF-C01 Exam 1st Edition by Ben Piper, David Clinton

40. Architecting Cloud Computing Solutions: Build cloud strategies that align technology and economics while effectively managing risk by Kevin L. Jackson, Scott Goessling, May 30, 2018

41. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems by Martin Kleppmann, April 18, 2017

42. Бушуев С.Д., Бушуева Н.С. Развитие технологической зрелости в управлении проектами.// Управління проектами та розвиток виробництва. Збірник наукових праць. Під ред. В.А.Рач. – 2003. – № 2(7). – С.5-12.

43. Teslia I.M., Khlevna I.L., Yehorchenkov O.V., Yehorchenkova N.I. Organizational bases of implementation of Specified project management. Methodologies. Sciences of Europe. Technical sciences. Vol 1,No 34(2018).P.12–18.

44. The Elements of Statistical Learning: Data Mining, Inference, and Prediction by T. Hastie, R. Tibshirani, J. Friedman - Springer 2009, 764 p.

45. Introduction To Machine Learning by Nils J Nilsson – 1997, 209 p.

46. Inductive Logic Programming: Theory and Methods by Stephen Muggleton, Luc de Raedt - ScienceDirect 1994, 51 p.

47. Information Theory, Inference, and Learning Algorithms by David J. C. MacKay - Cambridge University Press 2003, 640 p.

48. Gaussian Processes for Machine Learning by Carl E. Rasmussen, Christopher K. I. Williams - The MIT Press 2005, 266 p.

49. База даних транзакцій по платіжній системі. Режим доступу: http://mlg.ulb.ac.be/

50. Bayesian Reasoning and Machine Learning by David Barber - Cambridge University Press 2011, 644 p.

51. Лінійна регресія [Електронний ресурс] – 2018. – Режим доступу: uk.wikipedia.org/wiki/Лінійна_регресія. – (дата звернення 05.02.2019). – Назва з екрана.

52. Задача класифікації [Електронний ресурс] – 2018. – Режим доступу: uk.wikipedia.org/wiki/Задача_класифікації. – (дата звернення 05.02.2019). – Назва з екрана.

53. Автокодувальник [Електронний ресурс] – 2018. – Режим доступу: uk.wikipedia.org/wiki/Автокодувальник. – (дата звернення 05.02.2019). – Назва з екрана.

54. Платіжна система [Електронний ресурс] – 2018. – Режим доступу: uk.wikipedia.org/wiki/Платіжна_система. – (дата звернення 05.02.2019). – Назва з екрана.

55. Логістична регресія [Електронний ресурс] – 2018. – Режим доступу: uk.wikipedia.org/wiki/ Логістична_регресія – (дата звернення 05.02.2019). – Назва з екрана.

56. Метод опорних векторів [Електронний ресурс] – 2018. – Режим доступу: uk.wikipedia.org/wiki/Метод_опорних_векторів. – (дата звернення 05.02.2019). – Назва з екрана.

57. Дерево ухвалення рішень [Електронний ресурс] – 2018. – Режим доступу: uk.wikipedia.org/wiki/Дерево_ухвалення_рішень. – (дата звернення 05.02.2019). – Назва з екрана.

58. Попко. А.В. Магістерська дисертація на тему: «Методи виявлення образ в коротких текстах» [Електронний ресурс] – 2018. – Режим доступу: http://ela.kpi.ua/bitstream/123456789/23001/3/Popko_magistr.pdf. – (дата звернення 05.02.2019). – Назва з екрана.

59. A Course in Machine Learning by Hal Daumé III - ciml.info 2012, 189 p.

60. Machine Learning, Neural and Statistical Classification by D. Michie, D. J. Spiegelhalter - Ellis Horwood 1994, 298 p.

