НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Навчально-науковий інститут прикладного системного аналізу

Кафедра штучного інтелекту

До захисту допущено:

В. о. завідувачки кафедри

_____ Ірина ДЖИГИРЕЙ

«___»_____20__ р.

Дипломна робота

на здобуття ступеня бакалавра
за освітньо-професійною програмою «Системи і методи штучного інтелекту»
спеціальності 122 «Комп'ютерні науки» на тему:
«Моделі машинного навчання для виявлення шахрайських транзакцій у фінансових системах»

Виконав:
Студент IV курсу, групи КІ-13
Пилипенко Артем Володимирович _____

Керівник:
доцент кафедри штучного інтелекту, к.ф-м.н., доцент,
Пишнограєв Іван Олександрович _____

Консультант з економічного розділу:
доцент кафедри економічної кібернетики, к.е.н., доцент,
Рощина Надія Василівна _____

Консультант з нормоконтролю:
фахівець першої категорії кафедри штучного інтелекту,
к.т.н., доцент,
Комариста Богдана Миколаївна _____

Рецензент:
доцент кафедри економічної кібернетики, к.ф-м.н., доцент,
Лазаренко Ірина Сергіївна _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу
Кафедра штучного інтелекту

Рівень вищої освіти – перший(бакалаврський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Системи та методи штучного інтелекту»

ЗАТВЕРДЖУЮ
В. о. завідувачки кафедри
_____ Ірина ДЖИГИРЕЙ
«15» січня 2025 р.

ЗАВДАННЯ
на дипломну роботу студенту
Пилипенку Артему Володимировичу

1. Тема роботи «Моделі машинного навчання для виявлення шахрайських транзакцій у фінансових системах», керівник роботи Пишнограєв Іван Олександрович, доцент кафедри ШІ, затверджені наказом по університету від «___»_____ 20__ р. №_____
2. Термін подання студентом роботи «09» червня 2025 року.
3. Вихідні дані до роботи: набір фінансових транзакцій.
4. Зміст роботи: Аналіз предметної області дослідження та даних, вибір доцільних методів та моделей для роботи, розробка алгоритму обраними методами, оцінка отриманих результатів.
5. Перелік ілюстративного матеріалу: електронна презентація.
6. Консультанти розділів роботи.

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Рощина Надія Василівна, доцент, к. е. н.		

7. Дата видачі завдання: «03» лютого 2025 року.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Огляд літератури за темою	21.04.2025	Виконано
2	Підготовка першого розділу	28.04.2025	Виконано
3	Підготовка другого розділу	05.05.2025	Виконано
4	Розробка програмного продукту	12.05.2025	Виконано
5	Підготовка третього розділу	19.05.2025	Виконано
6	Оформлення розділів відповідно до нормоконтролю	26.05.2025	Виконано
7	Оформлення дипломної роботи	02.06.2025	Виконано
8	Підготовка презентації доповіді	09.06.2025	Виконано

Студент Артем ПИЛИПЕНКО

Керівник Іван ПИШНОГРАЄВ

РЕФЕРАТ

Дипломна робота: 84 ст., 25 рис., 2 табл., 14 посилань, 2 додатки.

МОДЕЛІ МАШИННОГО НАВЧАННЯ, ШАХРАЙСЬКІ ТРАНЗАКЦІЇ, ФІНАНСОВІ СИСТЕМИ, КАСКАДНЕ МОДЕЛЮВАННЯ, АРХІТЕКТУРА МОДЕЛІ.

У роботі досліджується застосування моделей машинного навчання для виявлення шахрайських транзакцій у фінансових системах. Основну увагу приділено каскадній моделі, яка включає кілька етапів з різними методами класифікації та обробки даних. Описано побудову моделі — від попередньої обробки до застосування складних алгоритмів. Мета дослідження — створити ефективну каскадну модель для точного й своєчасного виявлення шахрайства, що сприятиме зниженню фінансових ризиків і підвищенню якості моніторингу.

Об’єктом дослідження є процес виявлення шахрайських транзакцій у фінансових системах шляхом застосування машинного навчання. Предметом дослідження є каскадна модель для класифікації транзакцій на основі статистичних ознак, класифікаційних алгоритмів та нейронних мереж. Актуальність цієї роботи зумовлена стрімким зростанням кількості онлайн-платежів та цифрових фінансових операцій, що у свою чергу, робить проблему виявлення шахрайських транзакцій надзвичайно актуальною для забезпечення безпеки користувачів і стабільного функціонування фінансових систем.

Проаналізовано підходи до обробки транзакцій, формування ознак і виявлення аномалій у контексті боротьби з шахрайством. Показано переваги каскадного моделювання, яке забезпечує ефективну роботу з великими фінансовими даними та підвищує точність виявлення шахрайських операцій.

ABSTRACT

Thesis: 64 pages, 25 figures, 2 tables, 14 sources, 2 appendices.

MACHINE LEARNING MODELS, FRAUDULENT TRANSACTIONS, FINANCIAL SYSTEMS, CASCADE MODELING, MODEL ARCHITECTURE.

This work explores the application of machine learning models for detecting fraudulent transactions in financial systems. The primary focus is on a cascade model that involves multiple stages with various classification and data processing methods. The model development is described from preprocessing to the use of advanced algorithms. The aim of the research is to create an effective cascade model for accurate and timely fraud detection, which will help reduce financial risks and improve monitoring quality.

The object of the study is the process of detecting fraudulent transactions in financial systems using machine learning. The subject is the cascade model for classifying transactions based on statistical features, classification algorithms, and neural networks. The relevance of this work is driven by the rapid growth of online payments and digital financial operations, which makes fraud detection a pressing issue for ensuring user security and the stable functioning of financial systems.

The study analyzes approaches to transaction processing, feature engineering, and anomaly detection in the context of fraud prevention. It highlights the advantages of cascade modeling, which enables efficient handling of large financial datasets and enhances the accuracy of fraud detection.

ЗМІСТ

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ

ML – Machine Learning.

DNN – Deep Neural Network.

CNN – Convolutional Neural Network.

RNN – Recurrent Neural Network.

PCA – Principal Component Analysis.

SMOTE – Synthetic Minority Over-sampling Technique.

AUPRC – Area Under Precision-Recall Curve.

ROC – Receiver Operating Characteristic.

AUC – Area Under Curve.

API – Application Programming Interface.

JWT – JSON Web Token.

GDPR – General Data Protection Regulation.

PCI DSS – Payment Card Industry Data Security Standard.

FFIEC – Federal Financial Institutions Examination Council.

EBA – European Banking Authority.

UI – User Interface.

ВСТУП

Шахрайські[1] транзакції є серйозною проблемою в сучасних фінансових системах, оскільки вони становлять загрозу для як бізнесів, так і для споживачів. Такі транзакції передбачають умисні дії з метою обману або введення в оману фінансових установ, що призводить до фінансових втрат і підриває довіру до фінансових систем. Зі швидкою цифровізацією фінансових послуг частота та складність шахрайських схем значно зросла, що робить необхідним розроблення нових моделей та методів для ефективного виявлення і запобігання шахрайства.

В Україні фінансовий сектор зазнає значних змін, і зростає кількість транзакцій, які здійснюються через цифрові канали. Однак зростання обсягів онлайн-транзакцій також призводить до збільшення кількості випадків шахрайства. Фінансові установи в Україні, як і в інших країнах, стикаються з проблемами у виявленні та боротьбі з шахрайськими операціями, оскільки традиційні методи часто є повільними та неефективними для виявлення складних схем шахрайства. Більш того, відсутність сучасних інструментів для виявлення шахрайства в реальному часі підвищує уразливість фінансових систем.

Нинішній стан системи виявлення шахрайства в Україні характеризується залежністю від традиційних правил та ручного контролю. Ці методи виявляються все менш ефективними для вирішення проблеми динамічних і складних шахрайських схем. Незважаючи на деякі досягнення в застосуванні машинного навчання, фінансовий сектор ще не повністю використовує потенціал штучного інтелекту та машинного навчання для

[1]Тут і нижче використані такі інструменти штучного інтелекту як чат-бот з генеративним штучним інтелектом ChatGPT, виключно для корегування та редагування тексту, створеного автором цієї дипломної роботи, на основі автоматизованої перевірки граматики, структури та стилю, що відповідає Політиці використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського (протокол №11 Вченої ради КПІ ім. Ігоря Сікорського від 11 грудня 2023 р.).

виявлення шахрайства. Як результат, багато фінансових установ продовжують покладатися на застарілі моделі, які не здатні адаптуватися до нових шахрайських схем.

Моделі машинного навчання, особливо ті, що використовують передові нейронні мережі, показали перспективні результати в виявленні шахрайства, оскільки вони забезпечують більш адаптивний та точний підхід. Такі моделі можуть аналізувати величезні обсяги транзакційних даних, виявляти складні закономірності та надавати попередження в реальному часі щодо підозрілих операцій. Однак все ще є багато аспектів, які потребують удосконалення, таких як точність, ефективність та інтерпретованість цих моделей у контексті виявлення шахрайства в фінансових системах.

Отже, метою цієї дипломної роботи є дослідження застосування моделей машинного навчання для виявлення шахрайських транзакцій у фінансових системах. Дослідження зосередиться на розробці моделі, яка може бути використана для підвищення точності виявлення шахрайства. Аналізуючи архітектуру таких моделей і застосовуючи їх до реальних фінансових даних, ця робота зробить внесок у розробку більш надійних та ефективних систем виявлення шахрайства.

РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМИ ШАХРАЙСЬКИХ ТРАНЗАКЦІЙ У ФІНАНСОВИХ СИСТЕМАХ

1.1 Загальна характеристика шахрайських транзакцій

Шахрайські транзакції є важливою та актуальною проблемою у фінансових системах по всьому світу. Вони полягають у навмисному введенні в оману чи обмані фінансових установ, спрямованому на отримання вигоди за рахунок інших учасників економічної діяльності. З розвитком технологій, а особливо в умовах цифровізації фінансових послуг, таких транзакцій стало значно більше, що зумовлює необхідність розробки нових ефективних методів їх виявлення та запобігання.

Шахрайство може проявлятися в різних формах, від використання крадених даних банківських карток до складних схем з використанням фальшивих платіжних систем і маніпулювання алгоритмами кредитування. Основними цілями таких транзакцій є незаконне отримання коштів, обман споживачів чи партнерів, а також нанесення шкоди фінансовим установам.

Зокрема, серед найбільш поширених видів шахрайства можна виокремити: крадіжку особистих даних, фішинг, шахрайство з платежами, мультиаккаунтинг і синтетичні ідентичності [1].

Крадіжка особистих даних відбувається тоді, коли шахраї отримують доступ до конфіденційної інформації, включаючи номери кредитних карток, паролі та інші персональні дані, з метою здійснення незаконних транзакцій. Одним з найпоширеніших методів є фішинг. Фішинг — це використання підроблених вебсайтів або електронних листів для збору даних користувачів. Шахрайство з платежами охоплює різні схеми, у яких зловмисники маніпулюють платіжними системами, створюючи вигляд легітимних операцій для незаконного отримання коштів. Ще однією поширеною схемою є мультиаккаунтинг і використання синтетичних ідентичностей, тобто шахраї створюють кілька облікових записів або

вигадані особи на основі фальшивих даних для отримання вигоди у сфері кредитування чи фінансів.

Всі ці види шахрайських транзакцій мають серйозний вплив на фінансові установи та споживачів. Вони підривають довіру до фінансових систем, призводять до значних фінансових втрат і навіть можуть спровокувати банкрутство окремих компаній або навіть цілих секторів економіки.

Завдяки швидкому розвитку цифрових технологій і збільшенню обсягу фінансових операцій, традиційні методи виявлення шахрайства, які базуються на ручному контролі та простих правилах, стали менш ефективними. Тому важливою є розробка нових, автоматизованих методів, таких як використання моделей машинного навчання, які здатні аналізувати великі обсяги даних, виявляти складні патерни та надавати точніші результати в реальному часі.

Машинне навчання може допомогти виявити шахрайські транзакції, аналізуючи не лише явні ознаки шахрайства, але й складні взаємозв'язки між різними характеристиками транзакцій, такими як час, сума, географічне розташування, історія операцій тощо [1]. Важливим аспектом є також здатність таких моделей адаптуватися до нових схем шахрайства, що постійно змінюються.

Таким чином, розвиток ефективних систем для виявлення шахрайських транзакцій є важливим завданням для фінансових установ, оскільки забезпечує не лише безпеку клієнтів, але й стабільність і довіру до всього фінансового сектору.

1.2 Методи та технології боротьби з фінансовим шахрайством

Фінансове шахрайство — це незаконні дії, спрямовані на отримання неправомірної вигоди або обману в сфері фінансів, таких як крадіжка, маніпулювання фінансовими звітами, махінації з грошовими потоками тощо. В Україні, як і в інших країнах, це питання набуває все більшої актуальності через розвиток цифрових технологій та зростання фінансових послуг. Проблема фінансового шахрайства має серйозний вплив на економіку, а також на довіру громадян до фінансових установ і організацій.

Для боротьби з фінансовим шахрайством використовуються різноманітні методи та технології, спрямовані на запобігання, виявлення та розслідування правопорушень [2]. Одним з основних підходів є застосування сучасних інформаційних технологій, таких як системи моніторингу фінансових операцій, програмне забезпечення для виявлення аномальних дій та штучний інтелект для аналізу фінансових даних [1].

Одним з основних напрямків боротьби з фінансовим шахрайством є розробка та впровадження систем антифродових технологій. Ці системи дозволяють виявляти підозрілі фінансові операції на основі аналізу великих обсягів даних, використовуючи алгоритми машинного навчання та аналізу поведінки користувачів. Наприклад, система може відстежувати транзакції, порівнюючи їх з історичними даними, виявляючи аномалії, які можуть свідчити про шахрайські дії [2].

Ще одним важливим методом є використання технологій блокчейн, які дозволяють створювати прозорі та незмінні записи про фінансові транзакції. Блокчейн дозволяє уникнути фальсифікації та забезпечує високий рівень безпеки в фінансових операціях. Завдяки цій технології можна значно знизити ризик шахрайства, особливо в цифрових платформах.

Використання біометричних технологій є ще одним важливим інструментом у боротьбі з фінансовим шахрайством. Це включає в себе

біометричну ідентифікацію, таку як розпізнавання відбитків пальців, сканування райдужної оболонки ока або обличчя [2]. Це дозволяє забезпечити більш високий рівень безпеки при доступі до фінансових ресурсів, унеможливлюючи їхнє незаконне використання третіми особами.

Крім того, фінансові установи застосовують методи аудиту та перевірки фінансових звітів, що включають в себе регулярні перевірки бухгалтерських документів та фінансових операцій. Це дозволяє виявляти ознаки маніпуляцій або несанкціонованих дій, а також забезпечувати високий рівень прозорості в діяльності організацій.

Одним із важливих кроків у боротьбі з фінансовим шахрайством є співпраця між державними та приватними установами, зокрема фінансовими організаціями, банками, правоохоронними органами та державними органами регулювання. Створення національних баз даних про шахраїв та співпраця з міжнародними організаціями дозволяють підвищити ефективність боротьби з фінансовими злочинами на глобальному рівні.

Важливу роль відіграють також законодавчі ініціативи, спрямовані на посилення покарання за фінансове шахрайство та запровадження механізмів, які забезпечують захист споживачів фінансових послуг. Це включає в себе впровадження нових стандартів безпеки та вимог до банківських та фінансових установ, а також надання відповідних правових інструментів для розслідування фінансових злочинів.

Загалом, боротьба з фінансовим шахрайством є складним і багатогранним процесом, який вимагає комплексного підходу, що включає технологічні інновації, законодавчі зміни та ефективну співпрацю між усіма зацікавленими сторонами. Тільки за таких умов можна забезпечити належний рівень захисту від фінансових злочинів і зміцнити довіру до фінансових систем України.

1.3 Використання машинного навчання у фінансовій сфері

Машинне навчання (ML) стало потужним інструментом у фінансовій сфері, здатним вирішувати складні завдання, пов'язані з прогнозуванням, оптимізацією та автоматизацією процесів [11]. Відкриття нових можливостей для аналізу великих обсягів даних дозволяє фінансовим установам приймати більш обґрунтовані рішення, що в свою чергу знижує ризики і збільшує ефективність операцій. Основні напрямки використання машинного навчання в фінансах.

Прогнозування фінансових ринків. Машинне навчання дозволяє створювати моделі, здатні аналізувати історичні дані та прогнозувати рухи на фінансових ринках, зокрема на фондових біржах або валютних ринках. Алгоритми можуть допомогти передбачити зміни цін на акції, облігації або інші фінансові інструменти.

