Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеню бакалавра

студента ***Бедловського Дмитра Станіславовича***

академічної групи ***125-20-1***

спеціальності ***125 Кібербезпека***

спеціалізації[1]

за освітньо-професійною програмою ***Кібербезпека***

на тему ***Аналіз сучасних методів виявлення шахрайства у банківській сфері***

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
кваліфікаційної роботи	к.т.н., доц. Сафаров О.О.	94	відмінно	
розділів:				
спеціальний	к.т.н., доц. Сафаров О.О.	94	відмінно	
економічний	к.е.н., доц. Пілова Д.П.	90	відмінно	
Рецензент				
Нормоконтролер	ст. викл. Мєшков В.І.			

Дніпро
2024

ЗАТВЕРДЖЕНО:
завідувач кафедри
безпеки інформації та телекомунікацій
_____ д.т.н., проф. Корнієнко В.І.

«_____»_____20___року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавра

студенту ***Бедловському Дмитру Станіславовичу*** **академічної групи** ***125-20-1***
(прізвище ім'я по-батькові) (шифр)

спеціальності *125 Кібербезпека*
(код і назва спеціальності)

на тему ***Аналіз сучасних методів виявлення шахрайства у банківській сфері***

затверджену наказом ректора НТУ «Дніпровська політехніка» від 23.05.2024 №469-с

Розділ	**Зміст**	**Термін виконання**
Розділ 1	Дослідити сучасний стан проблеми шахрайства в банківській сфері, оцінити основні фактори, що призводять до зростання шахрайства, та провести аналіз ефективності існуючих методів протидії цьому	15.03.2024
Розділ 2	Провести аналіз методів виявлення шахрайства та аномалій у банківських транзакціях. Розробити систему виявлення аномалій	10.05.2024
Розділ 3	Розрахувати економічну доцільність та ефективність розробки системи виявлення аномалій	11.06.2024

Завдання видано _____ Олександр САФАРОВ
(підпис керівника) (ім'я, прізвище)

Дата видачі: 01.04.2024р.

Дата подання до екзаменаційної комісії: 28.06.2024р.

Прийнято до виконання _____ Дмитро БЕДЛОВСЬКИЙ
(підпис студента) (ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 89 с., 24 рис., 2 табл., 5 додатків, 19 джерел.

Об'єкт розробки: методи виявлення шахрайства при обробці платежів підприємств банківської сфери

Предмет розробки: заходи захисту інформаційної системи банку та її клієнтів-користувачів під час користування його засобами

Мета роботи: проаналізувати й вдосконалити систему виявлення шахрайства, тим самим підвищити безпеку проведення банківських операцій та користування банком його клієнтами

У першому розділі було досліджено сучасний стан проблеми, проведено аналіз наявних видів онлайн та фізичного типів шахрайства у банківській сфері, також було проаналізовано сучасні методи виявлення шахрайства та її протидії в світі та Україні.

У другому розділі було проведено аналіз методів виявлення аномалій й шахрайства у банківських операціях та їх підвиди та розроблено один з видів виявлення аномалій у транзакціях.

У третьому розділі було розраховано витрати на реалізацію методик, щодо виявлення та протидії шахрайства та досліджено ефективність такої реалізації.

Практична цінність розробки полягає у впровадженні розроблених методик виявлення шахрайства з платіжними картками та операціями за допомогою них для підвищення рівня безпеки банку та їх клієнтів банків у питанні такого виду злочинних дій та впровадженні методів захисту від них.

ШАХРАЙСТВО, БЕЗПЕКА БАНКІВСЬКИХ ОПЕРАЦІЙ, ФРОД-МОНІТОРИНГ, ВИЯВЛЕННЯ АНОМАЛІЙ, МАШИННЕ НАВЧАННЯ

ABSTRACT

Explanatory note: 89 pp., 24 pic., 2 tables, 5 apps, 19 sources.

Object of development: methods for detecting fraud in the processing of payments by banking enterprises

Subject of development: measures to protect the bank's information system and its customer-users when using its facilities

Purpose: to analyse and improve the fraud detection system, thereby increasing the security of banking operations and the use of the bank by its customers

The first section examined the current state of the problem, analysed the existing types of online and physical fraud in the banking sector, and analysed modern methods of fraud detection and counteraction in the world and Ukraine.

The second section analyses the methods of detecting anomalies and fraud in banking transactions and their subtypes, and develops one of the types of detection of anomalies in transactions.

The third section calculates the costs of implementing methods for detecting and combating fraud and studies the effectiveness of such implementation.

The practical value of the research lies in the implementation of the developed methods for detecting fraud with payment cards and transactions with them to increase the level of security of banks and their bank customers in terms of this type of criminal activity and the introduction of methods of protection against them.

FRAUD, SECURITY OF BANKING OPERATIONS, FRAUD MONITORING, ANOMALY DETECTION, MACHINE LEARNING

СПИСОК УМОВНИХ СКОРОЧЕНЬ

ADASYN – Adaptive Synthetic Sampling;

API – Application Programming Interface;

P2P – Person-To-Person;

SMOTE – Minority Oversampling Technique;

SMS – Short Message Service;

ККМ – коефіцієнт кореляції Меттьюза;

КСІБ – комп'ютерна система інформаційної безпеки;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер.

ЗМІСТ

с.

ВСТУП

Розвиток банківських систем та їх інтеграції до інтернет простору призвела до спрощення доступу до користування фінансовими послугами в повсякденному житті. Це дозволило банкам більше показати свою прозорість до користувачів – і це є однією з умов конкурентоспроможності банку.

В сучасному світі стає все важче уявити банківські установи в яких би не було доступу до використання коштів шляхом онлайн-банкінгу з пристроїв користувачів банку. Віддалений доступ до послуг банку дав можливість використання коштів майже в кожному місці світу де є інтернет і навіть не завжди обов'язково мати пластикову картку для користування – і ці операції проходять швидко та зручно, без важливості використання відділень банку що використовує клієнт – які зазвичай є не у кожній країні.

Але разом з розвитком систем електронної комерції також виникла більша проблема в тому що через більшу відкритість функцій – збільшилась й кількість випадків шахрайства та їх зловживань у банківській сфері. Через збільшення шахрайства та збільшення їх різноманіття й виникла потреба у активній розробці та покращенні систем моніторингу операцій зі сторони банків – використання ними передових технологій та алгоритмів штучного інтелекту для захисту коштів своїх клієнтів та ефективного виявлення шахрайства.

Та з цим виникла велика потреба у виявленні найкращої з методів захисту від шахрайства у банківській сфері для збереження репутації для користувачів та надійності збереження коштів в цьому банку. Тому потрібен був аналіз цих методів - через те що виявити усі чіткі дії та операції шахраїв неможливо для усіх видів шахрайства - при цьому намагатись захопити якомога більшу кількість шахрайства, але щоб від цього постраждало якомога менша кількість клієнтів-користувачів банку – потрібно було виявити один з найкращих способів.

Об'єктом дослідження кваліфікаційної роботи є аналіз сучасних методів безпеки по виявленню шахрайства у підприємствах банківської сфері.

Предметом дослідження роботи є сучасні методи безпеки банків від шахрайства.

Мета роботи – дослідити сучасні методи виявлення (моніторингу) різних видів шахрайства – рівні їх показників, аналіз самих шахрайства та розробка покращень у боротьбі з ними для забезпечення безпеки операцій.

Завдання роботи включають:

1. розглянути шахрайства що пов'язані з банківською сферою;

2. провести аналіз сучасних існуючих методів виявлення шахрайства у банку;

3. розробити систему для виявлення шахрайства на основі виявлення аномалій у банківських операціях;

4. оцінити ефективність методу виявлення.

Практичне значення кваліфікаційної роботи полягає у тому що одержані результати та покращення можуть використовувати банківські установи при аналізі для захисту своїх користувачів від шахрайства.

РОЗДІЛ 1. СТАН ПИТАННЯ. ПОСТАНОВКА ЗАДАЧІ

1.1. Актуальність проблеми

Стрімкий зріст використання інтернет-ресурсів спричинив до поширеності використання банківських послуг шляхом онлайн методів. Тим більш в останні роки - відбулось спрощення багатьох транзакцій – все менша кількість видів платежів проходить саме з детальним аналізом банку – які проходять тільки через них, а не лише з ініціативи користувача. Це призвело до поширеності фактору самостійності оплати в інтернеті – що дозволило шахраям мати все більшу причетність до операцій.

Шахрайство – це будь-яка діяльність, що ґрунтується на обмані з метою отримання вигоди [15]. Іншими словами, якщо ви вводите в оману людину або організацію з метою отримання грошей або іншої важливої інформації – це і є вчинення шахрайства.

Проблеми шахрайських дій у різних сферах життя людини та особливо у сфері банківської діяльності – завжди була актуальною, оскільки призводить зазвичай тільки до негативних наслідків – як для фізичної особи – так і до підприємства яке мало б по справжньому заробити ці кошти. Через поширення технологій в шахраїв збільшилась кількість шахрайства, наприклад таких як фішингові посилання в інтернеті – по яким зазвичай люди переходять та навіть не помічають що оплачуюсь на шахрайському сайті.

Фішинг (Phishing) – вид шахрайства або форма атаки з використанням соціальної інженерії – в ході якого зловмисник (шахрай), маскуючись під надійний суб'єкт, виманює конфіденційну інформацію жертви [7] – такі як наприклад дані банківської картки.

Шахрайства з фінансовими ресурсами – мають негативні наслідки як для користувача так і для самого банку, бо це впливає на прибутковість та його репутацію. Через це банки втрачають свою спроможність виконувати свої кредитні та й інші зобов'язання перед своїми клієнтами – що призводить до зниження економічної вартості установи.

Наприклад під час свого глобального дослідження компанія KPMG International у 2018-2019 роках [1] в якому було перевірено 43 банки - 61 % респондентів повідомили про зростання загальної кількості випадків зовнішнього шахрайства, 59 % – про збільшення сум шахрайських операцій.

Також по новим даним по Україні – а саме від 04.06.2024 від Ощадбанку повідомлено – "На круглому столі було оголошено загальну статистику НБУ: кількість незаконних дій з платіжними картками, які зафіксували банки в Україні в 2023 році, зросла на чверть та становить 272 тисяч операцій. Водночас сума збитків від платіжного шахрайства становить майже 833 млн грн, і це на 73% більше від показника попереднього року. Середня сума незаконної операції перевищила 3 тис. грн, що на 39% більше ніж у 2022 році"[5]. Тобто з цього виходить що сума збитків починає збільшуватись майже в два рази від показників минулого року. Тому банки постійно повинні покращувати свої системи виявлення - та робити їх на велике випередження – бо збільшується кількість схем та їх якість використання.

Найефективніший метод боротьби з шахрайськими операціями – це недопущення і виявлення причин, які сприяли виникненню проблеми. Але все одно кожного року десятки тисяч українців стикаються з шахрайствами. Проблема у виявленні постає в його ефективності – оскільки при якомусь обмеженні – дуже великий шанс що більшість з цих операцій будуть звичайними через те що в банків немає можливості одразу перевірити – чи це людина просто переводить кошти знайомому або купує якийсь товар – чи це саме шахрайство.

Наприклад якщо проходить 1 тисяча шахрайських операцій – банку для того щоб зупинити хоча б половину з них – доведеться зупинити майже 10 тисяч звичайних операцій клієнтів – що не є шахрайськими заради безпеки.

1.2. Шахрайства та його види у банківській сфері

Шахрайства у сфері банку поширюються з кожним роком та навіть збільшують свої суми. можна поділити на два великі групи:

- шахрайства в онлайн-банкінгу – тобто через інтернет;
- шахрайства саме з банківськими картками – тобто фізичні.

1.2.1. Модель порушника.

Типологія порушників відповідно підготовленості до подолання системи охорони.

Тип порушника характеризує його ставлення до захищеного об'єкта і його можливості щодо подолання системи охорони.

Категорія відображає соціальне становище порушника.

1. "спеціаліст" - професіонали, які мають спеціалізовані знання і навички у сфері кібербезпеки або банківських технологій. Методи атак: Використовують складні технічні методи для вторгнень, мають можливість оминути передові системи захисту;

2. "аматор" можна віднести людей які мають основні технічних знань або можуть залучити зовнішній допоміжній персонал для проведення атак. Методи атак: Часто використовують прості соціально-інженерні методи або базові технічні засоби;

3. “дилетант” – можна віднести некваліфікованих осіб, які вчиняють протиправні дії без попередньої підготовки або серйозних технічних знань. Методи атак: Часто мотивовані особистими або малозначними корисливими цілями, діють імпульсивно;

4. "співробітник" – це людина, що працює безпосередньо на об'єкті захисту, та мають доступ до внутрішніх систем або інформації банку через робочі обов'язки. Методи атак**:** Використовують свій доступ для злочинних цілей, що може включати зловживання довірою, даними карток або недобросовісні дії.

Підготовленість порушника характеризується рядом параметрів, основними з яких є психологічні особливості особистості, фізичний стан, технічна оснащеність і рівень обізнаності про об'єкт і системі охорони. Ці характеристики знаходяться у взаємодії, підсилюючи або послаблюючи один одного.

Типологія порушників за характером поведінки. В цілому особу порушника можна визначити як особистість людини, яка йде на вчинення злочину внаслідок властивих йому психологічних особливостей,

антигромадських поглядів, негативного ставлення до моральних цінностей і внаслідок вибору суспільно небезпечного шляху для задоволення своїх потреб або невияву необхідної активності в запобіганні негативного результату.

Специфічна сутність особистості порушника полягає в особливостях його психічного складу, які висловлюють собою внутрішні передумови антисоціальної поведінки. Суспільна небезпека висловлюється потенційною особистості до злочинної поведінки, яка розуміється як внутрішня можливість здійснення за певних умов злочинних дій.

Можна виділити дві групи порушників, що відрізняються характером поведінки при вчиненні протиправних дій на об'єкті, - обережні і необережні.

Обережні порушники характеризуються:

- виявляються низькою тривожністю та здатністю до соціальної адаптації;
- часто намагаються встановлювати контакти для досягнення своїх цілей.

Необережні порушники:

- характеризуються високим рівнем тривожності та невпевненістю в собі;
- реагують емоційно та не завжди раціонально під час стресових ситуацій.

1.2.2. Шахрайства в онлайн-банкінгу

В онлайн-банкінгу шахрайства зазвичай передбачає собою крадіжку інформації про жертву – наприклад дані користувача для входу в застосунок онлайн-банкінгу з ціллю входу туди для повного використання його банкінгом. Або ж з найпоширенішого – крадіжка самих даних картки (реквізитів) – номера картки, строку дії, CVV коду. За допомогою цієї групи шахрайства зловмисникам вдається залишатись повністю невідомими для жертви.

До найпоширеніших шахрайства в онлайн-банкінгу – входить зазвичай – фішинг. При таких видах жертвам пропонують самостійну увійти до свого застосунку інтернет-банкінгу або надати дані до них під виглядом звичайної ідентифікації на сайті або для отримання поповнення на картку. Або ж при

викраденні вже інформації про картку – вони вже самостійно використовуючи цю інформацію роблять оплати без згоди власника.

Так само одним з великих видів підшахрайствадля фішингу – є смішинг[15] – як поєднання фішингу та SMS-повідомлень. Коли людям надсилають текстові повідомлення такі як смс – або вже стали поширеними для цього месенджери – куди теж надсилають такі повідомлення – щоб вивідати інформацію. Зазвичай в такому повідомленні вказується перейти по посиланню чи перетелефонувати на номер представляючись співробітниками служби безпеки банку. Іноді в повідомленні вказано що по посиланню людина зможе отримати виплату від держави, але вони ведуть на підроблений сайт в якому вже жертва вводить свої дані.

Ну і звісно є й інший вид цього пов'язаний з дзвінками – Вішинг – від слова "Voice" – голос та фішинг. Цей вид шахрайства полягає в дзвінку до жертви з метою вивідати інформацію або одразу переказати кошти[15] – наприклад – "Ви маєте для підтвердження ідентифікації в нашому магазині спочатку зробити переказ на картку – і після цього вони повернуться та ви зможете зробити купівлю".

Також існує такий вид шахрайства – як внутрішній – тобто дані краде саме працівник банківської установи – бо вони можуть переглядати інформацію про користувача – клієнта банку для власних цілей – але він не є дуже поширеним оскільки з ним банки швидко борються та є дуже ризикованим – бо в банків є можливість бачити що саме робив їх співробітник.

1.2.3. Шахрайства з банківськими картками (фізично)

Також звісно не варто забувати й про шахрайства з банківськими картками - які здійснюються за допомогою фізичного контакту зловмисника з жертвою. До них входять такі банальні як:

1. крадіжка картки в громадському місці або її втрата:

Тобто звичайна крадіжка картки або її втрата людиною з карману, сумочки або навіть рюкзака людини. Та потім зловмисник використовує цю картку в своїх

цілях – зазвичай вони знають обмеження що встановлюються на картки і намагаються робити оплати так щоб їх не побачили.

2. крадіжка даних працівниками магазинів:

Крадіжка даних працівниками магазинів або кафе є ще одним розповсюдженим видом шахрайства. Зловмисники іноді підкуповують працівників, щоб ті скопіювали банківські дані клієнтів під час транзакції. Ці дані потім передаються шахраям для подальшого використання у крадіжках.

3. використання програм для зламу платіжних терміналів в магазинах:

Використання програм для зламу платіжних терміналів в магазинах є ще одним поширеним методом шахрайства. Зловмисники встановлюють програми-шпигуни на платіжних терміналах магазинів або навіть супермаркетів, щоб отримати доступ до даних карток покупців. Ці програми дозволяють зловмисникам перехоплювати і збирати інформацію, включаючи номери карток і особисті дані, які потім використовуються для крадіжки грошей або здійснення ідентифікаційних крадіжок.

4. клонування картки або її копіювання (скімінг):

Скімінг – це повне копіювання даних картки жертви або навіть клонування всієї картки поки жертва здійснює транзакцію. Зловмисники використовують спеціальні пристрої, які непомітно зчитують інформацію з магнітної смуги картки або чіпа.

Крім того, скімінг може бути особливо небезпечним при знятті коштів у старих банкоматах, які працюють лише з магнітними смугами. У таких випадках зловмисники вміло використовують спеціальне обладнання, що здатне швидко і без помітної для жертви перешкоди скопіювати всі необхідні дані. Використання чіпів у картках зменшує вразливість до таких атак, адже чіп генерує унікальний код для кожної транзакції, який є надзвичайно складним для підробки навіть у разі фізичного доступу до картки.

5. втрата або крадіжка пристрою на якому обліковий запис онлайн-банкінгу:

Зловмисники при крадіжці пристрою зазвичай мають навіть легший доступу до облікового запису банкінгу жертви. Іноді на телефоні жертви може бути не встановлений пароль – тоді зловмиснику лише потрібно буде дізнатись дані для входу в застосунок банку. Дані для входу зловмисники частіше всього можуть знайти просто переглядаючи інформацію в пристрої – чати, записники або навіть просто може бути в календарі – дата дня народження – й за допомогою цього увійти в застосунок і використовувати його вже для усіх оплат з повним доступом у шахрая.

Таблиця 1.1 – Сума збитків клієнтів по місяцях в Monobank [2]

Тематика	Березень	Квітень	Травень	Червень	Липень 1–20.07
Соцінжиніринг; під тиском	7 524 470	6 970 855	7 981 791	7 680 025	5 779 332
Соцінжиніринг; фейкові збори	134 445	73 260	58 441	98 856	71 220
Соцінжиніринг; весь	7 660 915	7 044 115	8 040 232	7 778 881	5 850 552
Брокери/біржа	4 500 059	1 134 067	1 947 808	3 605 185	2 889 638
Фішинг	1 807 582	1 702 058	1 732 035	3 448 079	2 729 656
Втрата картки	288 206	188 335	207 606	302 498	221 655
Виманювання даних	936 999	1 329 396	1 914 510	2 335 318	1 035 317
Дружне/Сімейне шахрайство	584 126	258 698	338 419	454 371	241 143
Втрата картки та пристрою	120 818	91 093	34 378	26 268	190 203
Втрата пристрою	41 461	100 814	194 645	137 428	148 726
Перевипуск сім-карт	34 635	60 060	52 709	222 316	35 250

Продовження таблиці 1.1

Тематика	Березень	Квітень	Травень	Червень	Липень 1-20.07
Сума	15 974 805 грн	11 908 641 грн	14 462 346 грн	18 310 348 грн	13 342 144 грн
Відсоток додання	+11,35%	-25,45%	+21,44%	+26,61%	
Ненадання товару	9 224 622	6 959 527	8 264 091	7 689 149	5 205 957
Загальна сума	25 199 428 грн	18 868 168 грн	22 727 818 грн	25 999 497 грн	18 578 101 грн

Аналізуючи дані наглядно видно що суми шахрайства у банківській сфері тільки зростає з кожним роком – хоч й іноді бувають й деякі спади в цьому.

1.2.4. Шахрайство з ненаданням товару та фейковий заробіток

Великий обсяг шахрайства в банківській сфері мають шахрайства з ненаданням товару людині. Це шахрайство стає популярним з кожним днем бо люди все більше приймають за звичку робити купівлі в інтернеті. Самі купівлі в інтернеті це добре – але люди через це часто переглядають одразу багато товарів на різних сайтах – та через це збільшується ризик що на товарах з низькими цінами буде саме шахрай.

Зазвичай такі сайти з товаром мають дуже низьку ціну в порівнянню з іншими магазинами – та потрібно сплатити передплату за товар при доставці – зазвичай саме це насторожує покупця – і він задумується чи точно товар справжній чи можливо його не відправлять.