61. Професіональний інформаційно-аналітичний ресурс, присвячений машинному навчанню, розпізнаванню образів й інтелектуальному аналізу даних MachineLearning.ru. Режим доступу: http://www.machinelearning.ru

62. Python Data Analysis, 2nd Edition by Armando Fandango - Packt Publishing 2017, 330 p.

63. Платформа онлайн-курсів Coursera. Режим доступу: https://www.coursera.org/

64. Платформа онлайн-курсів Prometheus. Режим доступу: https://prometheus.org.ua/

65. Microsoft Developer Network. Режим доступу: https://msdn.microsoft.com/

66. Платформа онлайн-курсів Prometheus. Режим доступу: https://www.edx.org/

67. Machine Learning & Data Science Landscape by Christina Voskoglou, Mark Wilcox, Stijn Schuermans – VisionMobile 2017, 46 p.

68. Big Data Visualization by James D. Miller - Packt Publishing 2017, 304 p.

69. Python for Data Analysis, 2E By Wes McKinney – O'Reilly Media 2016, 550 p.

70. Practical Data Analysis, 2nd Edition by Hector Cuesta, Sampath Kumar - Packt Publishing 2016, 338 p.

71. Python Data Analysis Cookbook by Ivan Idris – Packt Publishing 2016, 462 p.

72. Mastering Python Data Analysis by Magnus Vilhelm Persson, Luiz Felipe Martins - Packt Publishing 2016, 284 p.

73. Learning Predictive Analytics with Python by Ashish Kumar - Packt Publishing 2016, 354 p.

74. Practical Machine Learning by Sunila Gollapudi – Packt Publishing 2016, 468 p.

75. Анісімов В.В. Математична статистика: [навч. посібник] / В.В. Анісімов, О.І.Черняк – К.: МП «Леся». – 1995.

76. Черняк О.І. Теорія ймовірностей та математична статистика. Збірник задач: [навч. посібник] / О.І. Черняк, О.М. Обушна, А.В. Ставицький.- 2-ге вид. – К.: Знання, КОО, 2002.

77. Черняк О.І. Динамічна економетрика: [навч. посібник] / О.І.Черняк, А.В.Ставицький . – К.: КВІЦ, 2000.

78. Карнаух Т.О., Ставровський А.Б. Теорія графів у задачах – Київ, 90 с.

79. Офіційний веб-сайт Державного комітету статистики України. [Електронний ресурс]. – Режим доступу: http://www.ukrstat.gov.ua/

80. С. Л. Брю, К. Р. Макконел. Экономикс: Принципы, проблемы и политика / С. Л. Брю, К. Р. Макконел. – 1992. – 800 с.

81. Naive Bayes classifier [Електронний ресурс] – 2018. – Режим доступу: https://en.wikipedia.org/wiki/Naive_Bayes_classifier. – (дата звернення 02.03.2019). – Назва з екрана.

82. Рекомендательные системы: теорема Байеса и наивный байесовский классификатор [Електронний ресурс] / Николенко С. // Блог компании Surfingbird. – 2012. – Режим доступу: https://habr.com/company/surfingbird/blog/150207/ – (дата звернення 18.08.2018). – Назва з екрана.

83. Рекомендательные системы: SVD, часть I [Електронний ресурс] / Николенко С. // Блог компании Surfingbird. – 2012. – Режим доступу: https://habr.com/company/surfingbird/blog/139863/ – (дата звернення 17.08.2018). – Назва з екрана.

84. Лекции по логическим алгоритмам классификации [Електронний ресурс] / Воронцов К. В. – 2010. – С. 17-21. – Режим доступу: http://www.machinelearning.ru/wiki/images/3/3e/Voron-ML-Logic.pdf – (дата звернення 17.09.2018). – Назва з екрана.

85. Персептроны [Електронний ресурс] / Минский М., Пейперт С. – 1971. – Режим доступу: https://sheba.spb.ru/delo/perseptrony-1969.htm – (дата звернення 17.09.2018). – Назва з екрана.

86. Метод k-найближчих сусідів [Електронний ресурс] – 2018. – Режим доступу: uk.wikipedia.org/wiki/Метод_k-найближчих_сусідів. – (дата звернення 05.02.2019). – Назва з екрана.