Машинне навчання активно застосовується у фінансовій сфері для вирішення низки важливих завдань. Зокрема, кредитний скоринг дозволяє банкам точніше оцінювати кредитоспроможність клієнтів, знижуючи кількість помилок у процесі надання кредитів. Алгоритми аналізують великі обсяги даних про позичальників, зокрема історію погашення кредитів, рівень доходу, кількість попередніх позик та інші фактори, щоб надати точну оцінку ризику. У сфері виявлення шахрайства машинне навчання допомагає виявляти аномалії у фінансових операціях, що дозволяє вчасно реагувати на незаконні транзакції чи фальсифікацію даних. Алгоритми, навчені на великих масивах інформації, здатні оперативно розпізнавати підозрілі дії.

Алгоритмічна торгівля — ще один напрям, де машинне навчання демонструє високу ефективність: системи аналізують ринки в реальному часі та здійснюють автоматичні угоди, що підвищує швидкість і точність реакції на ринкові зміни. Персоналізація фінансових послуг також значно

покращується завдяки машинному навчанню — фінансові установи можуть пропонувати індивідуальні продукти, такі як кредити, інвестиції чи страхування, які найкраще відповідають потребам конкретного клієнта.

Незважаючи на численні переваги, використання машинного навчання у фінансовому секторі має свої обмеження та виклики [11]. Одним з основних є необхідність в обробці великих обсягів даних, що вимагає потужних обчислювальних ресурсів і висококваліфікованих фахівців для створення і підтримки моделей. Крім того, моделі машинного навчання можуть мати "чорну скриньку" — результат, який важко пояснити або інтерпретувати, що може створювати проблеми з регуляторами або клієнтами.

У майбутньому використання машинного навчання у фінансах буде лише зростати, особливо з урахуванням розвитку таких технологій, як блокчейн та інші розподілені реєстри. Очікується, що нові підходи до автоматизації аналізу даних, глибоке навчання і нейронні мережі зроблять фінансові послуги ще більш персоналізованими та ефективними, а також дозволять знизити операційні витрати та ризики.

1.4 Постановка задачі

У сучасному цифровому середовищі питання забезпечення фінансової безпеки набуває особливої актуальності. Стрімке зростання кількості онлайн-платежів, розширення можливостей мобільного банкінгу та електронної комерції створює сприятливі умови для появи нових видів шахрайських схем. У зв’язку з цим виникає потреба в побудові ефективних систем автоматичного виявлення аномальних транзакцій, які можуть свідчити про потенційне шахрайство. Основне завдання полягає у створенні гнучкої та надійної моделі машинного навчання, здатної в реальному часі

аналізувати великий потік фінансових даних і приймати рішення з високою точністю.

Ключовою проблемою є складність структури транзакційних даних, нерівномірний розподіл класів (із значним домінуванням легітимних транзакцій над шахрайськими), а також постійна еволюція методів шахрайства. Це обумовлює необхідність розробки багаторівневого підходу до аналізу, який дозволяє на кожному етапі застосовувати спеціалізовані алгоритми для попередньої обробки, виявлення ознак і уточнення результатів класифікації.

Таким чином, формулюється задача побудови поетапної, або каскадної, моделі виявлення шахрайства, яка використовує комбінацію класичних статистичних методів і сучасних алгоритмів машинного навчання. Такий підхід дає змогу не лише підвищити точність класифікації, але й зменшити кількість помилкових спрацьовувань та покращити масштабованість рішення. У рамках дослідження планується розробка та експериментальна перевірка запропонованого підходу, результати якого будуть розглянуті у наступних розділах.

Висновки до розділу 1

Отже шахрайські транзакції є серйозною загрозою для фінансових установ, оскільки вони призводять до значних фінансових втрат та підривають довіру клієнтів. Основні види шахрайства включають крадіжки даних, фішинг, маніпуляції з платіжними картками та використання підроблених документів. Виявлення таких транзакцій ускладнюється через їхню різноманітність та швидкість виконання.

Сучасні підходи до боротьби з шахрайством включають як традиційні методи (правила та експертні системи), так і інноваційні технології, такі як

аналіз великих даних та поведінкова аналітика. Комбінація різних підходів дозволяє забезпечити більш ефективний захист від шахрайства. Зокрема, використання багаторівневої аутентифікації, моніторинг транзакцій у режимі реального часу та аналіз аномалій є ключовими елементами сучасних систем безпеки.

Машинне навчання відіграє вирішальну роль у виявленні шахрайських транзакцій завдяки здатності обробляти великі обсяги даних та виявляти приховані закономірності. Алгоритми класифікації та кластеризації дозволяють розпізнавати підозрілі патерни в даних, а методи глибокого навчання підвищують точність прогнозів [12].

Поєднання традиційних методів боротьби з шахрайством та сучасних технологій машинного навчання є найбільш ефективним підходом для забезпечення фінансової безпеки. Подальший розвиток алгоритмів та збільшення обсягів даних сприятимуть підвищенню точності моделей та зниженню рівня шахрайства у фінансовій сфері.

РОЗДІЛ 2 ПОБУДОВА АРХІТЕКТУРИ МОДЕЛІ ДЛЯ ВИЯВЛЕННЯ ШАХРАЙСЬКИХ ТРАНЗАКЦІЙ

2.1 Основи машинного навчання у виявленні аномалій

Виявлення шахрайських транзакцій у фінансових системах є складним завданням, оскільки шахраї постійно вдосконалюють свої методи. Одним із ефективних підходів до вирішення цієї проблеми є використання методів машинного навчання, зокрема алгоритмів виявлення аномалій. Отже давайте спочатку розглянемо поняття машинного навчання.

Машинне навчання включає різні методи обробки даних для автоматичного пошуку закономірностей. Один із найбільш ефективних підходів – нейронні мережі, які імітують роботу людського мозку. Архітектура нейронної мережі, зображена на рисунку 2.1.

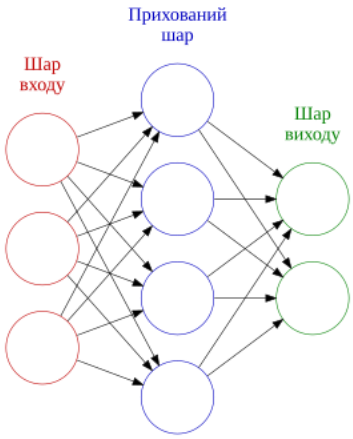


Рисунок 2.1 – Архітектура нейронної мережі[2]

[2] Джерело зображення: https://www.wikiwand.com/uk/articles/Штучна_нейронна_мережа#/media/Файл:Colored_neural_network_uk.svg

Відповідно до рисунку 2.1 архітектура нейронної мережі складається з нейронів (вузлів), об’єднаних у шари [5]. Вона містить три основні типи шарів, які описані нижче.

Вхідний шар передає вхідні характеристики далі у мережу. У контексті виявленні аномалій у фінансових транзакціях вхідні дані можуть містити: Суму транзакції, Час проведення, Геолокацію, Тип картки.

Прихований шар це фактично проміжні рівні мережі, які виконують обчислення та розпізнають приховані закономірності у вхідних даних, функція яких обчислення нелінійних залежностей у даних. Формула (2.1) продемонстрована для базового розуміння, обчислення одного нейрона.

$$z = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b, \tag{2.1}$$

де w – визначають важливість кожного вхідного сигналу;

b – додає додаткову гнучкість;

x – визначає, які сигнали передавати далі.

Вихідний шар це фінальний шар мережі, який формує підсумковий прогноз. Функція видає результат, який може бути таким.

1. Двійковим (0 або 1) – чи є транзакція аномальною.
2. Ймовірністю – наприклад, 80% того, що транзакція шахрайська.
3. Класами – якщо потрібно класифікувати тип аномалії.

Типи вихідних шарів.

1. Sigmoid – використовується для бінарної класифікації.
2. Softmax – використовується для багатокласової класифікації.

Аномалії у фінансових транзакціях можуть визначатися як відхилення від нормальної поведінки користувачів. Це можуть бути незвичайні суми переказів, нетипові місця проведення операцій, різкі зміни в частоті платежів тощо. Існує два основних типи методів виявлення аномалій, які зображені на рисунку 2.2.

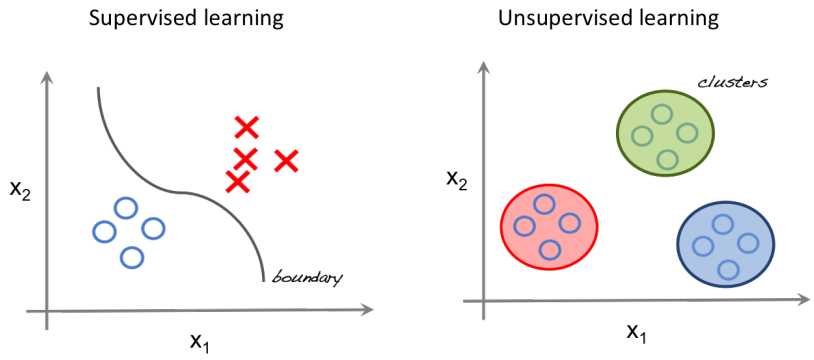


Рисунок 2.2 – Основні типи методів виявлення аномалій[3]

Методи без учителя (Unsupervised Learning), як правило використовуються, коли відсутні мітки "шахрайство/не шахрайство". До них належать методи кластеризації, наприклад: DBSCAN, K-Means та методи пошуку викидів (Isolation Forest, One-Class SVM). Відповідно, методи з учителем (Supervised Learning), більше ґрунтуються на використанні розмічених даних для навчання моделей класифікації. До них можуть належати такі алгоритми, як логістична регресія, дерева рішень, випадковий ліс та глибокі нейронні мережі.

Давайте віддалено сформуємо задачу. Нехай фінансова транзакція представлена як вектор ознак, формула (2.2) продемонстрована нижче.

$$X = (x_1, x_2, \dots, x_n), \tag{2.2}$$

де x_i – характеризує певний аспект транзакції.

Завдання моделі машинного навчання полягає в побудові функції $f(x)$, яка оцінює ймовірність того, що транзакція є шахрайською, вказано у формулі (2.3) [12].

[3] Джерело зображення: https://cdn.dida.do/blog/20230707_sp_supervised_vs_unsupervised_learning/illustration-supervised-vs-unsupervised.png

$$f(x) = P(y = 1|X) \quad (2.3)$$

2.2 Огляд відомих алгоритмів

Основні алгоритми, що можуть використовуються для класифікації шахрайських транзакцій це логістична регресія, дерева ухвалення рішень, випадковий ліс та градієнтний бустинг. Нижче розглянуто кожен з алгоритмів.

Логістична регресія — це статистичний метод, який використовується для бінарної класифікації. Вона обчислює ймовірність того, що транзакція є шахрайською, використовуючи сигмоїдну функцію активації. Нижче наведено принцип роботи логістичної регресії, на рисунку 2.3, а також її формула (2.4).

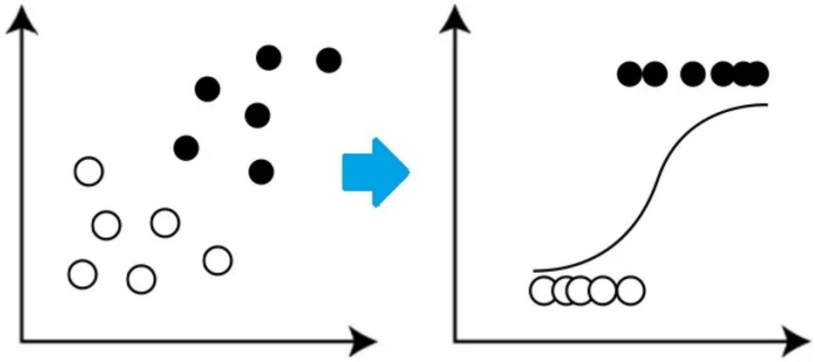


Рисунок 2.3 – Принцип роботи логістичної регресії[4]

$$P(\,y = 1 \mid X\,) = \frac{1}{1+e^{-(w_0+w_1x_1+w_2x_2+\cdots+w_nx_n)}} \quad (2.4)$$

[4] Джерело зображення: https://media2.dev.to/dynamic/image/width=800%2Cheight=%2Cfit=scale-down%2Cgravity=auto%2Cformat=auto/https%3A%2F%2Fdev-to-uploads.s3.amazonaws.com%2Fi%2Fc0zndgp3o6xfy6hhb0zo.png

Дерева ухвалення рішень — це ієрархічна модель, яка використовує послідовні розгалуження для ухвалення рішень. Основний принцип роботи полягає в тому, що кожен вузол дерева представляє тест над певною ознакою, а гілки відповідають можливим результатам тесту. Листові вузли (кінцеві) містять передбачені класи або значення. Для кращого розуміння, принцип роботи дерев рішень, зображений на рисунку 2.4.

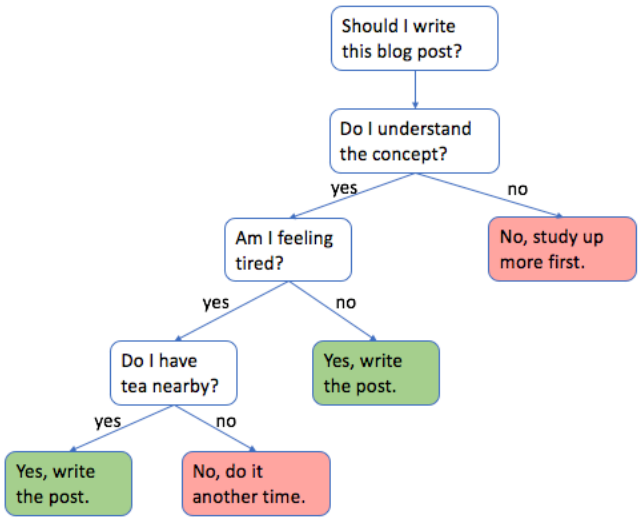


Рисунок 2.4 – Принцип роботи дерев прийнятя рішень[5]

Однак розбиття вузлів дерева виконується на основі декількох критеріїв описаних нижче.

Критерій Джині (Gini Index) – критерій, що використовується для визначення "чистоти" вузла в дереві рішень, формула (2.5). Він оцінює ймовірність того, що випадковий елемент буде неправильно класифікований, якщо його класифікація буде визначена випадково відповідно до розподілу класів у вузлі.

$$\text{Gini } = 1 - \sum_{i=1}^{C} p_i^2. \qquad (2.5)$$

[5] Джерело зображення: https://miro.medium.com/v2/resize:fit:1074/format:webp/1*KGUUGBvtT-bwFvt0jIe_Ug.png

Ентропія (Entropy) – це міра невизначеності або безладу у вибірці, формула (2.6). Вона використовується для визначення інформаційного приросту, тобто наскільки сильно зменшується невизначеність при розбитті вибірки за певною ознакою.

$$\text{Entropy } = -\sum_{i=1}^{C} p_i \log_2(p_i). \tag{2.6}$$

Інформаційний приріст (Inf Gain) – цей показник відображає, наскільки ознака зменшує ентропію після розбиття, формула (2.7). Це різниця між ентропією до і після поділу.

$$\text{Inf Gain } = \text{ Entropy (parent) } - \sum_{i=1}^{k} \frac{n_i}{n} \text{ Entropy (child }_i). \tag{2.7}$$

Зниження дисперсії (Variance Reduction, VR) – цей критерій використовується в регресійних деревах, а не класифікаційних, формула (2.8). Він оцінює, наскільки ефективно поділ зменшує варіацію (розкид) у цільовій змінній.

$$\text{VR } = \text{Var(parent)} - \left(\frac{n_{\text{left}}}{n}\text{Var(left)} + \frac{n_{\text{right}}}{n}\text{Var(right)}\right) \tag{2.8}$$

Випадковий ліс – ансамблевий метод, що об'єднує кілька дерев рішень для підвищення точності класифікації та зменшення перенавчання, формула (2.9). Фактично створюється кілька дерев рішень із випадковими підмножинами даних де кожне дерево приймає своє рішення щодо шахрайства [13]. І потім Відбувається голосування: якщо більшість дерев передбачає «шахрайство», то транзакція класифікується як шахрайська [8].

$$P(y = 1 \mid X) = \frac{1}{T}\sum_{t=1}^{T} h_t(X), \tag{2.9}$$

де T – кількість дерев;

$h_t(X)$ – прогноз окремого дерева.

Візуально зображено, як саме працює ансамблевий метод – випадковий ліс на рисунку 2.5.