Зловмисниками в таких шахрайствах перебувають не якісь великі магазини, а звичайна фізична особа або фізична особа – підприємець в не дуже поширених ситуаціях. Клієнт зв'язується з продавцем та збирається придбати деякі товари або послуги що надає продавець. Продавець зі своєї сторони

повідомляє ціну та усі дані товару – та коли покупець відправить кошти – його блокують або просто не надсилають товар зовсім – зазвичай в таких ситуаціях товару взагалі немає.

Також з кінця 2023 року поширився ще один спосіб шахрайства пов'язаний з заробітком в інтернеті. В цьому виді шахрайства – тисячі людей щодня отримують повідомлення в месенджерах з пропозицією заробити грошей або навіть просто реклама в соціальних мережах про великі заробітки – ніби на дистанційній основі працювати і заробляти багато грошей на день. При цьому в таких повідомленнях вказується що для заробітку потрібно буде просто підключеним до інтернету. Завдання в таких заробітках дають прості – наприклад переглядати сторінки або робити коментарі на різних сайтах які повідомлять вам шахраї – і за кожен цей коментар вам будуть сплачувати кошти. Надаються ці завдання на сайтах де ви реєструєтесь – та прикидаються зазвичай якимись відомими інтернет-магазинами такими як – Amazon, Rozetka. Comfy і так далі.

Шахрайство полягає в тому що користувачу на початку за виконання цих завдань сплачують справжні кошти і відправляють на картку – й ти ніби заробляєш 150 грн за 5 хвилин. Потім же щоб далі продовжити працювати – клієнту повідомлять що потрібно сплатити за нову кількість завдань суму трохи більше – приблизно в 2 рази в порівнянні з тим що йому сплатили раніше за роботу – й в нових завданнях вже буде змога заробити ще більше коштів. Якщо клієнт це теж виконує – йому теж сплатять суму. Це робиться, щоб викликати у жертви довіру.

Зазвичай після двох виплат в цьому шахрайстві – пропонується ще завдання – за яких заплатять вже майже дві тисячі – клієнт звісно погоджується бо в минулі рази його не обманули на кошти. Але після виконання вже цих завдань – він не зможе отримати кошти за роботу – та для цього потрібно буде ніби сплатити за реєстрацію користувача преміум класу – і навіть після цього виплачують кошти. На цьому етапі людина вже повністю впевнена в ресурсі – й тому готова сплатити навіть суму в 5 разів більше минулої – клієнт виконує ці завдання – але вже з'являються ніби комісії за вивід та податок. Та вже після

цього його обманюють та не видають кошти. Для тих, хто відразу виявиться не готовий до «заробітку» на підвищені ставок, можуть запропонувати «роботу з переказу коштів» на картки інших людей. В такий спосіб, користуючись звичайними людьми як посередниками, кіберзлочинці переказують собі вкрадені гроші і заплутують сліди. По цих слідах саме на вас, в першу чергу, і вийде слідчий, який буде ці сліди розплутувати.

В такому способі клієнта можуть обманути на майже будь-якому етапі цього заробітку – як на початкових так і тягнути більшу суму. При цьому в таких шахрайствах в багатьох випадках клієнт надсилає кошти не саме шахраю – а між собою надсилають кошти інші клієнти. Сама схема може тривати й навіть тиждень. Про цю схему та те що вона поширюється в інтернеті повідомив співзасновник “Monobank” Олег Гороховський в своїх соціальних мережах – та що за цією схемою в них лише за зверненнями клієнтів у підтримку за підсумками листопада біля 3000 клієнтів втратило 65 мільйонів гривень – при тому що він повідомляє що ним вдається виявляти за запобігати понад 70% таких спроб. [3]

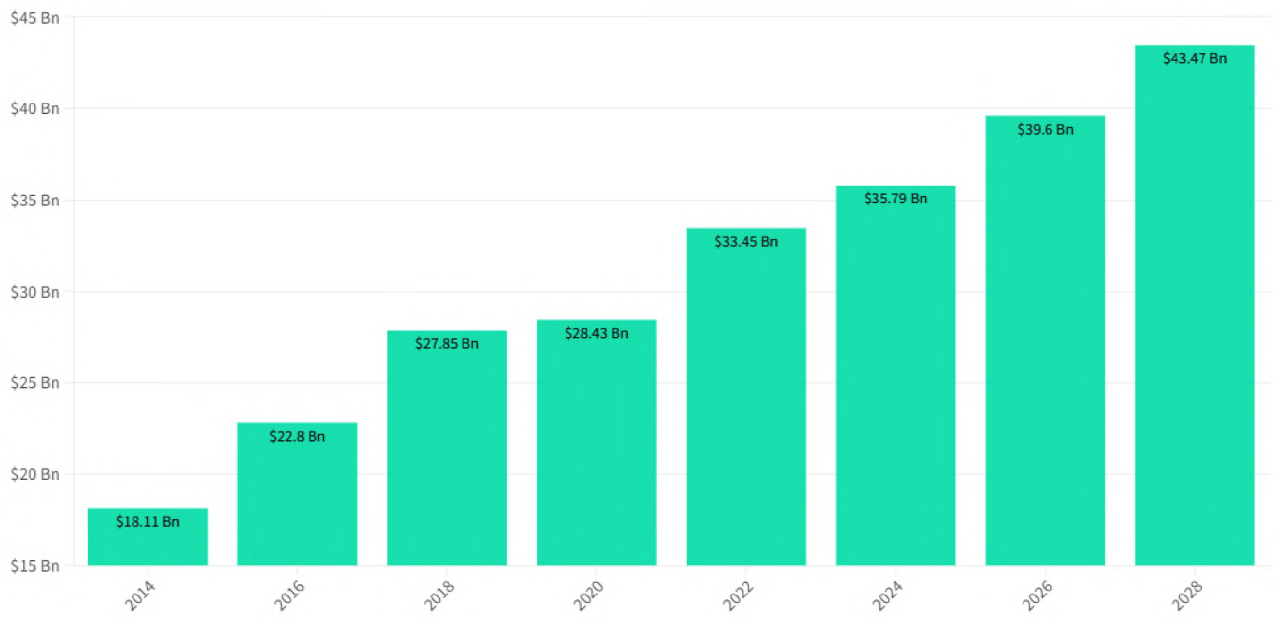

Рисунок 1.1 – Карткове шахрайство з 2014 та його прогнозування до 2028

Також проаналізувавши дані Федеральної торгової комісії США [9] за 2023 рік – згідно їх даних споживачі повідомили про втрату понад 10 мільярдів доларів США через шахрайство – та це більше на 14% від цього в 2022 році. Також було повідомлено про найбільші втрати – які стались саме через інвестиційні шахрайства (з фейковим заробітком) – всього було втрачено суму на понад 4,6 мільярдів доларів.

1.3. Сучасні типи та підходи для виявлення шахрайства

Зі зростом видів та кількості шахрайства й зрості кіберзлочинності з кожним роком яка навіть стає більш автоматизованою – в інтернеті з'являються навіть спеціалізовані набори для цього – що навіть цим може скористатись шахрай з малими навичками. Тому виникла потреба й розробці підходів до виявлення цих злочинів.

Для виявлення шахрайських дій використовуються спеціалізовані програмні комплекси що допомагають виявити частину операцій пов’язаних з оманою в операціях з різними платіжними картками. На самому початку зароджень банків – ці рішення не мали великих інвестицій та вкладень технологій. Але в останні роки до цих інструментів все більше намагаються залучати просунуті технології – такі як розробка машинного навчання та виявлення саме за допомогою неї шахрайських дій – бо за допомогою цього можна буде проаналізувати більшу кількість тонкощів в кожній дії – наприклад таку як сума та час операцій.

Але звісно для її розробки потрібно спочатку більш стандартні алгоритми аналізу – статистичні – аналітика для виявлення аномалій у транзакціях. Тобто використовується близько 2 актуальних технік на сьогоднішній день – це використання штучного інтелекту для аналізу та виявлення шахрайства – але їх розробка коштує великих грошей та часу. Також є й інший тип – аналіз статистичних даних в якому обов’язково аналізувати та шукати самостійно шаблони та співставляти дані для аналізу. Застосування цих методів дозволяє підвищити точність виявлення шахрайських схем. Однак, розвиток і впровадження таких систем вимагає постійного вдосконалення та інвестицій.

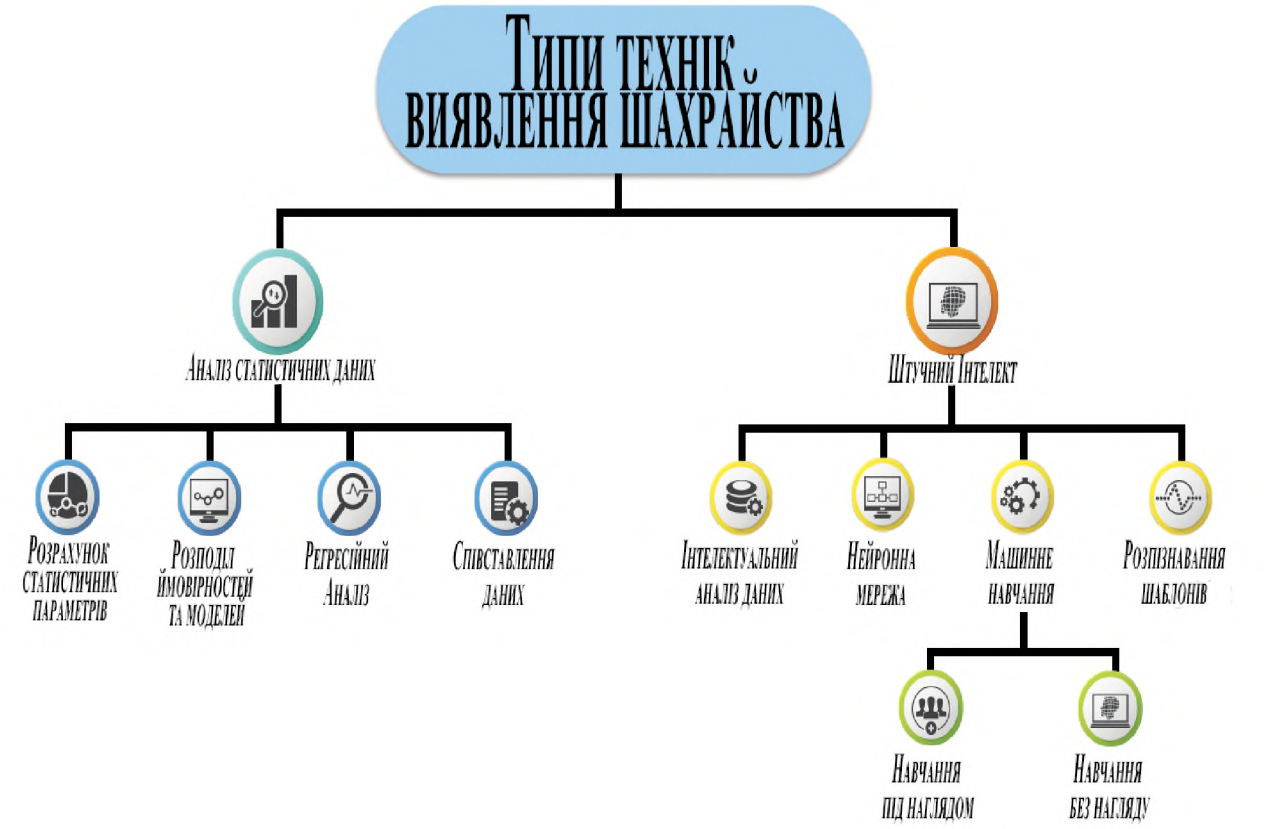

Рисунок 1.2 – Типи сучасних методів виявлення шахрайства

Головні проблеми на шляху таких систем полягають у тому, що дуже важливо не тільки зупиняти дії шахраїв – але й випадково не протидіяти звичайним користувачам щоб не зменшити свою репутацію. Тому для цього важливо знати усі випробування таких систем:

1. Кількість помилкових спрацювань

При малій кількості додаткових перевірок та покращенні блокування системою – дуже часто веде до збільшення й кількості помилкових спрацювань – яких дуже важко повністю позбавитись. Тому для цього при розробці системи важливо враховувати більшу кількість параметрів – для зменшення кількості просто схожих випадків – які б погіршували ставлення клієнтів до банку – й потребувало їх робити додаткові дії для розблокування своєї транзакції чи навіть повністю облікового запису

2. Обробка транзакцій та їх швидкість

В суспільстві та міжбанківській конкуренції – дуже важливо мати гарні показники швидкості обробки платежів. В останній час ці показники намагаються досягнути стандартів обробки близької до миттєвих – таких як просто P2P-перекази з картки на картку – в сучасних банках вони оброблюються за секунду – та якщо вони займають годину – це вже дуже погано й може вплинути на те що користувач змінить банк.

3. Зростання витонченості видів шахрайства

Нові атаки, такі як соціальна інженерія, бот-мережі та шкідливе програмне забезпечення, обходять традиційні системи виявлення, засновані на правилах залежних від відомих сигнатур шахрайства. Протистояти їм майже неможливо – але все ж є деякі правила які під це підпадають й гарним показником є їх виявлення.

4. Витрати на розробку системи

Дуже важливо постійно покращувати систему виявлення, бо шахраї теж покращують схеми шахрайства для того щоб їх не виявляли. Для цього ж потрібно банкам покращувати системи та збільшувати витрати на них – для того щоб вкладатись у найбільш сучасні методи. Це відбувається оскільки застарілі системи не зможуть протистояти новим загрозам – й шахраї просто не будуть користуватись тим способом що блокується.

5. Можлива анонімність через віддалені транзакції

Кожного дня відбувається дуже багато транзакцій – і через те що вони стали можливі всі дистанційно – без обов'язкової присутності та ідентифікацією банком – ставить під питання чи робить цю транзакцію саме клієнт – чи можливо клієнт втратив вже телефон і їм користується шахрай.

1.4. Наявні системи детектування шахрайських операцій

На сьогоднішній день існують системи для виявлення шахрайства. Всього серед них актуальних на сьогоднішній день цих систем лише близько 8-10. Та потрібно розглянути деякі з них – для того щоб розуміти яка з них краще й для чого підходить в кожній ситуації при боротьбі та детектування шахрайства. Серед

актуальних та найкращих згідно [13] та [19] займають такі системи детектування та моніторингу шахрайських операцій:

- SEON

Це інструмент що спеціалізується саме на аналізі в реальному часі, який гарно підходить для бізнесу – але малого та середнього. Система використовує машинне навчання для аналізу та збирає дані з різних джерел для створення комплексів користувача – та допомогаючи виявити аномалії. Ця система гарна своєю швидкістю обробки операцій та визначенням ризику кожної з них. Переваги – використовує різні дані та моделі машинного навчання. Пропонує для швидкого реагування свої системи моніторингу та оповіщення про можливе шахрайство.

- Kount

Це рішення корпоративного ступеня – яке так само використовує штучний інтелект для аналізу даних та прийняття рішень – та дуже гарне для моніторингу операцій та захисту від шахрайства де не присутня картка. Так само вона надає параметри для свого налаштування та свої підходи для боротьби з шахрайськими діями – крадіжкою персональних даних. Й система має щорічно дані про мільярди взаємодій користувачів – що надає їй за допомогою цього перевагу саме на стадії ідентифікації клієнтів. Переваги – інтеграція з внутрішніми інструментами та API відбувається доволі легко та без помилок. Пропонує свої розширені правила та функції відзвітування для вдосконалення в режимі реального часу.

- Cybersource

Система надає розширені можливості для боротьби з шахрайством під назвою – менеджер рішень. Cybersource розробив комплексні модулі, які пропонують розгалужені можливості для торговців у плані розширення їхнього бізнесу. Ці модулі включають в себе інструменти для зручної конвертації валют, забезпечення відповідності з глобальним податковим законодавством та ефективного управління термінами обслуговування клієнтів. Використання цих інноваційних функцій дозволяє торговцям оптимізувати їхні процеси, підвищити

конкурентоспроможність та забезпечити більш ефективне обслуговування своїх клієнтів. Переваги – відома своєю високою точністю, гнучкістю налаштувань, глобальним охопленням, якісною підтримкою клієнтів та надійністю в своєму аналізу.

- Sift

Динамічно захищає бізнес від різних загроз в режимі реального часу використовуючи передові можливості штучного інтелекту та велику мережу даних. Також пропонує багато рішень для захисту платежів, цілісності та захисту облікового запису. Використовуючи штучний інтелект та велику мережу даних, вона забезпечує надійний захист платежів та облікових записів. Переваги – має гарні можливості у відловлюванні ботів та робить повністю автоматичну перевірку на шахрайство. Надає дані що є легкими для перевірки та читання з пошуком інформації.

- Ekata

Це рішення для підтвердження особи та запобігання шахрайства по всьому світу та має до себе гарну довіру – оскільки ця компанія підвласна Міжнародній Платіжній Системі – Mastercard. Також система надає різні функції для різних випадків – як запобігання фальшивим обліковим записам та перевірка платежів на автентичність. Також вони надають такі API від себе API Ризикових транзакцій, API ризикованих адрес та API телефонної розвідки та також інтерфейси для роботи з API для аналітиків – що полегшує додаткові перевірки при його використанні.

Переваги – пропонує оцінку ризиків, надаючи детальну інформацію про те, наскільки ризикованою є навіть просто номер телефону та її ідентифікація користувачів працює в різних країнах. Має зручний інтерфейс для навігації. Й зменшує ризик крадіжки особистих даних, перевіряючи правильність адреси, місцезнаходження та інших ідентифікаційних даних.

1.5. Системи антишахрайського обміну інформацією в Україні

Зазвичай в країнах не дуже розвинена можливість обміну інформацією про шахраїв між собою. Але в Україні є система для обміну такими користувачами –

“EMA Anti fraud Hub” – це захищений портал, через який різні компанії та корпорації роблять анти-шахрайський обмін даними або їх транзит із застосуванням API та інших способів комунікації. Це реалізовано у формі ПЗ (програмного забезпечення) – та за допомогою цього компанії між собою обмінюються новими даними про шахраїв та їх методи.

ЄМА – Українська міжбанківська Асоціація членів платіжних систем, створена банками України 1999 року з метою всебічного сприяння розвитку безготівкових розрахунків в Україні.

За допомогою цієї системи між банками налагоджене співробітництво у сфері захисту. Між ними передається інформація про чорний список банку та фіксація користувача та запис інцидентів. Автоматизація збору і перевірки інформації про інциденти платіжного і кредитного шахрайства. Також в цих системах було створено – вебдодаток для автоматизованої передачі інформації про місця зняття готівки з карток, використовуваних в шахрайських схемах, від банків до кіберполіції.

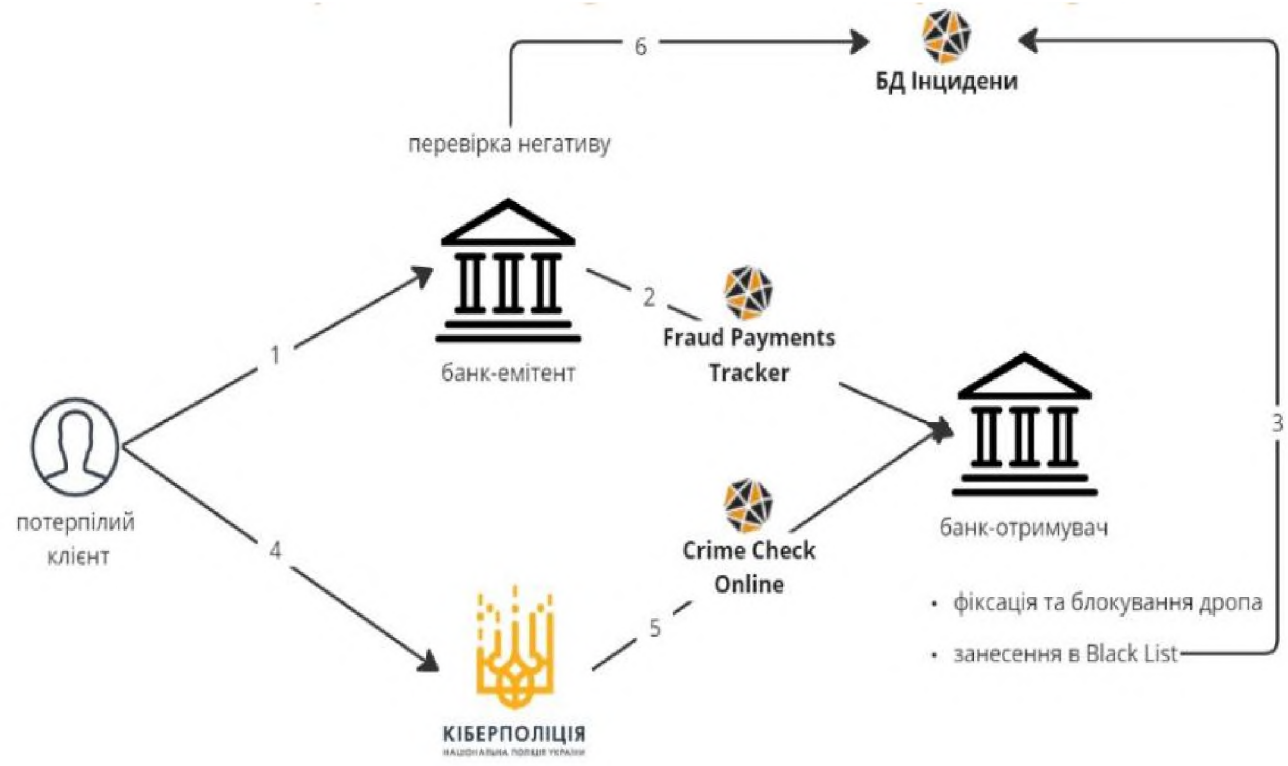

Рисунок 1.3 – Використання сервісу “AntiFraud Hub” банками

Завдяки цій ініціативі банки можуть швидко реагувати на потенційні загрози та вчасно вживати заходів для запобігання фінансовим втратам. Це підвищує загальний рівень безпеки у фінансовому секторі України та сприяє довірі клієнтів до банківських установ. Крім того, такий обмін інформацією допомагає кіберполіції у виявленні та затриманні злочинців, що використовують шахрайські схеми.