87. Штучна нейронна мережа [Електронний ресурс] – 2018. – Режим доступу: uk.wikipedia.org/wiki/Штучна_нейронна_мережа. – (дата звернення 05.02.2019). – Назва з екрана.

88. Чайковський Я.І. Платіжні системи : навчальний посібник. Тернопіль : Карт-бланш, 2016. 210 с.

ДОДАТКИ

Додаток А. Фрагменти програмного коду процесу виявлення шахрайства в платіжних сервісах

1.

```
fraudTransactions = df.loc[df.isFraud == 1].type.drop_duplicates().values
print(list(fraudTransactions))
```

['TRANSFER', 'CASH_OUT']

2.

```
dfTransactions = df.loc[(df.type == 'TRANSFER') | (df.type == 'CASH_OUT')]
dfFraud = dfTransactions['isFraud']
del dfTransactions['isFraud']
```

3.

```
dfTransactions.loc[dfTransactions.type == 'TRANSFER', 'type'] = 0
dfTransactions.loc[dfTransactions.type == 'CASH_OUT', 'type'] = 1
dfTransactions.type = dfTransactions.type.astype(int)
```

4.

```
dfTransactionsFraud = dfTransactions.loc[dfFraud == 1]
dfTransactionsNonFraud = dfTransactions.loc[dfFraud == 0]
fractionAnomalyTransactionsInFraud = len(dfTransactionsFraud.loc[
  (dfTransactionsFraud.oldBalanceDest == 0)
  & (dfTransactionsFraud.newBalanceDest == 0)
  & (dfTransactionsFraud.amount)
]) / (1.0 * len(dfTransactionsFraud))
print(
  "Частка аномальних транзакцій серед шахрайських: ",
  fractionAnomalyTransactionsInFraud
)
fractionAnomalyTransactionsInNonFraud = len(dfTransactionsNonFraud.loc[
  (dfTransactionsNonFraud.oldBalanceDest == 0)
  & (dfTransactionsNonFraud.newBalanceDest == 0)
  & (dfTransactionsNonFraud.amount)
]) / (1.0 * len(dfTransactionsNonFraud))
print(
  "Частка аномальних транзакцій серед звичайних (справжніх): ",
  fractionAnomalyTransactionsInNonFraud
)
```

Частка аномальних транзакцій серед шахрайських: 0.4955558261293072
Частка аномальних транзакцій серед звичайних (справжніх): 0.0006176245277308345

5.

```
dfTransactions.loc[
  (dfTransactions.oldBalanceDest == 0)
  & (dfTransactions.newBalanceDest == 0)
  & (dfTransactions.amount != 0), \
   ['oldBalanceDest', 'newBalanceDest']] = - 1
```

6.

```
dfTransactions.loc[
  (dfTransactions.oldBalanceOrig == 0)
  & (dfTransactions.newBalanceOrig == 0)
  & (dfTransactions.amount != 0), \
   ['oldBalanceOrig', 'newBalanceOrig']] = np.nan
```

7.

```
dfTransactions['errorBalanceDest'] = \
  dfTransactions.oldBalanceDest + dfTransactions.amount \
  - dfTransactions.newBalanceDest
dfTransactions['errorBalanceOrig'] = \
  dfTransactions.newBalanceOrig + dfTransactions.amount \
  - dfTransactions.oldBalanceOrig
```

8.

```
dfTransactions = dfTransactions.drop(
  ['nameOrig', 'nameDest', 'isFlaggedFraud'],
  axis = 1
)
```

9.

```
list(dfTransactions)
```

['step', 'type', 'amount', 'oldBalanceOrig', 'newBalanceOrig', 'oldBalanceDest',
 'newBalanceDest', 'errorBalanceDest', 'errorBalanceOrig']

10.

