ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Штучний інтелект у виявленні шахрайських транзакцій у фінансових установах»

на здобуття освітнього ступеня магістр
зі спеціальності 124 Системний аналіз
освітньо-професійної програми Інтелектуальні системи управління

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Денис Зарніцин
(підпис) *(ім'я, ПРІЗВИЩЕ здобувача)*

Виконав:
здобувач вищої освіти
група САДМ-61 Денис Зарніцин
(ім'я, ПРІЗВИЩЕ)
Керівник Михайло Кузьміч
Доцент кафедри ІСТ *(ім'я, ПРІЗВИЩЕ)*
Рецензент: _____
(ім'я, ПРІЗВИЩЕ)

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій
Ступінь вищої освіти магістр
Спеціальність 124 Системний аналіз
освітньо-професійної програми Інтелектуальні системи управління

ЗАТВЕРДЖУЮ
Завідувач кафедрою ІСТ
_____ Каміла СТОРЧАК

«____» _____2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Зарніцин Денис Володимирович
(прізвище, ім'я, по батькові здобувача)

1.Тема кваліфікаційної роботи: Штучний інтелект у виявленні шахрайських транзакцій у фінансових установах
керівник кваліфікаційної роботи: Михайло Кузьміч к.т.н., доцент кафедри ІСТ
(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» жовтня 2025 р. № 467

2. Строк подання кваліфікаційної роботи «26» грудня 2025 р.

3. Вихідні дані кваліфікаційної роботи

1. Сучасні методи штучного інтелекту (AI), зокрема алгоритми машинного навчання (Logistic Regression, Random Forest, Gradient Boosting) та глибинного навчання (MLP, RNN), що застосовуються для аналізу та оброблення транзакційних даних.
2. Технології роботи з великими масивами даних (Big Data), включно з методами потокової обробки даних у реальному часі та техніками корекції дисбалансу вибірки (SMOTE, undersampling).
3. Концепції забезпечення фінансової кібербезпеки (антифрод), які включають моделі поведінкового аналізу транзакцій, системи виявлення аномалій та метрики оцінювання ефективності AI-моделей (Precision, Recall, F_1-score, ROC-AUC).

4. Відкриті фінансові транзакційні набори даних, зокрема Credit Card Fraud Detection Dataset, що використовуються для навчання та тестування моделі.
5. Науково-технічні джерела інформації: матеріали конференцій (IEEE), статті з баз Scopus і Web of Science та монографії у галузях штучного інтелекту, Data Science та фінансової безпеки.

4. Зміст розрахунково-пояснювальної записки

1. Дослідити теоретичні основи застосування штучного інтелекту у фінансовому моніторингу; здійснити огляд сучасних антифрод-систем та визначити актуальні проблеми виявлення шахрайських транзакцій (дисбаланс класів, складність виявлення аномалій, адаптивність шахрайських схем).
2. Розробити методику підготовки та аналізу фінансових даних, включаючи вибір, очищення, нормалізацію та розширений feature engineering транзакційних ознак, а також обґрунтувати застосування методів корекції дисбалансу вибірки.
3. Розробити, навчити та оптимізувати гібридну AI-модель, що поєднує методи машинного навчання та нейронні мережі, сформувавши ансамблевий підхід для підвищення точності та стійкості класифікації.
4. Провести експериментальне оцінювання ефективності розробленої AI-моделі за допомогою метрик F_1-score, Recall, Precision і ROC-AUC; здійснити порівняння з альтернативними моделями.
5. Сформувати рекомендації щодо впровадження AI-моделі у фінансові системи з урахуванням вимог кіберстійкості, а також можливістю застосування пояснюваного штучного інтелекту (Explainable AI, XAI) для підвищення прозорості прийняття рішень.

5.Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання «30» жовтня 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Підбір, систематизація та аналітичний огляд науково-технічної літератури, джерел даних і сучасних підходів до виявлення фінансового шахрайства	08.11.2025	Виконано
2.	Дослідження теоретичних засад використання методів штучного інтелекту	15.11.2025	Виконано

	(AI) у фінансових антифрод-системах; визначення проблем і завдань дослідження		
3.	Підготовка даних: очищення, нормалізація та feature engineering транзакційних ознак; усунення дисбалансу вибірки (SMOTE)	22.11.2025	Виконано
4.	Розроблення та навчання гібридної AI-моделі (Random Forest + XGBoost + MLP); оптимізація параметрів і формування ансамблевого підходу	03.12.2025	Виконано
5.	Оцінювання ефективності розробленої моделі за метриками F_1-score,Recall,Precision,ROC-AUC; порівняння з альтернативними алгоритмами; формування аналітичних висновків	10.12.2025	Виконано
6.	Формулювання загальних висновків і рекомендацій; оформлення розрахунково-пояснювальної записки (РПЗ) відповідно до вимог ДСТУ	18.12.2025	Виконано
7.	Підготовка демонстраційних матеріалів (презентація, доповідь, ілюстрації); остаточне редагування та подання магістерської роботи до захисту	26.12.2025	Виконано

Здобувач вищої освіти _____ Денис Зарніцин
(підпис) *(ім'я, ПРІЗВИЩЕ)*
Керівник кваліфікаційної роботи _____ Михайло Кузьміч
(підпис) *(ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина магістерської кваліфікаційної роботи: 106 стор., 14 рис., 13 табл., 48 джерел.

Мета роботи – розроблення, навчання та валідація гібридного підходу на основі методів штучного інтелекту для автоматизованого виявлення шахрайських транзакцій у фінансових установах в умовах суттєвого дисбалансу вибірки.

Об’єкт дослідження – процес виявлення шахрайських транзакцій у фінансових системах із застосуванням інтелектуальних методів аналізу даних.

Предмет дослідження – алгоритми машинного навчання та глибинного навчання (ансамблеві моделі та штучні нейронні мережі), а також методи оброблення незбалансованих фінансових даних у задачах класифікації транзакцій.

Короткий зміст роботи.

У першому розділі проведено аналітичний огляд сучасних кіберзагроз у фінансовому секторі та типових схем шахрайства в цифровому банкінгу (CNP-операції, захоплення облікових записів, відмивання коштів). Розглянуто еволюцію систем фінансового моніторингу – від статичних rule-based-рішень до адаптивних AI-систем – та проаналізовано можливості алгоритмів машинного навчання (Random Forest, XGBoost, MLP) для задач бінарної класифікації транзакцій. Особливу увагу приділено проблемі критичного дисбалансу класів і вимогам нормативно-правового регулювання та Explainable AI у фінансовому секторі.

У другому розділі запропоновано методику підготовки транзакційних даних для побудови антифродмодель. Описано розвідувальний аналіз обраного датасету, очищення, масштабування та нормалізацію ознак, а також розроблення розширених поведінкових характеристик (часові інтервали, частотні показники, агреговані профілі клієнтів). Для усунення дисбалансу між шахрайським та не шахрайським класами застосовано метод SMOTE, що

дозволило сформувати більш репрезентативну навчальну вибірку та підвищити чутливість моделей до рідкісних шахрайських подій.

У третьому розділі розроблено та досліджено набір моделей штучного інтелекту для виявлення шахрайських транзакцій: Random Forest, XGBoost, багатошаровий персептрон (MLP) та гібридну стекінг-схему (Stacking Ensemble) на їх основі. Навчання й оптимізація моделей виконувалися з використанням Grid Search CV та метрик Precision, Recall, F_1-score і ROC-AUC. Експерименти показали, що найкращий баланс між точністю і повнотою забезпечує модель MLP із показниками Precision = 0.7431, Recall = 0.8265, F_1-score = 0.7826, and ROC-AUC = 0.9595. Стекінг-ансамбль у поточній конфігурації не продемонстрував суттєвої переваги над найкращою базовою моделлю, однак розглядається як перспективний напрям подальшої оптимізації. Для підвищення прозорості рішень застосовано інструменти Explainable AI (зокрема SHAP-аналіз), що дозволило оцінити внесок окремих ознак та поведінкових груп у формування прогнозу ризику.

У четвертому розділі розроблено концепцію інтеграції AI-моделі у банківську систему моніторингу в режимі, наближеному до реального часу. Запропоновано архітектуру взаємодії з потоками транзакцій, схеми маршрутизації інцидентів і ролі аналітичних підрозділів. На основі розрахунків показано потенційну економічну ефективність впровадження моделі (скорочення збитків від шахрайства, зменшення обсягу ручних перевірок і кількості хибних спрацювань), а також окреслено перспективи розвитку системи на основі графових нейронних мереж, потокової Big Data-аналітики та подальшого розширення застосування Explainable AI.

Розроблений підхід забезпечує ефективне виявлення шахрайських транзакцій, демонструючи збалансований компроміс між точністю, чутливістю та кількістю хибних спрацювань і може бути інтегрований у практичні фінансові системи моніторингу.