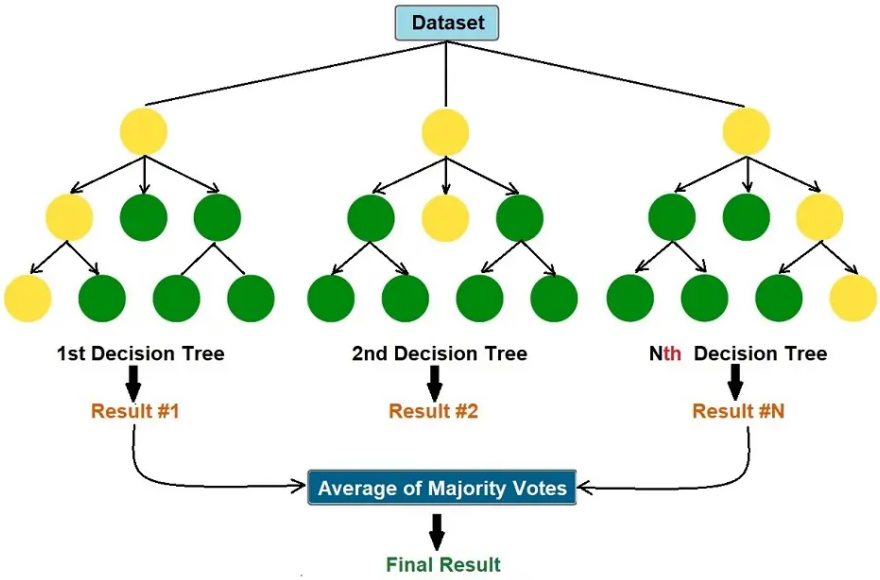


Рисунок 2.5 – Принцип роботи випадкового лісу[6]

Глибокі нейронні мережі (DNN) складаються з великої кількості прихованих шарів (іноді десятки або навіть сотні), які дозволяють моделі поступово виділяти та інтерпретувати складні структури в даних, формула (2.10). Кожен наступний шар навчається дедалі абстрактнішим ознакам. ReLU — найпопулярніша функція активації в DNN, особливо для прихованих шарів. Вона проста, швидка у розрахунках і допомагає уникати проблеми зникаючого градієнта [12].

$$f(z) = \max(0, z), \tag{2.10}$$

[6] Джерело зображення: https://www.ejable.com/wp-content/uploads/2023/11/Random-forest-6.webp

де $x < 0$: вихід 0;

$x \geq 0$: вихід x.

Softmax застосовується на вихідному шарі, коли модель повинна вибрати один з кількох класів (наприклад, класифікація зображень), формула (2.11).

$$P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}, \tag{2.11}$$

де z_i — вагова сума для класу i;

n — кількість класів;

z_j — виходи всіх нейронів вихідного шару перед активацією.

Градієнтний бустинг (XGBoost, LightGBM, CatBoost) — це ансамблевий метод, що покращує точність прогнозування, об'єднуючи багато слабких моделей (зазвичай дерев рішень). Градієнтний бустинг оптимізує функцію втрат за допомогою градієнтного спуску. Метод складається з декількох етапів, нижче продемонстровані основні кроки [4].

Ініціалізація передбачень константним значенням. Починаємо з простої базової моделі наприклад, постійного значення — середнього значення для регресії або логарифма odds для класифікації, формула (2.12).

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma). \tag{2.12}$$

Ітеративне побудова нових моделей (дерев) для корекції помилок попередніх. На кожній ітерації обчислюємо градієнт функції втрат наприклад, MSE або логарифмічної втрати для кожного прикладу, формула (2.13).

$$r_i = -\frac{\partial L\left(y_i, F_{m-1}(x_i)\right)}{\partial F_{m-1}(x_i)}. \tag{2.13}$$

Навчаємо дерево рішень або інший простий алгоритм на цих залишках (градієнтах). Оновлення моделі з урахуванням коефіцієнта навчання, додаванням нової моделі до ансамблю з певним коефіцієнтом навчання η (learning rate). Повторюємо кроки 2 – 4 певну кількість ітерацій або до досягнення прийнятного рівня похибки, формула (2.14).

$$F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x). \quad (2.14)$$

Існує три основних бібліотек для реалізації градієнтного бустингу, а саме: XGBoost, LightGBM і CatBoost. Давайте порівняємо їх і оберемо найкращу для роботи з виявленням шахрайських транзакцій [6].
Порівняння бібліотек наведено в таблиці 2.1.

Таблиця 2.1 – Порівняння бібліотек бустингу

Характеристика	*XGBoost*	*LightGBM*	*CatBoost*
Швидкість	Висока, але повільніша за LightGBM.	Дуже швидкий завдяки Leaf-wise стратегії.	Швидкий, але трохи повільніший за LGBM.
Тип росту дерева	Level-wise.	Leaf-wise (більш гнучкий, швидкий).	Symmetric tree.
Підтримка категоріальних ознак	Ні (треба one-hot).	Частково (автообробка через параметри).	Так, нативна підтримка.
Модель за замовчуванням	Хороша, але потребує тюнінгу.	Добра, але чутлива до learning rate.	Дуже стабільна.
Параметри	Багато, потребує ретельного тюнінгу.	Менше, але чутлива до гіперпараметрів.	Просте налаштування.

Серед трьох найпопулярніших фреймворків градієнтного бустингу — XGBoost, LightGBM та CatBoost — саме CatBoost найкраще відповідає вимогам даної проблеми. Він дозволяє уникнути складного попереднього перетворення (наприклад, one-hot або label encoding), зберігаючи при цьому якість інформації. CatBoost менш чутливий до гіперпараметрів і менш схильний до переобучення. Це особливо важливо в умовах обмеженого часу

на моделювання, коли потрібно отримати добру якість швидко. Завдяки, інтерпретованості моделі (через аналіз важливості ознак, SHAP-значення тощо) є критично важливою у фінансовому секторі, де кожне автоматичне рішення повинно бути обґрунтованим для аудиту та дотримання нормативних вимог [11].

Таким чином, у задачі виявлення шахрайських транзакцій використання CatBoost дозволяє досягти високої точності, надійності, а також спростити процес розробки моделі. Це робить його оптимальним вибором як для досвідчених аналітиків, так і для команд, які працюють у режимі реального часу.

2.3 Обґрунтування вибору підходу

У цьому пункті ми розглянемо два підходи до побудови моделей машинного навчання для виявлення шахрайських транзакцій у фінансових системах: каскадну модель та модифікацію архітектури нейромережі. Для кожного з підходів ми проаналізуємо переваги та недоліки, визначимо їхні сильні сторони в контексті конкретних задач.

Каскадна модель — це підхід, при якому для розв'язання задачі використовується кілька етапів класифікації або прогнозування. Кожен етап фокусується на вирішенні конкретної частини задачі і поступово відсіює або уточнює результати, в залежності від специфіки кожного етапу [7]. Вона базується на принципі, що можна почати з простих рішень, а потім переходити до складніших, залежно від отриманих результатів на попередніх етапах. Це підходить для задач, де на перших етапах можна ефективно відсіяти очевидні, легко розпізнавані випадки, а більш складні випадки вимагають додаткового аналізу. Візуалізація каскадного підходу продемонстрована на рисунку 2.6.

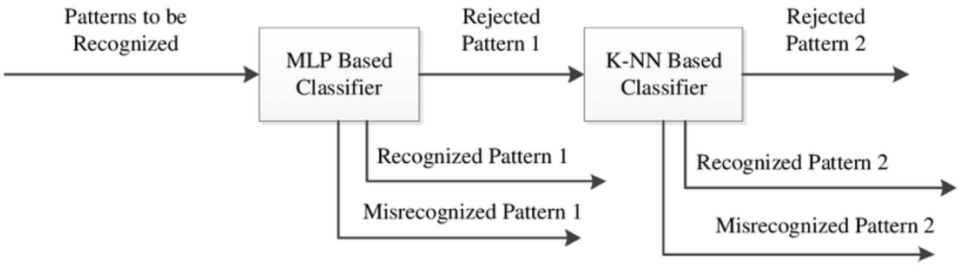


Рисунок 2.6 – Принцип роботи каскадного підходу[7]

Модифікація архітектури нейромережі передбачає вдосконалення існуючих моделей глибокого навчання з метою підвищення їхньої ефективності при розв’язанні конкретних прикладних задач. Такий процес може охоплювати як структурні зміни в самій архітектурі, так і оновлення методів її тренування. Наприклад, може здійснюватися впровадження нових типів нейронних мереж — зокрема згорткових мереж CNN для обробки просторових структур у зображеннях або рекурентних мереж RNN для аналізу послідовностей у часових рядах чи текстах. Завдяки цьому забезпечується здатність моделі краще виявляти й інтерпретувати складні, нелінійні закономірності в даних. Усі ці зміни спрямовані на підвищення точності прогнозування, стійкості до шуму в даних та зменшення витрат обчислювальних ресурсів при роботі з великими обсягами інформації.

У таблиці 2.2 представлено детальне порівняння між двома підходами до навчання, що дозволяє оцінити переваги та недоліки кожного з них у контексті вирішуваної задачі.

[7] Джерело зображення: https://ars.els-cdn.com/content/image/3-s2.0-B9780128167182000021X-gr002.gif

Таблиця 2.2 – Порівняння двох підходів навчання: каскадного і модифікація архітектури

Критерій	*Каскадна модель*	*Модифікація архітектури*
Складність моделі	Простота, менше етапів, але треба врахувати кілька класифікацій	Складніша архітектура, часто з великою кількістю шарів
Час тренування	Швидше через простіші етапи та меншу кількість параметрів	Повільніше, оскільки моделі можуть мати багато параметрів та складні архітектури
Простота налаштування	Легше налаштувати та адаптувати для задач, що включають відсівання легко розпізнаваних даних	Складніше налаштування та оптимізація нейромереж, потребує більше досвіду
Гнучкість у обробці складних даних	Обмежена: каскадні моделі можуть бути ефективними лише при простих шаблонах	Висока: нейромережі можуть виявляти складні взаємозв'язки в даних, особливо у великих обсягах
Застосування для виявлення шахрайства	Добре працює для розпізнавання шаблонів шахрайства з очевидними ознаками (наприклад, дуже великі суми або підозрілі платіжні реквізити)	Краще для виявлення складних аномалій, коли шахрайські дії не можна виявити лише за допомогою явних шаблонів
Інтерпретованість	Вища інтерпретованість, оскільки кожен етап моделі можна проаналізувати	Нейромережі важче інтерпретувати, особливо при глибинних моделях

Для кращого розуміння чи підходить той чи інший підхід у нашому випадку, давайте розглянемо технічні приклади та кейси.

Каскадну модель можна застосувати для виявлення шахрайських транзакцій у фінансових системах, де спочатку проводиться загальний

відсів очевидних «нормальних» транзакцій, а потім на кожному наступному етапі модель перевіряє більш складні транзакції. Наприклад, на першому етапі можна використовувати прості правила для відсіву великих транзакцій або тих, що відповідають певним шаблонам. На другому етапі можна застосувати складніші моделі для перевірки транзакцій з підозрілими реквізитами, а на третьому етапі — більш складні моделі для перевірки транзакцій з мінімальною інформацією, які вимагають більш ретельного аналізу.

Якщо ми розглядаємо модифікацію архітектури нейромережі, то вона може включати використання глибоких нейронних мереж для виявлення неочевидних шахрайських патернів, таких як нелінійні взаємозв'язки між різними атрибутами транзакцій. Наприклад, можна застосувати згорткові нейромережі для виявлення локальних патернів у даних транзакцій або рекурентні нейромережі для обробки послідовностей транзакцій та виявлення аномальних поведінкових патернів.

Після аналізу двох підходів можна зробити висновок, що каскадна модель є найбільш підходящим вибором для виявлення шахрайських транзакцій у фінансових системах. Вона дозволяє швидко та ефективно обробляти транзакції, поступово відсіюючи очевидні випадки та концентруючи ресурси на складніших. Крім того, каскадна модель є простішою в налаштуванні та інтерпретації, що є важливим аспектом у фінансових системах, де необхідно швидко реагувати на підозрілі транзакції. Модифікація архітектури нейромережі, хоча й може забезпечити високу точність при виявленні складних аномалій, потребує більше обчислювальних ресурсів та складнішої настройки. Тому каскадна модель виглядає оптимальним вибором для задачі виявлення шахрайства в умовах фінансових систем.

Висновки до розділу 2

У цьому розділі були розглянуті основи машинного навчання та їх застосування для виявлення аномалій, зокрема у фінансових системах. Початково було вивчено структуру моделей машинного навчання, включаючи вхідний, прихований та вихідний шари, що є основою для побудови ефективних моделей для виявлення аномальних або шахрайських транзакцій. Далі було здійснено огляд популярних алгоритмів, таких як логістична регресія, дерева рішень, випадковий ліс, градієнтний бустинг, які використовуються для класифікації та прогнозування. Відповідно по аналізу було обрано наступні моделі: Дерева ухвалення рішень, Випадковий ліс та Градієнтний бустинг.

На основі цього огляду, було обґрунтовано вибір підходу для вирішення задачі виявлення шахрайських транзакцій. Порівнявши каскадну модель та модифікацію архітектури нейромережі, було прийнято рішення про використання каскадної моделі, яка дозволяє ефективно обробляти транзакції, поступово відсіюючи очевидні випадки та зосереджуючи ресурси на більш складних. Каскадна модель є більш простою в налаштуванні та інтерпретації, що є важливим аспектом у контексті фінансових систем, де швидка реакція на підозрілі транзакції є критичною.

РОЗДІЛ 3 РОЗРОБКА МОДЕЛІ ДЛЯ ВИЯВЛЕННЯ ШАХРАЙСЬКИХ ТРАНЗАКЦІЙ

3.1 Підготовка та аналіз даних

Набір даних містить інформацію про транзакції, здійснені за допомогою кредитних карток у вересні 2013 року європейськими власниками карт. У вибірці представлені транзакції, що відбулися протягом двох днів, серед яких 492 випадки шахрайства із загальної кількості 284 807 транзакцій. Співвідношення шахрайських транзакцій до правильних продемонстровано на рисунку 3.1.

Рисунок 3.1 – Співвідношення шахрайських транзакцій до загальної кількості операцій

Даний набір є суттєво незбалансованим, оскільки позитивний клас (шахрайські транзакції) становить лише 0.172% від загальної кількості операцій. Набір даних містить лише числові вхідні змінні, які є результатом перетворення методом головних компонент (PCA). Всі ознаки, за винятком

двох, були трансформовані за допомогою цього методу. Матриця кореляції зображено нижче, на рисунку 3.2.

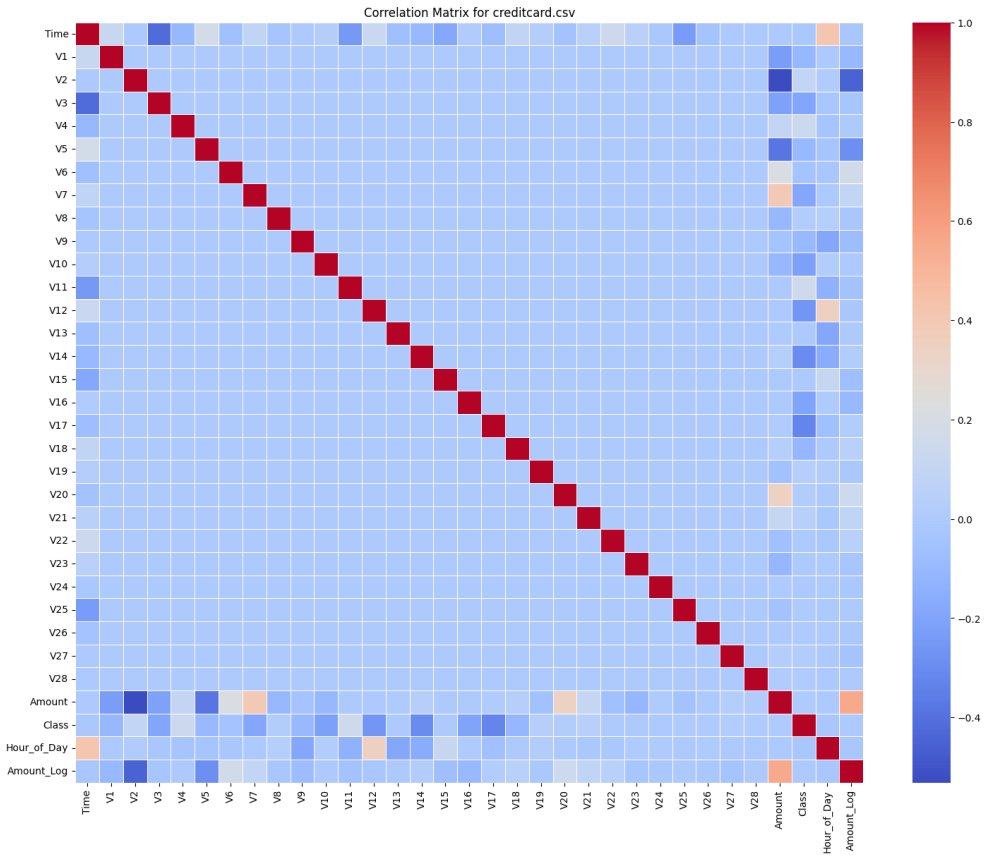


Рисунок 3.2 – Кореляційна матриця creditcard.csv

Ознаки V1, V2, … V28 – це головні компоненти, отримані після застосування PCA.