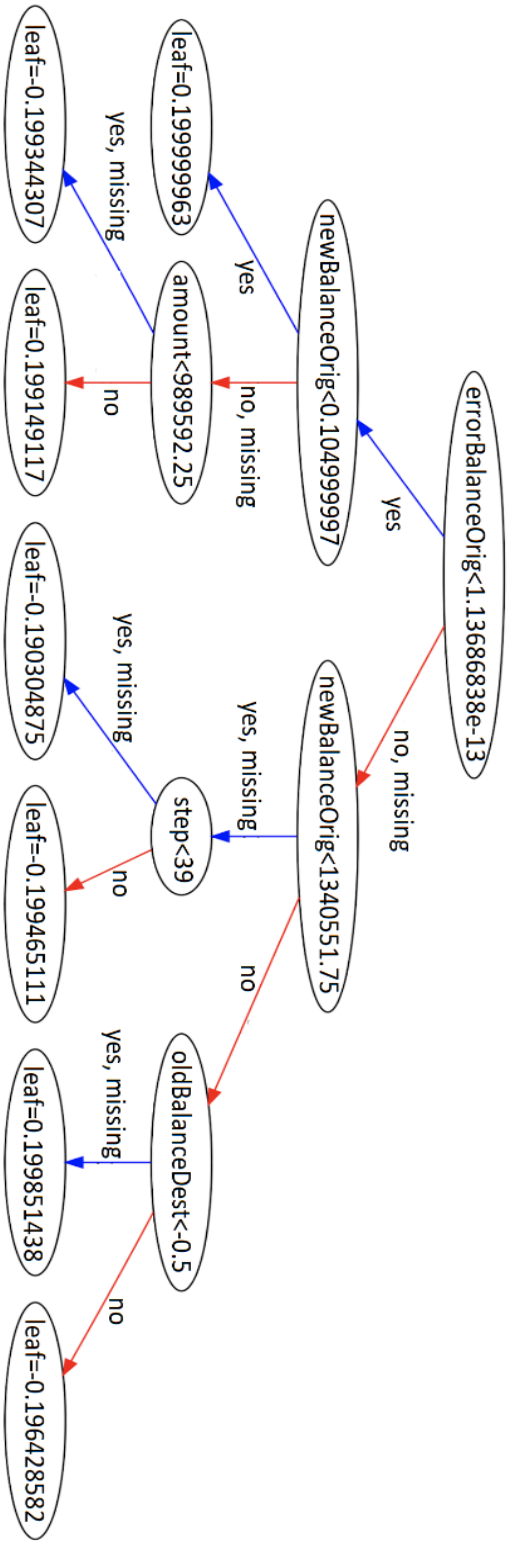
```
randomState = 5
np.random.seed(randomState)

trainX, testX, trainY, testY = train_test_split(
  dfTransactions, dfFraud, test_size = 0.2, random_state = randomState
)

weights = (dfFraud == 0).sum() / (1.0 * (dfFraud == 1).sum())
classifier = XGBClassifier(max_depth = 3, scale_pos_weight = weights, \
        n_jobs = 4)
predictions = classifier.fit(trainX, trainY).predict_proba(testX)
AURPC = average_precision_score(testY, predictions[:, 1])
AURPC
```

0.9986361116985445

Додаток Б. Дерево прийняття рішень реалізованої моделі виявлення шахрайства



Додаток В. Python код ініціалізації набору даних

1. Завантаження необхідних бібліотек:

```
import numpy as np # лінійна алгебра
import pandas as pd # обробка данних (I/O, csv)
import tensorflow as tf
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.manifold import TSNE
from sklearn.decomposition import PCA, TruncatedSVD
import matplotlib.patches as mpatches
import time

# бібліотеки класифікаторів
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
import collections


from imblearn.datasets import fetch_datasets
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline
from imblearn.pipeline import make_pipeline as imbalanced_make_pipeline
from imblearn.over_sampling import SMOTE
from imblearn.under_sampling import NearMiss
from imblearn.metrics import classification_report_imbalanced
from collections import Counter
from sklearn.model_selection import KFold, StratifiedKFold
```

2. Виведення зразку записів транзакцій, що розміщені в DataFrame:

```
df.head()
```

3. Перевірка DataFrame на відсутність даних:

```
df.isnull().sum().max()
```

```
0
```