КЛЮЧОВІ СЛОВА: КІБЕРБЕЗПЕКА, МАШИННЕ НАВЧАННЯ, ВИЯВЛЕННЯ ШАХРАЙСТВА, АНОМАЛІЇ, НЕЙРОННІ МЕРЕЖІ, АНСАМБЛЕВЕ МОДЕЛЮВАННЯ, ДИСБАЛАНС ВИБІРКИ, ТРАНЗАКЦІЙНІ ДАНІ, КЛАСИФІКАЦІЯ, EXPLAINABLE AI.

ABSTRACT

The text part of the master's thesis: 106 pages, 14 figures, 13 tables, 48 references.

The purpose of the study is to develop, train, and validate a hybrid artificial intelligence (AI)–based approach for automatic detection of fraudulent transactions in financial institutions under conditions of significant class imbalance.

Object of the study – the process of detecting fraudulent transactions in financial systems using intelligent data analysis methods.

Subject of the study – machine learning and deep learning algorithms (ensemble models and neural networks), as well as methods for processing imbalanced financial data in transaction classification tasks.

Summary.

The first chapter provides an analytical overview of current cybersecurity threats in the financial sector and typical fraud schemes in digital banking, including card-not-present (CNP) operations, account takeover (ATO), and money laundering. The evolution of financial monitoring systems is considered – from static rule-based solutions to adaptive AI-driven platforms. The chapter analyzes the applicability of Random Forest, XGBoost, and Multilayer Perceptron (MLP) for binary transaction classification, highlights the critical problem of class imbalance, and discusses regulatory and ethical requirements, including the need for Explainable AI in the financial domain.

The second chapter describes the methodology for preparing transactional data for fraud detection modelling. It covers exploratory data analysis, data cleaning,

scaling and normalization, as well as extended behavioral feature engineering (time intervals between operations, frequency indicators, aggregated customer profiles). To mitigate the imbalance between fraudulent and non-fraudulent classes, the SMOTE technique is applied, which enables the formation of a more representative training set and improves model sensitivity to rare fraudulent events.

The third chapter presents the development and experimental evaluation of several AI models for fraud detection: Random Forest, XGBoost, Multilayer Perceptron (MLP), and a stacking ensemble built on top of these base classifiers. Model training and hyperparameter optimization are performed using Grid Search CV and evaluated with Precision, Recall, F_1-score, and ROC-AUC metrics. The experiments show that MLP provides the best trade-off between precision and recall, achieving Precision = 0.7431, Recall = 0.8265, F_1-score = 0.7826, and ROC-AUC = 0.9595. The stacking ensemble in its current configuration does not significantly outperform the best single model but is analyzed as a promising direction for further improvement. Explainable AI tools, in particular SHAP-based analysis, are used to quantify the contribution of individual features and behavioral groups to the predicted fraud risk.

The fourth chapter proposes an integration concept for the AI model within a real-time banking monitoring infrastructure. It introduces a reference architecture for interaction with transaction streams, incident routing, and analyst workflows. The chapter also provides an economic effectiveness assessment, demonstrating the potential to reduce fraud-related losses, decrease manual review workload, and limit the number of false positives. Future research directions include the use of graph neural networks for modeling transaction networks, real-time stream processing on Big Data platforms, and extended application of Explainable AI methods.

The proposed approach ensures effective detection of fraudulent transactions, achieving a balanced compromise between accuracy, sensitivity, and false alarm rate, and can be integrated into practical financial monitoring systems.

KEYWORDS: CYBERSECURITY, FRAUD DETECTION, MACHINE LEARNING, NEURAL NETWORKS, ENSEMBLE LEARNING, DATA IMBALANCE, TRANSACTION CLASSIFICATION, TRANSACTIONAL DATA, EXPLAINABLE AI.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 12
ВСТУП 14
РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ЗАСАДИ ВИЯВЛЕННЯ ШАХРАЙСТВА У ФІНАНСОВИХ СИСТЕМАХ 17
1.1. Аналіз сучасного ландшафту кіберзагроз та типологія шахрайства у цифровому банкінгу (CNP, ATO, Money Laundering) 17
1.2. Еволюція систем фінансового моніторингу: від статичних правил до адаптивних систем штучного інтелекту 22
1.3. Аналіз методів машинного навчання (Random Forest, XGBoost, MLP) у контексті задач бінарної класифікації транзакцій 27
1.4. Проблема критичного дисбалансу класів у фінансових даних та методи її вирішення (SMOTE, Undersampling) 31
1.5. Нормативно-правові та етичні аспекти використання ШІ у фінансовому секторі 36
Висновки до Розділу 1 40
РОЗДІЛ 2. МЕТОДИКА ДОСЛІДЖЕННЯ ТА ПІДГОТОВКА ДАНИХ ДЛЯ ПОБУДОВИ АНТИФРОДМОДЕЛІ 43
2.1. Характеристика та розвідувальний статистичний аналіз (EDA) обраного датасету транзакцій 43
2.2. Методика попередньої обробки даних: очищення від шумів, масштабування та нормалізація ознак 49
2.3. Feature Engineering: Розробка та обґрунтування поведінкових ознак (часові інтервали, частотні характеристики, агрегації) 53
2.4. Алгоритмічна реалізація балансування навчальної вибірки методом SMOTE для підвищення чутливості моделі 58
Висновки до Розділу 2 61
РОЗДІЛ 3. ПРОЕКТУВАННЯ, НАВЧАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ГІБРИДНОЇ AI-СИСТЕМИ 63

3.1. Розробка архітектури гібридної моделі Stacking Ensemble: обґрунтування вибору базових класифікаторів та мета-моделі 63
3.2. Процес навчання моделей та оптимізація гіперпараметрів (Grid Search CV) для максимізації метрик 73
3.3. Порівняльний аналіз ефективності моделей (RF, XGBoost, MLP vs Stacking) на тестовій вибірці 77
3.4. Оцінка результатів за метриками Precision, Recall, F_1-score та візуалізація (ROC-AUC, Confusion Matrix) 80
3.5. Інтерпретація рішень моделі методами Explainable AI (XAI): аналіз важливості ознак (SHAP) 85
Висновки до Розділу 3 89
РОЗДІЛ 4. ПРАКТИЧНІ АСПЕКТИ ВПРОВАДЖЕННЯ ТА ПЕРСПЕКТИВИ РОЗВИТКУ СИСТЕМИ 92
4.1. Розробка архітектури інтеграції AI-моделі у банківську систему моніторингу реального часу 93
4.2. Алгоритм взаємодії моделі з потоками транзакцій та сценарії реагування на інциденти 98
4.3. Оцінка економічної ефективності від впровадження розробленої системи (розрахунок запобігання збиткам) 101
4.4. Перспективи розвитку: використання графових нейронних мереж та потокової аналітики (Big Data) 106
Висновки до Розділу 4 112
ВИСНОВКИ 114
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 118
ДОДАТКИ 122
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація) 142

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AI (Artificial Intelligence) – штучний інтелект.

ML (Machine Learning) – машинне навчання.

DL (Deep Learning) – глибинне навчання.

XAI (Explainable Artificial Intelligence) – пояснюваний штучний інтелект; підхід, що забезпечує інтерпретацію рішень моделей.

ANN (Artificial Neural Network) – штучна нейронна мережа.

MLP (Multilayer Perceptron) – багатошаровий перцептрон, тип штучної нейронної мережі.

RNN (Recurrent Neural Network) – рекурентна нейронна мережа.

RF (Random Forest) – алгоритм випадкового лісу, ансамблевий метод класифікації.

XGBoost (Extreme Gradient Boosting) – алгоритм градієнтного бустингу, що використовує ансамбль дерев рішень.

SMOTE (Synthetic Minority Oversampling Technique) – метод синтетичного збільшення меншого класу для балансування вибірки.

ROC-AUC (Receiver Operating Characteristic – Area Under Curve) – метрика якості класифікації, що відображає співвідношення між рівнем хибних і правильних спрацьовувань.

Precision – точність класифікації; частка правильно передбачених позитивних випадків серед усіх позитивних прогнозів.

Recall (Sensitivity) – повнота або чутливість моделі; частка правильно знайдених позитивних випадків серед усіх реальних позитивних.

F_1-score – гармонійне середнє між Precision і Recall; показник балансу між точністю та повнотою.

Accuracy – загальна точність класифікації; частка правильно класифікованих спостережень.

Confusion Matrix – матриця неточностей; таблиця, що відображає кількість правильних і помилкових класифікацій.

TP (True Positive) – правильно передбачені позитивні випадки.

FP (False Positive) – неправильно передбачені позитивні випадки (хибні спрацювання).

FN (False Negative) – неправильно передбачені негативні випадки (пропущені шахрайства).

TN (True Negative) – правильно передбачені негативні випадки.

Stacking Ensemble – ансамблевий метод, що поєднує результати кількох базових моделей для покращення загальної ефективності.

ROC Curve – крива, що показує залежність між чутливістю та специфічністю моделі.

PR Curve (Precision-Recall Curve) – крива, що демонструє взаємозв’язок між точністю та повнотою.