1.6. Висновок першого розділу

В даному розділі було наведено наявність та актуальність проблеми шахрайських дій у банках та розглянуто основні види шахрайства та загроз у банківській сфері. Також було проаналізовано приклади шахрайства та їх видів – та перевірено актуальну інформацію по збиткам та поширеності проблем. В розділі ще було наведено сучасні типи та підходи для боротьби та виявлення шахрайства – які вони існують та як між собою відрізняються.

Також було наведено актуальні системи для боротьби з шахрайством та проаналізовано їх плюси. Це також є дуже важливою темо скільки банківські операції що кожен рік підпадають під шахрайства – є одним з найбільших видів збитків у сучасному світі. Саме тому також проаналізовано наявні в Україні методи обміну між собою інформацією про різні види омани та злочинів.

РОЗДІЛ 2. СПЕЦІАЛЬНИЙ РОЗДІЛ

2.1.Різниця методів виявлення шахрайства

Надійне виявлення шахрайства в реальному часі в банківській сфері ґрунтується на синергії між автоматизованими інструментами виявлення шахрайства та людськими аналітиками. Але також варто пам'ятати що в сфері банківського шахрайства розглядають в основному зовнішнє середовище, як ініціатора шахрайства, що є не зовсім правильно. Тому можливості вторгнення повинні враховувати також і внутрішні аспекти загрози.

Як правило виявлення шахрайства автоматизовані так що спочатку виконують функції – використовуючи біометричні дані – відбитки пальців і методи моніторингу транзакцій. Вже після цього вони передаються на аналіз шахрайства або в журнал операцій – й там вже аналізується на аномалії. Якщо ж це аномалія – то в такому випадку одразу запускається різні реакції в залежності від рівня аномалії та її загрози. Реакція та прийняття рішень вже приймаються аналітиком – в якого вже є детальна інформація від системи виявлення шахрайства.

Після аналізу інформації про виявлення шахрайства було виявлено що для виявлення шахрайств згідно [10] зазначено, що на сьогоднішній день широко використовується:

- логістична регресія – вона здатна вирішувати задачі по категоріям; метод дозволяє оцінити ймовірність події, що відбувається. Це один з найпростіших і найефективніших методів для виявлення аномалій у даних. Аналізуючи різні характеристики транзакцій, такі як сума, місце та частота, моделі логістичної регресії можуть передбачити ймовірність того, що транзакція є шахрайською, на основі вивчених закономірностей з історичних даних.
- метод опорних векторів – за допомогою нього є можливість обробляти між собою дані зі складними зв'язками між змінними; він відомий своєю здатністю працювати з високорозмірними даними. Метод особливо ефективний у випадках, коли класи важко відділити лінійно.

– випадковий ліс (random forest) – це ансамблевий метод машинного навчання, що складається з множини дерев рішень. Він забезпечує високу точність і стійкість до перенавчання, що робить його ефективним для виявлення шахрайства. Дерева рішень – це інтуїтивно зрозумілі моделі машинного навчання, які можуть ефективно виявляти складні закономірності в даних про транзакції. Розбиваючи дані на підмножини на основі значень різних ознак, дерева рішень можуть класифікувати транзакції як законні або шахрайські.

– само-організовані карти – вони використовуються для класифікації та кластеризації інформації; ці карти можуть самостійно навчатися без попередніх міток даних. Це корисно для виявлення нових, невідомих раніше шаблонів шахрайства.

– нечітка логіка, яка підвищує ефективність управлінських рішень. вона дозволяє працювати з неточними і неповними даними, надаючи гнучкіші моделі прийняття рішень. Це важливо для виявлення складних і непередбачуваних схем шахрайства.

Для виявлення фінансового шахрайства важливо моніторити поведінку власників карткових рахунків. У роботі [11] застосовується прихована марківська модель (HMM), яка навчається на основі нормальних дій власника картки і виявляє шахрайські операції. Робота [10] використовує теорію нечіткої логіки для цієї ж мети.

Генетичний алгоритм, представлений у роботі [14], враховує втрати від помилкової класифікації транзакцій, надаючи пріоритет правильній класифікації важливих операцій. Початкові дані для алгоритму включають використання картки, місцезнаходження, баланс і середньодобову суму знятих грошей. Алгоритм визначає критичні значення цих змінних, які потім застосовуються разом із технологіями Data Mining для виявлення шахрайства, включаючи шахрайські дії з боку банківського персоналу.

Порівняльний аналіз методів виявлення шахрайства виділяє чотири основні групи: якісні методи, кількісні методи, методи машинного навчання та гібридні методи. Гібридні методи виявилися найбільш ефективними, оскільки

поєднують сильні сторони різних підходів. Подальші дослідження можуть сприяти створенню інтелектуальної системи протидії шахрайству, що підвищить загальну ефективність захисту банків і запобігання шахрайству з боку персоналу.

Результати аналізу порівняння математичних методів виявлення шахрайства у банківській сфері, що здійснює персонал банку можна провести у вигляді таблиці:

Таблиця 2.1 – Порівняльний аналіз методів виявлення шахрайств у банках, що здійснюються персоналом банку

Група методів виявлення шахрайства у банках	Основні характеристики	Урахування невизначеності
Якісна (нечітка логіка)	Базуються на експертних оцінках	Невизначеність враховується за допомогою експертних оцінок
Машинне навчання (нейронні мережі, дерево рішень)	На технологіях штучного інтелекту (навчання з учителем і без нього)	Невизначеність враховується за допомогою засобів статистики та теорії ймовірностей
Кількісні (асоціативний аналіз, логістична регресія)	Базується на традиційній математичній системі	Враховується за допомогою засобів статистики та теорії ймовірностей
Гібридні (нейро-чіткі системи)	Базуються на поєднаному підході (використовуються сильні сторони різних методів)	Невизначеність враховується за допомогою кількісного та якісного математичного аналізу

2.2. Методи виявлення аномалій

Виявлення аномалій – це метод аналізу даних, який використовується для виявлення точок даних або шаблонів, що значно відхиляються від очікуваної поведінки. Ці відхилення, або аномалії, сигналізують про потенційно небезпечні тенденції, на які слід звернути увагу дослідникам.

Ці методи є одними з найважливіших та найпростіших – бо виявляючи аномальну поведінку можна запобігати шахрайським операціям до того як вони завдадуть певної шкоди. Ці шахрайські можуть бути – просто транзакціями, спроби входу в обліковий запис чи застосунок – як саме вони були – через відбиток чи можливо через пін-код, незвична поведінка або інша підозріла діяльність.

2.2.1. Методи виявлення аномалій за допомогою статистики й аналізу

Для виявлення аномалій використовується статистичний аналіз із застосуванням таких показників, як Z-бали, щоб точно визначити відхилення в даних. Виявлення статистичних аномалій ґрунтується на аналізі статистичних властивостей даних для виявлення випадків, які значно відхиляються від норми. Ці методи припускають, що аномалії мають відмінні статистичні характеристики порівняно з нормальними точками даних. Використовуючи статистичні методи, можна виявити аномалії на основі таких показників, як середнє значення, стандартне відхилення .

Метод Z-балів є дуже поширеним статистичним методом виявлення аномалій. Точки даних з Z-балами, що перевищують певний поріг, вважаються аномаліями. Також створюються профілі для представлення поведінки користувачів – з якими й порівнюються ці бали – ці порівняння можуть бути наприклад зі швидкістю діяльності користувача. Зазвичай кожен користувач має свій запис – в якому прописано його модуль поведінки зазвичай та поточний профіль – який періодично обчислюється та порівнюється для виявлення аномалії. Показник при виявленні аномалії виявляє незвичність кожної дії чи навіть пакету даних від мережі для перевірки.

Якщо показник аномальності в якомусь місці перевищує поріг – то буде надіслано оповіщення співробітникам або одразу буде заблоковано деякі дії користувача. Ця система має гарні переваги у виявленні нових шахрайських дій – яких зазвичай ніколи не було. Після виставлення стандартного параметру – система одразу може перейти в режим обробки інформації та сповіщати про злочини з доволі великою ймовірністю. Але визначити параметр по якому вирівнюватись дуже важко – бо якщо його виставити дуже низько то буде великий процент помилкових спрацювань – тому потрібно встановлювати вище але тоді збільшується кількість шахрайських дій що пройде крізь цей метод

Однак серед недоліків системи є ще те що статистика може зламатись якщо шахраї будуть потрохи змінювати поведінку – й для системи вже те що спочатку було найбільш високою аномалією – буде звичайним. Й тому використовувати тільки статистичний метод виявлення не є доцільним – й разом з ним має бути обов'язково інший. Реалізувавши цей метод, буде можливість ефективно виявляти та аналізувати аномалії у наборах даних.

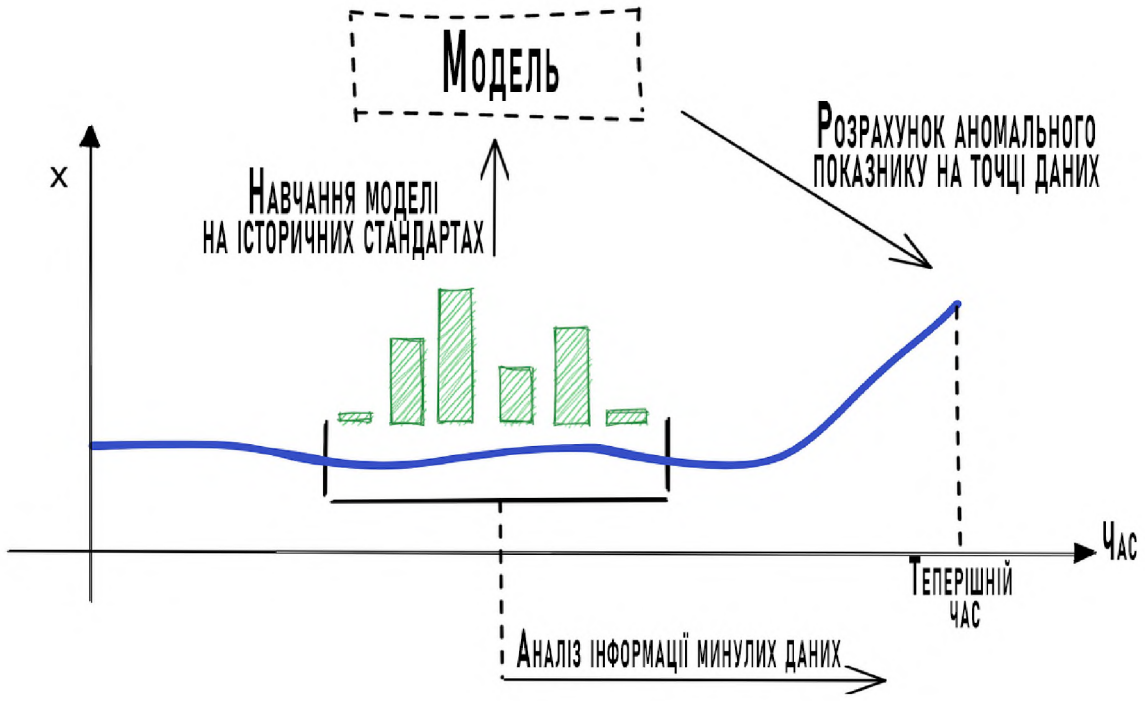

Рисунок 2.1 – Аналіз аномалій за допомогою статистичного виявлення

Також цей метод можна реалізувати дуже просто навіть за допомогою Python – досліджувати набір даних, щоб зрозуміти його структуру, розподіл та будь-які потенційні аномалії. За допомогою бібліотек Python, таких як Matplotlib або Seaborn, можна отримати уявлення про їхні закономірності та характеристики. Наприклад це було наведено в статті [16].

Виявлення аномалій у реальному часі важливе з кількох причин. По-перше, це може допомогти швидко виявити та виправити помилки до того, як вони спричинять значні пошкодження або збої в роботі. По-друге, це може дозволити проактивно моніторити системи на предмет потенційних проблем і швидко реагувати, якщо проблеми виявлено.

2.2.2. Контрольовані методи навчання виявлення аномалій

Контрольоване виявлення аномалій використовує марковані дані для навчання класифікатора, який може розрізняти нормальні та аномальні екземпляри. аномалій будує моделі нормальної поведінки на основі історичних прикладів нормальних і аномальних точок даних. Алгоритми класифікації можуть передбачити, чи є нові точки даних аномаліями.

Переваги такого навчання під наглядом:

- висока точність: оскільки моделі навчаються на маркованих даних, вони можуть робити дуже точні прогнози на нових небачених даних. Наприклад, моделі класифікації зображень можуть ідентифікувати об'єкти на зображеннях з точністю понад 90%.
- менш схильні до перенавчання: Оскільки навчальні дані позначені, моделі мають меншу схильність до перенастроювання на навчальні дані.
- широкий вибір алгоритмів: Існує багато добре працюючих алгоритмів, таких як лінійна регресія, випадковий ліс, SVM, нейронні мережі тощо.

Недоліки керованого навчання:

- потребує великих маркованих навчальних даних: Створення маркованих навчальних даних є дорогим і трудомістким процесом. Для деяких додатків може бути неможливо отримати тисячі мічених прикладів.

– не підходить для неструктурованих даних: Керовані моделі погано працюють з неструктурованими даними, такими як текст, аудіо та відео, які важко і дорого маркувати.

Ще однією з проблем у контрольованому виявленні аномалій є робота з незбалансованими даними. Класифікаційні моделі прагнуть розподіляти дані по різних категоріях. У незбалансованих наборах даних одна категорія займає значну частину навчального набору (клас більшості), тоді як інша категорія є менш представленою (клас меншості). Проблема з моделлю, навченою на таких незбалансованих даних, полягає в тому, що вона може досягти високої точності, постійно передбачаючи клас більшості, навіть якщо розпізнавання класу меншості є надзвичайно важливим у реальних сценаріях.

Рисунок 2.2 – Графік звичайних та шахрайських транзакцій з незбаланованими даними

Наприклад, якщо зібрати дані для моделі, що прогнозує Шахрайські та звичайні транзакції, то більшість зразків, скажімо, 99%, будуть даними що не були шахрайськими, тоді як дані зареєштовані складатимуть набагато меншу частину. Під час навчання модель може досягти 99-відсоткової точності, просто передбачаючи "Звичайну" для кожного випадку. Це створює серйозну проблему,

оскільки тоді від цієї моделі не буде великого сенсу у виявленні шахрайства. Для прикладу я створив код (Рис. 2.3) який би проаналізував набір даних з транзакціями і збудував би графік з ними (Рис. 2.2). Й видно за допомогою цього яскравий приклад незбалансованості даних.

```
import numpy as np
import pandas as pd
import seaborn as sns
import plotly.express as px
from matplotlib import pyplot as plt
from sklearn.model_selection import cross_val_score

from sklearn import metrics
from collections import Counter

try:
    raw_df = pd.read_csv('../input/creditcardfraud/creditcard.csv')
except:
    raw_df = pd.read_csv('creditcard.csv')

labels=["Звичайні","Шахрайські"]

fraud_or_not = raw_df["Class"].value_counts().tolist()
values = [fraud_or_not[0], fraud_or_not[1]]

fig = px.pie(values=raw_df['Class'].value_counts(), names=labels , width=700, height=400, color_discrete_sequence=["blue","red"]
             ,title="Шахрайські vs Звичайні транзакції")
fig.show()
```

Рисунок 2.3 – Код для аналізу набору даних та побудови графіку

2.2.2.1. Вирішення проблеми незбалансованості даних

Для розв'язання проблеми незбалансованих даних існують різноманітні стратегії. Одним із методів є повторна вибірка, яка допомагає врівноважити класи. Це може передбачати збільшення кількості даних класу меншості або зменшення кількості даних класу більшості. Вирівнювання набору даних спрощує навчання моделі, запобігаючи її упередженості до конкретного класу. Тобто, модель перестане віддавати перевагу класу з більшою кількістю даних лише через його чисельність.

Однак, при недостатній вибірці може бути втрачено потенційно важливі дані. Надмірна вибірка, з іншого боку, означає збільшення кількості зразків класу меншості шляхом їх копіювання. Збільшення вибірки класу меншості може бути корисним, якщо даних недостатньо.

Недоліком же недостатньої вибірки є ризик перенавчання та погане узагальнення на тестовому наборі даних та збільшення часу навчання моделі завдяки новому більшому набору даних який потрібно проаналізувати.

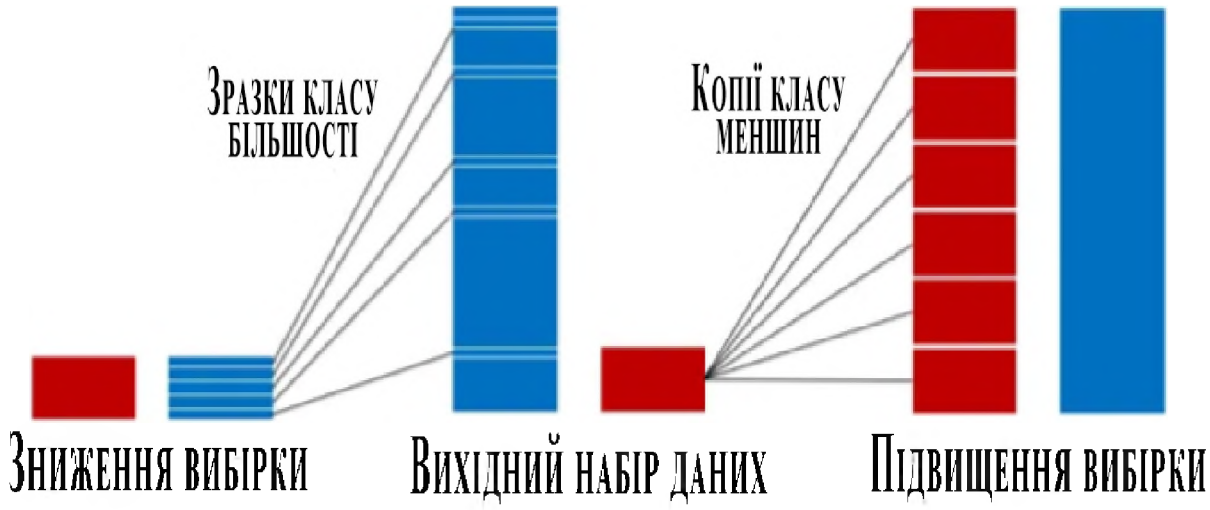

Рисунок 2.4 – Зразок результату зниження вибірки та її підвищення

Основна концепція методів цієї категорії полягає в балансуванні наборів даних до пропорції 50 на 50. Це досягається шляхом повторного вибору вихідних даних, створення нових зразків для менш представленого класу або комбінування обох підходів.

Зазначені методи включають випадкове видалення або повторення зразків. Щоб уникнути можливих проблем, які можуть виникнути після таких маніпуляцій, можна створювати штучні зразки для менш представленого класу. Це можна здійснити за допомогою різних систематичних підходів. Розглянемо кілька відомих методів.

Метод синтетичного створення надмірної вибірки з меншості – SMOTE (Synthetic Minority Oversampling Technique) - це техніка збільшення вибірки, яка генерує синтетичні зразки для менш представленого класу, використовуючи наступні кроки - Зразок з класу меншості випадково вибирається, після чого використовується метод K-найближчих сусідів (KNN) для вибору K зразків з цього ж класу. Потім випадково обирається зразок у межах обраної області і

генерується новий зразок, проводячи лінію між двома зразками (помаранчева зірочка на рисунку). Приклад цього методу зображений на рисунку 2.3.

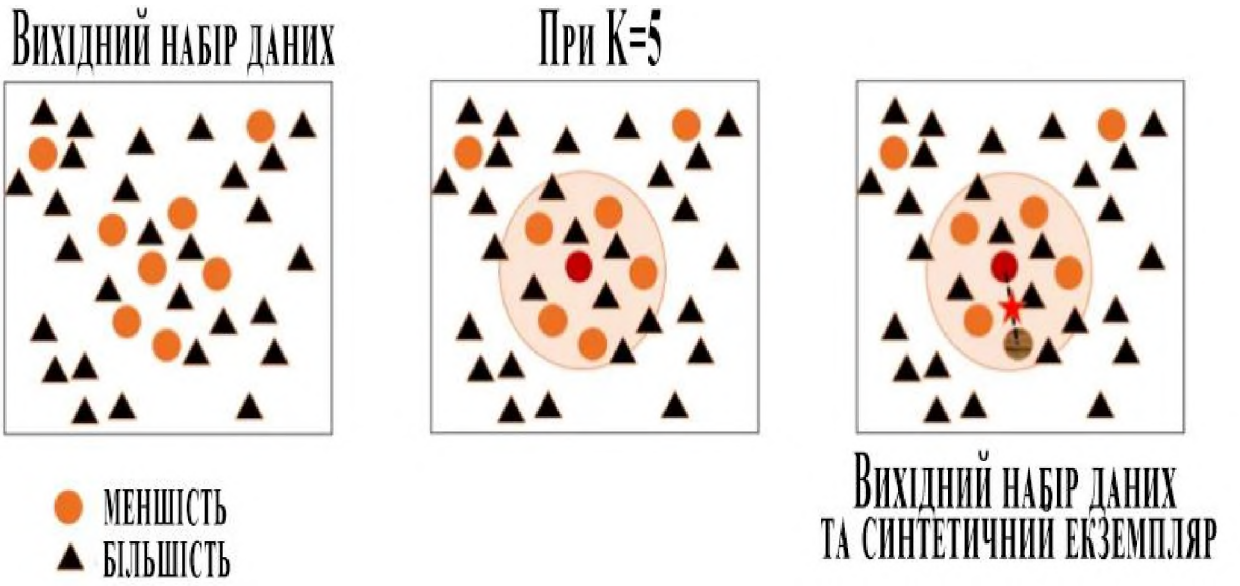

Рисунок 2.5 – Демонстрація методу SMOTE при виборі 5 сусідів

Цей підхід визначає, що точки даних з більшою відстанню до свого k-го найближчого сусіда ймовірніше визначаються як аномалії. Іншим підходом є використання середніх відстаней до kнайближчих сусідів, що має свої переваги, враховуючи локальну щільність точок.