Ознака 'Time' – кількість секунд, що минули між кожною транзакцією та першою транзакцією у наборі даних.

Ознака 'Amount' – сума транзакції, яка може використовуватися для навчання моделі з урахуванням вартості конкретних операцій.

Ознака 'Class' – цільова змінна, що приймає значення 1 у випадку шахрайства та 0 для легітимних транзакцій.

Перед тренуванням моделі машинного навчання здійснюється обробка та балансування класів, що дозволяє покращити якість класифікації.

Видаляється цільова змінна Class із набору вхідних ознак. Дані розподіляються на тренувальний (80%) та тестовий (20%) набори, при цьому зберігається початковий розподіл класів. Оскільки кількість шахрайських транзакцій значно менша, використовується метод SMOTE (Synthetic Minority Over-sampling Technique) для генерації синтетичних зразків меншого класу. Це дозволяє досягти балансу у тренувальному наборі даних. До застосування SMOTE у тренувальній вибірці містилося 227 451 легітимна транзакція та лише 394 випадки шахрайства. Після балансування кількість прикладів кожного класу стала однаковою — по 227 451. Таким чином, було згенеровано 227 057 синтетичних зразків для класу шахрайських транзакцій. У результаті створено збалансовану вибірку, що дозволяє моделі ефективніше розпізнавати шахрайські дії, не втрачаючи точності на легітимних операціях.

У результаті виконаної підготовки даних було створено збалансовану вибірку, що дозволяє навченій моделі краще розпізнавати шахрайські транзакції, не втрачаючи при цьому точності на легітимних операціях. Це є важливим кроком для покращення загальної продуктивності класифікаційної моделі.

3.2 Архітектура моделі

Каскадна модель — це метод, у якому кілька моделей використовуються послідовно для етапного покращення класифікації. На кожному рівні моделі відсіюють частину даних, передаючи лише найбільш підозрілі транзакції на наступний етап.

У запропонованій каскадній архітектурі класифікація здійснюється поетапно, причому кожна модель виконує свою функцію.

1. Перший рівень – швидка фільтрація підозрілих транзакцій.

2. Другий рівень – уточнення прогнозів за допомогою більш складної моделі.
3. Третій рівень – остаточний аналіз найскладніших випадків.

На початковому рівні використовується Decision Tree Classifier як базова швидка модель для попереднього оцінювання ймовірності шахрайства. Легкість та швидкість побудови дерева забезпечують ефективну обробку великого масиву даних. Далі транзакції, передаються на наступний рівень для більш детальної обробки.

На другому рівні використовується більш складна модель – Random Forest, для підвищення точності класифікації. Модель обробляє лише транзакції, позначені як підозрілі на першому етапі. Використовується поріг (0.7) для відбору транзакцій. Завдяки ансамблю дерев модель забезпечує підвищену стійкість до шуму та покращену загальну узагальнювальну здатність.

Для найбільш складних та неоднозначних випадків використовується CatBoost Classifier – потужний градієнтний бустинг для обробки структурованих даних. Використовується поріг (0.8) для остаточного рішення. CatBoost ефективно працює з дисбалансом класів і категоріальними даними. Висока інтерпретованість та контроль над процесом навчання.

Отже каскадна модель дозволяє оптимізувати процес виявлення шахрайських транзакцій завдяки етапному аналізу та поєднанню різних алгоритмів: Decision Tree забезпечує швидке початкове фільтрування. Random Forest підвищує точність класифікації. CatBoost виконує фінальний аналіз найбільш складних випадків. Такий підхід оптимізує використання ресурсів та дозволяє підвищити точність виявлення шахрайства, мінімізуючи помилкові позитивні результати.

3.3 Вибір метриків для подальшого оцінювання результатів

У задачах виявлення шахрайських транзакцій у фінансових системах правильний вибір метрик оцінки моделі має критично важливе значення через специфіку проблеми. Дані у таких задачах зазвичай є високодисбалансованими, оскільки кількість нормальних транзакцій значно перевищує кількість шахрайських. Крім того, висока вартість помилки робить задачу ще складнішою — як пропуск шахрайської транзакції, так і хибне спрацьовування можуть мати серйозні фінансові та репутаційні наслідки. Саме тому при оцінці ефективності моделей у задачах виявлення шахрайства необхідно звертати увагу на такі метрики, як Precision, Recall, F1-score та AUPRC [14].

Precision (точність) — це частка транзакцій, які були класифіковані як шахрайські та насправді виявилися шахрайськими. Висока точність критична для фінансових систем, оскільки помилкове блокування легітимної транзакції може призвести до втрати клієнта через незадоволеність, негативного впливу на репутацію банку чи платіжної системи та юридичних наслідків у разі неправомірного блокування коштів. Висока Precision забезпечує мінімізацію кількості хибно позитивних спрацьовувань, що важливо для зниження кількості оскаржень від клієнтів і уникнення зайвих перевірок. Розрахунок цього параметру, вказано у формулі (3.1).

$$\text{Precision} = \frac{TP}{TP+FP}. \tag{3.1}$$

Recall (повнота) — це частка шахрайських транзакцій, які модель правильно ідентифікувала серед усіх фактичних шахрайських транзакцій. Висока повнота критична у виявленні шахрайства, оскільки пропуск шахрайської транзакції призводить до прямих фінансових втрат,

репутаційних ризиків і порушень регуляторних норм, що може спричинити серйозні санкції. Висока повнота забезпечує виявлення максимальної кількості шахрайських транзакцій, що допомагає зменшити фінансові втрати від пропуску шахрайства. Розрахунок цього параметру, вказано у формулі (3.2).

$$\text{Recall} = \frac{TP}{TP+FN}. \tag{3.2}$$

F1-score — це середньогармонійне значення між Precision та Recall. Високе значення F1-score свідчить про відмінний баланс між точністю та повнотою. Важливо отримати високе значення F1, оскільки висока Precision без високої Recall означає, що модель не пропускає шахрайські транзакції, але може пропускати частину з них, тоді як висока Recall без Precision призводить до великої кількості хибно позитивних спрацьовувань, що викликає незадоволеність клієнтів та зниження ефективності бізнесу. Таким чином, F1-score особливо важливий у задачах виявлення шахрайства через дисбаланс класів, оскільки дозволяє оцінити компроміс між виявленням шахрайства та збереженням точності класифікації. Розрахунок цього параметру, вказано у формулі (3.3).

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3.3}$$

AUPRC (Area Under Precision-Recall Curve) — це площа під кривою "Точність – Повнота", яка відображає здатність моделі зберігати високу точність при різних рівнях повноти. У задачах з дисбалансом класів стандартна ROC-крива (Receiver Operating Characteristic) може давати завищені оцінки якості через велику кількість негативних зразків. AUPRC є більш чутливою до ефективності виявлення рідкісного класу, тобто шахрайських транзакцій. Високе значення AUPRC означає, що модель

зберігає високу точність при високій повноті, що свідчить про ефективне виявлення шахрайських транзакцій без значної втрати точності. Розрахунок цього параметру, вказано у формулі (3.4).

$$\text{AUPRC} = \int \text{Precision}(r)dr. \tag{3.4}$$

У задачах виявлення шахрайства важливо мінімізувати пропуски шахрайських транзакцій (високий Recall), зберігати високу точність класифікації (високий Precision), а також збалансувати ці дві метрики для загальної продуктивності (F1-score). Метрика AUPRC показує, наскільки добре модель зберігає ефективність класифікації при зміні рівня чутливості та специфічності, що є ключовим для роботи з дисбалансом класів у фінансових транзакціях. Оптимальна модель повинна демонструвати високі значення Recall, Precision, F1 та AUPRC для забезпечення надійного та ефективного виявлення шахрайства у фінансових системах.

3.4 Реалізація моделі на Python

Код реалізує етапи побудови моделі для виявлення шахрайських транзакцій з використанням різних машинних навчальних підходів, таких як Decision Tree, Random Forest та CatBoost Classifier. Давайте розглянемо всі ключові етапи процесу.

У проєкті використовуються такі бібліотеки:

- pandas, numpy — для обробки та маніпуляції з даними;
- matplotlib, seaborn — для візуалізації даних;
- scikit-learn — для машинного навчання та оцінки моделей;
- imblearn — для боротьби з класовим дисбалансом за допомогою методу SMOTE;

– catboost — для побудови моделі градієнтного бустингу;
– warnings — для ігнорування зайвих попереджень під час виконання коду.

Набір даних завантажується за допомогою pandas.read_csv() з локального файлу. Дані містять інформацію про транзакції з кредитних карт, охоплюючи такі поля.

1. Time це час транзакції (в секундах від початку збору даних).
2. Amount це сума транзакції.
3. Class це мітка класу (0 — не шахрайство, 1 — шахрайство).

Формування нових ознак.

1. Hour_of_Day, перетворює час транзакції в годину дня.
2. Amount_Log, логарифмує суму транзакції для зменшення впливу великих значень та нормалізації даних.

Виконується базова візуалізація для розуміння структури даних, кількість шахрайських та не-шахрайських транзакцій.

1. Показує дисбаланс між класами.
2. Отримується матриця кореляції.
3. Застосовується логарифмічна шкала через сильний дисбаланс.
4. Розподіл сум транзакцій за класами.
5. Показує як суми транзакцій пов'язані з шахрайством.

Вибір ознак — всі колонки, окрім Class, використовуються як вхідні ознаки (X), а колонка Class — як мітка цільового класу (y).

Розділення на тренувальну та тестову вибірки — поділ на навчальну та тестову вибірки за допомогою train_test_split з розміром тестової вибірки 20%.

Балансування даних — використовується метод SMOTE для створення нових синтетичних зразків класу меншості та покращення якості навчання моделі.

На першому етапі використовується модель DecisionTreeClassifier, яка виконує базову класифікацію для відсіву більшої частини некоректних

транзакцій. Використовується як швидкий фільтр для усунення найбільш очевидних некоректних транзакцій. Модель приймає рішення на основі критеріїв розбиття ознак, прагнучи мінімізувати ентропію.

Вибір гіперпараметрів:

- criterion='entropy' — критерій для визначення чистоти вузла в дереві;
- min_samples_split=10 — мінімальна кількість зразків для поділу вузла;
- min_samples_leaf=2 — мінімальна кількість зразків у листовому вузлі;
- class_weight='balanced' — балансування ваг для врахування дисбалансу класів.

Основні функції:

- fit(X_train_res, y_train_res) — навчає модель на навчальній вибірці;
- predict(X_test) — прогнозує клас для тестових даних;
- predict_proba(X_test) — повертає ймовірності кожного класу;
- classification_report(y_test, y_pred) — генерує звіт про якість моделі (точність, recall, F1-score).

На другому етапі використовується RandomForestClassifier для подальшої фільтрації транзакцій після дерева рішень. Роль моделі полягає в комбінація кількох дерев рішень для створення ансамблевої моделі з кращою узагальнювальною здатністю. Покращує точність моделі шляхом зменшення перенавчання.

Вибір гіперпараметрів:

- n_estimators=100 — кількість дерев у лісі;
- random_state=42 — фіксація випадковості для відтворюваності результатів.

Основні функції:

- fit(X_train_res, y_train_res) — навчає модель випадкового лісу;

- predict(X_test_tree) — прогнозує класи для відфільтрованих транзакцій;
- predict_proba(X_test_tree) — повертає ймовірності для кожного класу;
- classification_report(y_test_tree, rf.predict(X_test_tree)) — оцінка якості моделі.

На третьому етапі використовується CatBoostClassifier — потужний алгоритм градієнтного бустингу на основі дерев рішень. Роль моделі полягає в тому, щоб вона виконує остаточну класифікацію на основі попередньо відібраних транзакцій. Модель має високу точність та здатність до узагальнення навіть за умов дисбалансу класів.

Вибір гіперпараметрів:

- iterations=500 — кількість ітерацій для навчання моделі;
- learning_rate=0.1 — швидкість навчання;
- depth=6 — глибина дерев у ансамблі;
- random_state=42 — фіксація випадковості для відтворюваності результатів.

Основні функції:

- fit(X_train_res, y_train_res) — навчає модель градієнтного бустингу;
- predict(X_test_rf) — прогнозує класи для остаточно відібраних транзакцій;
- predict_proba(X_test_rf) — повертає ймовірності для кожного класу;
- classification_report(y_test_rf, catboost_preds) — оцінка якості моделі.

Для оцінки роботи кожної моделі використовуються такі графіки та матриці. ROC-крива: відображає співвідношення між частотою хибних позитивів (FPR) та правильних позитивів (TPR). Показує ефективність класифікації. Матриця сплутаності (Confusion Matrix): допомагає оцінити

точність моделі за допомогою кількості хибних і правильних прогнозів. Precision-Recall крива: для оцінки точності та відгуку моделі на дисбалансованих даних.

3.5 Результати роботи моделі

На першому етапі каскадної моделі використовується DecisionTreeClassifier для первинного відсіювання більшості нешахрайських транзакцій. Нижче, наведено основні результати класифікації, а саме на рисунку 3.3.

Classification Report for DecisionTreeClassifier (Stage 1):

	precision	recall	f1-score	support
0	1.00	1.00	1.00	56864
1	0.47	0.81	0.59	98
accuracy			1.00	56962
macro avg	0.73	0.90	0.80	56962
weighted avg	1.00	1.00	1.00	56962

Рисунок 3.3 – Результати класифікації за допомогою DecisionTreeClassifier

Accuracy : 0.98 — модель демонструє високу загальну точність на всіх даних, що є дуже хорошим показником. Це свідчить про те, що модель здатна правильно класифікувати майже всі транзакції, але потрібно враховувати значний дисбаланс класів у вихідних даних.

Precision для класу 0: 1.00 — модель дуже добре класифікує клас 0 (основний клас). Для класу 1: 0.47 — відносно низька точність для шахрайських транзакцій, оскільки багато з них модель класифікує як нешахрайські. Це типова проблема при дисбалансі класів.

Recall для класу 0: 1.00 — модель добре знаходить приклади класу 0. Для класу 1: 0.81 — значно краща повнота для рідкісного класу, але все ще є місце для поліпшень.

F1-score для класу 0: 1.00 — відмінний результат для основного класу. Для класу 1: 0.59 — низький показник через слабку здатність до виявлення шахрайських транзакцій.

Візуалізація матриці сплутаності для DecisionTreeClassifier, зображено на рисунку 3.4.

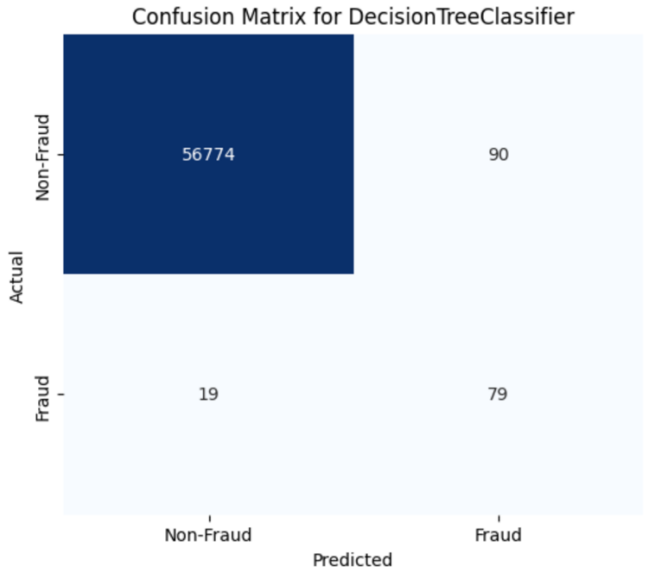


Рисунок 3.4 – Матриця сплутаності для DecisionTreeClassifier

Модель показує високу точність у виявленні нешахрайських транзакцій (56774 правильно класифікованих). Однак, вона має проблеми з виявленням шахрайських транзакцій (79 правильно класифікованих, але 19 помилково класифікованих як нешахрайські). На цьому етапі модель ефективно відсіює більшість нешахрайських транзакцій, але залишає багато шахрайських транзакцій, які потрібно виявити на наступних етапах. Графіки Precision-Recall та ROC Curve для DecisionTreeClassifier, зображено на рисунку 3.5.