ВСТУП

Розвиток фінансового сектору в останні десятиліття характеризується стрімким переходом до цифрових технологій, що докорінно змінюють способи взаємодії клієнтів із банківськими установами. Масове впровадження інтернет-банкінгу, мобільних застосунків, безконтактних платежів та цифрових платформ обслуговування сформували нову екосистему фінансових сервісів, у межах якої щодня здійснюються мільйони транзакцій у режимі реального часу. В таких умовах виникає нагальна потреба у впровадженні інноваційних методів захисту, оскільки традиційні механізми контролю дедалі менше відповідають сучасним викликам, обумовленим швидкоплинною та динамічною природою фінансових кіберзагроз.

Розширення сфери електронних платежів і інтеграція різних секторів бізнесу з віддаленими платіжними сервісами зумовили появу нового типу шахрайства, що базується на викраденні цифрових ідентичностей, компрометації облікових записів, маніпуляції даними клієнтів та експлуатації недоліків систем багатофакторної автентифікації. До ключових типів шахрайства у сучасному цифровому банкінгу належать атаки без фізичної присутності картки (CNP), захоплення облікових записів клієнтів (Account Takeover), операції з використанням підроблених персональних даних, а також складні міжбанківські схеми відмивання коштів, що реалізуються через автоматизацію мікротранзакцій. Складність і швидкість таких атак вимагають від фінансових установ переходу до систем, здатних працювати у режимі постійної адаптації та глибокого поведінкового аналізу.

Традиційні системи виявлення шахрайства, побудовані на статичних правилах і фіксованих шаблонах поведінки, не встигають за розвитком методів атак. Вони забезпечують високу точність у виявленні відомих сценаріїв, але демонструють низьку ефективність при зіткненні з новими, раніше невідомими патернами шахрайських дій. Саме тому сфера

фінансової безпеки активно переходить до інтелектуальних технологій аналізу транзакцій, застосовуючи машинне навчання, нейронні мережі та ансамблеві підходи для побудови гнучких антифродмодель.

Однією з основних труднощів під час розробки AI-моделей у фінансовій сфері є виражений дисбаланс класів у даних. У реальних історичних вибірках частка шахрайських операцій зазвичай не перевищує 0,2–0,5 % від загальної кількості транзакцій. Це істотно ускладнює процес навчання моделей і зумовлює їхню схильність класифікувати переважну більшість операцій як легітимні. За таких умов особливої важливості набувають методи балансування даних, зокрема SMOTE та різні підходи до undersampling, які дозволяють сформувати більш репрезентативну навчальну вибірку та підвищити здатність алгоритмів ідентифікувати рідкісні шахрайські події.

Цілком закономірним у таких умовах є зростання популярності гібридних підходів, зокрема стекінг-ансамблів (Stacking Ensemble), які об’єднують результати кількох базових класифікаторів і дають змогу підвищити точність, стійкість та здатність моделі коректно працювати на нових, раніше не бачених даних. Комбінація дерев рішень (Random Forest), градієнтного бустингу (XGBoost) та багатошарових персептронів (MLP) дозволяє краще враховувати різні типи структур у даних, зменшувати ризик перенавчання та формувати більш достовірні прогнози у складних середовищах, де на кожну шахрайську транзакцію припадають сотні тисяч легітимних.

Важливим аспектом розробки AI-рішень у фінансовому секторі є те, щоб їхні рішення були прозорими для користувачів і регуляторів та відповідали етичним нормам. Регуляторні акти Європейського Союзу, зокрема *AI Act* та директиви щодо фінансового моніторингу, вимагають від банківських установ забезпечення прозорості рішень штучного інтелекту, надання можливості відтворюваності результатів, а також недопущення використання моделей, які можуть порушувати права клієнтів. У цьому контексті методи Explainable AI

(XAI), зокрема SHAP-аналіз, є ключовим інструментом для визначення внеску кожної ознаки у прийняття рішення та контролю справедливості моделі.

Зважаючи на зазначені тенденції, ця робота спрямована на комплексне дослідження процесів виявлення шахрайства у фінансових системах та розроблення та порівняльну оцінку гібридної стекінг-моделі поряд із базовими класифікаторами, яка поєднує сучасні методи машинного навчання з практичними підходами до обробки даних із суттєво нерівномірним розподілом класів. У межах дослідження проаналізовано актуальні загрози у сфері цифрового банкінгу, розглянуто та реалізовано методики підготовки й трансформації даних для задачі виявлення шахрайства (зокрема масштабування ознак та балансування вибірки методом SMOTE), побудовано й навчено базові класифікатори (Random Forest, XGBoost, MLP), а також гібридну стекінг-модель та здійснено їх порівняльну оцінку на тестовій вибірці за ключовими метриками Precision, Recall, F_1-score та ROC-AUC. Окрему увагу приділено побудові архітектури інтеграції моделі у банківську систему моніторингу реального часу, оцінці економічного ефекту від її впровадження, а також окресленню перспектив удосконалення системи на основі графових нейронних мереж і потокової обробки даних.

Структура дослідження охоплює теоретичний аналіз проблематики, розроблення методології підготовки даних, побудову та навчання моделей, експериментальну оцінку їх ефективності та практичні рекомендації щодо впровадження у фінансову інфраструктуру. Такий комплексний підхід дозволяє отримати цілісне уявлення про механізми виявлення шахрайства та забезпечує наукову і прикладну цінність роботи.

РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ЗАСАДИ ВИЯВЛЕННЯ ШАХРАЙСТВА У ФІНАНСОВИХ СИСТЕМАХ

У першому розділі розглядаються ключові теоретичні питання, пов’язані з виявленням шахрайства у сучасних фінансових системах. Спочатку аналізується сучасна картина кіберзагроз та основні типи шахрайства у цифровомубанкінгу, далі розглядається перехід від простих правил до адаптивних AI-рішень у системах фінансового моніторингу. Окрему увагу приділено характеристиці методів машинного навчання, що застосовуються для бінарної класифікації транзакцій, а також проблемі критичного дисбалансу класів у фінансових даних і підходам до її подолання (зокрема SMOTE та undersampling). Завершується розділ аналізом нормативно-правових та етичних вимог до використання штучного інтелекту у фінансовому секторі, що визначають рамки для подальшої практичної реалізації антифрод-рішень.

1.1 Аналіз сучасного ландшафту кіберзагроз та типологія шахрайства у цифровому банкінгу (CNP, ATO, Money Laundering)

Стрімка цифровізація банківських послуг призвела до того, що основна частина взаємодії клієнтів із фінансовими установами перемістилася в онлайн-середовище: інтернет-банкінг, мобільні застосунки, дистанційне відкриття рахунків, миттєві перекази та оплата товарів і послуг у режимі 24/7. З одного боку, це підвищує зручність і доступність фінансових сервісів, з іншого — суттєво розширює поверхню атаки для кіберзлочинців. Традиційні підходи до безпеки, орієнтовані переважно на фізичні канали обслуговування (відділення, банкомати, POS-термінали), виявляються недостатніми в умовах, коли критична частина операцій здійснюється без фізичної присутності клієнта та платіжного інструменту.

Сучасну картину кіберзагроз для банківського сектору можна описати як комплекс взаємопов’язаних ризиків, у яких технічні атаки поєднуються з елементами соціальної інженерії та зловживанням легітимними сервісами. До

найпоширеніших загроз належать фішингові кампанії, розповсюдження шкідливого програмного забезпечення (malware, трояни для мобільного банкінгу), атаки типу man-in-the-middle, компрометація облікових даних через витоки з інших сервісів, бот-мережі для масового підбору паролів (credential stuffing), а також шахрайські схеми з використанням «фінансових посередників» (money mules).

На цьому тлі особливу актуальність набувають саме шахрайські операції у цифровому банкінгу, де зловмисник намагається ініціювати чи провести фінансову транзакцію, маскуючись під легітимного клієнта або використовуючи підроблені чи вкрадені платіжні інструменти. Для побудови ефективних систем протидії важливо не лише фіксувати окремі інциденти, а й розуміти типологію шахрайства, тобто класифікувати його за механізмом реалізації, каналом здійснення операції та роллю учасників.

У контексті цифрового банкінгу до ключових категорій шахрайських схем зазвичай відносять:

- шахрайство з платіжними картками без фізичної присутності картки (Card-Not-Present, CNP);
- шахрайство, пов’язане із захопленням облікових записів (Account Takeover, ATO);
- операції, пов’язані з відмиванням коштів (Money Laundering), у тому числі з використанням онлайн- та міжбанківських каналів.

Усі ці типи мають спільну рису: вони реалізуються в цифровому середовищі, часто в автоматизованому режимі, і здатні генерувати великий обсяг дрібних транзакцій, що ускладнює їх виявлення за допомогою лише простих правил.

Шахрайство категорії Card-Not-Present (CNP) охоплює всі операції, за яких платіжна картка фізично не використовується в момент транзакції. Йдеться насамперед про інтернет-еквайринг, мобільні застосунки, платіжні шлюзи електронної комерції тощо. У таких сценаріях рішення про авторизацію

ґрунтується на введенні реквізитів картки (номер, строк дії, CVV/CVC-код, інколи — одноразових паролів), а не на пред'явленні фізичного пластику.