Хоча SMOTE реалізується просто, його основна слабкість полягає в тому, що він присвоює однакову вагу всім зразкам з класу меншості, а зразки, які складніше вивчити, можуть бути недоступними у певних випадках.

Адаптивне синтетичне створення зразку – ADASYN (Adaptive Synthetic Sampling). ADASYN використовує ідею адаптивного створення вибірок для класів меншин відповідно до їх розподілу. Цей метод іноді усуває головний недолік алгоритму SMOTE, створюючи складніші синтетичні дані для навчання класу меншин. На рисунку 2.4 демонструється, як ADASYN відрізняється від SMOTE у процесі генерації синтетичних даних.

ADASYN частіше використовує зразки з менш представлених класів, які характеризуються більш складними властивостями. Використання ADASYN в машинному навчанні призводить до покращеної продуктивності класифікації за

рахунок генерації синтетичних зразків і балансування набору даних, а також зменшує зсув в бік більшості класів у незбалансованих наборах даних. Крім того, ADASYN сприяє покращенню здатності моделей машинного навчання до узагальнення шляхом акценту на генерацію зразків для складних для вивчення екземплярів класів меншин, що робить його застосовним у різних сферах, таких як виявлення вторгнень та шахрайства.

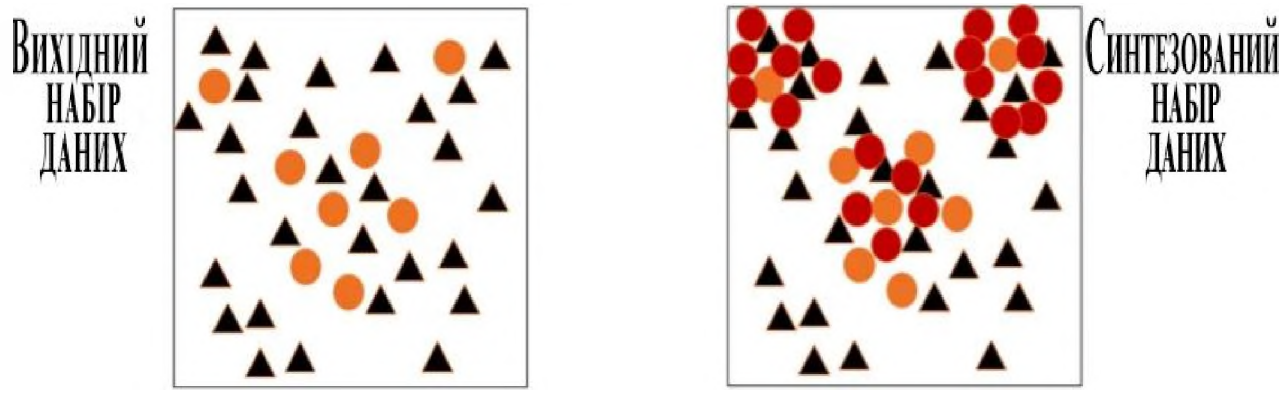

Рисунок 2.6 – Демонстрація методу ADASYN

ADASYN і SMOTE є методами надвибірки для роботи з незбалансованими даними. Основна відмінність між ними полягає в тому, що ADASYN адаптивно генерує синтетичні вибірки, враховуючи розподіл щільності класів меншості, тоді як SMOTE створює синтетичні вибірки за допомогою інтерполяції між вже існуючими вибірками класів меншин. Цей адаптивний підхід ADASYN спрямований на збільшення уваги до складних для навчання вибірок, що може призвести до поліпшення якості класифікації.

2.2.3. Напівконтрольовані методи навчання виявлення аномалій

У сучасних дослідженнях для створення моделей виявлення кібератак в інтелектуальних мережах переважно використовуються алгоритми з наглядом (контрольовані). Ці алгоритми потребують як нормальних даних, так і даних про атаки для навчання. Проте зібрати репрезентативні приклади різних атак часто складно або навіть неможливо, що може призвести до низької ефективності у виявленні певних типів атак, особливо тих, що не були представлені у навчальних

даних. Крім того, постійна еволюція методів кібератак ускладнює підтримку актуальності навчальних даних, що також може негативно вплинути на точність та швидкість виявлення нових загроз.

Але існує ще метод виявлення кібератак в інтелектуальних мережах, заснований на напівконтрольованому виявленні аномалій. Напівконтрольовані алгоритми виявлення аномалій використовують лише дані звичайних подій для навчання моделі, здатної розпізнавати нові типи атак. На рисунку 2.2 представлено порівняння між контрольованими та напівконтрольованими алгоритмами виявлення аномалій.

Такі методи, як самонавчання, можуть починатися з контрольованої моделі, а потім використовувати її прогнози на немаркованих даних, щоб поступово вдосконалюватися, не потребуючи більше маркованих даних.

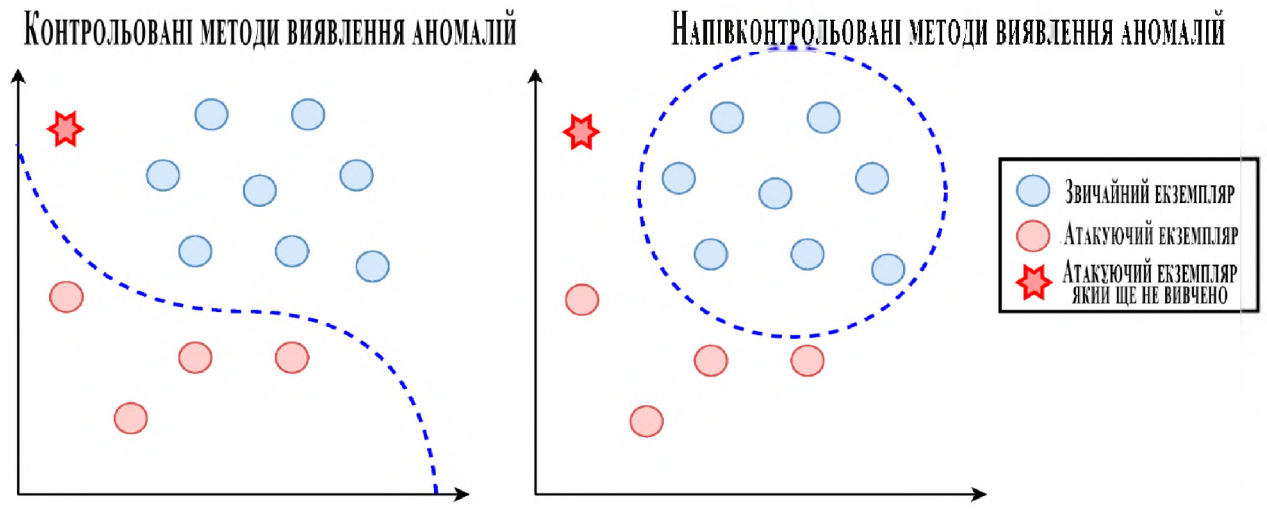

Рисунок 2.7 – Порівняння контрольованого та напівконтрольованого методу

Також в статті [12] проаналізували декілька напівконтрольованих алгоритмів виявлення аномалій і визначили найефективніші для виявлення кібератак в інтелектуальних мережах. Ефективність напівконтрольованих алгоритмів була порівняна з популярними контрольованими алгоритмами, щоб продемонструвати їх переваги у виявленні атак. Крім того, ними було використано глибоке навчання для виділення дискримінантних ознак, що дозволило ще більше покращити точність виявлення атак.

2.2.4. Неконтрольовані методи навчання виявлення аномалій

Неконтрольоване виявлення аномалій не потребує маркованих даних для виявлення відхилень. Натомість воно покладається на статистичні або засновані на відстані показники, щоб оцінити, наскільки екземпляр відрізняється від решти даних. Це виявлення використовує такі методи, як кластеризація та аналіз викидів, щоб змоделювати розподіл більшості точок даних. Нові точки, що виходять далеко за межі вивченого розподілу, позначаються як аномалії.

Переваги неконтрольованого навчання:

- виявлення прихованих закономірностей: Методи, такі як кластеризація та видобуток правил асоціацій, можуть виявляти цікаві зв'язки та групування у великих немаркованих наборах даних;
- відсутність потреби у маркуванні: Оскільки дані для навчання не вимагають позначення, методи неконтрольованого навчання легко застосовуються у нових областях.

Недоліки неконтрольованого навчання:

- результати є суб'єктивними, оскільки відсутні об'єктивні метри точності. Корисність результатів значною мірою залежить від того, як людина їх інтерпретує;
- схильні до надмірного переналаштування – без зворотного зв'язку у вигляді міток істини неконтрольовані моделі схильні до надмірного налаштування на хибні шаблони.

2.3. Методи машинного навчання

Машинне навчання знаходить все більше застосувань у виявленні та запобіганні шахрайству, оскільки воно може аналізувати об'ємні дані, виявляти закономірності і адаптуватися до нових даних.

Алгоритми машинного навчання можуть виявляти незвичайні відмінності в даних про транзакції, які відхиляються від звичайної поведінки. Вони адаптуються до історичних даних, вчаться розпізнавати легітимні транзакції та відзначати підозрілі активності, які можуть сигналізувати про можливість

шахрайства. Вони також можуть враховувати контекст та інші фактори, щоб підтвердити або відхилити підозрілі дії з високою точністю. Це дозволяє зменшити кількість фальшивих спрацювань і підвищити ефективність виявлення шахрайства.

Глибоке навчання використовує штучні нейронні мережі (ANN) і здатне аналізувати складні моделі та зв'язки у наборах даних без необхідності повного програмування. Ця технологія набуває популярності завдяки зростанню обчислювальних можливостей та доступності великих обсягів даних, особливо через використання глибоких нейронних мереж (DNN), які є її основою.

Кероване навчання описує сценарій, в якому досвід описується навчальним датасетом, що містить важливу інформацію (бажаний результат, мітку), якої, натомість немає в 29 майбутніх тестових прикладах, до яких планується застосовувати «навчений досвід» В такому випадку може розглядати середовище як вчителя, який керує навчанням, надаючи додаткову інформацію (мітки). В некерованому навчанні відсутня різниця між навчальним та тестовим датасетом. В такому випадку навчання полягає в тому, щоб отримати коротшу (стиснену) версію даних або ж прийти до певного висновку на основі даних.

Прихована марковська модель є статистичною моделлю, де система, яку аналізують, уявляється як марковський процес з прихованими станами.

При навчанні з підкріпленням агент взаємодіє з навколишнім середовищем, виконуючи дії, і навчається за допомогою зворотного зв'язку. Зворотній зв'язок надається агенту у вигляді винагороди, наприклад, за кожну хорошу дію він отримує позитивну винагороду, а за кожну погану – негативну . Нагляд за агентом відсутній.

Лінійна регресія – один з найпопулярніших і найпростіших алгоритмів машинного навчання, який використовується для предиктивного аналізу. Тут предиктивний аналіз визначає передбачення чогось, а лінійна регресія робить прогнози для безперервних чисел, таких як зарплата, вік тощо.

Лінійна регресія є одним з найпопулярніших і найпростіших алгоритмів машинного навчання, призначеним для прогнозування безперервних числових

значень, таких як вік чи зарплата. При цьому предиктивний аналіз передбачає майбутній стан даних.

Логістична регресія, у свою чергу, є алгоритмом керованого навчання, спрямованим на прогнозування категоріальних або дискретних значень. Вона застосовується для класифікації даних, які можуть бути виражені у форматі "Так" або "Ні", "0" або "1", "Червоний" або "Синій".

Хоча логістична регресія подібна до лінійної регресії, вони використовуються для вирішення різних завдань: лінійна регресія – для прогнозування безперервних значень, логістична регресія – для класифікації та прогнозування дискретних категорійних даних.

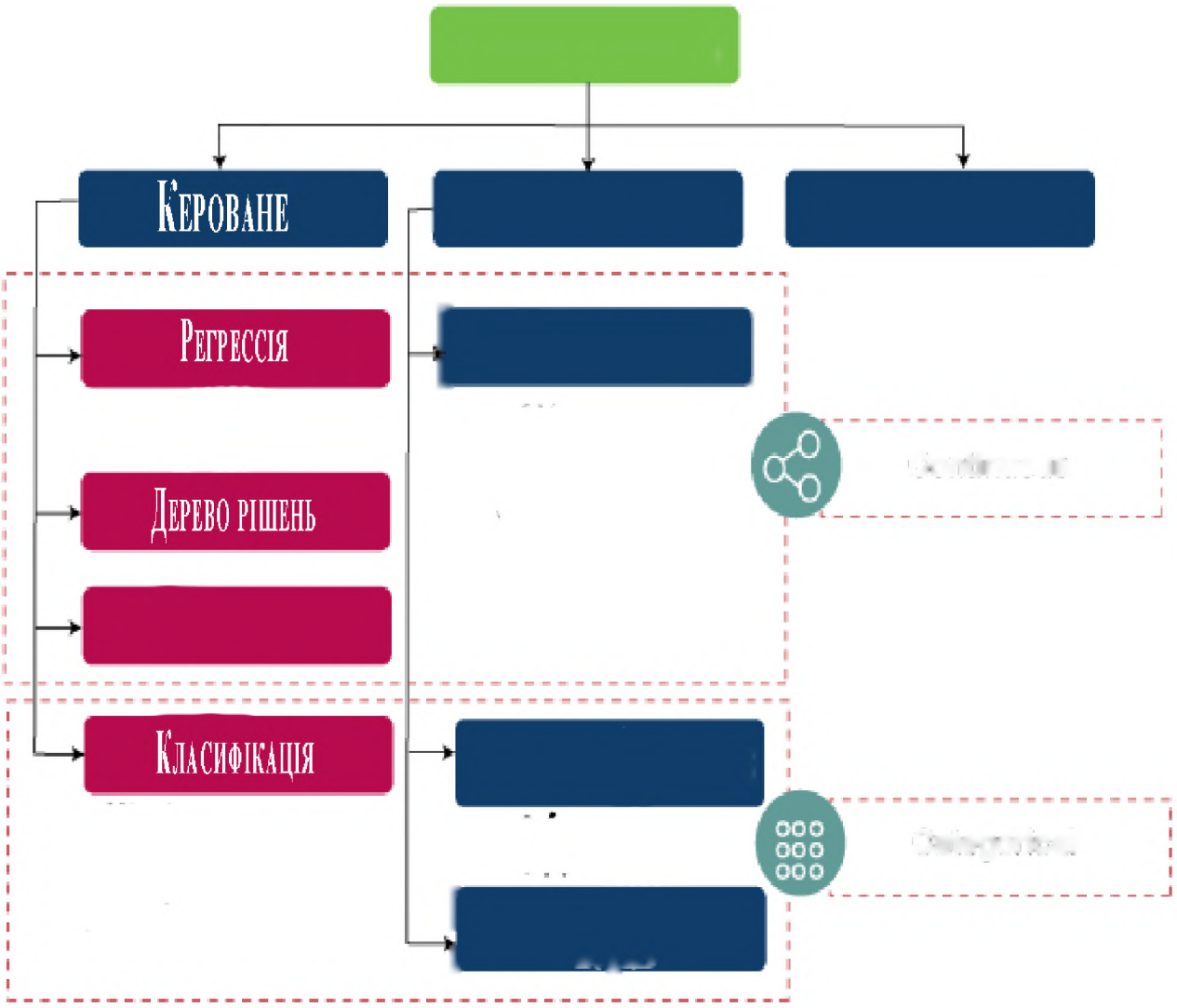

Рисунок 2.8 – Види та методи машинного навчання

Дерево рішень – це метод навчання з нагляду, який в основному застосовується для класифікації, а також може застосовуватися для регресії. Воно працює з категоріальними і числовими змінними, будуючи структуру з вузлів і гілок, починаючи з кореневого вузла, який ділиться на гілки і закінчується листяними вузлами. Внутрішні вузли представляють характеристики даних, гілки відображають правила рішень, а листяні вузли - результати аналізу.

Найголовніший етап – збір набору даних. Якщо немає доступу до експерта для визначення найбільш інформативних полів, можна скористатися методом «грубої сили», що передбачає вимірювання всіх доступних параметрів у сподіванні визначити правильні ознаки. Однак такий підхід часто містить шум та пропуски в даних, що потребують значної попередньої обробки перед індукцією.

2.4. Розробка системи виявлення аномалій в банківських транзакціях

Потрібно побудувати систему виявлення аномалій для виявлення транзакцій, які в певному сенсі відрізняються від звичайних, автентичних транзакцій. Ці спостереження позначаються як потенційно шахрайські і підлягають подальшій перевірці.

Оскільки дані про шахрайські транзакції часто є конфіденційними і не доступними для публічного використання. Тому було обрано файли даних creditcard.csv, доступні на GitHub, через їх потенціал для вивчення шахрайських дій у сфері кредитних карт. Цей набір даних містить інформацію про транзакції, здійснені європейськими власниками кредитних карт у вересні 2013 року. Додатково варто відзначити, що те, що цей набір даних стосується транзакцій 2013 року, не робить його дуже застарілим для багатьох аспектів досліджень. Виявлення шахрайських транзакцій має актуальність протягом тривалого часу, оскільки основні методи шахрайства з кредитними картами часто зберігають свою актуальність.

Також цей набір даних містить як легітимні, так і шахрайські транзакції, що робить його цінним ресурсом для аналізу моделями машинного навчання. Особливість полягає в тому, що шахрайські дії становлять лише дуже малу

частину загального обсягу, що відображає реальний характер таких даних в реальних умовах.

Набір даних є вельми несбалансованим, оскільки містить лише 492 випадки шахрайських транзакцій серед 284,807 усіх транзакцій. Шахрайські транзакції становлять лише 0.172% від усіх транзакцій, що ускладнює завдання класифікації через недостатню кількість позитивних прикладів – й дасть змогу сбалансувати їх при розробці.

Датасет містить лише числові вхідні змінні, які є результатом трансформації методом головних компонент (PCA). Оригінальні функції звісно не надаються через питання конфіденційності. Однак, доступні змінні, які не були піддані PCA, включають 'Time' (час транзакції) і 'Amount' (сума транзакції), що можуть бути корисними для розробки моделей, наприклад, для вивчення залежності від витрат.

Аналіз цих даних може допомогти виявити візуальні та структурні особливості шахрайських транзакцій, що сприяє розробці ефективних моделей виявлення шахрайства.

При виборі середовища програмування було обрано Python оскільки він надає багато можливостей для розробки моделей машинного навчання та їх використання. Важливим аспектом Python є наявність різноманітних бібліотек, таких як NumPy, Pandas, які спеціалізуються на числових обчисленнях, аналізі даних і глибокому навчанні. Це робить Python популярним інструментом у сфері машинного навчання.

Плюсом Python є його активна спільнота, яка постійно розширює функціональність мови через створення нових бібліотек і фреймворків. Це забезпечує величезний вибір інструментів для різних завдань, від веброзробки до наукових обчислень.

Завдяки активному розвитку і підтримці спільноти, Python постійно оновлюється і адаптується до нових вимог і технологічних тенденцій. Це забезпечує його актуальність і конкурентоспроможність в галузі розробки

програмного забезпечення і сприяє розвитку новітніх інновацій для вирішення складних завдань у сучасному технологічному світі.

Вибір бібліотек є одним з найважливіших початків програмування – бо від них залежить сама розробка методу. При виборі бібліотек для розробки програми, яка включає обробку даних, візуалізацію та машинне навчання, важливо враховувати кілька ключових аспектів. Зазначені бібліотеки, такі як NumPy і Pandas, є основними інструментами для ефективної роботи з числовими даними та табличними структурами. Вони надають широкі можливості для обробки і аналізу даних, що дозволяє розробникам зосередитися на аналізі замість розробки базових алгоритмів. Вони були обрані за їхню потужність і ефективність у роботі з великими обсягами даних.

```
import time, psutil, os, gc
# Додання бібліотеки математичних функцій
import math
# Додання головний бібліотек для маніпуляцій над данимим
import numpy as np
import pandas as pd
# Побудова графіків та візуалізація
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import seaborn as sns
sns.set_theme()
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
# Розбиття набору даних на тренувальні тести
from sklearn.model_selection import train_test_split
```

Рисунок 2.9 – Ініціалізація та імпорт бібліотек

Для візуалізації даних бібліотеки, такі як Matplotlib, Seaborn і Plotly, надають потужні засоби для створення різноманітних графіків і діаграм, що допомагають у зрозумінні структури та зв'язків в даних. Кожна з цих бібліотек має свої унікальні особливості: Matplotlib — для базових і змінених графіків,

Seaborn — для статистичної візуалізації, Plotly — для інтерактивних інтерфейсів. Вони були обрані за їхню здатність створювати якісні і професійні візуалізації, що є важливим для аналізу та ефективного представлення результатів програми.