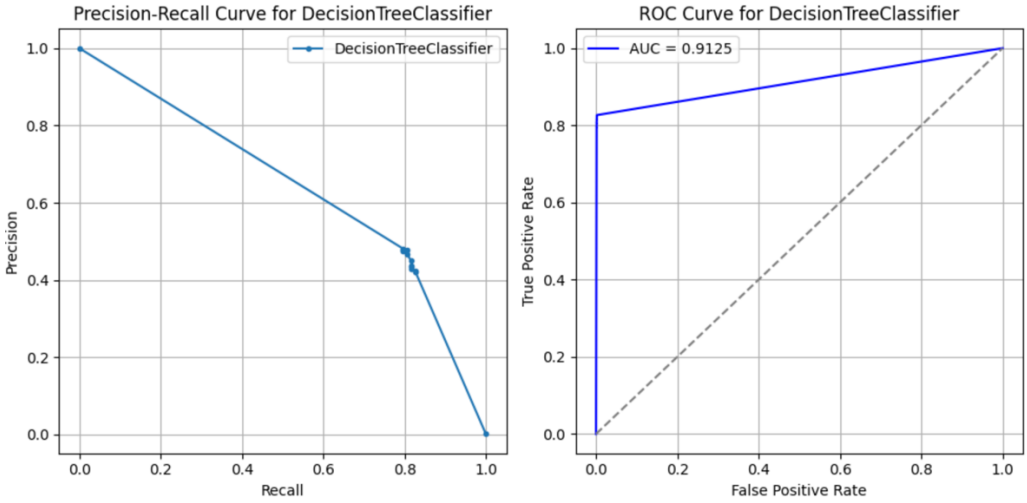


Рисунок 3.5 – Графіки Precision-Recall та ROC Curve для DecisionTreeClassifier

Крива демонструє, що модель досягає непоганої точності при низькій повноті. Однак із підвищенням рівня повноти точність помітно знижується. Це свідчить про компроміс між здатністю до виявлення шахрайських транзакцій і загальною точністю класифікації.

Площа під кривою (AUC) складає 0.9125, що вказує на високу здатність моделі відрізняти шахрайські транзакції від нешахрайських.

Високе значення AUC підтверджує, що модель добре працює при відокремленні класів, незважаючи на дисбаланс у даних.

На другому етапі каскадної моделі використовується Random Forest для покращення виявлення шахрайських транзакцій після первинного відсіювання моделлю DecisionTreeClassifier. Основні результати класифікації зображено на рисунку 3.6.

```
Classification Report for Random Forest (Stage 2):
              precision    recall  f1-score   support

           0       0.98      0.94      0.96        86
           1       0.94      0.97      0.96        79

    accuracy                           0.96       165
   macro avg       0.96      0.96      0.96       165
weighted avg       0.96      0.96      0.96       165
```

Рисунок 3.6 – Результати класифікації за допомогою Random Forest

Accuracy: 0.96 — модель показує дуже високу загальну точність на перевірочних даних. Вона працює дещо гірше, ніж DecisionTreeClassifier, для основного класу, але значно краще виявляє шахрайські транзакції.

Precision: для класу 0: 0.98 — модель дуже добре класифікує основний клас. Для класу 1: 0.94 — точність для рідкісного класу покращена порівняно з DecisionTreeClassifier.

Recall для класу 0: 0.94 — модель добре ідентифікує більшість нешахрайських транзакцій. Для класу 1: 0.97 — дуже добра повнота для шахрайських транзакцій, що показує, що модель ефективно виявляє майже всі позитивні випадки.

F1-score для класу 0: 0.96 — дуже хороший показник для основного класу. Для класу 1: 0.96 — суттєве покращення порівняно з попередньою моделлю, що свідчить про кращий баланс між точністю та повнотою.

Візуалізація матриці сплутаності для Random Forest, зображено на рисунку 3.7.

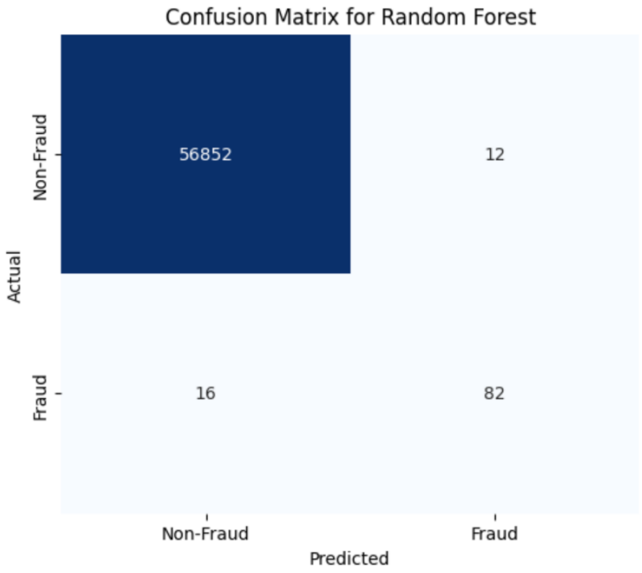


Рисунок 3.7 – Матриця сплутаності для Random Forest

Модель Random Forest демонструє дуже високу точність у класифікації "Не шахрайства", з великою кількістю істинно негативних результатів (56852) і дуже малою кількістю хибно позитивних (12). Модель також покращила виявлення "Шахрайства" порівняно з DecisionTreeClassifier, з 82 правильно класифікованими шахрайськими транзакціями та 16 помилково класифікованими як нешахрайські. Загалом, Random Forest показує значно кращу продуктивність, ніж DecisionTreeClassifier, у виявленні як нешахрайських, так і шахрайських транзакцій. Графіки Precision-Recall та ROC Curve для Random Forest, зображено на рисунку 3.8.

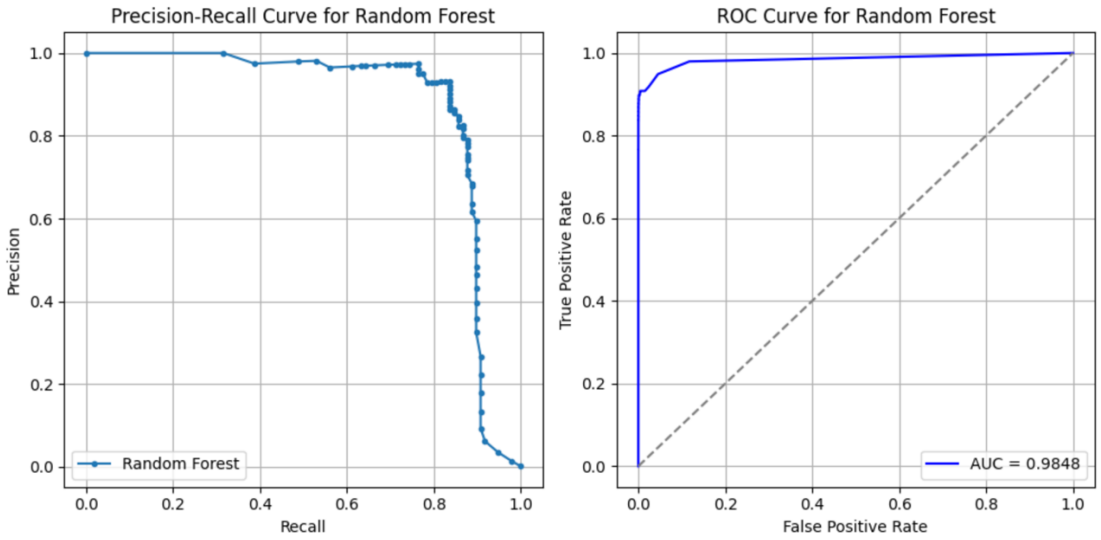


Рисунок 3.8 – Графіки Precision-Recall та ROC Curve для Random Forest

Модель демонструє хорошу точність навіть при високій повноті. Відсутнє різке зниження точності при збільшенні повноти, що вказує на стабільність роботи моделі у виявленні шахрайських транзакцій.

Площа під кривою (AUC) складає 0.9848, що вказує на майже ідеальну здатність моделі відокремлювати класи.

Високе значення AUC підтверджує, що модель добре справляється із завданням класифікації навіть у разі сильного дисбалансу класів.

На третьому етапі каскадної моделі використовується CatBoost для остаточного вдосконалення виявлення шахрайських транзакцій після попереднього відсіювання моделями DecisionTreeClassifier та Random Forest. Основні результати класифікації, зображено на рисунку 3.9.

```
Classification Report for CatBoost (Stage 3):
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         2
           1       0.97      1.00      0.99        69

    accuracy                           0.97        71
   macro avg       0.49      0.50      0.49        71
weighted avg       0.94      0.97      0.96        71
```

Рисунок 3.9 – Результати класифікації за допомогою CatBoost

Precision для класу 0: 0.00 — модель не змогла коректно класифікувати жодну нешахрайську транзакцію, що пояснюється дуже малою кількістю прикладів у цьому класі (всього 2 випадки). Для класу 1: 0.97 — висока точність, що свідчить про те, що 97% транзакцій, класифікованих як шахрайські, дійсно були такими.

Recall для класу 0: 0.00 — жодна нешахрайська транзакція не була ідентифікована правильно. Для класу 1: 1.00 — модель виявила всі шахрайські транзакції, без жодного пропущеного випадку.

F1-score для класу 0: 0.00 — через відсутність правильних класифікацій цей показник дорівнює нулю. Для класу 1: 0.99 — чудовий баланс між точністю та повнотою, що вказує на дуже ефективне виявлення шахрайських дій.

Візуалізація матриці сплутаності для CatBoost, зображено нижче на рисунку 3.10.

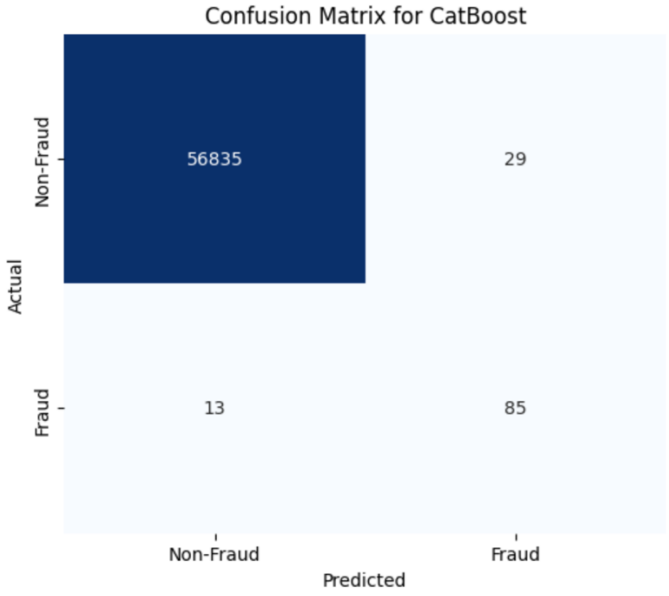


Рисунок 3.10 – Матриця сплутаності для CatBoost

Модель чудово ідентифікує нешахрайські транзакції — лише 29 хибнопозитивних випадків. Ідентифікація шахрайських транзакцій покращилася, правильно класифіковано 85 із 98 шахрайських транзакцій. Лише 13 шахрайських транзакцій залишаються невиявленими, що є найкращим результатом серед усіх моделей. Графіки Precision-Recall та ROC Curve для CatBoost, зображено на рисунку 3.11.

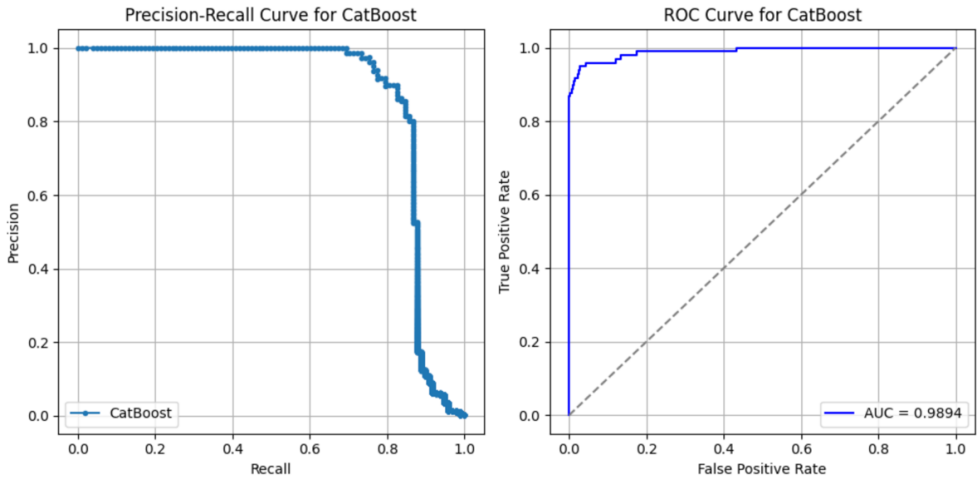


Рисунок 3.11 – Графіки Precision-Recall та ROC Curve для CatBoost

Модель демонструє високу точність навіть при високій повноті. Відсутнє значне зниження точності при збільшенні повноти, що підтверджує стабільність роботи моделі.

Площа під кривою (AUC) = 0.9894 — дуже високе значення, що свідчить про чудову здатність до класифікації.

ROC-крива наближається до ідеальної, що свідчить про відмінне відокремлення класів.

AUC (площа під ROC-кривою): 1.00 — модель досягла ідеального результату у відокремленні класів. AUC поступово покращувалась з кожною епохою і досягла максимального значення в кінці навчання.

Результати AUPRC для кожної із моделей, зображено на рисунку 3.12.

```
AUPRC for Logistic DecisionTreeClassifier: 0.6402
AUPRC for Random Forest: 0.9800
AUPRC for CatBoost: 0.9985
```

Рисунок 3.12 – Результати AUPRC для кожної із моделей

AUPRC показує, наскільки добре модель класифікує позитивний клас (шахрайство) у випадку незбалансованих класів. Високе значення вказує на хорошу здатність моделі виявляти шахрайські транзакції.

DecisionTreeClassifier (0.6402) — значно гірший результат через низьку точність у виявленні шахрайських транзакцій, але в цілому допустимо для першого шару.

Random Forest (0.9800) —показує чудову продуктивність, хоча трохи нижчу за видно що модель добре навчилась на основі фільтрації в DecisionTreeClassifier.

CatBoost (0.9985) — майже ідеальний результат, свідчить про те, що модель добре справляється з класифікацією навіть за сильно незбалансованих класів.

Висновки до розділу 3

Застосування каскадного підходу у виявленні шахрайських транзакцій у фінансових системах продемонструвало високу ефективність і доцільність такого рішення в умовах складного та дисбалансованого середовища. Кожен етап цього підходу виконує свою унікальну функцію: початкова модель бере на себе завдання швидкої фільтрації більшості очевидно «чистих» транзакцій, зменшуючи навантаження на наступні рівні аналізу. Подальші моделі, поступово звужуючи фокус, зосереджуються на дедалі більш підозрілих випадках, підвищуючи точність і впевненість у прийнятих рішеннях.

Завдяки такому поетапному аналізу, система демонструє здатність не лише виявляти потенційно шахрайські дії, а й робити це з урахуванням балансу між ефективністю, швидкістю та гнучкістю обробки даних. Каскадне розгортання моделей дозволяє уникнути ситуацій, коли цінна інформація втрачається через надто загальне або, навпаки, надмірно жорстке первинне фільтрування.

У підсумку, такий підхід формує надійну основу для побудови адаптивних та інтелектуальних систем протидії фінансовому шахрайству, які здатні не тільки реагувати на відомі загрози, а й ефективно адаптуватися до нових викликів у постійно змінному середовищі.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ ДЛЯ ВИЯВЛЕННЯ ШАХРАЙСЬКИХ ТРАНЗАКЦІЙ

4.1 Постановка задачі проектування

У цій роботі для проведення техніко-економічного аналізу застосовується метод функціонально-вартісного аналізу (ФВА). Об’єктом розробки є система виявлення шахрайських транзакцій у фінансових системах, побудована на основі каскадного підходу використання моделей машинного навчання.

Враховуючи, що кожен компонент розроблюваної системи істотно впливає на загальну продуктивність та якість прийняття рішень, усі модулі повинні відповідати встановленим технічним та експлуатаційним вимогам. Функціонально-вартісний аналіз передбачає оцінку функцій програмного продукту, спрямованого на збір, попередню обробку та багаторівневу класифікацію транзакцій для визначення ймовірності шахрайства. Основні технічні вимоги до системи.

1. Забезпечення коректної роботи на персональних комп’ютерах або серверних системах із базовими обчислювальними ресурсами.
2. Високу швидкість обробки фінансових транзакцій на етапі початкової фільтрації.
3. Високу точність класифікації завдяки каскадному використанню моделей різної складності.
4. Можливість масштабування системи відповідно до обсягу оброблюваних даних.
5. Зниження витрат на впровадження, налаштування та подальшу підтримку програмного продукту.

4.2 Обґрунтування технічного завдання

Розробка ефективної системи виявлення шахрайських транзакцій у фінансових системах вимагає ретельного підбору методів машинного навчання та їх організації у відповідну архітектуру. Враховуючи високу динамічність фінансових потоків, обмеженість обчислювальних ресурсів та необхідність оперативного прийняття рішень, було обрано каскадний підхід до побудови моделі.