Основою для CNP-шахрайства зазвичай є компрометація карткових даних, яка може відбуватися різними шляхами:

- отримання реквізитів через фішингові сайти або підроблені форми оплати;
- перехоплення даних уразливими або шкідливими мобільними застосунками;
- придбання масивів карткових даних на «темних» форумах унаслідок попередніх витоків;
- використання шкідливих скриптів (web skimming), вбудованих у сторінки інтернет-магазинів.

Після отримання доступу до реквізитів картки зловмисники зазвичай діють за одним із двох сценаріїв. Перший — це «тестові» транзакції на невеликі суми з метою перевірки життєздатності картки та відсутності жорстких обмежень. У разі успішного тесту здійснюються більші покупки або серії мікротранзакцій. Другий сценарій — масове генерування платежів невеликих сум через автоматизовані інструменти, що ускладнює відокремлення шахрайських операцій від звичайної платіжної активності клієнта.

Особливістю CNP-шахрайства є те, що в транзакційному потоці відсутній фізичний контакт з інфраструктурою банку, а отже, традиційні механізми захисту (EMV-чип, PIN-код, перевірка підпису) не застосовуються. Безпека в такому середовищі значною мірою залежить від:

- коректної реалізації протоколів автентифікації (наприклад, 3D Secure);
- налаштувань лімітів і правил моніторингу;
- здатності аналітичних систем розпізнавати аномалії у поведінці клієнта (час, геолокація, пристрій, тип покупки тощо).

Саме тому моделі на основі машинного навчання, здатні виявляти нестандартні патерни поведінки в сукупності з даними про попередні транзакції, є критично важливими для протидії CNP-шахрайству.

Шахрайство типу Account Takeover (ATO) передбачає, що зловмисник отримує контроль над чинним обліковим записом клієнта у системі інтернет- чи мобільного банкінгу і здійснює операції від імені жертви. На відміну від CNP-сценаріїв, де використовуються реквізити картки, при ATO атакують саме облікові дані для входу (логін, пароль, одноразові коди, токени).

Основні способи захоплення облікових записів включають:

- використання облікових даних, викрадених із зовнішніх сервісів (credential stuffing) за принципом повторного використання паролів;
- фішингові атаки, що змушують клієнта самостійно розкрити логін, пароль та одноразові коди;
- встановлення шкідливого ПЗ на пристрій жертви (keylogger, трояни для мобільного банкінгу);
- атаки типу SIM-swapping, коли зловмисник отримує контроль над номером мобільного телефону і перехоплює SMS-коди;
- шахрайські дзвінки від «псевдопрацівників банку», під час яких клієнта переконують виконати потрібні дії в застосунку.

Після успішного захоплення облікового запису злочинці, як правило, змінюють критичні параметри профілю (контактний номер телефону, електронну пошту, ліміти, список шаблонів платежів), а також здійснюють перекази на підконтрольні рахунки або електронні гаманці. Важливо, що в більшості таких випадків операції формально виглядають як ініційовані справжнім власником рахунку, адже виконуються через звичайний інтерфейс інтернет- чи мобільного банкінгу.

Для виявлення ATO недостатньо аналізувати лише факт авторизації — потрібен поведінковий аналіз:

- аномальна геолокація або різка зміна IP-адреси;
- незвичний пристрій або браузер;

- нетипова швидкість навігації та послідовність дій;
- раптові зміни параметрів облікового запису перед операціями з підвищеним рівнем ризику;
- спроби доступу після численних невдалих входів.

Таким чином, боротьба з АТО передбачає не лише удосконалення процедур автентифікації, але й побудову моделей, здатних розпізнавати нетипову поведінку сесій у реальному часі.

Ще однією критичною загрозою для фінансових систем є відмивання коштів (Money Laundering) — комплекс дій, спрямованих на легалізацію злочинних доходів через фінансову інфраструктуру. Цей процес традиційно описують як послідовність етапів: розміщення, розшарування (layering) та інтеграція. У цифровому банкінгу відмивання коштів набуває специфічних форм, пов’язаних із використанням онлайн-каналів, миттєвих переказів та транскордонних операцій.

Серед типових сценаріїв можна виділити:

- багаторазові дрібні перекази (smurfing, structuring), що дають змогу уникати автоматичних порогових спрацювань;
- використання великої кількості тимчасових або підставних рахунків (money mules), оформлених на підставних осіб;
- швидке переказування коштів між рахунками різних банків та платіжних систем із подальшим зняттям готівки або виведенням коштів у фінансові інструменти з обмеженою прозорістю;
- комбінацію операцій із платіжними картками, електронними гаманцями та криптоактивами.

На відміну від «класичного» шахрайства, метою якого є безпосереднє списання коштів із рахунку жертви, при відмиванні коштів ключовим є маскування джерела походження фінансових потоків. Виявлення таких схем вимагає аналізу ланцюжків транзакцій, взаємозв’язків між клієнтами, часових патернів та типових «маркерів» підозрілої активності (часті вхідні/вихідні

перекази без економічного змісту, незвична географія контрагентів, повторювані маршрути коштів тощо).

У зв’язку з цим дедалі частіше застосовуються підходи, що поєднують класичну аналітику з методами машинного навчання та побудовою графових моделей взаємозв’язків між рахунками. Такі рішення дають змогу не лише виявляти окремі підозрілі транзакції, а й ідентифікувати цілі схеми відмивання, у яких беруть участь десятки або сотні пов’язаних рахунків.

Цифровий банкінг сьогодні характеризується різноманіттям шахрайських схем, у яких CNP-операції, захоплення облікових записів та відмивання коштів є ключовими типами ризиків. Усі ці схеми об’єднує висока швидкість змін, масовість та сильний зв’язок із поведінкою клієнтів і параметрами транзакційних потоків. Це, своєю чергою, обумовлює потребу у використанні інтелектуальних систем моніторингу, здатних аналізувати великі обсяги даних у реальному часі та виявляти нетипові патерни, які виходять за межі можливостей традиційних rule-based-рішень. Розмаїття описаних схем шахрайства показує, що фінансовим установам недостатньо покладатися лише на жорсткі правила та ручний аналіз інцидентів. Для своєчасного виявлення CNP-операцій, захоплення облікових записів та складних схем відмивання коштів потрібні системи моніторингу, здатні працювати з великими потоками транзакцій у реальному часі та враховувати поведінкові особливості клієнтів. Це, своєю чергою, актуалізує перехід до інтелектуальних моделей аналізу транзакцій і визначає подальший напрям дослідження у межах даної роботи.

1.2. Еволюція систем фінансового моніторингу: від статичних правил до адаптивних систем штучного інтелекту

Поява масових електронних платежів та цифрових каналів обслуговування поступово змінювала не лише характер шахрайських загроз, а й підходи фінансових установ до їх виявлення. Якщо перші системи протидії шахрайству базувалися переважно на простих порогових умовах і ручній

перевірці операцій, то сучасні рішення дедалі частіше використовують складні аналітичні моделі, алгоритми машинного навчання та елементи штучного інтелекту. Така еволюція є закономірною відповіддю на зростаючу складність і динамічність шахрайських схем у цифровому середовищі.

На початкових етапах розвитку платіжних систем боротьба з шахрайством спиралася насамперед на організаційні процедури та ручний контроль. Аналітики служби безпеки переглядали звіти про операції, спиралися на власний професійний досвід, звертали увагу на надто великі суми, підозрілих контрагентів та нетипові для клієнта операції.Зі зростанням кількості транзакцій ручний аналіз став фізично неможливим, тому банки почали автоматизувати хоча б базові перевірки. Так з’явилися перші порогові правила: операції понад певну суму або з певних країн вимагали додаткового погодження, а окремі типи транзакцій могли блокуватися автоматично.

Подібні рішення мали очевидні переваги — простота реалізації, прозорість логіки, зрозумілість для аудиторів і регуляторів. Водночас їхні недоліки теж стали швидко проявлятися. Жорстко фіксовані пороги погано враховували контекст: те, що для одного клієнта є великою сумою, для іншого — цілком звичайна щоденна операція. Це спричиняло велику кількість хибних спрацювань, через що знижувався комфорт і зручність користування банківськими сервісами. Ще більш критичним було те, що статичні правила не встигали за розвитком шахрайських схем: зловмисники швидко підлаштовувалися під встановлені ліміти, дробили операції, змінювали маршрути переказів.

Наступним етапом розвитку стали rule-based системи та скорингові моделі, які дали змогу перетворити досвід аналітиків на формалізовану, розгалужену систему правил. Замість одного/двох порогів почали використовуватися складні комбінації умов: аналізувались сума, валюта, канал проведення операції, країна відправника й одержувача, історія взаємодії клієнта з банком. На цьому етапі з’явилися і скорингові підходи, коли кожній транзакції присвоювався ризиковий бал, а подальша доля операції визначалася

порівнянням цього бала з налаштованими порогами. Такі системи суттєво підвищили швидкість прийняття рішень та дозволили застосовувати єдиний підхід до великої кількості клієнтів і продуктів.