Після ініціалізації бібліотек важливим є перевірка того як виглядають дані що в нас є – для наглядності я вирішив в коді перевірити перші 20 сторінок даних (Рис. 2.5) за допомогою коду. Та там є такі параметри:

- time: Час (у секундах), що минув між транзакцією та найпершою транзакцією;
- параметри від V1 до V28: Отримано в результаті перетворення аналізу головних компонент на вихідні ознаки, які не є доступними з міркувань конфіденційності;
- amount: Сума транзакції;
- class: статус автентичної (або шахрайської) транзакції приймається як 0 (відповідно 1).

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0
5	2.0	-0.425966	0.960523	1.141109	-0.168252	0.420987	-0.029728	0.476201	0.260314	-0.568671	...	-0.208254	-0.559825	-0.026398	-0.371427	-0.232794	0.105915	0.253844	0.081080	3.67	0
6	4.0	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.005159	0.081213	0.464960	...	-0.167716	-0.270710	-0.154104	-0.780055	0.750137	-0.257237	0.034507	0.005168	4.99	0
7	7.0	-0.644269	1.417964	1.074380	-0.492199	0.948934	0.428118	1.120631	-3.807864	0.615375	...	1.943465	-1.015455	0.057504	-0.649709	-0.415267	-0.051634	-1.206921	-1.085339	40.80	0
8	7.0	-0.894286	0.286157	-0.113192	-0.271526	2.669599	3.721818	0.370145	0.851084	-0.392048	...	0.073425	-0.268092	-0.204233	1.011592	0.373205	-0.384157	0.011747	0.142404	93.20	0
9	9.0	-0.338262	1.119593	1.044367	-0.222187	0.499361	-0.246761	0.651583	0.069539	-0.736727	...	-0.246914	-0.633753	-0.120794	-0.385050	-0.069733	0.094199	0.246219	0.083076	3.68	0
10	10.0	1.449044	-1.176339	0.913860	-1.375667	-1.971383	-0.629152	-1.423236	0.048456	-1.720408	...	-0.009302	0.313894	0.027740	0.500512	0.251367	-0.129478	0.042850	0.016253	7.80	0
11	10.0	0.384978	0.616109	-0.874300	-0.094019	2.924584	3.317027	0.470455	0.538247	-0.558895	...	0.049924	0.238422	0.009130	0.996710	-0.767315	-0.492208	0.042472	-0.054337	9.99	0
12	10.0	1.249999	-1.221637	0.383930	-1.234899	-1.485419	-0.753230	-0.689405	-0.227487	-2.094011	...	-0.231809	-0.483285	0.084668	0.392831	0.161135	-0.354990	0.026416	0.042422	121.50	0
13	11.0	1.069374	0.287722	0.828613	2.712520	-0.178398	0.337544	-0.096717	0.115982	-0.221083	...	-0.036876	0.074412	-0.071407	0.104744	0.548265	0.104094	0.021491	0.021293	27.50	0
14	12.0	-2.791855	-0.327771	1.641750	1.767473	-0.136588	0.807596	-0.422911	-1.907107	0.755713	...	1.151663	0.222182	1.020586	0.028317	-0.232746	-0.235557	-0.164778	-0.030154	58.80	0
15	12.0	-0.752417	0.345485	2.057323	-1.468643	-1.158394	-0.077850	-0.608581	0.003603	-0.436167	...	0.499625	1.353650	-0.256573	-0.065084	-0.039124	-0.087086	-0.180998	0.129394	15.99	0
16	12.0	1.103215	-0.040296	1.267332	1.289091	-0.735997	0.288069	-0.586057	0.189380	0.782333	...	-0.024612	0.196002	0.013802	0.103758	0.364298	-0.382261	0.092809	0.037051	12.99	0
17	13.0	-0.436905	0.918966	0.924591	-0.727219	0.915679	-0.127867	0.707642	0.087962	-0.665271	...	-0.194796	-0.672638	-0.156858	-0.888386	-0.342413	-0.049027	0.079692	0.131024	0.89	0
18	14.0	-5.401258	-5.450148	1.186305	1.736239	3.049106	-1.763406	-1.559738	0.160842	1.233090	...	-0.503600	0.984460	2.458589	0.042119	-0.481631	-0.621272	0.392053	0.949594	46.80	0
19	15.0	1.492936	-1.029346	0.454795	-1.438026	-1.555434	-0.720961	-1.080664	-0.053127	-1.978682	...	-0.177650	-0.175074	0.040002	0.295814	0.332931	-0.220385	0.022298	0.007602	5.00	0

Рисунок 2.10 – Таблиця з першими 20 стовпцями даних

Метою програми є виявлення аномалій в операціях з кредитними картками. Точніше, маючи дані про час, суму та параметри V1-V28, потрібно побудувати розподіл ймовірностей на основі автентичних транзакцій, а потім використати

його для правильної ідентифікації нової транзакції як автентичної чи шахрайської. Важливо зазначити, що невідомі параметри не беруть участі у формуванні ймовірнісного розподілу.

Передбачення в системі можуть бути 4 типів:

- істинно позитивний: Модель класифікації правильно прогнозує, що результат буде позитивним;
- істинно негативний: Модель класифікації правильно передбачає, що результат буде від'ємним;
- хибно позитивний: Модель класифікації помилково передбачає, що результат буде позитивним;
- хибно негативний: Модель класифікації помилково прогнозує від'ємний результат.

А самі параметри оцінки такі:

- accuracy (точність): Це відношення кількості правильних передбачень до загальної кількості передбачень. Ця метрика визначає, наскільки правильно модель передбачає класифікацію. Вона враховує як істинно позитивні, так і істинно негативні передбачення. Висока точність означає, що модель робить менше помилок;
- reliable (надійність): Відношення кількості істинно позитивних передбачень до загальної кількості позитивних передбачень. Ця метрика вказує, як часто модель правильно передбачає позитивний клас. Вона корисна, коли важливо уникнути хибно позитивних передбачень);
- recall (повнота): Ця метрика показує, як часто модель виявляє всі фактичні позитивні випадки. Вона важлива, коли важливо не пропустити жодного позитивного випадку. Та визначається відношенням кількості істинно позитивних передбачень до загальної кількості фактичних позитивних випадків;
- бали F1: параметр що показує дані середнього між надійністю та повнотою. Вона допомагає збалансувати обидві метрики;
- бали F_b – це метрика, яка використовується для оцінки ефективності бінарних моделей класифікації. Ця метрика дозволяє гнучко налаштовувати

оцінку моделі в залежності від конкретних вимог та контексту задачі. Якщо b>1, то більший акцент робиться на recall. Якщо b<1, то більший акцент робиться на надійність. При b=1 буде отримано звичайний F_1 Score:

$$F_b = \frac{(1 + b^2) * reliable * recall}{b^2 * reliable * recall}, \quad (2.1)$$

Використовується як міра якості бінарних (двокласових) класифікацій

$$F_b = \frac{(\text{ІП} * \text{ІН}) - (\text{ХП} * \text{ХН})}{\sqrt{(\text{ІП} + \text{ХП}) * (\text{ІП} + \text{ХН}) * (\text{ІН} + \text{ХП}) * (\text{ІН} + \text{ХН})}}, \quad (2.2)$$

де ІП – істинно позитивне; ІН – істинно негативне; ХП – хибно позитивне; ХН – хибно негативне.

На відміну від попередніх метрик, ККМ змінюється від -1 (найгірший сценарій) до 1 (найкращий сценарій: ідеальне передбачення).

Серед розглянутих метрик для оцінки моделей, особливо для незбалансованих наборів даних, добре підходять ККМ і F_1, тоді як reliable і recall також дають корисну інформацію. Тому не буде надано великого значення метриці точності в цьому проєкті, оскільки вона призводить до хибних висновків, коли класи не збалансовані.

В даній проблемі хибно негативний результат (шахрайська транзакція класифікується як автентична) є більш небезпечним, ніж хибно позитивний (автентична транзакція класифікується як шахрайська). У першому випадку шахрай може завдати ще більших фінансових збитків. У другому випадку банк може здійснити перехресну перевірку автентичності транзакції від користувача картки після вжиття необхідних заходів для захисту картки.

Ефективність моделі в значній мірі залежить від її здатності точно розрізняти шахрайські та автентичні транзакції, мінімізуючи при цьому кількість хибно позитивних та хибно негативних результатів. Важливо розуміти, що

ефективна модель повинна мінімізувати ризик для користувача та забезпечити безпеку фінансових операцій. Враховуючи цей факт, використано Бали F_2 для налаштування порогового параметру та відбору ознак у цій роботі:

$$F_2 = \frac{5 * \text{ІП}}{5 * \text{ІП} + 4 * \text{ХН} + \text{ІП}}, \tag{2.3}$$

де ІП – істинно позитивне;

ХН – хибно негативне.

Всі згадані метрики звітуються як для валідаційного, так і для тестового набору. Це дозволяє оцінити продуктивність моделі на різних етапах розробки.

Тому ці дані розподілено на їх класи – щахрайські та звичайні й ділимо їх на функції, цілі та додатково ділимо їх на:

- тренування;
- валідація;
- тестування.

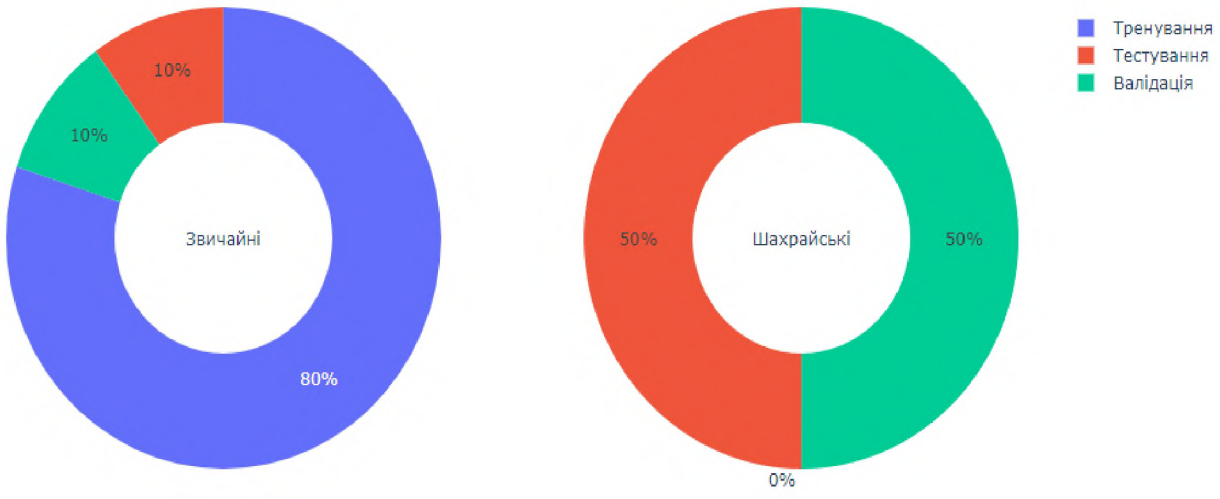

Рисунок 2.11 – Діаграми розподілу транзакцій між тренувальними, валідаційними та тестовими наборами

Після цього, розподіляються дані за їх класами — шахрайські та звичайні (тобто не шахрайські) — і поділено їх на тренувальні, валідаційні та тестові

набори. Далі виконується побудова діаграми (див. Рисунок 2.6), яка візуально демонструє, як ці дані розподіляються між вказаними наборами.

```
# Розподіл даних за цільовими класами
data_0, data_1 = data[data['Class'] == 0], data[data['Class'] == 1]

# Розподіл функцій та цілей
X_0, y_0 = data_0.drop('Class', axis = 1), data_0['Class']
X_1, y_1 = data_1.drop('Class', axis = 1), data_1['Class']

# Розбиття звичайного класу та побудова навчальної вибірки
X_train, X_test, y_train, y_test = train_test_split(X_0, y_0, test_size = 0.2, random_state = 40)
X_val, X_test, y_val, y_test = train_test_split(X_test, y_test, test_size = 0.5, random_state = 40)
data_val_1, data_test_1 = pd.concat([X_val, y_val], axis = 1), pd.concat([X_test, y_test], axis = 1)

# Поділ класу шахрайства
X_val, X_test, y_val, y_test = train_test_split(X_1, y_1, test_size = 0.5, random_state = 40)
data_val_2, data_test_2 = pd.concat([X_val, y_val], axis = 1), pd.concat([X_test, y_test], axis = 1)

# Об'єднання даних для побудови вибірки для валідації та тестової вибірки
data_val, data_test = pd.concat([data_val_1, data_val_2], axis = 0), pd.concat([data_test_1, data_test_2], axis = 0)
X_val, y_val = data_val.drop('Class', axis = 1), data_val['Class']
X_test, y_test = data_test.drop('Class', axis = 1), data_test['Class']

# Розподіл звичайних та шахрайських транзакцій між тренувальними, валідаційним та тестовим набором
labels = ['Тренування', 'Валідація', 'Тестування']
values_0 = [len(y_train[y_train == 0]), len(y_val[y_val == 0]), len(y_test[y_test == 0])]
values_1 = [len(y_train[y_train == 1]), len(y_val[y_val == 1]), len(y_test[y_test == 1])]
fig = make_subplots(rows = 1, cols = 2, specs = [[{'type': 'domain'}, {'type': 'domain'}]])
fig.add_trace(go.Pie(values = values_0, labels = labels, hole = 0.5, textinfo = 'percent', title = "Звичайні"),
              row = 1, col = 1)
fig.add_trace(go.Pie(values = values_1, labels = labels, hole = 0.5, textinfo = 'percent', title = "Шахрайські"),
              row = 1, col = 2)
text_title = "Розподіл звичайних та шахрайських транзакцій між тренувальними, валідаційним та тестовим набором"
fig.update_layout(height = 500, width = 1000, showlegend = True, title = dict(text = text_title, x = 0.5, y = 0.95))
fig.show()
```

Рисунок 2.12 – Лістинг частини коду про розподіл транзакцій

Після цього було розподілено дані про транзакції по часу. Цей процес включає побудову гістограми, де кожний стовпчик представляє певний інтервал часу – години дня. Такий аналіз дозволяє виявити залежності між часом та частотою транзакцій, ідентифікувати піки активності, а також виявляти можливі аномальні патерни, що відбуваються в певні періоди.

Це дозволяє краще розуміти динаміку операцій та приймати обґрунтовані рішення щодо захисту фінансових транзакцій. Дані аналізу часових вибірок можуть бути корисними для вдосконалення стратегій моніторингу та запобігання шахрайству.

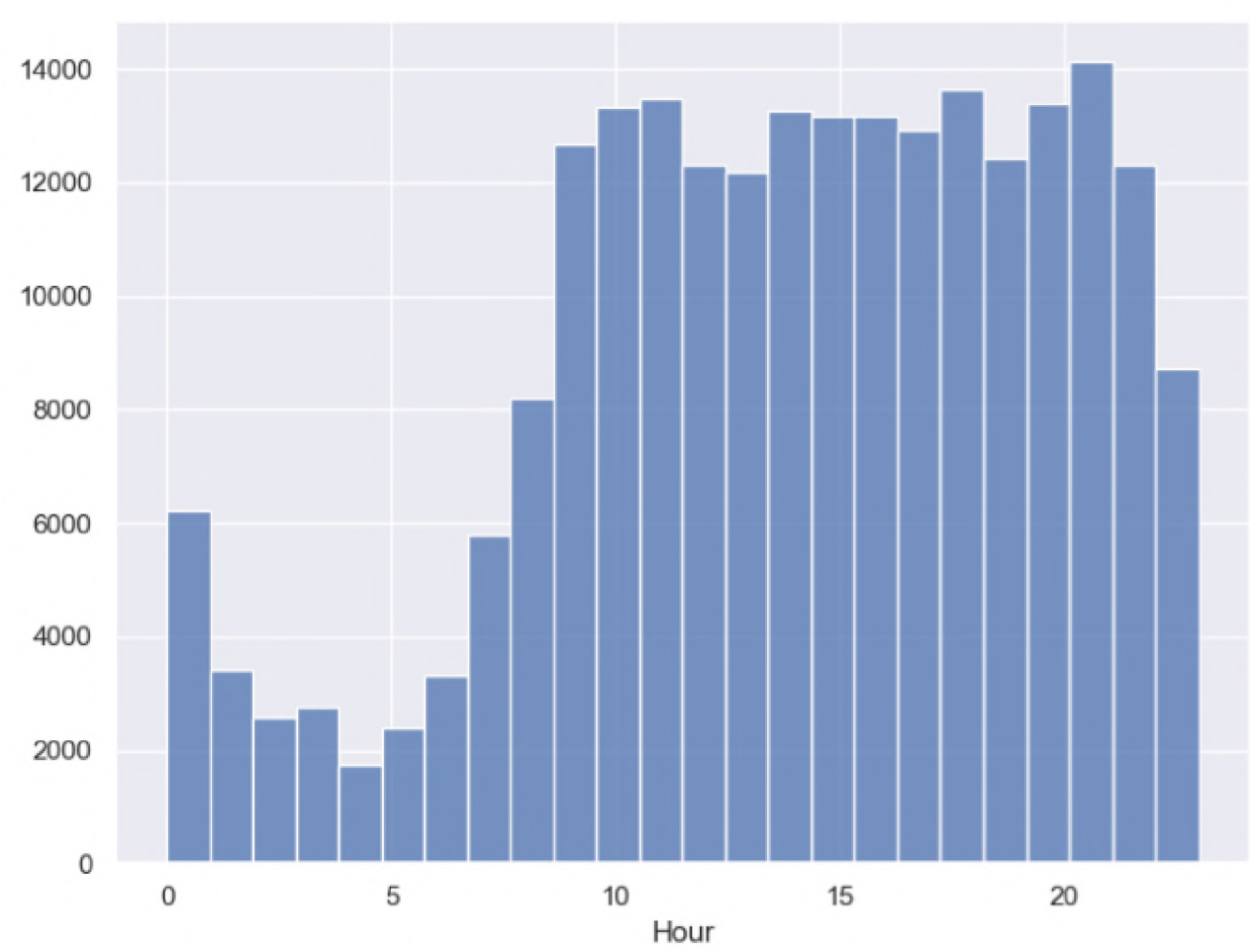

Рисунок 2.13 – Гістограма розподілу транзакцій по годинам та її кількості

Вивчення динаміки транзакцій не обмежується лише часовими аспектами. Після аналізу часових вибірок, розподілено дані про транзакції по їхнім обсягам. Цей процес включає побудову гістограми, де кожен стовпчик відображає розподіл сум транзакцій, збережених після застосування логарифмічного перетворення. Такий підхід дозволяє краще розуміти, як обсяги транзакцій розподілені впродовж доби.

Логарифмічне перетворення використовується для згладжування розподілу даних, особливо коли значення x містять нулі або дуже малі значення. Додавши позитивну константу 0.001 до x перед застосуванням логарифму, було уникнено виникнення неозначеності при спробі обчислити log(0), що є нескінченністю. Це перетворення сприяє стабілізації даних та поліпшенню їхнього розподілу для подальшого аналізу і моделювання.

Аналіз обсягів транзакцій в різний час доби допомагає ідентифікувати найактивніші періоди та виявляти можливі аномальні патерни, що можуть свідчити про шахрайські дії. Це не лише дозволяє покращити моніторинг фінансових операцій, але і забезпечує ефективнішу захист від потенційних загроз.

Отримані під час аналізу дані створюють основу для розроблення стратегій захисту та оптимізації фінансових процесів. З їх допомогою можна приймати обґрунтовані рішення щодо управління ризиками та вдосконалення загальної ефективності операційної діяльності.

Після обробки часових і обсягових аспектів транзакцій, наступним логічним кроком є видалення зайвих або непотрібних стовпців з набору даних. Серед таких стовпців можуть бути інформація про час (години, день, хвилини, секунди) та сума транзакцій, які перед цим піддалися логарифмічному перетворенню Ці кроки важливі для ефективної підготовки даних до подальшого аналізу та моделювання, оскільки спрямовані на збереження лише суттєвих параметрів, які суттєво впливають на остаточні результати аналізу.

У розглянутій задачі справа мається з набором даних, що містить 30 ознак. Основною метою є відбір ознак, які допоможуть чітко відрізнити автентичні транзакції від шахрайських. Для досягнення цієї мети було порівняно розподіл кожної ознаки в обох цільових класах. Якщо ознака має подібний розподіл для автентичних і шахрайських транзакцій, вона може бути менш інформативною для процесу класифікації.

У випадку, коли розподіл ознаки суттєво відрізняється між цільовими класами, ця ознака виявляється важливою для правильного класифікаційного процесу. Для визначення таких ключових ознак було збудовано графіки їх розподілу і проаналізовано, як вони відрізняються між класами.

Аналіз розподілу ознак дозволяє ефективно відбирати параметри, які максимально відповідають вимогам задачі. Цей процес важливий не лише для підвищення точності класифікації, але й для забезпечення глибокого розуміння взаємозв'язків між параметрами та цільовими змінними.

Зробивши відбір суттєвих ознак для класифікації транзакцій, наступним кроком є порівняння їхніх розподілів для різних цільових класів. Це допомагає визначити, які параметри мають значущий вплив на відміну між автентичними та шахрайськими транзакціями. В коді використовується графічний метод для порівняння щільності розподілу кожної ознаки між двома класами. Це дозволяє виявити ознаки, що мають різний характер розподілу і, отже, можуть бути корисними для побудови ефективної моделі класифікації. Детальний аналіз розподілів допомагає відібрати ті ознаки, які найбільше впливають на точність класифікації та забезпечують найкращі результати моделювання.