Додаток Г. Python код первинного аналізу даних

1. Перевірка даних у DataFrame на розмічені класи:

```
round(df['Class'].value_counts()[0]/len(df) * 100,2)
round(df['Class'].value_counts()[1]/len(df) * 100,2)
```

```
Звичайні транзакції 99.83 % від усіх
Шахрайські транзакції 0.17 % від усіх
```

2. Виведення графіку розподілу класів:

```
sns.countplot('Class', data=df, palette=colors)
```

```
Text(0.5,1,'Розподіл класів \n (0: Звичайні || 1: Шахрайські)')
```

3. Побудова графіків розподілу транзакцій за часом та сумою:

```
fig, ax = plt.subplots(1, 2, figsize=(18,4))

amount_val = df['Amount'].values
time_val = df['Time'].values

sns.distplot(amount_val, ax=ax[0], color='r')
ax[0].set_title('Розподіл транзакцій за сумою', fontsize=14)
ax[0].set_xlim([min(amount_val), max(amount_val)])

sns.distplot(time_val, ax=ax[1], color='b')
ax[1].set_title('Розподіл транзакцій за часом', fontsize=14)
ax[1].set_xlim([min(time_val), max(time_val)])



plt.show()
```

Додаток Д. Python код підготовки даних до подальшого використання

1. Розділення початкового набору даних на тренувальний та тестовий. Застосування техніки балансування даних до тренувального набору:

```
from sklearn.model_selection import StratifiedShuffleSplit

X = df.drop('Class', axis=1)
y = df['Class']

sss = StratifiedShuffleSplit(n_splits=5,
                             test_size=0.2,
                             random_state=42)

for train_index, test_index in sss.split(X, y):
    original_Xtrain = X.iloc[train_index]
    original_Xtest = X.iloc[test_index]
    original_ytrain = y.iloc[train_index]
    original_ytest = y.iloc[test_index]

original_Xtrain = original_Xtrain.values
original_Xtest = original_Xtest.values
original_ytrain = original_ytrain.values
original_ytest = original_ytest.values
```

2. Застосування випадкової супердискретизації:

```
df = df.sample(frac=1)

fraud_df = df.loc[df['Class'] == 1]
non_fraud_df = df.loc[df['Class'] == 0][:492]

normal_distributed_df = pd.concat([fraud_df, non_fraud_df])

new_df = normal_distributed_df.sample(frac=1, random_state=42)

new_df.head()
```

Додаток Ж. Python код імплементації й оцінювання класифікаторів

1. Визначення базових регресійних та дискретних моделей:

```
classifiers = {
    "LogisiticRegression": LogisticRegression(),
    "KNearest": KNeighborsClassifier(),
    "Support Vector Classifier": SVC(),
    "DecisionTreeClassifier": DecisionTreeClassifier()
}
```

2. Визначення точності базових класифікаторів:

```
from sklearn.model_selection import cross_val_score

for key, classifier in classifiers.items():
    classifier.fit(X_train, y_train)
    training_score = cross_val_score(classifier,
                                     X_train,
                                     y_train,
                                     cv=5)
```

```
Класифікатор:  LogisticRegression отримав  94.0 % точності
Класифікатор:  KNeighborsClassifier отримав  92.0 % точності
Класифікатор:  SVC отримав  92.0 % точності
Класифікатор:  DecisionTreeClassifier отримав  90.0 % точності
```

3. Отримання оцінок кросс-валідації базових класифікаторів:

```
log_reg_score = cross_val_score(log_reg, X_train,
                                y_train, cv=5)

knears_score = cross_val_score(knears_neighbors, X_train,
                               y_train, cv=5)

svc_score = cross_val_score(svc, X_train,
                            y_train, cv=5)

tree_score = cross_val_score(tree_clf, X_train,
                             y_train, cv=5)
```

```
Логістична регресія оцінка кросс-валідації:  93.52%
Метод k-найближчих сусідів оцінка кросс-валідації:  93.14%
Метод опорних векторів оцінка кросс-валідації:  93.78%
Класифікатор дерев рішень оцінка кросс-валідації:  91.84%
```