Проте зі збільшенням обсягу та різноманіття операцій rule-based моделі поступово втрачали ефективність через надмірне ускладнення їхньої структури. Кількість правил у деяких банках сягала сотень і тисяч, їх підтримка перетворювалася на окремий проєкт: одні правила конфліктували з іншими, частина переставала бути актуальною, а внесення змін займало тижні. Крім того, такі системи залишалися реактивними: нові сценарії шахрайства потрапляли в поле зору лише після того, як завдали збитків, і тільки тоді для них створювалися нові правила. У ситуації, коли зловмисники активно використовують автоматизацію, бот-мережі та міжбанківські схеми, подібна інерційність стає критичним недоліком.

Цифровий банкінг висунув до систем моніторингу нові вимоги. Вони мають працювати з великими потоками транзакцій у режимі, наближеному до реального часу, враховувати індивідуальну поведінку кожного клієнта, гнучко реагувати на зміни в ринку та тактиці зловмисників. На цьому етапі логічним стало впровадження підходів машинного навчання. На відміну від жорстко прописаних правил, моделі машинного навчання навчаються на даних про попередні транзакції і здатні самостійно виявляти складні залежності між ознаками, які важко або неможливо описати вручну.

У практиці фінансового моніторингу почали широко застосовуватися дерева рішень та їх ансамблі, зокрема Random Forest та XGBoost, а також нейронні мережі для моделювання складних поведінкових патернів.Ці моделі добре працюють із табличними даними, природними для банківських транзакцій, дозволяють враховувати десятки й сотні ознак та досягати високих показників точності. Паралельно з’явилися та почали активно використовуватися нейронні мережі, зокрема багатошарові персептрони, які краще вловлюють складні, нелінійні патерни та придатні для побудови гібридних ансамблів. У сегментах, де маркування даних є частковим або

неповним, застосовуються методи пошуку аномалій і кластеризації, що дозволяють фокусувати увагу аналітиків на нетипових транзакціях.

Важливо, що перехід до моделей машинного навчання торкнувся не лише математичного ядра систем, а також і організації всього процесу моніторингу. З'явилися централізовані платформи, де в одному контурі інтегруються: модуль збору і попередньої обробки даних, модуль оцінки ризику транзакцій, модуль маршрутизації алертів і робочі місця аналітиків, які приймають фінальні рішення. Одна й та сама транзакція може послідовно оброблятися кількома рівнями контролю: від простих технічних фільтрів та нормативних правил до складних AI-моделей, а результат їх роботи фіксується в єдиній системі підтримки рішень.

Попри впровадження машинного навчання, повна відмова від правил виявилася ні теоретично, ні практично доцільною. У багатьох регуляторних вимогах, особливо у сфері фінансового моніторингу та протидії відмиванню коштів, наголошується на тому, що критерії спрацювання контролів мають бути прозорими та чітко задокументованими. Тому сучасні антифродмодель зазвичай мають гібридну архітектуру. На базовому рівні функціонують rule-based механізми, які забезпечують виконання мінімально необхідних нормативних вимог і відсікають очевидно підозрілі операції. Вище розташовуються скорингові модулі та AI-компоненти, що аналізують широкий спектр ознак і забезпечують більш тонку градацію ризику.

У такій архітектурі саме моделі машинного навчання виступають аналітичним ядром — вони визначають, які операції є типовими для того чи іншого клієнта, а які суттєво відхиляються від звичної поведінки. При цьому правила залишаються важливим «захисним шаром», який гарантує, що навіть у разі деградації моделі певний рівень контролю буде збережений. Додатковим виміром стає пояснюваність: фінансові установи змушені враховувати вимоги регуляторів та внутрішніх політик щодо можливості обґрунтувати причини блокування чи відмови в операції, що безпосередньо впливає на вибір класів моделей та підходів до їх інтерпретації.

Ще однією ключовою характеристикою сучасних систем моніторингу є їхня адаптивність. У міру накопичення нових даних моделі потребують регулярного оновлення. Зміни у поведінці клієнтів, поява нових продуктів або каналів, а також еволюція шахрайських схем призводять до явища концептуального дрейфу, коли розподіл ознак та цільових змінних поступово змінюється, а отже, моделі, навченої на старих даних, вже недостатньо. Відповідно до цього в системах моніторингу впроваджуються процедури періодичного перенавчання моделей, оновлення наборів ознак, моніторингу якості прогнозів та автоматичного виявлення деградації.

Особливу роль у цьому відіграють часові та поведінкові ознаки. Сучасні антифродмодель аналізують не лише базові параметри самої транзакції (сума, валюта, країна, канал), а й узагальнені показники: частоту операцій за певний проміжок часу, середній і максимальний розмір платежів, типові для клієнта часові вікна активності, географію його переказів та історію взаємодії з одними й тими самими контрагентами.По суті, замість абстрактної моделі типової поведінки формується індивідуальний профіль клієнта, відхилення від якого використовується як один із ключових сигналів ризику. Підсумовуючи, еволюцію систем фінансового моніторингу можна описати як шлях від ручного аналізу та простих порогових перевірок до гібридних інтелектуальних платформ, де поєднуються rule-based контроль, скорингові підходи та моделі машинного навчання. Сучасніантифродмодель вже не сприймаються як набір ізольованих правил, а розглядаються як динамічні, адаптивні рішення, здатні працювати з великими потоками транзакцій у реальному часі, враховувати поведінкові особливості клієнтів і забезпечувати баланс між вимогами регуляторів, комфортом користувачів та ефективністю виявлення шахрайства. Подальший розвиток таких систем пов’язаний із ширшим застосуванням ансамблевих і стекінг-моделей, методів Explainable AI, а також переходом до графових уявлень транзакційних мереж і потокової обробки даних, що створює передумови для наступних етапів розвитку антифрод-рішень.

1.3. Аналіз методів машинного навчання (Random Forest, XGBoost, MLP) у контексті задач бінарної класифікації транзакцій

Виявлення шахрайських операцій у цифровому банкінгу за своєю природою є задачею бінарної класифікації, у межах якої кожній транзакції необхідно надати один із двох статусів: шахрайська або така, що не містить ознак шахрайства. На практиці моделі зазвичай повертають не лише кінцевий клас, а й числову оцінку ймовірності того, що операція належить до ризикової категорії. Далі фінансова установа визначає порогове значення цієї ймовірності залежно від власної політики ризику. Зміна порога безпосередньо впливає на те, скільки звичайних операцій буде помилково позначено як підозрілі, а скільки справді шахрайських транзакцій залишаться непоміченими, тому для оцінки якості моделей особливого значення набувають метрики Precision, Recall, F_1-score та ROC-AUC.

Специфіка транзакційних даних у банківському секторі полягає у великій кількості ознак, суттєво нерівномірному розподілі класів (коли шахрайські операції становлять лише незначну частку від загального обсягу), наявності шуму, помилок та пропусків у даних. У таких умовах алгоритми машинного навчання мають забезпечувати стійкість до аномалій, працездатність на великих масивах табличних даних, здатність вловлювати складні взаємозв’язки між параметрами операцій і водночас зберігати прийнятний рівень інтерпретованості. У цьому контексті особливий інтерес становлять ансамблеві методи на основі дерев рішень — Random Forest та XGBoost, а також нейромережевий підхід на базі багатошарового персептрону (Multilayer Perceptron, MLP), які в даній роботі обрано як базові класифікатори для подальшого побудови гібридної моделі.

Random Forest (довільний ліс) є класичним прикладом ансамблевого методу, в якому замість одного великого дерева рішень формується множина відносно простих дерев, навчання кожного з яких відбувається на випадковій підвибірці спостережень і підмножині ознак. Результат для окремої транзакції

отримують шляхом агрегування прогнозів усіх дерев, зазвичай за принципом більшості голосів. Такий підхід зменшує ризик перенавчання окремих дерев і підвищує стійкість моделі до шуму та локальних аномалій. У задачі виявлення шахрайських транзакцій Random Forest є привабливим завдяки тому, що добре працює з великою кількістю вхідних характеристик, може враховувати різні типи ознак (як прості параметри операції, так і агреговані поведінкові показники) і дозволяє оцінювати важливість кожної ознаки для прийняття рішення. Це полегшує подальший аналіз моделі аналітиками та сприяє відповідності вимогам регуляторів. Водночас у разі значного дисбалансу класів Random Forest потребує додаткових заходів: налаштування ваг класів або поєднання з методами попереднього балансування вибірки, зокрема методом SMOTE.

Методи градієнтного бустингу, на відміну від Random Forest, формують ансамбль не з незалежних, а з послідовно пов’язаних моделей, де кожне наступне дерево спрямоване на виправлення помилок попередніх. Однією з найпоширеніших та найбільш ефективних реалізацій цього підходу є XGBoost, оптимізований варіант градієнтного бустингу над деревами рішень. Під час навчання XGBoost зосереджується на транзакціях, які попередні ітерації класифікували неправильно, поступово зменшуючи залишкову помилку. Це дозволяє моделі вловлювати тонкі закономірності у поведінці клієнтів, що є особливо важливим у випадках, коли шахрайські операції маскуються під звичайну активність. Завдяки вбудованим механізмам контролю складності моделі, гнучким налаштуванням функції втрат та підтримці коригування ваг класів XGBoost добре придатний для роботи з вибірками з нерівномірним розподілом класів. Крім того, він ефективно поєднується з інструментами пояснення рішень, такими як SHAP-аналіз, що дає змогу оцінити внесок окремих ознак у прогноз ризику.