Порівняння розподілів ознак для різних цільових класів проводиться за допомогою ядерних оцінок щільності. Було побудовано матрицю графіків, де кожен стовпець представляє розподіл однієї ознаки для обох класів. Такий аналіз дозволяє візуально порівняти форму та положення розподілів, виявити схожості або відмінності між ними. Чітке розрізнення розподілів може свідчити про важливість ознаки для вирішення задачі класифікації. Цей підхід допомагає здійснити інформативний відбір ознак перед побудовою моделі машинного навчання.

```
# Порівняння розподілів ознак для різних цільових класів
data_val = pd.concat([X_val, y_val], axis = 1)
data_val_0, data_val_1 = data_val[data_val['Class'] == 0], data_val[data_val['Class'] == 1]
cols, ncols = list(X_val.columns), 3
nrows = math.ceil(len(cols) / ncols)
fig, ax = plt.subplots(nrows, ncols, figsize = (4.5 * ncols, 4 * nrows))
for i in range(len(cols)):
    sns.kdeplot(data_val_0[cols[i]], ax = ax[i // ncols, i % ncols])
    sns.kdeplot(data_val_1[cols[i]], ax = ax[i // ncols, i % ncols])
    if i % ncols != 0:
        ax[i // ncols, i % ncols].set_ylabel(" ")
plt.tight_layout()
plt.show()
```

Рисунок 2.14 – Частина коду для порівняння розподілів ознак для різних класів

Після побудови графіків можна провести детальний аналіз та відбір тих ознак, які найбільше сприяють відмінності між класами. Також цей процес

важливий для підготовки даних перед використанням їх у моделі машинного навчання.

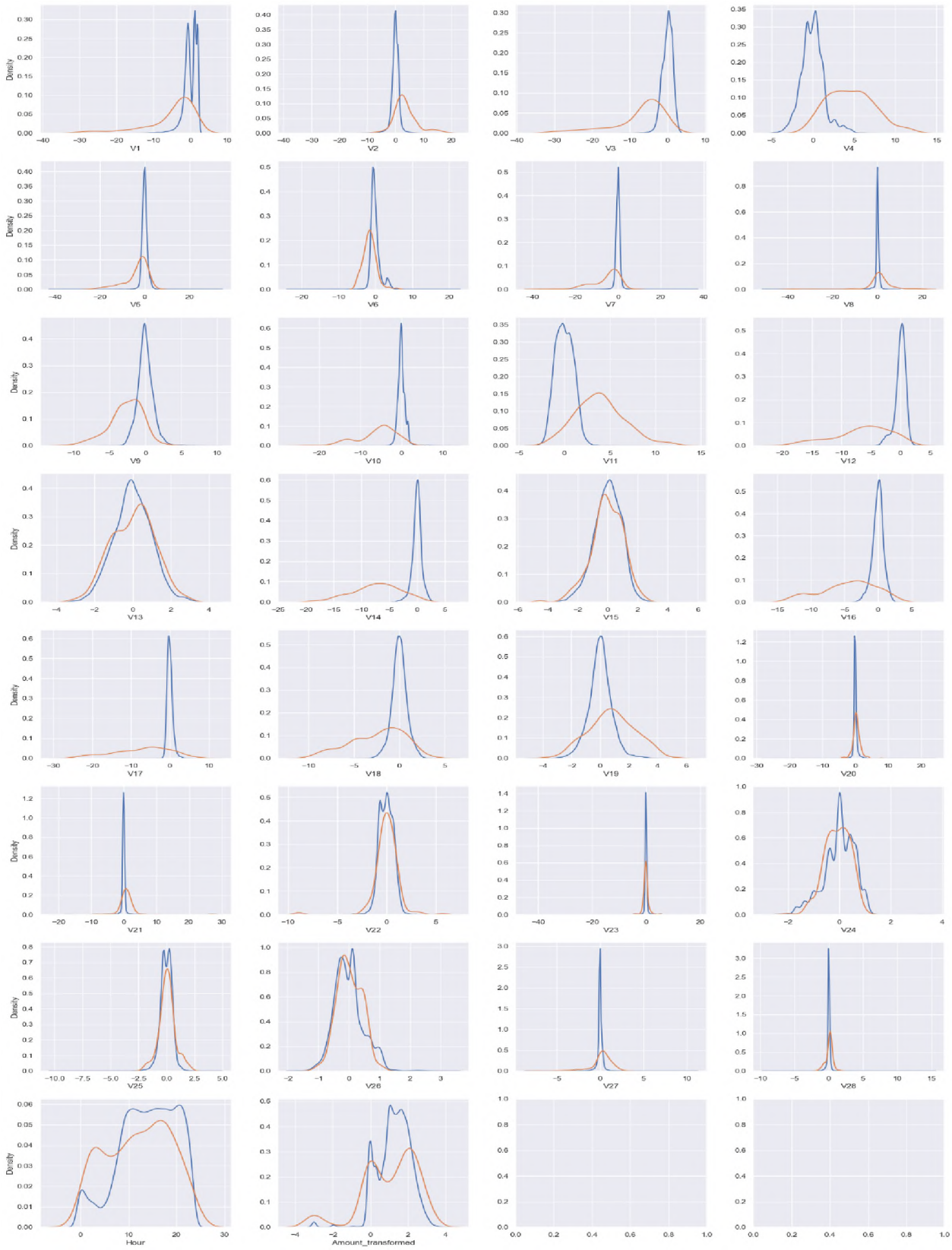

Рисунок 2.15 – Графіки для порівняння розподілів ознак для різних класів

З усіх ознак отриманих після цього було обрано 9 ознак, які суттєво відрізняються за розподілом для різних цільових класів: V4, V11, V12, V14, V16, V17, V18, V19 та Hour

Отримана інформація допомагає не лише покращити точність нашої моделі, але і забезпечити глибше розуміння впливу окремих ознак на результати класифікації.

Функція щільності ймовірності одновимірного нормального розподілу з середнім значенням μ та середньоквадратичним відхиленням σ згідно [18] має вигляд :

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right), \tag{2.4}$$

Для $x \in R$, де $\mu \in R$ та $\sigma > 0$.

Наступна функція що потрібна – це добуток нормальної щільності розподілу ймовірностей – тобто функція обчислює добуток таких одновимірних нормальних щільностей. Це можна розглядати як об'єднаний pdf ряду змінних ознак, кожна з яких має одновимірний нормальний розподіл і є статистично незалежною від інших ознак.

$$g(x;\mu,\sigma) = \prod_{i=1}^{n} f(x_i;\mu_i,\sigma_i), \tag{2.5}$$

Для $x=(x_1,x_2,\ldots,x_n) \in R^n$, де $\mu=(\mu_1,\mu_2,\ldots,\mu_n) \in R$ та $\sigma > 0$.

В наступному кроці обчислено вектор середніх значень і вектор стандартних відхилень для ознак у навчальній вибірці. Ці статистичні параметри визначають спільну функцію щільності ймовірності ознак, яка використовується для виявлення аномальних спостережень. Чим вищі стандартні відхилення, тим більша ймовірність, що спостереження відхиляється від звичайного розподілу даних.

Далі йде прогнозування аномалій на основі заданого порогу для щільності ймовірності. Цей поріг встановлюється з урахуванням статистичних властивостей розподілу даних і може бути налаштований залежно від специфіки проблеми. Виявлення аномальних спостережень є важливою задачею в аналізі даних, особливо у фінансових або медичних доменах, де навіть невеликі аномалії можуть мати серйозні наслідки.

```python
#  Нормальна щільність розподілу ймовірностей
def normal_density(x, mu, sigma):
    """
    Обчислює одновимірну нормальну щільність розподілу ймовірностей із середнім значенням u(мю), стандартним відхиленням сігма
    Аргументи:
    x: вхідне спостереження
    mu: середнє значення
    sigma: стандартне відхилення (> 0)
    Повертає
    f: значення одновимірної нормальної шільності розподілу ймовірностей
    """
    assert sigma > 0, "Середньоквадратичне відхилення має бути позитивним"
    f = (1 / (sigma * np.sqrt(2 * np.pi))) * np.exp(- (1 / 2) * ((x - mu) / sigma)**2)
    return f

# Добуток нормальної щільності розподілу ймовірностей
def normal_product(x_vec, mu_vec, sigma_vec):
    """
    Обчислює добуток одновимірних нормальних густин
    Аргументи:
      x_vec (array_like, shape (n,)) : вектор вхідних спостережень
      mu_vec (array_like, shape (n,)) : вектор середніх
      sigma_vec (array_like, shape (n,)): вектор стандартних відхилень (> 0)
    Повертає
      f: добуток одновимірних нормальних щільностей
    """
    assert min(sigma_vec) > 0, "Середньоквадратичне відхилення має бути позитивним"
    assert len(mu_vec) == len(x_vec), "Довжина середнього вектора не відповідає довжині вхідного вектора"
    assert len(sigma_vec) == len(x_vec), "Довжина вектора середньоквадратичного відхилення не відповідає довжині вхідного вектора"
    f = 1
    for i in range(len(x_vec)):
        f = f * normal_density(x_vec[i], mu_vec[i], sigma_vec[i])
    return f

# Підбір моделі
mu_train, sigma_train = X_train_fs.mean().values, X_train_fs.std().values

# Функція для прогнозування аномалії на основі порогового значення щільності ймовірності
def model_normal(X, epsilon):
    """
    Модель виявлення аномалій
    Аргументи:
      X (DataFrame, shape (m, n)): DataFrame ознак
      epsilon : порогове значення щільності (> 0)
    Повертає
      y (array_like, shape (m,)): передбачені мітки класів
    """
    y = []
    for i in X.index:
        prob_density = normal_product(X.loc[i].tolist(), mu_train, sigma_train)
        y.append((prob_density < epsilon).astype(int))
    return y
```

Рисунок 2.16 – Код з функціями для впровадження виявлення аномалій

Далі потрібно провести налаштування порогу на валідаційному наборі. Для цього спочатку збудовано деякі функції для обчислення і відображення матриці непередбачуваності, а також для обчислення F2 балів, враховуючи істинні мітки та передбачувані мітки цілі.

```python
# Функція для обчислення матриці невідповідностей
def conf_mat(y_test, y_pred):
    """
    Обчислює матрицю плутанини
    Аргументи:
      y_test: істинні двійкові (0 або 1) мітки
      y_pred: передбачені двійкові (0 або 1) мітки
    Повертається:
      confusion_mat: двовимірний масив, що представляє матрицю плутанини 2x2
    """
    y_test, y_pred = list(y_test), list(y_pred)
    count, labels, confusion_mat = len(y_test), [0, 1], np.zeros(shape = (2, 2), dtype = int)
    for i in range(2):
        for j in range(2):
            confusion_mat[i][j] = len([k for k in range(count) if y_test[k] == labels[i] and y_pred[k] == labels[j]])
    return confusion_mat

# Функція для відображення матриці невідповідностей
def conf_mat_heatmap(y_test, y_pred):
    confusion_mat = conf_mat(y_test, y_pred)
    labels, confusion_mat_df = [0, 1], pd.DataFrame(confusion_mat, range(2), range(2))
    plt.figure(figsize = (6, 4.75))
    sns.heatmap(confusion_mat_df, annot = True, annot_kws = {"size": 16}, fmt = 'd')
    plt.xticks([0.5, 1.5], labels, rotation = 'horizontal')
    plt.yticks([0.5, 1.5], labels, rotation = 'horizontal')
    plt.xlabel("Прогнозована мітка", fontsize = 14)
    plt.ylabel("Справжня мітка", fontsize = 14)
    plt.title("Матриця невідповідностей", fontsize = 14)
    plt.grid(False)
    plt.show()

# Функція для обчислення та повернення F2-балів
def f2_score(y_test, y_pred):
    """
    Обчислює F2-бали за істинними та передбаченими бінарними (0 або 1) мітками
    Аргументи:
      y_test (array_like): істинні двійкові (0 або 1) мітки
      y_pred (array_like): передбачені двійкові (0 або 1) мітки
    Повертається:
      f2: F2-бал, отриманий з y_test та y_pred
    """
    confusion_mat = conf_mat(y_test, y_pred)
    tn, fp, fn, tp = confusion_mat[0, 0], confusion_mat[0, 1], confusion_mat[1, 0], confusion_mat[1, 1]
    f2 = (5 * tp) / ((5 * tp) + (4 * fn) + fp)
    return f2
```

Рисунок 2.17 – Код з функціями для матриці невідповідностей та F_2-балів

На основі послідовності порогових значень alpha = 0.001, 0.002, ..., 0.005, призначених для щільності розподілу однієї ознаки, визначено відповідний поріг для спільної щільності ймовірності, що дорівнює alpha у степені n, де n – кількість ознак, використаних у моделі.

Для кожного порогу обчислено F2-метрику для оцінки ефективності моделі на валідаційному наборі даних. Графіки залежності F2 від значення alpha демонструють, як змінюється ефективність моделі в залежності від встановленого порогу alpha.

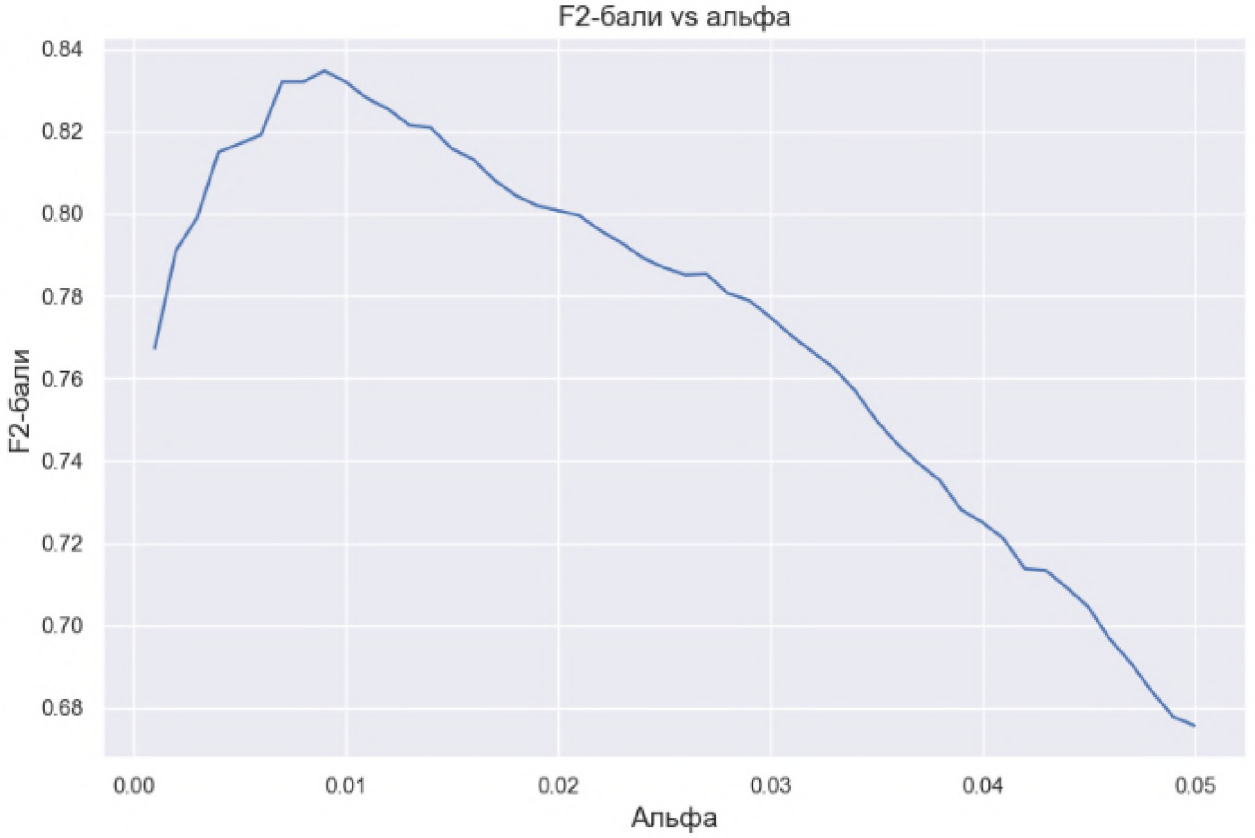

Рисунок 2.18 – Діграма залежності F_2 від значення альфа

На рисунку 2.12 представлена діаграма, що ілюструє залежність F_2 від значення альфа. Цей аналіз дозволяє визначити оптимальне значення альфа, яке максимізує показник F_2 для моделі. За результатами нашої валідації, оптимальне значення alpha становить приблизно 0.009, а відповідний до нього оптимальний бал F2 складає 0.834671.

Додатково, на основі отриманих результатів F2-балів для різних значень alpha, проаналізовано матрицю невідповідності для прогнозів на валідаційному наборі даних. Ця матриця надає деталізацію результатів класифікації, включаючи кількість правильних і неправильних прогнозів для кожного класу та загальну

точність моделі. Вона є важливим інструментом для оцінки того, як добре модель справляється з розпізнаванням шахрайських та автентичних транзакцій, і допомагає визначити додаткові можливості для покращення її результативності.

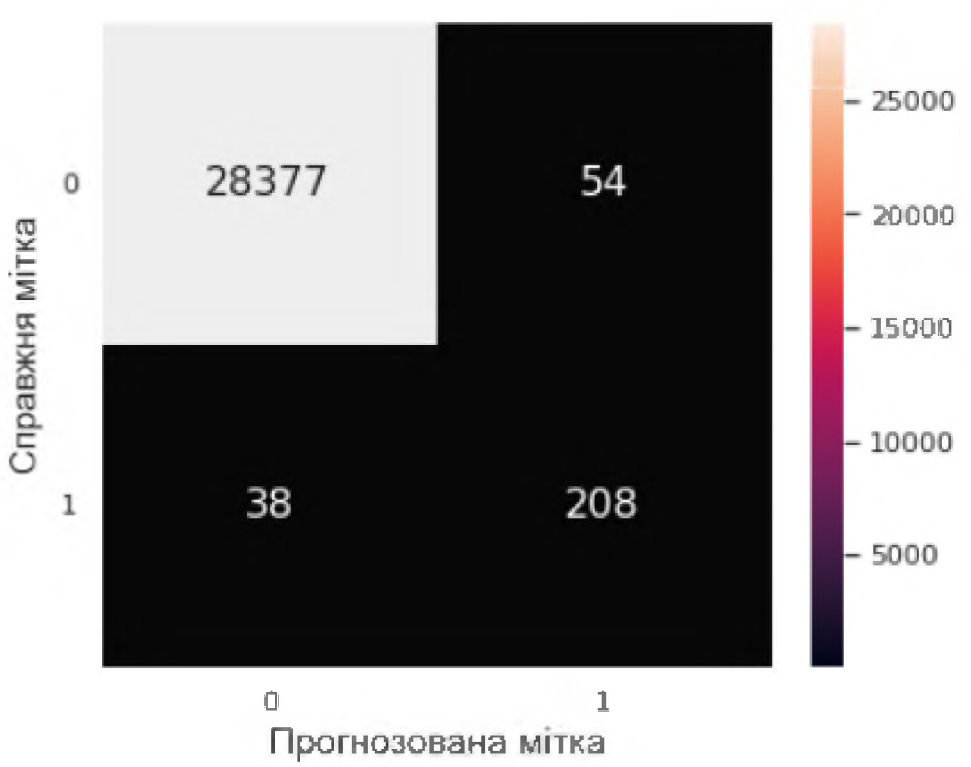

Рисунок 2.19 – Матриця невідповідності для прогнозів на валідаційному наборі

На основі обраного оптимального значення alpha, яке дорівнює приблизно 0.009, спрогнозовано аномалії на тестовому наборі даних. Для оцінки ефективності моделі на цьому наборі використано метрики, такі як точність, надійність, повнота та F-бали. Ці метрики надають повну картину того, як добре модель справляється з виявленням шахрайських транзакцій порівняно з автентичними.

Далі, було сбудовано матрицю невідповідності для прогнозів на тестовому наборі даних. Ця матриця дозволяє візуалізувати результати класифікації, показуючи кількість правильних і неправильних прогнозів для кожного класу (шахрайські та автентичні транзакції). Вона є важливим інструментом для оцінки загальної точності моделі та ідентифікації можливостей для її подальшого вдосконалення.

```python
# Функція для обчислення та друку метрик оцінювання
def evaluation(y_test, y_pred):
    confusion_mat = conf_mat(y_test, y_pred)
    tn, fp, fn, tp = confusion_mat[0, 0], confusion_mat[0, 1], confusion_mat[1, 0], confusion_mat[1, 1]
    print(pd.Series({
        "Accuracy (Точність)": (tp + tn) / (tn + fp + fn + tp),
        "Reliable (Надійність)": tp / (tp + fp),
        "Recall (Повнота)": tp / (tp + fn),
        "F1-бали": (2 * tp) / ((2 * tp) + fn + fp),
        "F2-бали": (5 * tp) / ((5 * tp) + (4 * fn) + fp),
        "Коефіцієнт кореляції Меттьюза (ККМ) ": ((tp * tn) - (fp * fn)) / np.sqrt((tp + fp) * (tp + fn) * (tn + fp) * (tn + fn))
    }).to_string())

# Прогнозування та оцінка на тестовому наборі
y_test_normal = model_normal(X_test_fs, epsilon = alpha_opt**X_test_fs.shape[1])
evaluation(y_test, y_test_normal)

# Матриця невідповідностей для прогнозів на тестовому наборі
conf_mat_heatmap(y_test, y_test_normal)
```

Рисунок 2.20 – Прогнозування, обчислення та друк метрик оцінювання

На рисунку 2.15 представлена матриця невідповідності для прогнозів на тестовому наборі даних, яка ілюструє ефективність моделі в розпізнаванні шахрайських та автентичних транзакцій.