Каскадна структура дозволяє оптимізувати розподіл обчислювальних витрат між різними рівнями аналізу. На першому рівні використовується Decision Tree Classifier — проста та швидка модель, яка дозволяє із мінімальними затратами обчислювальних ресурсів провести первинну фільтрацію транзакцій. Дерева рішень мають низьку латентність, тому ідеально підходять для обробки великих обсягів транзакцій у реальному часі. Їх основною перевагою є висока інтерпретованість і швидкість прийняття рішень, що є критичним на етапі первинного виявлення аномалій.

Другий рівень обробки використовує Random Forest Classifier — ансамблеву модель, що базується на великій кількості дерев рішень. Random Forest відзначається кращою точністю в порівнянні з окремим деревом завдяки процедурі "голосування" між моделями. На цьому етапі завданням є уточнення попередніх прогнозів: зниження кількості хибно позитивних та хибно негативних рішень. Застосування Random Forest забезпечує баланс між підвищенням точності і допустимим рівнем обчислювальних витрат, оскільки обробці підлягають лише попередньо відібрані підозрілі транзакції.

Третій рівень призначений для аналізу найбільш складних і суперечливих випадків. На цьому етапі використовується CatBoost Classifier — сучасна градієнтна бустингова модель, яка демонструє високу точність при роботі з табличними даними, особливо з категоріальними ознаками. На

відміну від Random Forest, CatBoost краще справляється з невеликими обсягами складних даних та менш схильний до переобучення. Використання CatBoost на фінальному етапі дозволяє забезпечити максимальну точність класифікації при мінімізації ризиків помилок у вирішальних випадках.

Як альтернативний варіант розробки можна було б застосувати однорівневу складу модель, наприклад, нейронних мереж або використання простого фільтраційного механізму без каскаду. Проте однорівневі складні моделі вимагають значно більше обчислювальних ресурсів навіть для обробки нескладних транзакцій, що робить їх непридатними для реального часу без значного підвищення вартості інфраструктури. З іншого боку, прості фільтраційні підходи не гарантують необхідного рівня точності, що може призвести до збитків через пропущені випадки шахрайства.

Таким чином, обґрунтованим є вибір саме каскадної архітектури, яка дозволяє таке.

1. Швидко обробляти більшість нормальних транзакцій з мінімальними витратами.
2. Ретельно перевіряти підозрілі випадки на подальших етапах.
3. Досягати високої загальної точності при оптимальному використанні ресурсів.

Отже, обрані алгоритми машинного навчання відповідають вимогам до швидкодії, точності, масштабованості та економічної доцільності впровадження системи у реальні фінансові процеси.

4.3 Економічний аналіз використання моделей в реальних умовах

Впровадження систем виявлення шахрайських транзакцій на основі моделей машинного навчання має пряму економічну доцільність, оскільки дозволяє суттєво зменшити фінансові збитки компаній, пов’язані з шахрайством, та оптимізувати витрати на обробку транзакцій. Для проведення економічного аналізу використовується підхід оцінки витрат і вигод (Cost-Benefit Analysis).

Основні економічні показники.

1. C_{int} — Витрати на впровадження системи.
2. C_{oper} — Річні операційні витрати на підтримку та обслуговування системи.
3. S_{ave} — Річна сума коштів, зекономлених завдяки виявленню та запобіганню шахрайським транзакціям.
4. P_e^{ff} — Ефективність системи виявлення шахрайства (у відсотках).
5. R_{oi} — Рентабельність інвестицій у систему.

Формула розрахунку чистого економічного ефекту та рентабельність інвестицій за рік, визначаються за формулами (4.1) та (4.2) відповідно.

$$E = S_{\text{ave}} \times P_{\text{eff}} - C_{\text{oper}}, \tag{4.1}$$

$$ROI = \frac{E - C_{\text{int}}}{C_{\text{int}}} \times 100\%. \tag{4.2}$$

Порівняно з використанням лише складної моделі (наприклад, нейронної мережі або CatBoost на всіх транзакціях), каскадний підхід має переваги які описані нижче.

Зниження обчислювальних витрат, як правило, моделі першого рівня (Decision Tree) працюють дуже швидко і обробляють 80-90% нормальних транзакцій без залучення ресурсомістких моделей.

Оптимізація витрат на інфраструктуру, зменшується потреба у дорогих серверних потужностях або хмарних обчисленнях.

Збільшення точності у критичних випадках за рахунок використання Random Forest та CatBoost лише для підозрілих транзакцій підвищується загальний рівень виявлення шахрайства без істотного навантаження на систему.

1. Річні витрати на підтримку — C_{oper} = 20 000 грн.
2. Впровадження системи коштує C_{int} = 50 000 грн.
3. Щорічні збитки від шахрайства без системи становлять 500 000 грн.
4. Система виявляє і запобігає 90% шахрайських випадків (P_e^{ff} = 0.9).

Тоді.

1. Save = 500 000 грн.
2. E = 500 000 × 0.9 – 20 000 = 430 000 грн.
3. ROI = $\frac{430\,000 - 50\,000}{50\,000}$ × 100% = 760%.

Отже, за перший рік роботи система приносить економічний ефект у 430 000 грн та забезпечує рентабельність інвестицій на рівні 760%, що свідчить про надзвичайно високу ефективність впровадження запропонованої архітектури.

4.4 Впровадження та підтримка програмного продукту

Інтеграція каскадної моделі для виявлення шахрайських транзакцій у фінансову систему є складним, але важливим етапом для забезпечення

високого рівня безпеки та мінімізації фінансових втрат. Впровадження моделі у реальну фінансову інфраструктуру вимагає врахування низки технічних та організаційних аспектів, щоб забезпечити високу продуктивність, надійність та швидкодію системи в умовах реального часу.

Першим кроком у процесі інтеграції є розгортання каскадної моделі у виробничому середовищі. Оскільки модель складається з трьох етапів (DecisionTreeClassifier → Random Forest → CatBoost), важливо забезпечити ефективний потік даних між кожною моделлю та налаштувати логіку послідовного виклику на основі отриманих результатів. Для цього модель можна реалізувати у вигляді мікросервісної архітектури, де кожна модель буде представлена як окремий сервіс із власним інтерфейсом виклику (API). Це дозволить масштабувати кожен етап незалежно від інших та оптимізувати обчислювальні ресурси [10]. Візуалізація мікросервісної архітектури, зображено на рисунку 4.1.

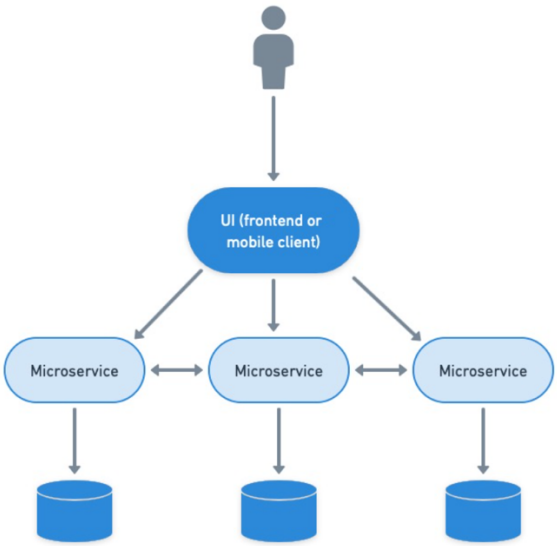


Рисунок 4.1 – Мікросервісна архітектура[8]

[8] Джерело зображення: https://miro.medium.com/v2/resize:fit:1400/format:webp/1*GNQWxkCS28CppGEU7m50Zg.png

При надходженні нової транзакції у фінансову систему перший рівень моделі (DecisionTreeClassifier) виконує базову класифікацію у режимі реального часу. Якщо транзакція класифікується як "нешахрайська" із високою впевненістю, вона одразу пропускається як безпечна і обробляється відповідною платіжною інфраструктурою. Якщо ж модель має сумніви щодо класифікації, транзакція передається на другий рівень (Random Forest), який здійснює більш точний аналіз, використовуючи глибшу структуру дерева рішень і більш складні шаблони. У разі невпевненості у класифікації навіть після другого рівня транзакція передається на третій рівень (CatBoost) — найпотужнішу модель, яка забезпечує максимальну точність у класифікації складних випадків завдяки використанню градієнтного бустингу на основі дерев рішень.

Реальна експлуатація каскадної моделі передбачає не лише ефективну класифікацію транзакцій, а й моніторинг її роботи, автоматичне донавчання та взаємодію з іншими системами банку. Успішна інтеграція такої системи вимагає комплексного підходу, що включає відстеження продуктивності, адаптацію до нових загроз та безперебійну комунікацію з банківськими платформами.

Перш за все необхідно реалізувати Моніторинг та оцінку продуктивності. Для забезпечення високої точності та стабільної роботи моделі необхідно реалізувати систему моніторингу, яка дозволить оцінювати її ефективність та швидко реагувати на можливі збої чи зниження продуктивності. Використання Prometheus та Grafana [3]. Ці інструменти забезпечують збір і візуалізацію метрик продуктивності моделі, таких як AUC, Precision (точність) та Recall (повнота). Вони дозволяють оперативно виявляти проблеми та вживати необхідних заходів.

Аналіз логів через ELK Stack (Elasticsearch, Logstash, Kibana). Використання централізованої системи логування дозволяє виявляти помилки, аналізувати підозрілі транзакції та отримувати аналітичні звіти в режимі реального часу.

Налаштування алертів. Автоматизовані сповіщення про критичні ситуації (наприклад, значне зниження точності моделі або підозріло висока кількість позитивних спрацьовувань) допоможуть швидко реагувати на можливі загрози. Візуалізація користувацького інтерфейсу сервісу Grafana, зображено на рисунку 4.2.

Рисунок 4.2 – UI вигляд сервісу Grafana[9]

Не менш важливо, завжди адаптуватися до нових шахрайських схем, адже шахраї постійно змінюють свої тактики, тому важливою складовою роботи моделі є її здатність до адаптації та донавчання на нових даних. Використання AutoML, автоматизоване налаштування гіперпараметрів дозволяє швидко оновлювати модель, не вимагаючи значних витрат часу на ручну оптимізацію. Щоб краще зрозуміти роботу AutoML, на рисунку 4.3, вказано як це працює.

[9] Джерело зображення: https://grafana.com/media/blog/export-visualizations/export_visualizations_sample_dashboard.png

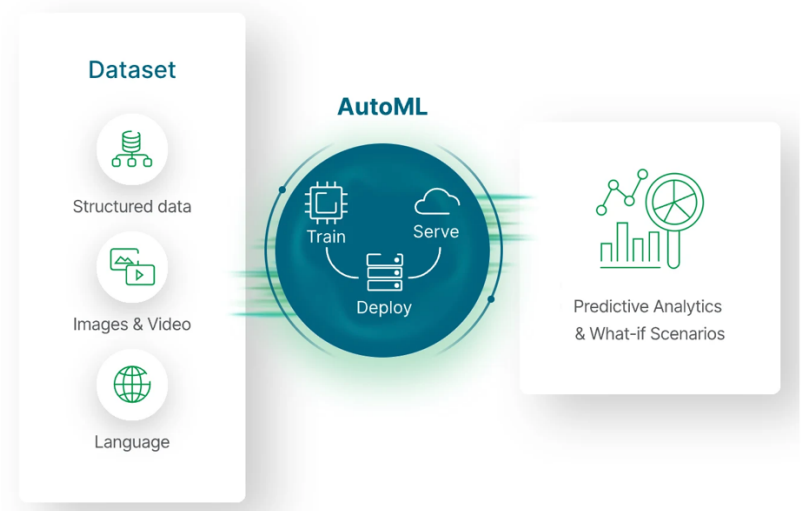


Рисунок 4.3 – Принцип роботи AutoML[10]

Також можна подумати про створення пісочниці для тестування нових версій моделі. Перед розгортанням оновлених моделей необхідно перевіряти їхню ефективність у спеціальному тестовому середовищі, що дозволить уникнути помилкових блокувань транзакцій реальних клієнтів. Також можно впровадити аналіз історичних даних. Постійне оновлення навчального набору даних на основі нових випадків шахрайства дозволяє моделі залишатися актуальною та ефективною.

Найбільш важливий етап це Взаємодія з іншими банківськими системами. Для забезпечення повної ефективності каскадна модель має інтегруватися з іншими банківськими сервісами та платформами. Платіжні шлюзи (Stripe, PayPal, Visa Secure). Завдяки інтеграції з платіжними провайдерами модель може оперативно виявляти шахрайські транзакції та запобігати їхньому виконанню в режимі реального часу.

Системи управління ризиками. Інтеграція з внутрішніми банківськими механізмами аналізу ризиків дозволяє автоматично блокувати підозрілі операції та відправляти їх на додаткову перевірку. Модулі

[10] Джерело зображення: https://res.cloudinary.com/talend/image/upload/w_1408/q_auto/qlik/glossary/augmented-analytics/seo-hero-automl_femm2t.webp

верифікації клієнтів (KYC, AML). Поєднання каскадної моделі з процедурами Know Your Customer (KYC) та Anti-Money Laundering (AML) дозволяє значно покращити рівень безпеки та запобігати фінансовим злочинам.

Таким чином, успішна реалізація каскадної моделі у банківській сфері вимагає продуманих механізмів моніторингу, адаптивності до нових загроз і тісної взаємодії з іншими фінансовими системами. Це дозволяє забезпечити високий рівень захисту транзакцій та ефективну боротьбу з шахрайством.

Незважаючи на високий потенціал каскадної моделі для виявлення шахрайських транзакцій, її впровадження у реальні банківські системи супроводжується низкою викликів. Це пов’язано як із технічними аспектами інтеграції та продуктивності, так і з високими вимогами до безпеки, конфіденційності та відповідності регуляторним нормам. Впровадження моделі вимагає ретельного опрацювання кожного етапу, починаючи від розподілу обчислювальних ресурсів і закінчуючи юридичними питаннями.

Одним із ключових викликів є забезпечення стабільної роботи моделі в умовах високого навантаження банківських систем. Щодня фінансові установи обробляють мільйони транзакцій, тому будь-яка затримка або збій у роботі моделі може спричинити негативні наслідки, зокрема невдоволення клієнтів або фінансові втрати. У цьому контексті особливо актуальним є питання продуктивності.

Банківські системи працюють у режимі реального часу, і будь-яке сповільнення в процесі обробки транзакцій може призвести до серйозних проблем, таких як зависання платежів, невчасне виконання операцій або навіть фінансові втрати. Високе навантаження на сервери та необхідність аналізу кожної транзакції вимагають ефективних механізмів масштабування та оптимізації.

Одним із рішень є розподіл навантаження між серверами за допомогою балансувальників навантаження, таких як HAProxy або Nginx,

які дозволяють рівномірно розподіляти запити між сервісами та запобігати перевантаженню окремих вузлів. Принцип роботи Nginx, вказано на рисунку 4.4.

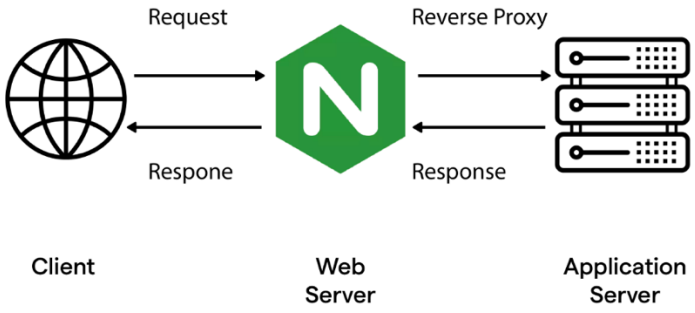


Рисунок 4.4 – Принцип роботи Nginx[11]

Окрім оптимізації програмних рішень, можна також застосовувати апаратні прискорювачі для пришвидшення обчислень. Використання графічних процесорів (GPU) або тензорних процесорів (TPU) дозволяє значно зменшити час обробки великих обсягів даних та прискорити процес класифікації транзакцій.

Захист даних є критично важливим фактором у банківських системах, оскільки будь-який витік інформації може призвести до серйозних фінансових і репутаційних втрат. Оскільки каскадна модель працює з чутливими фінансовими даними, її інтеграція вимагає суворого дотримання стандартів кібербезпеки.

Для захисту переданих даних широко використовуються протоколи шифрування, такі як TLS 1.3 та AES-256, які забезпечують захист інформації від несанкціонованого доступу під час її передачі через мережу. Крім того, необхідно забезпечити контроль доступу до моделі та

[11] Джерело зображення:
https://cdn.botpenguin.com/assets/website/NGINX_11e20532e2.webp

транзакційних даних. Для цього застосовуються захищені API з авторизацією на основі OAuth 2.0 або JWT-токенів, що гарантує, що лише авторизовані користувачі та системи можуть отримувати доступ до 56 критично важливих сервісів [9]. Структуру JWT токену, а також процес авторизації з використанням цієї технології, зображено на рисунках 4.5 і 4.6 відповідно.