Багатошаровий персептрон (MLP) реалізує інший підхід до класифікації транзакцій на основі штучних нейронних мереж. Вхідний шар MLP отримує числові значення ознак транзакції, які надалі проходять через один або кілька

прихованих шарів. На кожному з них сигнали лінійно комбінуються за допомогою вагових коефіцієнтів, до них додається зміщення, а потім застосовується нелінійна функція активації. У результаті формується внутрішнє, більш компактне представлення даних у прихованому просторі, а на виході мережа генерує оцінку ймовірності належності транзакції до шахрайського класу. Завдяки послідовному застосуванню нелінійних перетворень MLP здатний моделювати складні взаємозв'язки між сумою операції, часовими інтервалами, історією активності клієнта, географією платежів та іншими факторами. Водночас нейронна мережа є більш вибагливою до попередньої обробки даних: для стабільної роботи необхідні масштабування ознак, коректна робота з пропущеними значеннями, застосування прийомів обмеження складності моделі та контроль процесу навчання, щоб уникнути перенавчання.

Для зручності сприйняття основні характеристики моделей Random Forest, XGBoost та MLP, їхні переваги й обмеження у контексті задачі виявлення шахрайських транзакцій узагальнено в таблиці 1.1.

Таблиця 1.1Порівняльна характеристика моделей Random Forest, XGBoost та MLP у задачі виявлення шахрайських транзакцій

Характеристика	Random Forest	XGBoost	Багатошаровий персептрон (MLP)
Тип моделі	Ансамбль дерев рішень із використанням методів випадкової вибірки (bagging)	Ансамбль дерев рішень, побудованих послідовно за схемою градієнтного бустингу	Нейронна мережа прямого поширення з одним чи кількома прихованими шарами
Робота з табличними даними	Добре підходить для великих табличних вибірок	Особливо ефективний на великих та складних табличних вибірках	Потребує коректного кодування та масштабування ознак
Основні переваги	Стабільна якість,	Висока точність,	Висока гнучкість,

	стійкість до шуму, можливість оцінки важливості ознак	гнучкі налаштування, здатність виявляти складні патерни	здатність моделювати складні нелінійні залежності
Типові обмеження	За значного дисбалансу класів без додаткових заходів тяготіє до класу більшості	Чутливість до гіперпараметрів, ризик перенавчання за відсутності регуляризації	Вибагливість до якості даних, ризик перенавчання, нижча прозорість рішень
Чутливість до гіперпараметрів	Помірна, базові налаштування часто дають прийнятний результат	Висока, потрібен ретельний підбір параметрів навчання	Висока, важливі архітектура мережі, вибір функцій активації та режимів регуляризації
Інтерпретованість	Порівняно добра завдяки аналізу важливості ознак та окремих дерев	Середня; потребує застосування XAI-методів (наприклад, SHAP)	Низька без XAI; для пояснення потрібні додаткові інструменти
Робота з дисбалансом класів	Потребує налаштування ваг класів або балансування вибірки (наприклад, SMOTE)	Підтримує ваги класів та налаштування функції втрат під дисбаланс	Потребує попереднього балансування даних та налаштування функції втрат
Вимоги до підготовки даних	Відносно невисокі, масштабування ознак не є критичним	Помірні; важливо забезпечити коректність і повноту даних	Високі; необхідне масштабування ознак, обробка пропусків, ретельне очищення
Типова роль у системі	Базовий або еталонний класифікатор	Потужний основний класифікатор з високими показниками якості	Гнучкий додатковий класифікатор у складі ансамблю або стекінг-моделі

Як видно з узагальненої характеристики, жодна з моделей не є універсальною в усіх аспектах. Random Forest вирізняється простотою

налаштування, стійкістю до шуму та базовою інтерпретованістю, що робить його зручним для побудови еталонної моделі. XGBoost нерідко досягає кращих значень основних метрик на складних вибірках, однак для цього потребує тонкого налаштування гіперпараметрів та більш трудомісткого процесу оптимізації. Багатошаровий персептрон надає додаткову гнучкість у моделюванні нелінійних залежностей, проте є більш вимогливим до підготовки даних і менш прозорим з точки зору пояснення рішень.

У сучасній практиці виявлення шахрайства акцент поступово зміщується від пошуку однієї найкращої моделі до використання гібридних підходів, у межах яких поєднуються сильні сторони різних алгоритмів. Застосування ансамблів, зокрема стекінг-ансамблів із участю моделей типу Random Forest, XGBoost та MLP, дає змогу зменшити вплив обмежень окремих підходів і досягти кращого балансу між точністю, повнотою та стійкістю рішень в умовах нерівномірного розподілу класів. Такі гібридні схеми створюють підґрунтя для побудови більш надійних систем виявлення шахрайських транзакцій, придатних до застосування в реальних фінансових інфраструктурах.

1.4. Проблема критичного дисбалансу класів у фінансових даних та методи її вирішення (SMOTE, Undersampling)

Однією з ключових методологічних проблем під час побудови моделей виявлення шахрайства є критичний дисбаланс класів у вхідних даних. У типовій ситуації шахрайські операції становлять менше 1 % від усіх транзакцій: зазвичай їхня частка коливається в межах приблизно 0,2–0,5 % від загального обсягу операцій. Для фінансової установи така ситуація є природною: переважна більшість клієнтів здійснює коректні операції, тоді як випадки шахрайства трапляються нечасто, але можуть мати значні фінансові наслідки. Однак для алгоритмів машинного навчання така картина є проблемною, оскільки машина орієнтується не на важливість подій для бізнесу, а на їхню частоту у навчальних даних.

Якщо навчати модель без урахування диспропорції між класами, вона прагнутиме мінімізувати загальну кількість помилок насамперед за рахунок правильної класифікації більшості. У крайній ситуації класифікатор може «навчитися» майже завжди відносити транзакції до класу «не шахрайська» і при цьому демонструвати дуже високу формальну точність. Наприклад, якщо шахрайських операцій лише 0,2 %, то модель, яка позначає всі транзакції як «звичайні», матиме accuracy на рівні 99,8 %, але не виявить жодного реального випадку шахрайства. Подібний результат, хоч і виглядає привабливим за показником загальної точності, насправді не відповідає реальним цілям системи протидії шахрайству.

Таким чином, дисбаланс класів впливає як на процес навчання моделі, так і на інтерпретацію результатів. Класичні показники, зокрема загальна точність, стають малоінформативними, оскільки майже повністю визначаються якістю класифікації транзакцій більшості. Натомість зростає роль метрик, чутливих до роботи саме з рідкісним класом, — точності (Precision), повноти (Recall), F_1-score та площі під ROC-кривою. Саме ці метрики дають змогу оцінити, наскільки ефективно модель виявляє шахрайські операції, водночас не перевантажуючи аналітиків і службу підтримки надмірною кількістю помилкових сповіщень про звичайні транзакції.

Дисбаланс також впливає на формування межі розділення класів у просторі ознак. Оскільки точки класу більшості заповнюють цей простір значно щільніше, оптимізаційні процедури, що мінімізують сумарну помилку, прагнуть розмістити межу так, щоб максимально правильно відокремити саме звичайні операції. У підсумку зона, у якій модель дозволяє собі позначати транзакції як шахрайські, стає дуже вузькою, і внаслідок цього як чимало нетипових, але коректних операцій, так і значна частина справжніх випадків шахрайства можуть опинитися по один бік від межі розділення. Це призводить до низького Recall для шахрайського класу, тобто до пропуску значної частини небезпечних транзакцій.

Щоб пом’якшити вплив дисбалансу, у практиці машинного навчання застосовують кілька груп підходів. З одного боку, вплив дисбалансу можна враховувати на рівні самої моделі: налаштовувати ваги класів у функції втрат, задавати різну ціну помилок для кожного класу або коригувати поріг, за яким транзакція вважається шахрайською.З іншого боку, можна модифікувати навчальну вибірку, штучно змінюючи співвідношення між класами. У цьому підрозділі основна увага приділяється саме перетворенням вибірки, зокрема методам зменшення кількості прикладів класу більшості (undersampling) та синтетичного збільшення міноритарного класу (SMOTE).

Найпростішим способом боротьби з дисбалансом є undersampling класу більшості, тобто цілеспрямоване скорочення кількості звичайних транзакцій у навчальних даних. У найпростішому випадку частину записів класу звичайних транзакцій випадковим чином вилучають із вибірки, доки співвідношення між класами не наблизиться до бажаного. Такий підхід дозволяє сформувати більш компактну вибірку, прискорити навчання моделі та примусити алгоритм бачити більшу частку шахрайських прикладів у загальному наборі даних. Водночас ціна цієї простоти — втрата інформації: разом із зайвими спостереженнями видаляється і частина корисних варіантів нормальної поведінки клієнтів. Якщо випадково видалити цілі підтипи легальних сценаріїв (наприклад, специфічну поведінку окремих сегментів клієнтів або певних географічних регіонів), модель гірше працюватиме на реальних потоках транзакцій.