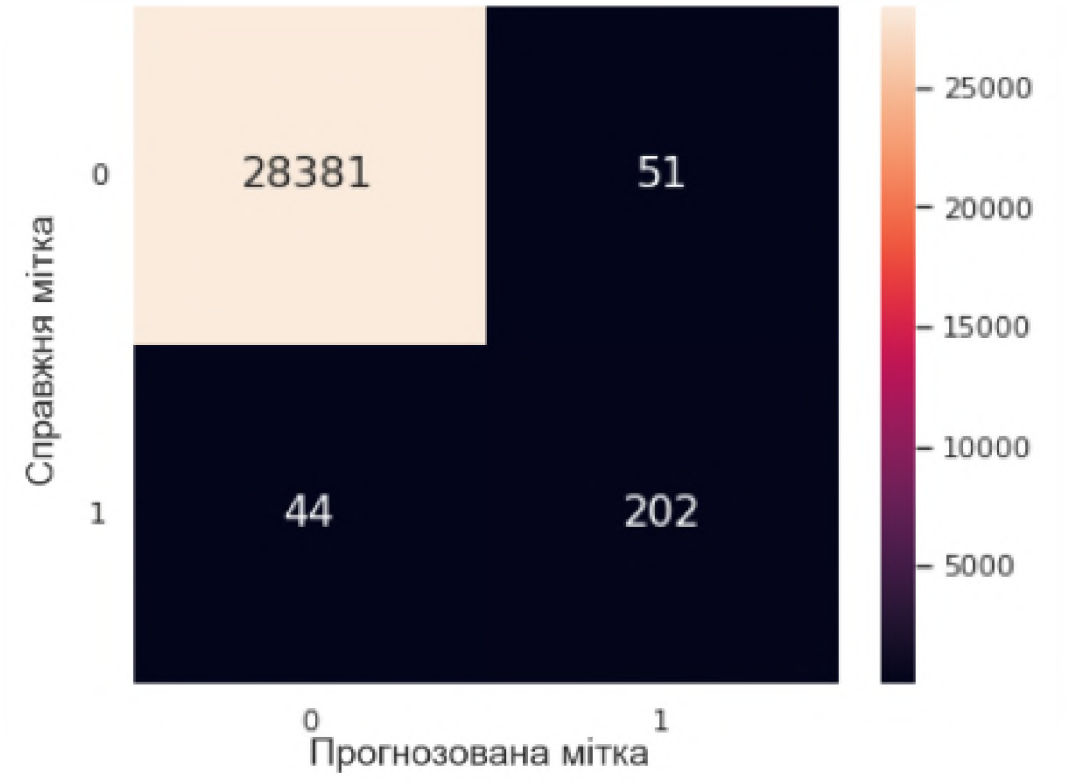

Рисунок 2.21 – Матриця невідповідності для прогнозів на тестовому наборі

У аналізі можна помітити значний дисбаланс у наборі даних, де шахрайські транзакції є рідкісними порівняно з автентичними. Цей дисбаланс створює проблеми, особливо враховуючи еволюційний характер шахрайської діяльності, яка може проявлятися в нових, небачених раніше способах. Щоб вирішити ці складнощі, було розроблено систему виявлення аномалій, призначену для виявлення транзакцій, які відхиляються від типових шаблонів.

Щоб покращити набір даних, проведено вибірку ознак, виділивши ознаку "Час" і застосувавши лог-перетворення (з невеликим зсувом) до сильно викривленої ознаки "Сума", створивши ознаку "Сума_трансформована". З початково спроєктованих 30 ознак було звужено вибірку до 9 ознак, які демонструють помітно відмінний розподіл між цільовими класами: V4, V11, V12, V14, V16, V17, V18, V19 та Hour.

Використовуючи навчальні дані, було підібрано багатовимірний нормальний розподіл після оцінки векторів середніх значень і векторів стандартних відхилень, припускаючи статистичну незалежність між ознаками (розумне припущення, враховуючи, що більшість ознак вже були оброблені методом PCA в наборі даних). Якщо значення щільності розподілу для нової транзакції падає нижче за попередньо визначений поріг, було класифіковано її як потенційно шахрайську. Процес вибору порогового значення включав оптимізацію F2-балів шляхом ітеративного прогнозування на валідаційному наборі.

За допомогою цього підходу було визначено оптимальне порогове значення приблизно $0{,}009^9 \approx 3.87*10^{-19}$, що відповідає дуже низькому значенню щільності. Також це дало оцінку F_2 у 0,834671 на валідаційному наборі. Цей показник також добре відображає надійність нашої моделі у виявленні шахрайських транзакцій за певних умов. Застосування тієї ж моделі до тестового набору підтвердило надійність нашого підходу, досягнувши результату F2 0,816492, що свідчить про його ефективність на різних наборах даних. Ці результати підтверджують ефективність нашої методології в практичних умовах і демонструють її потенціал для застосування у справжніх умовах для виявлення шахрайських операцій.

2.5. Висновок другого розділу

В розділі було розглянуто різноманітні методи виявлення аномалій в банківських транзакціях, що є ключовим завданням для забезпечення безпеки та ефективності фінансових установ. Особливу увагу приділено виявленню відмінностей між методами виявлення аномалій та шахрайства, включаючи статистичний аналіз, контрольовані та неконтрольовані методи навчання.

Зокрема, було виявлено, що статистичні методи та аналітичні підходи є ефективними для ідентифікації аномальних шаблонів у великих обсягах даних. Контрольовані методи навчання, такі як класифікація з учителем, показали високу точність у виявленні шахрайства. Для досягнення цього, було вирішено проблему незбалансованості даних шляхом застосування різних технік, що дозволило підвищити ефективність методів.

Також в цьому розділі особливе значення було надано розробці системи виявлення аномалій в банківських транзакціями – було розглянуто побудову системи виявлення аномалій для виявлення шахрайських транзакцій з використанням машинного навчання. Та за допомогою цієї системи виявлення було проаналізовано набір даних транзакцій європейських власників кредитних карт за вересень 2013 року. Цей набір даних включав як легітимні, так і шахрайські транзакції, і був взятим з реального життя, що підкреслює актуальність та практичну значущість проведеного аналізу.

РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ

В сучасних умовах банківські установи стикаються з різноманітними формами шахрайства, які становлять серйозну загрозу для їхньої фінансової стабільності. Ефективна протидія цим загрозам потребує значних капітальних інвестицій, що включають витрати на розробку та впровадження систем інформаційної безпеки, програмного та апаратного забезпечення, а також на навчання персоналу.

Детальний економічний аналіз включає оцінку вартості кожного з компонентів капітальних витрат, що дозволяє точно визначити необхідні ресурси для реалізації проекту. Аналіз фіксованих витрат є критичним етапом у впровадженні будь-якої системи безпеки, оскільки він надає фінансовим установам можливість планувати свої бюджети та оцінювати економічну доцільність інвестицій у захист від шахрайства.

3.1. Розрахунок фіксованих (капітальних) витрат

Капітальні інвестиції:

1. вартість розробки проєкту інформаційної безпеки (розробка схем пристроїв, політики функціонування системи тощо);

2. вартість створення основного й додаткового програмного забезпечення (ПЗ);

3. витрати на первісні закупівлі апаратного забезпечення;

4. витрати на навчання технічних фахівців і обслуговуючого персоналу.

Спершу було розраховано час, який буде витрачено на створення ПЗ:

$$t = t\text{тз} + t\text{в} + ta + t\text{пр} + t\text{опр} + t\text{д}, \text{годин} \quad (3.1)$$

де ttз – тривалість складання технічного завдання на розробку ПЗ;

tв – тривалість вивчення ТЗ, літературних джерел за темою тощо;

ta – тривалість розробки блок-схеми алгоритму;

tпр– тривалість програмування за готовою блок-схемою;

tопр – тривалість опрацювання програми на ПК;

tд – тривалість підготовки технічної документації на ПЗ.

Умовна кількість оперантів у програмі:

$$Q = q * c(1 + p), \text{штук} \quad (3.2)$$

де q – очікувана кількість операторів – 9;

c – коефіцієнт складності програми – 3;

p – коефіцієнт виправлення програми в процесі її використання – 1.

$$Q = 9 * 3(1 + 1) = 54 \text{ штук}$$

Оцінка часу на складання технічного завдання для розробки програмного забезпечення tтз – 6 годин. Час, витрачений на ознайомлення з технічним завданням:

$$t\text{в} = \frac{Q * B}{(75 \dots 85) * k} = \frac{54 * 1.5}{75 * 0.3} = 3.6 \text{ год} \quad (3.3)$$

де B – коефіцієнт, що показує збільшення тривалості етапу через недостатній опис завдання, B = 1,2...1,7;

k – коефіцієнт, який враховує кваліфікацію програміста що розроблює програмне забезпечення і залежить від стажу роботи за фахом: до 2 років – 1,0.

Тривалість розробки блок-схеми для алгоритму:

$$t\text{в} = \frac{Q}{(20 \dots 25) * k} = \frac{54}{20 * 0.3} = 9 \text{ год.} \quad (3.4)$$

Тривалість складання програми за готовою блок-схемою:

$$t\text{пр} = \frac{Q}{(20 \dots 25) * k} = \frac{54}{20 * 0.3} = 9 \text{ год.} \quad (3.5)$$

Тривалість опрацювання програми на ПК:

$$t\text{опр} = \frac{1.5Q}{(4 \dots 5) * k} = \frac{1.5 * 54}{4 * 0.3} = 67.50 \text{ год.} \quad (3.6)$$

Тривалість підготовки технічної документації на ПЗ:

$$t\text{д} = \frac{Q}{(15 \dots 20) * k} + \frac{Q}{(15 \dots 20)} * 0.75 = \frac{54}{15 * 0.3} + \frac{54}{15} * 0.75 = 14.70 \text{ год.} \quad (3.7)$$

Час, який буде витрачено на створення ПЗ:

$$t = 6 + 3.60 + 9 + 9 + 67.50 + 14.70 = 109.80 \text{ годин}$$

Розрахунок витрат на створення програмного продукту

$$K\text{пз} = \text{Ззп} + \text{Змч. грн}, \quad (3.8)$$

Оплата праці враховує основну заробітну плату, додаткову заробітну плату і відрахування на соціальні потреби (такі як пенсійне страхування, страхування в разі безробіття, соціальне страхування тощо), але для цього потрібно розрахувати спочатку средньогодинну заробітну плату програміста з нарахуваннями, грн/годину.

$$\text{Зпр} = \frac{\text{Зм}}{168} = \frac{46000}{168} = 273.809 \frac{\text{грн}}{\text{годину}} \quad (3.9)$$

де Зм – середня заробітна плата програміста з розробки штучного інтелекту на місяць – 46000 грн.

Сама ж оплата праці обчислюються за певною формулою:

$$Ззп = t * Зпр = 109.8 * 274.809 = 30174.03 \text{ грн} \quad (3.10)$$

де t – загальна тривалість створення ПЗ, годин;

Зпр– средньогодинна заробітна плата програміста з нарахуваннями, грн/годину.

Витрати машинного часу для налаштування та налагодження програми на ПКі розраховуються за формулою:

$$Змч = t\text{опр} * Смч + t\text{д} = (67.5 + 14.7) * 2.99 = 30174.03 \text{ грн} \quad (3.11)$$

Вартість однієї години використання обчислювального часу на комп'ютері розраховується за певною формулою:

$$Смч = P * Се + \frac{(Фзал * На)}{Fp} + \frac{(Клпз * Напз)}{Fp}$$

$$Смч = 0.5 * 4.32 + \frac{(6000 * 0.1)}{1920} + \frac{(2000 * 0.5)}{1920} = 2.99 \frac{\text{грн}}{\text{год}} \quad (3.12)$$

де Р – потужність комп'ютера, становить 0.5 кВт;

Се – тариф на електроенергію, дорівнює 2.1 грн/кВт·год;

Фзал – залишкова вартість ПК на поточний рік, складає 4000 грн;

На – річна норма амортизації на ПК, 0.1 частки одиниці;

Клпз – вартість ліцензійного програмного забезпечення, в гривнях;

Напз – річна норма амортизації на ліцензійне програмне забезпечення, також 0.1 частки одиниці;

Fp – річний фонд робочого часу (при 40-годинному робочому тижні, Fp = 1920 год).

$$Кпз = 30174.03 + 245.778 = 30419.81 \text{ грн}$$

Таким чином, капітальні (фіксовані) витрати на проєктування та впровадження проєктного варіанта системи інформаційної безпеки складають:

$$K = \text{Кпз} + \text{Кнавч} + \text{Кн}, \text{грн} \quad (3.13)$$

де Кпз – вартість створення програмного забезпечення, грн;

Кнавч – витрати на навчання технічних працівників та підтримку персоналу, в гривнях;

Кн – витрати на встановлення та налаштування системи інформаційної безпеки, в гривнях.

Витрати на тренінг технічних спеціалістів і підтримуючий персонал, включаючи курси з управління і підтримки системи виявлення вторгнень, становлять 3000 грн;

Кнавч = 9000 грн;

Витрати на встановлення обладнання та налаштування системи інформаційної безпеки складають 4500 грн;

Кн = 4500 грн.

$$K = 30500 + 9000 + 4500 = 44000 \text{ грн}$$

3.2. Експлуатаційні витрати

$$\text{Ск} = \text{Сн} + \text{Са} + \text{Сз} + \text{Сєв} + \text{Се} + \text{Сел} + \text{Стос}, \text{грн} \quad (3.14)$$

де витрати на навчання персоналу і користувачів (Сн) визначаються за даними організації з проведення тренінгів персоналу та курсів підвищення кваліфікації й займає – 20000 грн.

Річного фонду амортизаційних відрахувань в нас немає – оскільки він розраховується тільки якщо буде закупівля апаратного, програмного чи іншого забезпечення.

Річна зарплатня для інженерно-технічного персоналу, який працює з системою інформаційної безпеки (СЗ):

$$Сз = Зосн + Здод = 8000 * 12 + 8000 * 0.22 * 12 = 117120 \text{ грн} \quad (3.15)$$

де Зосн, Здод– основна мінімальна заробітна плата на 01.01.2018, грн на рік.

Единий соціальний внесок – 22% або ж 0.22, частк. одиниці.

Вартість електроенергії, яку споживає апаратура системи інформаційної безпеки протягом року (Се), визначається за формулою, яка враховує кількість електричної енергії, що використовується, та відповідні тарифи на електроенергію:

$$Сел = Р * Fр * Це = 0.5 * 365 * 24 * 4.32 = 18921.60 \text{ грн} \quad (3.16)$$

де Р – встановлена потужність апаратури інформаційної безпеки, кВт;

Fр – річний фонд робочого часу системи інформаційної безпеки;

Це – тариф на електроенергію, 4.32 грн/кВт·годин.

Витрати на адміністрування та сервіси для виявлення вторгнень становлять 20% від загальних капітальних витрат:

$$Стос = К * 0.2 = 44000 * 0.2 = 8800 \text{ грн} \quad (3.17)$$

$$Ск = 20000 + 117120 + 18921.6 + 8800 = 164841.6 \text{ грн}$$

3.3. Оцінка потенційного ушкодження від інциденту (атаки), що спрямований на вразливість вузла корпоративної мережі

Завершальним результатом впровадження та проведення заходів із забезпечення інформаційної безпеки є сума відвернутих збитків, що

обчислюється на основі ймовірності виникнення інциденту інформаційної безпеки та потенційних економічних збитків від нього. Ця величина відображає ту частину прибутку, яка могла бути втрачена.

Загалом можна виділити такі типи збитків, що можуть вплинути на ефективність комп'ютерної системи інформаційної безпеки (КСІБ):

1. порушення конфіденційності ресурсів КСІБ, яке може призвести до незаконного доступу або несанкціонованого використання каналів зв'язку;

2. порушення доступності ресурсів КСІБ, що може ускладнити доступ авторизованих суб'єктів до необхідних ресурсів у будь-який момент;

3. порушення цілісності ресурсів КСІБ, що може призвести до їхнього пошкодження;

4. порушення автентичності ресурсів КСІБ, що може вплинути на їхню дійсність і вірогідність підробки.

Для розрахунку можливих економічних збитків використано такі вихідні дані:

- tп= 72 години – час простою вузла корпоративної мережі через атаку, годин;
- tв= 12 годин – час відновлення роботи після атаки персоналом, годин;
- tви= 24 години – час повторного введення загубленої інформації співробітниками атакованого вузла корпоративної мереж, годин.;
- Зо= 9760 грн – місячна заробітна плата обслуговуючого персоналу з урахуванням єдиного соціального внеску, грн на місяць;
- Зс= 10980 грн – місячна заробітна плата співробітника атакованого вузла корпоративної мережі з урахуванням єдиного соціального внеску, грн на місяць;
- Чо= 10 – чисельність обслуговуючого персоналу, осіб;
- Чс= 35 – чисельність співробітників атакованого вузла корпоративної мережі, осіб;
- О= 8000000 грн – обсяг чистого прибутку/доходу від реалізації атакованого вузла корпоративної мережі на рік, грн у рік;

- Пзч= 8000 грн – вартість заміни обладнання або запасних частин, грн;
- I= 1 – кількість атакованих вузлів корпоративної мережі;
- N= 10 – середня кількість можливих атак на рік.

Упущена вигода від простою атакованого вузла корпоративної мережі становить:

$$U = \text{Пп} + \text{Пв} + V, \text{грн} \tag{3.18}$$

де Пп – оплачувані втрати робочого часу та простої співробітників атакованого вузла або корпоративної мережі, грн;

Пв – вартість відновлення працездатності вузла корпоративної мережі (переустановлення системи, зміна конфігурації та ін.), грн;

V – втрати від зниження обсягу продажів за час простою атакованого вузла корпоративної мережі, грн.

Втрати від зниження продуктивності співробітників в результаті атаки на вузол корпоративної мережі оцінюються витратами на їхню заробітну плату за 48 годин непрацездатності внаслідок цієї атаки.:

$$\text{Пп} = \frac{\sum \text{Зс} * \text{Чс}}{F} * t\text{п}, \text{грн} \tag{3.19}$$

де F – місячний фонд робочого часу (при 40-годинному тижні складає 168 годин).

$$\text{Пп} = \frac{10980 * 35}{168} * 72 = 164700 \text{ грн.}$$

Витрати на відновлення ефективності (працездатності) вузла корпоративної мережі включають декілька аспектів:

$$\text{Пв} = \text{Пви} + \text{Пвв} + \text{Пзч}, \text{грн} \tag{3.20}$$

де Пви – витрати на повторне уведення інформації, грн;

Ппв– витрати на відновлення вузла корпоративної мережі, грн;

Пзч – вартість заміни устаткування або запасних частин, грн.

Витрати на повторне введення інформації розраховуються згідно з оплатою праці 10980 грн 10 співробітників атакованого вузла корпоративної мережі, які витрачають 24 години на цю роботу.

$$\text{Пви} = \frac{10980}{168} * 24 = 1568.58 \text{ грн} \quad (3.21)$$

Витрати на відновлення після атаки вузла корпоративній мережі Ппв обчислюються часом відновлення після атаки tв = 12 і розміром середньогодинної заробітної плати обслуговуючого персоналу (адміністраторів):

$$\text{Пвв} = \frac{9760}{168} * 12 = 697.15 \text{ грн} \quad (3.22)$$

$$\text{Пв} = 1568.58 + 697.15 + 8000 = 10265.73 \text{ грн} \quad (3.23)$$

Збитки від зниження очікуваного виявлення шахрайства та його запобіганню у розмірі 1000000 грн протягом 108 годин через відмову вузла мережі.

$$V = \frac{\text{О}}{F\text{г}} * (t\text{п} + t\text{в} + t\text{ви}) = \frac{1000000}{9340} * (72 + 12 + 24) = 11563.16 \text{ грн} \quad (3.24)$$

де Fг – річний фонд часу роботи організації становить близько 9340 ч

$$U = 164700 + 10265.74 + 11563.16 = 186528.90 \text{ грн.}$$

Таким чином, загальний втрачений обсяг коштів в результаті нападу на вузол чи частину корпоративної мережі організації буде дорівнювати:

$$U = \Sigma_i \Sigma_n U = 186528.90 * 10 * 1 = \ 1865289 \ \text{грн} \qquad (3.25)$$

3.4. Загальний вплив впровадження системи інформаційної безпеки

Загальний вплив впровадження системи захисту інформації визначається ризиками порушень безпеки і складає:

$$E = B * R - C = 1865289 * 0.6 - 164841.6 = \ 954331.8 \ \text{грн} \qquad (3.26)$$

де В – загальний збиток від атаки на вузол корпоративної мережі, грн;

R – очікувана імовірність атаки на вузол корпоративної мережі, частки одиниці;

С – щорічні витрати на експлуатацію системи інформаційної безпеки, грн.

3.5. Визначення та аналіз показників економічної ефективності системи

Розглянемо коефіцієнт повернення інвестицій ROSI, який використовується для визначення економічної ефективності системи інформаційної безпеки:

$$ROSI = \frac{E}{K} = \frac{954331.8}{44000} = 21.69, \text{одиниці} \qquad (3.27)$$

де Е – загальний ефект від впровадження системи інформаційної безпеки, грн;

К – капітальні інвестиції за варіантами, що забезпечили цей ефект, грн.

Термін окупності:

$$\text{То} = \frac{K}{E} = \frac{1}{ROSI} = 0.046 \text{ років}, \qquad (3.28)$$

3.6 Висновки економічного розділу

Розрахунки в економічному розділі показали, що витрати на впровадження системи виявлення та запобіганню шахрайських операцій доволі швидко окуплюються. Проєкт є економічно вигідним і може бути успішно впроваджений на підприємстві.