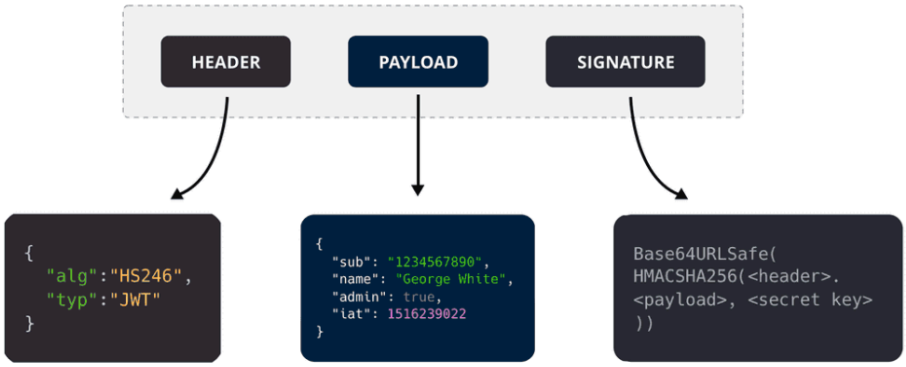


Рисунок 4.5 – Структура JWT токену[12]

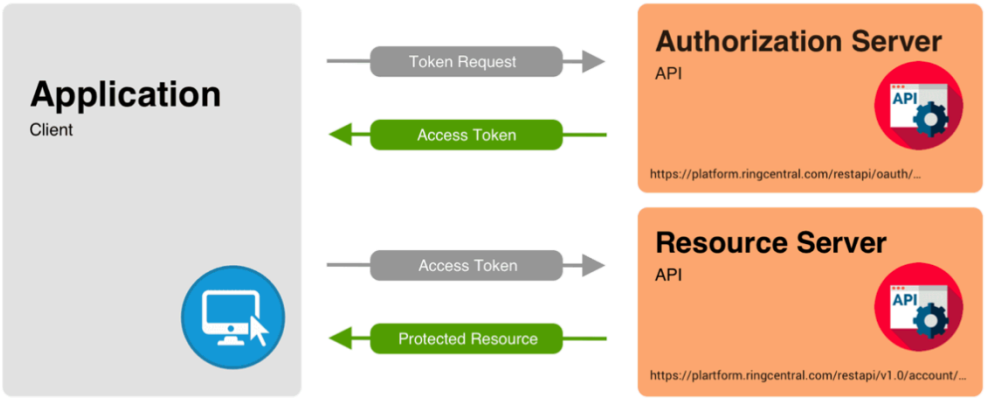


Рисунок 4.6 – Приклад авторизації через JWT[13]

[12] Джерело зображення: https://supertokens.com/static/b0172cabbcd583dd4ed222bdb83fc51a/9af93/jwt-structure.png
[13] Джерело зображення: https://assets-developers.ringcentral.com/dpw/guide/images/oauth-password-flow.png

Банківська діяльність суворо регулюється міжнародними та національними стандартами, тому будь-яка нова технологія, включаючи каскадну модель виявлення шахрайських транзакцій, повинна відповідати вимогам законодавства. Один із ключових нормативних актів, який регулює роботу з персональними даними, — це General Data Protection Regulation (GDPR), який встановлює жорсткі вимоги до обробки, зберігання та передачі даних клієнтів. Усі моделі машинного навчання, що працюють з фінансовими 57 транзакціями, повинні дотримуватися принципів прозорості, законності та цільового обмеження використання даних.

Ще одним важливим стандартом є PCI DSS (Payment Card Industry Data Security Standard), який визначає вимоги до захисту даних власників платіжних карток. Будь-яка система, що обробляє карткові транзакції, повинна забезпечувати захист даних від компрометації та мати механізми контролю доступу. Крім міжнародних стандартів, у різних країнах існують власні регуляторні вимоги, встановлені центральними банками та фінансовими наглядовими органами. Наприклад, у США це FFIEC (Federal Financial Institutions Examination Council), у Європі — EBA (European Banking Authority). Впровадження каскадної моделі у банківську систему має відповідати цим нормам, що може вимагати додаткових процедур сертифікації та аудиту.

Таким чином, інтеграція каскадної моделі в банківську систему — це не лише технічний процес, але й стратегічний виклик, що потребує комплексного підходу. Оптимізація продуктивності, дотримання вимог безпеки та регуляторних норм є ключовими факторами успіху впровадження цієї технології.

Висновки до розділу 4

У ході виконання цієї дипломної роботи було проведено детальне дослідження каскадної моделі та її застосування у фінансових системах, зокрема у сфері банківської безпеки. Було розглянуто всі аспекти інтеграції моделі, її використання в реальному середовищі, а також основні виклики, що виникають під час впровадження.

Було показано, що ефективне впровадження каскадної моделі потребує ретельного аналізу інфраструктури та вибору оптимальних рішень для її інтеграції. Важливими аспектами є вибір архітектури (мікросервісна або монолітна), налаштування API для взаємодії з іншими системами, а також використання сучасних баз даних для швидкого зберігання та обробки інформації. Підхід до інтеграції визначає не лише ефективність роботи моделі, а й її масштабованість та здатність адаптуватися до змін у бізнес-процесах.

Експлуатація моделі в реальному фінансовому середовищі вимагає постійного моніторингу її продуктивності та адаптації до нових викликів. Було встановлено, що моніторинг роботи моделі за допомогою Prometheus, Grafana та ELK Stack дозволяє оперативно виявляти аномалії та забезпечувати стабільну роботу. Крім того, автоматичне донавчання за допомогою AutoML та аналізу історичних даних є ключовим фактором у боротьбі з новими шахрайськими схемами. Також наголошено на важливості інтеграції моделі з іншими банківськими системами для ефективного управління ризиками та верифікації клієнтів.

Незважаючи на значні переваги каскадної моделі, її впровадження супроводжується рядом викликів. Основні труднощі включають проблеми продуктивності, що потребують масштабування інфраструктури та використання апаратних прискорювачів (GPU, TPU). Безпека та конфіденційність даних залишаються критичними питаннями, які

вирішуються шляхом шифрування, обмеженого доступу до моделі та дотримання принципів диференційної конфіденційності. Окремо розглянуто питання регуляторної відповідності, що вимагає дотримання стандартів GDPR, PCI DSS, ISO 27001 та вимог фінансових наглядових органів.

Таким чином, у роботі було проведено глубокий аналіз каскадної моделі у контексті фінансових систем. Дослідження підтвердило ефективність такого підходу для виявлення шахрайських транзакцій, але також виявило низку викликів, які необхідно враховувати при впровадженні. Успішна реалізація моделі залежить від комплексного підходу до її інтеграції, належного моніторингу, своєчасної адаптації та дотримання всіх стандартів безпеки та регуляторних вимог.

У майбутньому подальші дослідження можуть бути спрямовані на розширення функціоналу моделі, використання глибокого навчання для більш точних прогнозів, а також покращення механізмів автоматичного оновлення з урахуванням найновіших тенденцій у фінансовій сфері.

ВИСНОВКИ

У рамках цієї роботи комплексно досліджено проблему виявлення шахрайських транзакцій у фінансових системах за допомогою засобів машинного навчання. Проведено аналіз природи шахрайських операцій, сучасних підходів до боротьби з ними та можливостей автоматизованого виявлення таких транзакцій на основі алгоритмів класифікації. Зокрема, розглянуто переваги машинного навчання у контексті боротьби з фінансовим шахрайством, включаючи здатність до обробки великих обсягів даних, навчання на складних закономірностях і виявлення прихованих аномалій.

У ході роботи обґрунтовано доцільність використання каскадного підходу до побудови моделі виявлення шахрайства. Така стратегія дає змогу поступово уточнювати результати класифікації, підвищуючи точність виявлення підозрілих транзакцій і зменшуючи кількість хибнопозитивних результатів. Побудована модель охоплює три послідовні етапи: початкового аналізу даних із використанням простого Decision Tree Classifier, уточнення класифікації за допомогою ансамблевого методу Random Forest та фінального етапу, реалізованого на основі алгоритму градієнтного бустингу CatBoost, який продемонстрував високу ефективність на структурованих табличних даних.

Здійснено підготовку даних, а саме, проведено очистку, нормалізацію, балансування вибірки та побудову релевантних ознак. Обрано відповідні метрики оцінювання якості моделі, включаючи accuracy recall, F1 та AUC-ROC, що дозволило об'єктивно оцінити продуктивність розробленого рішення.

Отримані результати свідчать про успішну реалізацію каскадної архітектури, на кожному етапі вдалося досягти покращення класифікаційних показників, а фінальна модель CatBoost продемонструвала

найкращий баланс між точністю виявлення шахрайських транзакцій і стійкістю до перенавчання. Окрім технічних аспектів, у роботі також розглянуто економічну доцільність використання подібних систем у реальних умовах, а також аспекти впровадження і супроводу такого програмного забезпечення в рамках фінансових установ.

Таким чином, у результаті дослідження досягнуто поставленої мети — розроблено ефективну каскадну модель машинного навчання для виявлення шахрайських транзакцій, що може бути адаптована до реальних фінансових систем. Це уможливлює зниження ризиків фінансових втрат, покращення захисту користувачів і підвищення загальної ефективності моніторингу транзакцій.

ПЕРЕЛІК ПОСИЛАНЬ

1. Виявлення шахрайства у бізнесі та з платежами. *Payoneer*. URL: https://www.payoneer.com/uk/resources/general-payments/business-and-payment-fraud-detection-guide/ (дата звернення: 20.02.2025).

2. Шахрайство з банківськими картками: 4 способи захисту. *ZEN.COM*. URL: https://www.zen.com/ua/blog/personal-finance-uk/fraud-with-payment-cards/ (дата звернення: 20.02.2025).

3. Aleksandrov A. Getting Started with Grafana: Essential Tips for Beginners—Creating Dashboards, Connecting Data… *Medium*. URL: https://aleks-aleksandrov.medium.com/getting-started-with-grafana-essential-tips-for-beginners-creating-dashboards-connecting-data-94a70d019ab2 (дата звернення: 01.03.2025).

4. Alexander D. CatBoost Classifier: A Simple Guide for Everyone. *Medium*. URL: https://medium.com/@pwrxndr/catboost-classifier-a-simple-guide-for-everyone-48a2e3897251 (дата звернення: 07.03.2025).

5. Cascade Classifier. *APMonitor Optimization Suite*. URL: https://apmonitor.com/pds/index.php/Main/CascadeClassifier (дата звернення: 20.03.2025).

6. CatBoostClassifier. *CatBoost - open-source gradient boosting library*. URL: https://catboost.ai/docs/en/concepts/python-reference_catboostclassifier (дата звернення: 20.03.2025).

7. Dhadse A. Cascade Design Pattern. *adhadse*. URL: https://adhadse.com/cascade-design-pattern/ (дата звернення: 20.03.2025).

8. IBM. What Is Random Forest? | IBM. *IBM - United States*. URL: https://www.ibm.com/think/topics/random-forest (дата звернення: 01.04.2025).

9. JSON Web Tokens. *Auth0 Docs*. URL: https://auth0.com/docs/secure/tokens/json-web-tokens (дата звернення: 01.04.2025).

10. Kuncheva L. I. Combining Pattern Classifiers. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2014, ISBN: 9781118914564, doi: 10.1002/9781118914564.

11. Machine Learning, ML. *IT-Enterprise – your one-stop platform for digital transformation | www.it.ua*. URL: https://www.it.ua/knowledge-base/technology-innovation/machine-learning#:~:text=Машинне%20навчання%20(МО,%20Machine%20Learning,характеристики%20на%20основі%20отриманого%20досвіду». (дата звернення: 07.04.2025).

12. Olamendy J.C. Mastering the Cascade Design Pattern in ML/AI: Breaking Down Complexity into Manageable Steps. *Medium*. URL: https://medium.com/@juanc.olamendy/mastering-the-cascade-design-pattern-in-ml-ai-breaking-down-complexity-into-manageable-steps-98051f30fc48 (дата звернення: 07.04.2025).

13. RandomForestClassifier. *scikit-learn*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (дата звернення: 07.04.2025).

14. Rokach L. Ensemble Learning: Pattern Classification Using Ensemble Methods. World Scientific Publishing Co Pte Ltd, 2019. 300 c, ISBN: 9811201951.

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import SMOTE
from sklearn.metrics import precision_recall_curve, auc,
classification_report, confusion_matrix, roc_curve
from sklearn.ensemble import RandomForestClassifier
from catboost import CatBoostClassifier
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier
import warnings
warnings.filterwarnings("ignore")


data = pd.read_csv('/Users/apylypen/.cache/kagglehub/datasets/mlg-
ulb/creditcardfraud/versions/3/creditcard.csv')
print("Dataset loaded. Shape:", data.shape)


# Feature Engineering
data['Hour_of_Day'] = (data['Time'] // 3600) % 24  # Convert Time to
hours
data['Amount_Log'] = np.log1p(data['Amount'])
# --- Visualization 1: Fraud vs. Non-Fraud Count ---
plt.figure(figsize=(8, 5))
sns.countplot(x='Class', data=data)
plt.title("Fraud vs. Non-Fraud Transactions")
plt.xlabel("Class (0 = Non-Fraud, 1 = Fraud)")
plt.ylabel("Count")
plt.yscale('log')  # Log scale due to imbalance
plt.show()


# Розділення на ознаки та ціль
X = data.drop('Class', axis=1)
y = data['Class']

# Поділ на тренувальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Балансування даних за допомогою SMOTE
```

```
smote = SMOTE(random_state=42)
X_train_res, y_train_res = smote.fit_resample(X_train, y_train)


# Кількість прикладів до SMOTE
print("Before SMOTE:")
print(y_train.value_counts())

# Кількість прикладів після SMOTE
print("After SMOTE:")
print(y_train_res.value_counts())



# Обчислення кореляційної матриці
corr_matrix = data.corr()

# Візуалізація кореляційної матриці
plt.figure(figsize=(18, 14))
sns.heatmap(corr_matrix, cmap='coolwarm', annot=False, fmt='.2f',
linewidths=0.5)
plt.title('Correlation Matrix for creditcard.csv')
plt.show()


# 1. Етап 1: Decision Tree для первинної фільтрації
tree_model = DecisionTreeClassifier(
    criterion='entropy',
    min_samples_split=10,
    min_samples_leaf=2,
    class_weight='balanced',
    random_state=42
)

# Навчання моделі
tree_model.fit(X_train_res, y_train_res)

# Передбачення
y_pred = tree_model.predict(X_test)
y_prob = tree_model.predict_proba(X_test)[:, 1]


# Оцінка якості моделі
print("Classification Report for DecisionTreeClassifier (Stage 1):")
print(classification_report(y_test, y_pred))

# 2. Етап 2: Random Forest для подальшого фільтрування
```

```
threshold_tree = 0.5
mask_tree = y_prob >= threshold_tree
X_test_tree = X_test[mask_tree]
y_test_tree = y_test[mask_tree]

rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train_res, y_train_res)

# Прогнози для залишкових даних (передаем X_test_svm, а не y_test_svm)
rf_preds = rf.predict_proba(X_test_tree)[:, 1]

# Вибір транзакцій, які пройшли фільтрацію
print("Classification Report for Random Forest (Stage 2):")
print(classification_report(y_test_tree, rf.predict(X_test_tree)))


# 3. Етап 3: CatBoost для фінального виявлення шахрайських транзакцій
threshold_rf = 0.8
mask_rf = rf_preds >= threshold_rf
X_test_rf = X_test_tree[mask_rf]
y_test_rf = y_test_tree[mask_rf]

catboost = CatBoostClassifier(iterations=500, learning_rate=0.1,
depth=6, random_state=42, verbose=0)
catboost.fit(X_train_res, y_train_res)

# Прогнози для фінальних даних
catboost_preds = catboost.predict(X_test_rf)

print("Classification Report for CatBoost (Stage 3):")
print(classification_report(y_test_rf, catboost_preds))


# Оцінка результатів

# Обчислення AUPRC для кожної моделі
precision_log_reg, recall_log_reg, _ = precision_recall_curve(y_test,
tree_model.predict_proba(X_test)[:, 1])
auprc_log_reg = auc(recall_log_reg, precision_log_reg)

precision_rf, recall_rf, _ = precision_recall_curve(y_test_tree,
rf.predict_proba(X_test_tree)[:, 1])
auprc_rf = auc(recall_rf, precision_rf)

precision_catboost, recall_catboost, _ =
precision_recall_curve(y_test_rf, catboost.predict_proba(X_test_rf)[:,
1])
```

```
auprc_catboost = auc(recall_catboost, precision_catboost)

# Виведення AUPRC для кожної моделі
print(f"AUPRC for Logistic DecisionTreeClassifier:
{auprc_log_reg:.4f}")
print(f"AUPRC for Random Forest: {auprc_rf:.4f}")
print(f"AUPRC for CatBoost: {auprc_catboost:.4f}")