У відповідь на це були розроблені більш розумні схеми undersampling, коли для видалення обираються не випадкові спостереження, а такі, що найменше впливають на розділення класів. Наприклад, можуть відкидатися транзакції, розташовані в щільних «хмарах» звичайних операцій, тоді як спостереження, близькі до межі розділення з шахрайським класом, навпаки, зберігаються. Це дозволяє зберегти різноманіття поведінки класу більшості та водночас суттєво зменшити розмір вибірки. Однак базова проблема undersampling залишається: у будь-якому випадку йдеться про видалення

частини реальних даних, що не завжди прийнятна з погляду довгострокового аналізу або коли дані є дорогими з точки зору їх збору та зберігання.

Альтернативний підхід — це oversampling міноритарного класу, тобто штучне збільшення кількості шахрайських транзакцій у навчальній вибірці. Найпростіша реалізація — це дублювання наявних прикладів, наприклад багаторазове копіювання кожної шахрайської операції. Така стратегія дозволяє уникнути втрати інформації про звичайні транзакції, але має інший суттєвий недолік: модель починає надто точно відтворювати дубльовані приклади, що підвищує ризик перенавчання. У підсумку алгоритм може добре працювати на навчальних даних, але гірше — на нових, раніше не бачених транзакціях.

Щоб обійти цей недолік, були запропоновані методи синтетичного oversampling**,** у яких нові приклади міноритарного класу не копіюються, а генеруються на основі вже наявних. Одним із найвідоміших підходів цього типу є SMOTE (Synthetic Minority Over-sampling Technique). Його основна ідея полягає в тому, що нові шахрайські транзакції генеруються шляхом інтерполяції між близькими за ознаками прикладами того самого класу. Алгоритм обирає випадкову транзакцію міноритарного класу, знаходить для неї кілька найближчих за ознаками прикладів серед інших шахрайських операцій і генерує нові приклади, розміщені між цими точками в просторі ознак. Таким чином додаються синтетичні спостереження, які заповнюють проміжки між наявними випадками шахрайства, роблячи розподіл цього класу більш щільним і рівномірним.

Таке генерування має кілька переваг. По-перше, модель отримує більше інформації про те, як може виглядати шахрайський клас у різних комбінаціях ознак, а не лише про конкретні реалізації, що були зафіксовані в даних за попередні періоди. По-друге, на відміну від простого дублювання, SMOTE розширює різноманіття випадків міноритарного класу, що сприяє кращому узагальненню результатів на нові транзакції. По-третє, цей алгоритм дає змогу доволі точно керувати рівнем балансування, додаючи стільки синтетичних

прикладів, скільки потрібно для досягнення бажаного співвідношення між класами.

Водночас застосування SMOTE потребує обережності. Якщо у вихідних даних присутні некоректно промарковані спостереження або значний шум, синтетичне розширення навколо таких точок фактично поширює помилку на новостворені приклади. Крім того, у зонах, де класи сильно перекриваються, надмірне збільшення міноритарного класу може призвести до ще більшого змішування класів і ускладнити завдання моделі. На практиці SMOTE зазвичай не використовують ізольовано: його поєднують із попереднім очищенням даних, відбором найбільш інформативних ознак, а інколи також із помірним undersampling класу більшості, щоб досягти прийнятного балансу між обсягом даних та їхньою інформативністю.

Окремо варто зазначити, що вибір стратегії балансування тісно пов’язаний із бізнес-контекстом. Для одних фінансових установ пріоритетом може бути мінімізація пропущених шахрайських операцій (максимізація Recall), навіть ціною збільшення кількості помилкових сповіщень. Інші, навпаки, можуть прагнути зменшити навантаження на службу підтримки та клієнтів, приймаючи меншу кількість сповіщень про підозрілі операції, але водночас мирячись із тим, що частина незначних випадків шахрайства буде пропущена. Це відображається як у виборі метрик, за якими оцінюється модель, так і в налаштуваннях методів балансування, порогів рішень та ваг класів.

Проблема критичного дисбалансу класів у фінансових даних є фундаментальним викликом для побудови ефективних антифрод-рішень. Ігнорування цього чинника призводить до побудови моделей, які на папері мають високу загальну точність, але на практиці не справляються з ключовим завданням – виявленням шахрайських транзакцій. Застосування підходів до перетворення навчальної вибірки, зокрема undersampling класу більшості та синтетичного oversampling міноритарного класу на основі SMOTE, дає змогу краще представити рідкісні події в даних, підвищити чутливість моделей до випадків шахрайства та отримати більш реалістичну оцінку їхньої ефективності

в умовах реальних транзакційних потоків. У поєднанні з правильно обраними метриками якості та належним налаштуванням алгоритмів такі методи стають важливою складовою сучасної методології виявлення шахрайства у фінансових системах.

1.5. Нормативно-правові та етичні аспекти використання ШІ у фінансовому секторі

Використання систем штучного інтелекту у фінансовому секторі тісно пов’язане не лише з технічними обмеженнями, а й із комплексом нормативно-правових та етичних вимог. Якщо для розробника моделі першочерговими є питання якості даних, вибору алгоритмів, налаштування гіперпараметрів і метрик, то для фінансової установи загалом не менш важливими стають відповідність законодавству, захист прав клієнтів, прозорість прийняття рішень і чіткий розподіл відповідальності між людиною та автоматизованими системами.

Нормативне середовище, у якому працюють банки, формується на кількох рівнях. Національні регулятори фінансового ринку встановлюють вимоги до організації систем внутрішнього контролю, фінансового моніторингу, захисту прав споживачів, а також до процесів управління ризиками. Паралельно діють міжнародні та наднаціональні рамки – стандарти у сфері протидії відмиванню коштів, регламенти із захисту персональних даних, директиви щодо платіжних послуг, а останніми роками – окремі акти, присвячені регулюванню систем штучного інтелекту. У результаті будь-яка AI-система в банку має бути не лише технічно хорошою, а й вбудованою в цю багаторівневу правову конструкцію.

Окрему, дуже чутливу площину становить захист персональних даних. Підходи, подібні до загального регламенту про захист даних (GDPR), задають базові принципи: дані мають оброблятися законно, прозоро та з чітко визначеною метою; обсяг збирання має бути обмежений необхідним;

передбачені права суб’єкта даних на доступ, виправлення, обмеження обробки тощо. Для антифрод-систем це критично, оскільки моделі часто будуються саме на поведінкових та транзакційних профілях: історія платежів, географія операцій, часові патерни активності, взаємодія з різними каналами (інтернет-банкінг, мобільний додаток, карткові операції). Банк має чітко обґрунтувати правові підстави такої обробки даних, забезпечити їх належне знеособлення, запровадити необхідні технічні та організаційні заходи захисту, а також поінформувати клієнтів про загальні принципи використання їхньої інформації в ризик-орієнтованих моделях.

Не менш важливою є нормативна площина платіжних сервісів та відкритого банкінгу. Директиви на кшталт PSD2 встановлюють вимоги до сильної клієнтської автентифікації, захисту доступу третіх сторін до рахунків, безперервного моніторингу операцій у режимі, максимально наближеному до реального часу. Моделі ШІ вбудовуються в ці процеси як частина системи контролю: вони аналізують ризики окремих транзакцій, можуть впливати на рішення щодо додаткової автентифікації чи блокування операції. Водночас регулятор очікує, що банк зможе довести керованість таких рішень, наявність контролю з боку людини, документування логіки спрацювань і механізмів реагування на помилки моделей.

Особливий блок вимог стосується протидії відмиванню коштів та фінансуванню тероризму. Міжнародні стандарти (рекомендації FATF) та національне законодавство вимагають від банків застосування ризик-орієнтованого підходу: моніторингу операцій, виявлення підозрілої активності, фіксації та подальшого аналізу всіх рішень. Використання ШІ тут розглядається як інструмент посилення аналітичних можливостей, але не як заміна відповідального працівника. У більшості випадків остаточне рішення щодо кваліфікації операції, направлення повідомлення до органів фінансового моніторингу чи блокування рахунку залишається за людиною. Такий підхід забезпечує обов’язкову участь людини в ухваленні рішень, що є важливим як з правової, так і з етичної точки зору: жодна модель не повинна самостійно

приймати критично важливі рішення без можливості їх перегляду та оскарження.

В останні роки формується й окрема регуляторна рамка для систем штучного інтелекту. Документи на кшталт AI Act у ЄС пропонують класифікацію AI-рішень за рівнем ризику та встановлюють додаткові вимоги для систем, що використовуються у високоризикових сферах, до яких відносять і фінансові послуги. Для таких систем акцент робиться на прозорості, можливості аудиту, детальній документації моделі, використовуваних даних і процесу навчання, а також на проведенні оцінки впливу на права та свободи людини. Для банку це означає, що впровадження будь-якої складної антифродмодель має супроводжуватися формалізованими процедурами валідації, тестування та періодичного перегляду.