Щоб оцінити ефективність і доцільність проєкту, враховувалися не лише витрати на впровадження системи виявлення, а й очікувані економічні вигоди і строк окупності інвестицій який становить 0.046 років (17 днів). Результати розрахунків підтвердили його економічну доцільність (ROSI = 21.69) і показали, що проєкт може стати необхідним ресурсом для підприємства.

ВИСНОВКИ

У цій кваліфікаційній роботі було глибоко проаналізовано проблему шахрайства в банківській сфері, що є серйозним викликом для фінансових установ. Шахрайство виявляється в різних формах і є значною загрозою для економічної стабільності та довіри до банківської системи. Викладені дані та приклади демонструють широкий спектр методів, які зловживаються, а також вказують на важливість ефективних стратегій їх виявлення та протидії.

Аналіз сучасних методів виявлення шахрайства показав, що використання статистичних моделей, машинного навчання та систем класифікації є надзвичайно важливим для успішної боротьби з цим явищем.

Також було створено програму для виялення шахрайських операцій – та визначено оптимальне порогове значення приблизно 0,0099 ≈ 3.87*10-19 , що відповідає дуже низькому значенню щільності. Також це дало оцінку F2 у 0,834671 на валідаційному наборі. Застосування тієї ж моделі до тестового набору підтвердило надійність нашого підходу, досягнувши результату F2 0,816492, що свідчить про його ефективність на різних наборах даних

Подальший економічний аналіз вказав на те, що інвестиції у сучасні системи виявлення шахрайства швидко окупаються через зменшення втрат від шахрайських операцій та збільшення довіри клієнтів до фінансових установ. Це робить такі проєкти не лише стратегічно важливими для безпеки фінансових систем, але й економічно доцільними для підприємств.

Крім того, важливо підкреслити, що розвиток інформаційних технологій надає нові можливості для виявлення аномальних поведінок та шаблонів шахрайства, забезпечуючи більш точне і швидке реагування фінансових установ на потенційні загрози. Отже, впровадження сучасних методів виявлення шахрайства є критичним елементом стратегії управління ризиками в банківській сфері. Це дозволяє забезпечити стабільність і надійність фінансових установ у умовах постійно зростаючих загроз.

ПЕРЕЛІК ПОСИЛАНЬ

1. Глобальне дослідження з питань шахрайства у банківській сфері. Багатостороння загроза шахрайства: чи готові банки гідно протистояти виклику? KPMG International 2019. 24 с. URL: https://assets.kpmg.com/content/dam/kpmg/ua/pdf/2019/11/Global_Banking_Fraud_Survey.pdf

2. Гороховський вперше оприлюднив реальні цифри шахрайства з картками клієнтів. URL: https://minfin.com.ua/ua/2023/07/31/109897112/.

3. Гороховський О. У monobank розповіли про схему шахраїв із "заробітком в інтернеті". Хвиля. URL: https://hvylya.net/uk/news/283798-v-monobank-rasskazali-o-sheme-moshennikov-s-zarabotkom-v-internete.

4. EMA anti fraud hub. ЄМА. URL: https://www.ema.com.ua/business/antifraud-hub/.

5. Застосування штучного інтелекту допоможе знизити рівень шахрайства у фінансовому секторі. Ощадбанк. URL: https://www.oschadbank.ua/news/zastosuvanna-stucnogo-intelektu-dopomoze-zniziti-riven-sahrajstva-u-finansovomu-sektori.

6. Сигетова, К., Узікова, Л., Доценко, Т., & Бойко, А. (2022). ОСТАННІ ТЕНДЕНЦІЇ ФІНАНСОВОЇ ЗЛОЧИННОСТІ СВІТУ. Financial and Credit Activity Problems of Theory and Practice, 5(46), 258–270.

7. Фішинг – що це таке і яка мета фішингу?. ESET. URL: https://www.eset.com/ua/support/information/entsiklopediya-ugroz/fishing/.

8. Total value of losses due to card fraud, either credit card fraud or debit card fraud, worldwide from 2014 to 2022. Statista. URL: https://www.statista.com/statistics/1394119/global-card-fraud-losses/.

9. As nationwide fraud losses top $10 billion in 2023, FTC steps up efforts to protect the public. FEDERAL TRADE COMISSION. URL: https://www.ftc.gov/news-events/news/press-releases/2024/02/nationwide-fraud-losses-top-10-billion-2023-ftc-steps-efforts-protect-public.

10. Balamurugan M., Mathiazhagan P. Credit card transaction fraud detection

system using fuzzy logic and k-means algorithm. International journal of innovative research in technology. 2015.

11. Credit card fraud detection system using hidden Markov model and k-clustering / M. Zubair et al. International journal of advanced research in computer and communication engineering.

12. Detecting Cyber Attacks in Smart Grids Using Semi-Supervised Anomaly Detection and Deep Representation Learning / R. Qi et al. Information. 2021. Vol. 12, no. 8. P. 328.

13. Hilda H. Top 10 financial fraud detection software for online businesses. Hyperverge. URL: https://hyperverge.co/blog/financial-fraud-detection-software/.

14. Patel D., Singh D. Credit card fraud detection & prevention of fraud using genetic algorithm. International journal of soft computing and engineering (IJSCE). 2013.

15. Phishing_Vishing_UA_edited_EPOL. UKRSIBBANK | BNP PARIBAS GROUP. URL: https://ukrsibbank.com/wp-content/uploads/2021/12/recognize_vishing_phishing_smishing.pdf.

16. Pradhan A. K. Statistical Methods for Anomaly Detection using Python: A Comprehensive Guide. Medium. URL: https://medium.com/@akpradhn/statistical-methods-for-anomaly-detection-using-python-a-comprehensive-guide-96d7b95bec35.

17. Probability Density Function. NewCastle University. URL: https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/distribution-functions/probability-density-function.html.

18. Sirazinia A. Imbalanced Learning in Banking. Medium. URL: https://medium.com/analytics-ai-swedbank/imbalanced-learning-in-banking-1bd3868a496d.

19. Top fraud detection solutions 2024. Luxand Cloud. URL: https://luxand.cloud/face-recognition-blog/top-fraud-detection-solutions-2024.

ДОДАТОК А. Відомість матеріалів кваліфікаційної роботи

№	Формат	Найменування	Кількість листів	Примітка
1	А4	Реферат	2	
2	А4	Список умовних скорочень	1	
3	А4	Зміст	2	
4	А4	Вступ	2	
5	А4	1 Розділ	17	
6	А4	2 Розділ	35	
7	А4	3 Розділ	11	
8	А4	Висновки	1	
9	А4	Перелік посилань	2	
10	А4	Додаток А	1	
11	А4	Додаток Б	10	
12	А4	Додаток В	1	
13	А4	Додаток Г	1	
14	А4	Додаток Д	1	

ДОДАТОК Б. Лістинг коду програми виявлення аномалій

```
import time, psutil, os, gc
# Додання бібліотеки математичних функцій
import math
# Додання головний бібліотек для маніпуляцій над данимим
import numpy as np
import pandas as pd
# Побудова графіків та візуалізація
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
import seaborn as sns
sns.set_theme()
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
# Розбиття набору даних на тренувальні тести
from sklearn.model_selection import train_test_split
# Індикатор прогресу для циклу
from tqdm.contrib import itertools
# Завантаження датасету та його показ в таблиці
data = pd.read_csv('creditcard.csv')
print(pd.Series({"Використання памяті диску": "{:.2f}
MB".format(data.memory_usage().sum()/(1024*1024)),
                 "Форма набору даних": "{}".format(data.shape)}).to_string())
data.head(20)
# Розподіл даних за цільовими класами
```

```
data_0, data_1 = data[data['Class'] == 0], data[data['Class'] == 1]

# Розподіл функцій та цілей
X_0, y_0 = data_0.drop('Class', axis = 1), data_0['Class']
X_1, y_1 = data_1.drop('Class', axis = 1), data_1['Class']

# Розбиття звичайного класу та побудова навчальної вибірки
X_train, X_test, y_train, y_test = train_test_split(X_0, y_0, test_size = 0.2, random_state = 40)
X_val, X_test, y_val, y_test = train_test_split(X_test, y_test, test_size = 0.5, random_state = 40)
data_val_1, data_test_1 = pd.concat([X_val, y_val], axis = 1), pd.concat([X_test, y_test], axis = 1)

# Поділ класу шахрайства
X_val, X_test, y_val, y_test = train_test_split(X_1, y_1, test_size = 0.5, random_state = 40)
data_val_2, data_test_2 = pd.concat([X_val, y_val], axis = 1), pd.concat([X_test, y_test], axis = 1)

# Об'єднання даних для побудови вибірки для валідації та тестової вибірки
data_val, data_test = pd.concat([data_val_1, data_val_2], axis = 0), pd.concat([data_test_1, data_test_2], axis = 0)
X_val, y_val = data_val.drop('Class', axis = 1), data_val['Class']
X_test, y_test = data_test.drop('Class', axis = 1), data_test['Class']

# Розподіл звичайних та шахрайських транзакцій між тренувальними, валідаційним та тестовим набором
labels = ['Тренування', 'Валідація', 'Тестування']
```

```
values_0 = [len(y_train[y_train == 0]), len(y_val[y_val == 0]), len(y_test[y_test == 0])]
values_1 = [len(y_train[y_train == 1]), len(y_val[y_val == 1]), len(y_test[y_test == 1])]
fig = make_subplots(rows = 1, cols = 2, specs = [[{'type': 'domain'}, {'type': 'domain'}]])
fig.add_trace(go.Pie(values = values_0, labels = labels, hole = 0.5, textinfo = 'percent', title = "Звичайні"),
              row = 1, col = 1)
fig.add_trace(go.Pie(values = values_1, labels = labels, hole = 0.5, textinfo = 'percent', title = "Шахрайські"),
              row = 1, col = 2)
text_title = "Розподіл звичайних та шахрайських транзакцій між тренувальними, валідаційним та тестовим набором"
fig.update_layout(height = 500, width = 1000, showlegend = True, title = dict(text = text_title, x = 0.5, y = 0.95))
fig.show()

# Налаштування кількості розподілів на комірки
bins_train = math.floor(len(X_train)**(1/3))
# Час
for df in [X_train, X_val, X_test]:
    df['Day'], temp = df['Time'] // (24*60*60), df['Time'] % (24*60*60)
    df['Hour'], temp = temp // (60*60), temp % (60*60)
    df['Minute'], df['Second'] = temp // 60, temp % 60
X_train[['Time', 'Day', 'Hour', 'Minute', 'Second']].head(10)

# Візуалізація
fig, ax = plt.subplots(1, 2, figsize = (15, 6), sharey = False)
sns.histplot(data = X_train, x = 'Time', bins = bins_train, ax = ax[0])
```

```
sns.histplot(data = X_train, x = 'Hour', bins = 24, ax = ax[1])
ax[1].set_ylabel(" ")
plt.suptitle("Гістограми часу та години", size = 14)
plt.tight_layout()
plt.show()

# Трансформація "Amount"
for df in [X_train, X_val, X_test]:
    df['Amount_transformed'] = np.log10(df['Amount'] + 0.001)

# Порівняння розподілів ознак для різних цільових класів
data_val = pd.concat([X_val, y_val], axis = 1)
data_val_0, data_val_1 = data_val[data_val['Class'] == 0],
data_val[data_val['Class'] == 1]
cols, ncols = list(X_val.columns), 4
nrows = math.ceil(len(cols) / ncols)
fig, ax = plt.subplots(nrows, ncols, figsize = (4.5 * ncols, 4 * nrows))
for i in range(len(cols)):
    sns.kdeplot(data_val_0[cols[i]], ax = ax[i // ncols, i % ncols])
    sns.kdeplot(data_val_1[cols[i]], ax = ax[i // ncols, i % ncols])
    if i % ncols != 0:
        ax[i // ncols, i % ncols].set_ylabel(" ")
plt.tight_layout()
plt.show()
# Вибір функцій
cols = ['V4', 'V11', 'V12', 'V14', 'V16', 'V17', 'V18', 'V19', 'Hour']
X_train_fs, X_val_fs, X_test_fs = X_train[cols], X_val[cols], X_test[cols]
X_train_fs.head()

#  Нормальна щільність розподілу ймовірностей
```

```
def normal_density(x, mu, sigma):
    """
    Обчислює одновимірну нормальну щільність розподілу ймовірностей із середнім значенням u(мю), стандартним відхиленням сігма
    Аргументи:
    x: вхідне спостереження
    mu: середнє значення
    sigma: стандартне відхилення (> 0)
    Повертає
    f: значення одновимірної нормальної щільності розподілу ймовірностей
    """
    assert sigma > 0, "Середньоквадратичне відхилення має бути позитивним"
    f = (1 / (sigma * np.sqrt(2 * np.pi))) * np.exp(- (1 / 2) * ((x - mu) / sigma)**2)
    return f

# Добуток нормальної щільності розподілу ймовірностей
def normal_product(x_vec, mu_vec, sigma_vec):
    """
    Обчислює добуток одновимірних нормальних густин
    Аргументи:
      x_vec (array_like, shape (n,)) : вектор вхідних спостережень
      mu_vec (array_like, shape (n,)) : вектор середніх
      sigma_vec (array_like, shape (n,)): вектор стандартних відхилень (> 0)
    Повертає
      f: добуток одновимірних нормальних щільностей
    """
    assert min(sigma_vec) > 0, "Середньоквадратичне відхилення має бути позитивним"
```

```
    assert len(mu_vec) == len(x_vec), "Довжина середнього вектора не відповідає довжині вхідного вектора"
    assert len(sigma_vec) == len(x_vec), "Довжина вектора середньоквадратичного відхилення не відповідає довжині вхідного вектора"
    f = 1
    for i in range(len(x_vec)):
        f = f * normal_density(x_vec[i], mu_vec[i], sigma_vec[i])
    return f

# Підбір моделі
mu_train, sigma_train = X_train_fs.mean().values, X_train_fs.std().values

# Функція для прогнозування аномалії на основі порогового значення щільності ймовірності
def model_normal(X, epsilon):
    """
    Модель виявлення аномалій
    Аргументи:
      X (DataFrame, shape (m, n)): DataFrame ознак
      epsilon : порогове значення щільності (> 0)
    Повертає
      y (array_like, shape (m,)): передбачені мітки класів
    """
    y = []
    for i in X.index:
        prob_density = normal_product(X.loc[i].tolist(), mu_train, sigma_train)
        y.append((prob_density < epsilon).astype(int))
    return y

# Функція для обчислення матриці невідповідностей
```

```
def conf_mat(y_test, y_pred):
    """
    Обчислює матрицю плутанини
    Аргументи:
     y_test: істинні двійкові (0 або 1) мітки
     y_pred: передбачені двійкові (0 або 1) мітки
    Повертається:
     confusion_mat: двовимірний масив, що представляє матрицю плутанини 2x2
    """
    y_test, y_pred = list(y_test), list(y_pred)
    count, labels, confusion_mat = len(y_test), [0, 1], np.zeros(shape = (2, 2), dtype = int)
    for i in range(2):
       for j in range(2):
          confusion_mat[i][j] = len([k for k in range(count) if y_test[k] == labels[i] and y_pred[k] == labels[j]])
    return confusion_mat

# Функція для відображення матриці невідповідностей
def conf_mat_heatmap(y_test, y_pred):
    confusion_mat = conf_mat(y_test, y_pred)
    labels, confusion_mat_df = [0, 1], pd.DataFrame(confusion_mat, range(2), range(2))
    plt.figure(figsize = (6, 4.75))
    sns.heatmap(confusion_mat_df, annot = True, annot_kws = {"size": 16}, fmt = 'd')
    plt.xticks([0.5, 1.5], labels, rotation = 'horizontal')
    plt.yticks([0.5, 1.5], labels, rotation = 'horizontal')
    plt.xlabel("Прогнозована мітка", fontsize = 14)
```

```
        plt.ylabel("Справжня мітка", fontsize = 14)
        plt.title("Матриця невідповідностей", fontsize = 14)
        plt.grid(False)
        plt.show()

    # Функція для обчислення та повернення F2-балів
    def f2_score(y_test, y_pred):
        """
        Обчислює F2-бали за істинними та передбаченими бінарними (0 або 1) мітками
        Аргументи:
          y_test (array_like): істинні двійкові (0 або 1) мітки
          y_pred (array_like): передбачені двійкові (0 або 1) мітки
        Повертається:
          f2: F2-бал, отриманий з y_test та y_pred
        """
        confusion_mat = conf_mat(y_test, y_pred)
        tn, fp, fn, tp = confusion_mat[0, 0], confusion_mat[0, 1], confusion_mat[1, 0], confusion_mat[1, 1]
        f2 = (5 * tp) / ((5 * tp) + (4 * fn) + fp)
        return f2

    # Налаштування порогового значення щільності
    alpha_list, f2_list, f2_max, alpha_opt, y_val_pred_opt = [], [], 0.0, 0.0, np.zeros(len(y_val))
    for alpha, j in itertools.product(np.arange(0.001, 0.051, 0.001), range(1)):
        y_val_pred = model_normal(X_val_fs, epsilon = alpha**X_val_fs.shape[1])
        f2 = f2_score(y_val, y_val_pred)
        alpha_list.append(alpha)
        f2_list.append(f2)
```

```
        if f2 > f2_max:
            alpha_opt = alpha
            y_val_pred_opt = y_val_pred
            f2_max = f2

    # Побудова графіка F2-балів за альфа
    plt.figure(figsize = (9, 6))
    plt.plot(alpha_list, f2_list)
    plt.xlabel("Альфа", fontsize = 14)
    plt.ylabel("F2-бали", fontsize = 14)
    plt.title("F2-бали vs альфа", fontsize = 14)
    plt.tight_layout()
    plt.show()

    # Підсумок налаштування
    print(pd.Series({
        "Оптимальне альфа": alpha_opt,
        "Оптимальний бал F2": f2_score(y_val, y_val_pred_opt)
    }).to_string())
    # Матриця невідповідностей для прогнозів на валідаційному наборі
    conf_mat_heatmap(y_val, y_val_pred_opt)

    # Функція для обчислення та друку метрик оцінювання
    def evaluation(y_test, y_pred):
        confusion_mat = conf_mat(y_test, y_pred)
        tn, fp, fn, tp = confusion_mat[0, 0], confusion_mat[0, 1], confusion_mat[1,
0], confusion_mat[1, 1]
        print(pd.Series({
            "Accuracy (Точність)": (tp + tn) / (tn + fp + fn + tp),
            "Reliable (Надійність)": tp / (tp + fp),
```

```
            "Recall (Повнота)": tp / (tp + fn),
            "F1-бали": (2 * tp) / ((2 * tp) + fn + fp),
            "F2-бали": (5 * tp) / ((5 * tp) + (4 * fn) + fp),
            "Коефіцієнт кореляції Меттьюза (ККМ) ": ((tp * tn) - (fp * fn)) /
np.sqrt((tp + fp) * (tp + fn) * (tn + fp) * (tn + fn))
        }).to_string())

    # Прогнозування та оцінка на тестовому наборі
    y_test_normal = model_normal(X_test_fs, epsilon =
alpha_opt**X_test_fs.shape[1])
    evaluation(y_test, y_test_normal)

    # Матриця невідповідностей для прогнозів на тестовому наборі
    conf_mat_heatmap(y_test, y_test_normal)
```

ДОДАТОК В. Перелік документів на оптичному носії

Бедловський_ДС_125-20-1-КВ-РОБ.docx

Бедловський_ДС_125-20-1-КВ-РОБ.pdf

Бедловський_ДС_125-20-1-КВ-РОБ.pdf.p7s

Бедловський_ДС_125-20-1-ПРЕЗ-КВ-РОБ.pptx

Бедловський_ДС_125-20-1-ПРОГРАМА-КВ-РОБ.zip

ДОДАТОК Г. Відгук керівника економічного розділу:

Економічний розділ виконаний відповідно до вимог, які ставляться до кваліфікаційних робіт, та заслуговує на оцінку 90 б. («відмінно»).

Керівник розділу _____ Дар'я ПІЛОВА

(підпис) (ім'я, прізвище)

ДОДАТОК Д. ВІДГУК

на кваліфікаційну роботу бакалавра студента групи 125-20-1
Бедловського Дмитра Станіславовича
на тему: «Аналіз сучасних методів виявлення шахрайства у банківській сфері»

Пояснювальна записка складається зі вступу, трьох розділів і висновків, викладених на 89 сторінках.

Метою кваліфікаційної роботи є дослідження сучасних методів виявлення шахрайства у банківській сфері.

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності бакалавра спеціальності 125 "Кібербезпека". Для досягнення поставленої мети в кваліфікаційній роботі вирішуються наступні задачі: огляд існуючих типів шахрайства у банківський сфері, аналіз сучасних систем виявлення шахрайства та методів протидії ньому.

Практичне значення результатів кваліфікаційної роботи полягає у результатах проведеного дослідження алгоритмів виявлення аномалій у банківських транзакціях, що дозволяє автоматизувати процес виявлення загроз.

За час дипломування Бедловський Д.С. проявив себе фахівцем, здатним самостійно вирішувати поставлені задачі та заслуговує присвоєння кваліфікації бакалавра за спеціальністю 125 Кібербезпека, освітньо-професійна програма «Кібербезпека».

Рівень запозичень у кваліфікаційній роботі не перевищує вимог "Положення про систему виявлення та запобігання плагіату".

Кваліфікаційна робота заслуговує оцінки 94 - «відмінно».

Керівник
кваліфікаційної роботи
к.т.н., доц. каф. БІТ Олександр САФАРОВ