# Функція для побудови Confusion Matrix
def plot_confusion_matrix(cm, model_name):
    plt.figure(figsize=(6, 5))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False,
                xticklabels=['Non-Fraud', 'Fraud'],
                yticklabels=['Non-Fraud', 'Fraud'])
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.title(f'Confusion Matrix for {model_name}')
    plt.show()

# Функція для побудови Precision-Recall і ROC-кривої на одному полотні
def plot_combined_curves(y_test, y_prob, model_name):
    precision, recall, _ = precision_recall_curve(y_test, y_prob)
    fpr, tpr, _ = roc_curve(y_test, y_prob)
    roc_auc = auc(fpr, tpr)

    plt.figure(figsize=(10, 5))

    # Precision-Recall Curve
    plt.subplot(1, 2, 1)
    plt.plot(recall, precision, marker='.', label=f'{model_name}')
    plt.xlabel('Recall')
    plt.ylabel('Precision')
    plt.title(f'Precision-Recall Curve for {model_name}')
    plt.legend()
    plt.grid()

    # ROC Curve
    plt.subplot(1, 2, 2)
    plt.plot(fpr, tpr, color='blue', label=f'AUC = {roc_auc:.4f}')
    plt.plot([0, 1], [0, 1], color='grey', linestyle='--')
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title(f'ROC Curve for {model_name}')
    plt.legend()
    plt.grid()
```

```
    plt.tight_layout()
    plt.show()

# Моделі та передбачення
models = {
    "DecisionTreeClassifier": tree_model,
    "Random Forest": rf,
    "CatBoost": catboost
}

# Побудова графіків для кожної моделі
for model_name, model in models.items():
    y_pred = model.predict(X_test)
    y_prob = model.predict_proba(X_test)[:, 1]  # Ймовірності для класу
1

    # Confusion Matrix
    cm = confusion_matrix(y_test, y_pred)
    plot_confusion_matrix(cm, model_name)

    # Поєднана крива Precision-Recall + ROC
    plot_combined_curves(y_test, y_prob, model_name)
```

ДОДАТОК Б ЕЛЕКТРОННА ПРЕЗЕНТАЦІЯ

1. ОБ'ЄКТ ДОСЛІДЖЕННЯ

Фінансові транзакції в системах електронних платежів, зокрема процеси виявлення шахрайства в реальному часі за допомогою моделей машинного навчання

2. ПРЕДМЕТ ДОСЛІДЖЕННЯ

Моделі машинного навчання, зокрема каскадний підхід (Decision Tree, Random Forest, CatBoost). Досліджуються методи побудови таких моделей, їх точність, здатність до адаптації в умовах нових шахрайських схем та порівняння їх ефективності.

3. МЕТА ДОСЛІДЖЕННЯ

Розробка та впровадження моделей машинного навчання для ефективного виявлення шахрайських транзакцій у фінансових системах, використовуючи каскадний підхід, що дозволяє поетапно фільтрувати, уточнювати та аналізувати транзакції для підвищення точності виявлення шахрайства.

4. ПРОБЛЕМА

Проблема дослідження полягає в необхідності розробки ефективних методів виявлення шахрайських транзакцій, що враховують різноманітність шахрайських схем і забезпечують високий рівень точності при обробці великих обсягів транзакцій у реальному часі.

5. ЗАВДАННЯ

1. Зібрати відповідний набір даних для навчання моделей та виявлення шахрайських транзакцій.
2. Розробити архітектуру моделі, яка найбільш точно відповідає поставленій задачі.
3. Навчити модель в кілька етапів та підібрати найкращі гіперпараметри для кожної з моделей.
4. Провести експериментальне тестування та оцінити результати моделі.
5. Розробити інтерфейс користувача з інтегрованою системою.

ОБРАНІ ТЕХНОЛОГІЇ

МОВА ПРОГРАМУВАННЯ

Python, Середовище розробки: Jupyter Notebook

БІБЛІОТЕКИ

1. pandas, numpy — для обробки та маніпуляції з даними.
2. matplotlib, seaborn — для візуалізації даних.
3. scikit-learn — для машинного навчання та оцінки моделей (train_test_split, classification_report, confusion_matrix, roc_curve тощо).
4. imblearn — для боротьби з класовим дисбалансом за допомогою методу SMOTE.
5. CatBoost — для побудови моделі градієнтного бустингу.
6. warnings — для ігнорування зайвих попереджень під час виконання коду.

КАСКАДНИЙ ПІДХІД

ШВИДКА ФІЛЬТРАЦІЯ

Дозволяє використовувати простіші та швидші моделі, на першому етапі для попередньої фільтрації транзакцій. Це дозволяє швидко відсіяти очевидно не шахрайські транзакції, знижуючи навантаження на більш складні моделі на наступних етапах

ВИСОКА ТОЧНІСТЬ

Завдяки більш складних моделей, на наступних рівнях каскадного підходу, система може більш точно виявляти шахрайські транзакції, які не були відфільтровані на першому етапі. Це дозволяє поетапно уточнювати прогнози, досягаючи більш високої точності.

ЕФЕКТИВНІСТЬ НА ВЕЛИКИХ ДАНИХ

Підхід дозволяє розподілити навантаження на різні моделі за етапами. На першому етапі обробляються всі транзакції, а на більш складних етапах лише ті, що не були відсіяні. Це дозволяє знизити загальний час обробки без втрати точності.

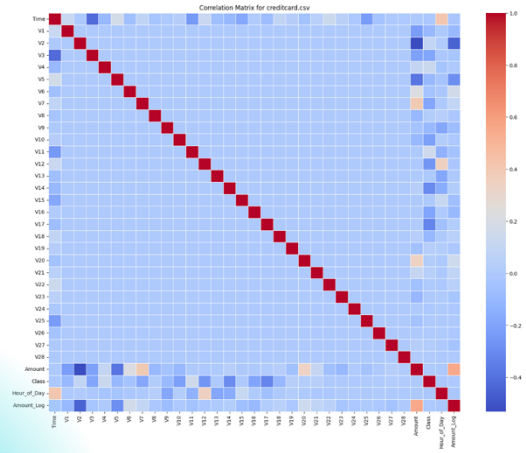
ОГЛЯД ДАТАСЕТУ

У дослідженні використано набір даних, опублікований на платформі **Kaggle**, що містить інформацію про транзакції, здійснені за допомогою кредитних карток європейськими користувачами у вересні 2013 року. Усього в наборі міститься **284807** записів, що охоплюють транзакції, здійснені протягом двох днів. Із них **492** транзакції (приблизно **0,17%**) класифіковані як шахрайські, що вказує на наявність суттєвого дисбалансу класів.

Особливістю цього датасету є попереднє застосування методу головних компонент (PCA) до більшості ознак — змінні **V1–V28** є результатами цього перетворення, що забезпечує анонімізацію та зменшення розмірності. Крім того, присутні змінні:

- Time — кількість секунд, що минули відносно першої транзакції;
- Amount — сума транзакції;
- Class — цільова змінна, яка приймає значення 1 у випадку шахрайства та 0 у випадку легітимної транзакції.

ПОПЕРЕДНЯ ОБРОБКА ДАНИХ



Перед навчанням моделей машинного навчання було здійснено попередню обробку даних, яка включала розподіл вибірки, де **80%** даних було відведено для тренування, а **20%** — для тестування, зі збереженням початкового співвідношення класів.

Оскільки кількість шахрайських транзакцій була значно меншою за кількість легітимних, для досягнення балансу між класами було застосовано метод **SMOTE**, що дозволив згенерувати **227057** синтетичних прикладів для меншого класу. У результаті тренувальна вибірка була збалансована і містила по 227451 прикладу для кожного з класів. Також було побудовано **кореляційну матрицю**, що дозволила візуалізувати зв'язки між ознаками та виявити найбільш інформативні змінні.

Такий підхід забезпечив значне покращення якості навчання моделей, оскільки дозволив їм ефективніше ідентифікувати рідкісні випадки шахрайства без втрати точності для основної маси легітимних транзакцій.

АРХІТЕКТУРА МОДЕЛІ

DECISION TREE CLASSIFIER

На початковому рівні використовується базова, швидка модель для попереднього оцінювання ймовірності шахрайства. Легкість та швидкість побудови дерева забезпечують ефективну обробку великого масиву даних.

RANDOM FOREST

На другому рівні використовується більш складна модель, для підвищення точності класифікації. Модель обробляє лише транзакції, позначені як підозрілі на першому етапі. Завдяки ансамблю дерев модель забезпечує підвищену стійкість до шуму та покращену загальну узагальнювальну здатність.

CATBOOST CLASSIFIER

Для найбільш складних та неоднозначних випадків використовується потужний градієнтний бустинг для обробки структурованих даних. Модель ефективно працює з дисбалансом класів і категоріальними даними.

ВИБІР ГІПЕРПАРАМЕТРІВ: DECISION TREE CLASSIFIER

CRITERION='ENTROPY'

Критерій "ентропія" обрано замість стандартного Gini, оскільки він краще підходить для задач з дисбалансом класів. Ентропія більш чутлива до неоднорідності, а отже, сприяє кращому поділу рідкісного шахрайського класу.

MIN_SAMPLES_LEAF=2

Мінімальна кількість прикладів у листі — 2 — дозволяє уникнути створення занадто специфічних вузлів, які можуть запам'ятовувати одиничні приклади, характерні лише для тренувального набору.

MIN_SAMPLES_SPLIT=10

Це значення дозволяє уникнути надмірного розгалуження дерева. Зменшуючи ризик переобучення, модель зосереджується на стабільних закономірностях, що критично важливо в умовах сильно дисбалансованих даних.

CLASS_WEIGHT='BALANCED'

Автоматичне зважування класів забезпечує адекватний вплив шахрайських транзакцій на функцію втрат. Це особливо важливо при суттєвому дисбалансі (менше 0.2% шахрайств).

ВИБІР ГІПЕРПАРАМЕТРІВ: RANDOM FOREST CLASSIFIER

N_ESTIMATORS=100

Вибір 100 дерев забезпечує достатню гнучкість і точність моделі без надмірного часу обчислення. Це емпірично перевірене значення, яке добре працює для більшості задач класифікації, включно з фінансовим шахрайством.

RANDOM_STATE=42

Фіксація початкового стану генератора випадкових чисел гарантує відтворюваність результатів, що особливо важливо для аналізу та порівняння ефективності моделей у дослідженні.

CATBOOST CLASSIFIER

ITERATIONS=500

Кількість ітерацій забезпечує достатню кількість циклів для ефективного навчання без перенавчання. Для задачі шахрайства, де необхідно уловити складні закономірності, це число забезпечує хороший баланс між якістю та швидкістю.

LEARNING_RATE=0.1

Оптимальний крок градієнтного спуску. Надто високі значення можуть призвести до стрибкоподібного навчання, а надто малі — до повільної збіжності. 0.1 — це широко визнане стандартне значення, яке забезпечує стабільне навчання.

DEPTH=6

Обмеження глибини дерев до 6 дозволяє моделі вловити складні взаємозв'язки між ознаками, не створюючи надто складної структури, що могла б призвести до перенавчання.

РЕЗУЛЬТАТИ РОБОТИ

Classification Report for DecisionTreeClassifier (Stage 1):

	precision	recall	f1-score	support
0	1.00	1.00	1.00	56864
1	0.47	0.81	0.59	98
accuracy			1.00	56962
macro avg	0.73	0.90	0.80	56962
weighted avg	1.00	1.00	1.00	56962

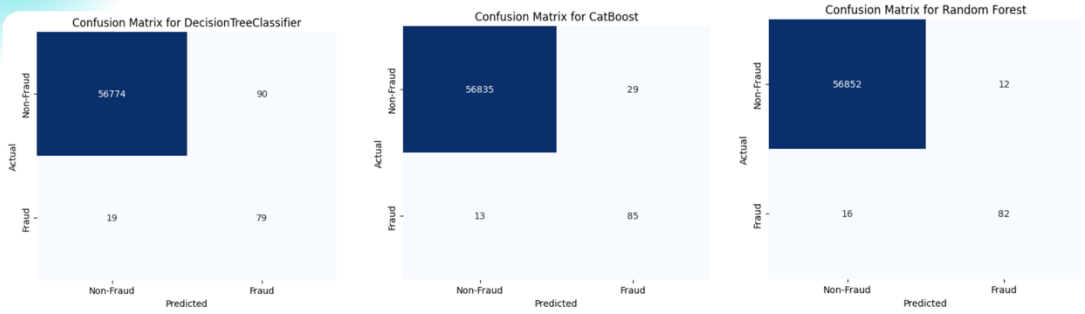
Classification Report for Random Forest (Stage 2):

	precision	recall	f1-score	support
0	0.98	0.94	0.96	98
1	0.93	0.97	0.95	80
accuracy			0.96	178
macro avg	0.95	0.96	0.95	178
weighted avg	0.96	0.96	0.96	178

Classification Report for CatBoost (Stage 3):

	precision	recall	f1-score	support
0	0.00	0.00	0.00	3
1	0.96	1.00	0.98	74
accuracy			0.96	77
macro avg	0.48	0.50	0.49	77
weighted avg	0.92	0.96	0.94	77

РЕЗУЛЬТАТИ РОБОТИ

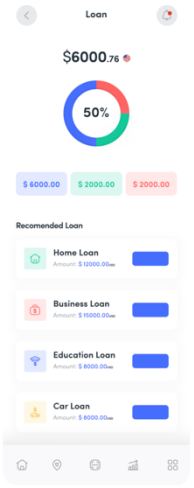


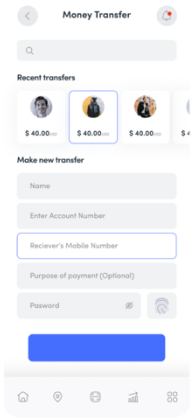
Загалом, завдяки такому поетапному аналізу, система демонструє здатність не лише виявляти потенційно шахрайські дії, а й робити це з урахуванням балансу між ефективністю, швидкістю та гнучкістю обробки даних. Каскадне розгортання моделей дозволяє уникнути ситуацій, коли цінна інформація втрачається через надто загальне або, навпаки, надмірно жорстке первинне фільтрування.

MPV ПРОДУКТУ

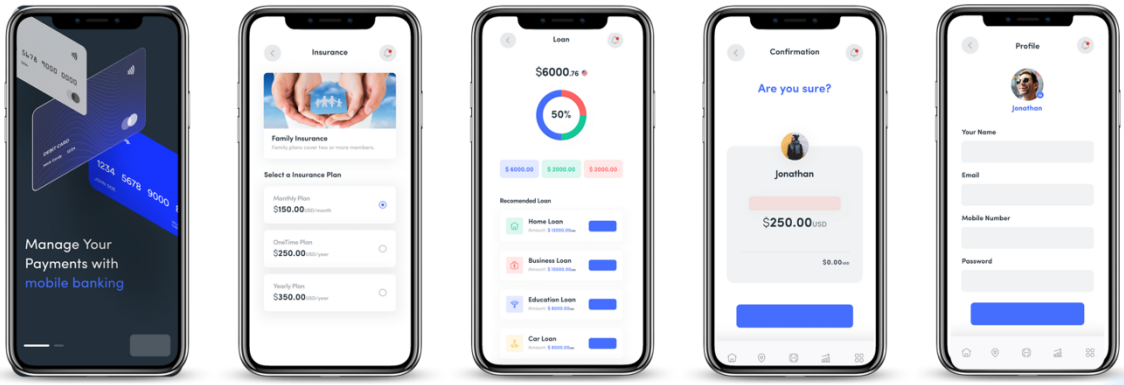
СПИСОК МІНІМАЛЬНИХ ФУНКЦІЙ ПРОДУКТУ

1. Система реєстрації та авторизації користувачів
2. Візуалізація результатів (Dashboard)
3. Налаштування персональних даних користувача
4. Плани підписки
5. Інтеграція платіжних систем (Оплата підписок через Stripe або PayPal.)
6. Сповіщення про про потенційні загрози





ІНТЕРФЕЙС ДОДАТКУ



ВИСНОВКИ

- ДОСЛІДЖЕННО СИТУАЦІЮ З ШАХРАЙСТВОМ У ФІНАНСОВИХ СИСТЕМАХ
- РОЗРОБЛЕНО КЛЮЧОВИЙ ФУНКЦІОНАЛ ДЛЯ РОЗПІЗНАВАННЯ ШАХРАЙСЬКИХ ТРАНЗАКЦІЙ
- ПРОАНАЛІЗОВАНІ ТА ПОРІВНЯННІ МЕТОДИ ТА ПІДХОДИ ШІ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ
- РОЗРОБЛЕНО МІНІМАЛЬНИЙ НАБІР ФУНКЦІЙ І UI/UX ПРОДУКТУ