Паралельно з юридичними вимогами постають етичні питання, які часто виявляються не менш важливими для довіри клієнтів та репутації установи. Одним із ключових принципів є справедливість. Алгоритми, що навчаються на історичних даних, можуть несвідомо відтворювати упередження, закладені в цих даних: наприклад, системно підвищувати ризикові оцінки для певних груп клієнтів через специфіку їхнього стилю використання продуктів, географії операцій чи соціально-економічних особливостей регіону. Навіть якщо формально модель не використовує заборонені або чутливі ознаки, поєднання непрямих показників може призводити до непрямої дискримінації. Відповідно, банки мають брати до уваги не лише формальні метрики якості, а й результати перевірок на наявність систематичних перекосів у рішеннях моделі.

Із цим тісно пов’язаний принцип прозорості та пояснюваності. Клієнт, чию операцію відхилено, або регулятор, що перевіряє роботу системи, очікують отримати хоча б загальне, але зрозуміле пояснення причин такого рішення: які фактори стали визначальними, чи не є логіка моделі внутрішньо суперечливою або надмірно чутливою до незначних змін вхідних даних. Це безпосередньо впливає на вибір типів моделей: надто закриті архітектури, що не підлягають жодному аналізу, можуть бути проблемними у сферах із підвищеними

вимогами до підзвітності. Саме тому дедалі ширше застосовуються методи Explainable AI, зокрема SHAP-аналіз, що дозволяють оцінити вклад окремих ознак у конкретне рішення та побудувати загальну картину роботи моделі на різних сегментах клієнтів.

Третій важливий етичний вимір – це відповідальність та підзвітність. Навіть якщо рішення технічно ухвалює автоматизована модель, відповідальність перед клієнтом та регуляторами несе фінансова установа. Це означає, що мають бути чітко прописані процеси життєвого циклу моделі: від етапу розробки та тестування до впровадження, супроводу, моніторингу якості, періодичної переоцінки та виведення з експлуатації. У практиці це відображається в концепціях на кшталт model risk management: банк має розуміти, які саме моделі використовуються, які ризики вони несуть, як часто перевіряється їхня адекватність і що відбувається у випадку виявлення суттєвих відхилень.

Окремого розгляду потребують питання конфіденційності та інформаційної безпеки. Для навчання, адаптації та моніторингу моделей використовується великий обсяг транзакційних даних, які містять фінансову та персональну інформацію про клієнтів. Витік таких даних або неконтрольований доступ до них може мати серйозні правові й репутаційні наслідки. Тому моделі ШІ не можна розглядати у відриві від загальної інфраструктури захисту інформації: важливими є шифрування, розмежування прав доступу, використання безпечних середовищ для навчання моделей, мінімізація копій даних і використання знеособлених або агрегованих наборів там, де це можливо.

Етичний дискурс також стосується балансу між автоматизацією та участю людини. З одного боку, ШІ дозволяє обробляти обсяги інформації, які неможливо охопити вручну, виявляти складні закономірності та реагувати на підозрілі операції впродовж секунд. З іншого боку, повна автоматизація без можливості людського втручання може пройти межу прийнятності для клієнтів і суспільства: помилки моделі, що не підлягають перегляду, сприймаються як

особливо несправедливі. Тому сучасні підходи орієнтуються на гібридні схеми: моделі виконують первинний аналіз і формують ризикові оцінки, а найбільш критичні рішення ухвалюються або підтверджуються відповідальними працівниками, які мають доступ до додаткового контексту.

У результаті нормативно-правові та етичні аспекти використання ШІ у фінансовому секторі формують складну рамку, у межах якої будь-яка антифрод-система має працювати як контрольований, прозорий і підзвітний інструмент. Йдеться не лише про формальне виконання законодавчих вимог, а й про дотримання принципів справедливості, пояснюваності, поваги до приватності та чіткого визначення відповідальності. Для систем виявлення шахрайства це означає, що моделі не можуть розглядатися виключно як технічні алгоритми; вони повинні бути інтегровані в комплексну систему управління ризиками, у якій технологічні рішення узгоджуються з правовими обмеженнями та етичними стандартами, а їхня поведінка підлягає постійному моніторингу та корекції. Такий підхід створює передумови для використання ШІ не лише як ефективного, а й як надійного та соціально прийнятного інструменту у фінансових системах.

Висновки до Розділу 1

У першому розділі було здійснено комплексний теоретико-методологічний аналіз проблематики виявлення шахрайства у фінансових системах в умовах цифровізації банківських послуг. Розглянуті аспекти дозволяють сформувати цілісне уявлення про те, в якому середовищі функціонують сучасні антифродмодель, які обмеження задають характеристики даних та регуляторні вимоги, а також які підходи машинного навчання є доцільними для використання у цій сфері.

По-перше, проаналізовано сучасну картину кіберзагроз для цифрового банкінгу та типові шахрайські схеми, до яких належать операції без фізичної присутності картки (CNP), захоплення облікових записів (Account Takeover) та

відмивання коштів через онлайн та міжбанківські канали. Показано, що ці види шахрайства мають спільні риси: вони реалізуються у цифровому середовищі, часто в автоматизованому режимі, характеризуються високою динамічністю та масовістю, а також тісно пов’язані з поведінковими характеристиками клієнтів і транзакційних потоків. Такі умови суттєво обмежують ефективність суто статичних та ручних підходів до контролю операцій.

По-друге, простежено еволюцію систем фінансового моніторингу: від переважно ручного аналізу та простих порогових правил до rule-based систем, скорингових моделей і, зрештою, до адаптивних рішень на основі штучного інтелекту. Було показано, що із зростанням обсягів транзакцій, різноманіття продуктів і каналів обслуговування жорстко зафіксовані правила перестають бути достатніми: вони погано масштабуються, потребують постійного ручного оновлення і часто генерують надмірну кількість помилкових спрацювань. Це обґрунтовує перехід до моделей, здатних автоматично вчитися на даних, враховувати контекст поведінки клієнта та адаптуватися до нових сценаріїв атак.

По-третє, у розділі розглянуто методи машинного навчання, які доцільно застосовувати для задачі бінарної класифікації транзакцій. Обґрунтовано вибір ансамблевих алгоритмів на основі дерев рішень (Random Forest, XGBoost) та багатошарового персептрону (MLP) як базових моделей, що добре працюють із табличними фінансовими даними, здатні враховувати складні взаємозв’язки між ознаками та досягати високих показників якості. Показано, що кожен із цих підходів має власні сильні сторони й обмеження, а тому особливий інтерес становлять гібридні рішення, зокрема стекінг-ансамблі, які поєднують переваги різних алгоритмів і забезпечують кращий баланс між точністю, повнотою та стійкістю моделі.

По-четверте, окреслено проблему критичного дисбалансу класів у фінансових даних, яка є визначальною для задач виявлення шахрайства. Пояснено, що за умов, коли шахрайські операції становлять лише незначну частку від загального обсягу транзакцій, моделі, побудовані без урахування

цього чинника, можуть демонструвати формально високу загальну точність, але майже не виявляти рідкісні шахрайські події. У зв’язку з цим розглянуто підходи до балансування навчальної вибірки, зокрема зменшення обсягу класу більшості (undersampling) та синтетичне розширення міноритарного класу методом SMOTE. Показано, що ці методи дозволяють покращити представлення рідкісних подій у даних та підвищити чутливість моделей до шахрайства, за умови їх обережного та усвідомленого застосування.

По-п’яте, проаналізовано нормативно-правові та етичні аспекти використання ШІ у фінансовому секторі. Наголошено на важливості дотримання вимог щодо захисту персональних даних, регулювання платіжних послуг, протидії відмиванню коштів, а також нових рамок, спрямованих безпосередньо на регулювання систем штучного інтелекту. Визначено ключові етичні принципи — справедливість, прозорість, пояснюваність, відповідальність та повага до приватності клієнтів. Підкреслено, що моделі ШІ мають розглядатися не як автономні чорні скриньки, а як елементи ширшої системи управління ризиками, у якій зберігається можливість людського контролю та оскарження рішень.

Загалом результати, отримані у розділі 1, дозволяють зробити кілька узагальнюючих висновків. По-перше, сучасні умови функціонування цифрового банкінгу об’єктивно вимагають переходу від статичних і суто правил-орієнтованих підходів до адаптивних антифрод-систем на основі машинного навчання. По-друге, для задачі виявлення шахрайства доцільним є використання ансамблевих та нейромережевих моделей, що працюють із транзакційними та поведінковими даними клієнтів, із можливістю їх подальшої інтеграції в гібридні схеми, такі як стекінг. По-третє, критичний дисбаланс класів та нормативно-правові обмеження задають жорсткі рамки для побудови моделей і вимагають комплексного підходу до підготовки даних, вибору метрик, пояснюваності рішень і організації контролю за роботою системи.

РОЗДІЛ 2. МЕТОДИКА ДОСЛІДЖЕННЯ ТА ПІДГОТОВКА ДАНИХ ДЛЯ ПОБУДОВИ АНТИФРОДМОДЕЛІ

Практична реалізація системи виявлення шахрайства неможлива без ретельного опрацювання вихідних даних та чітко сформованої методики дослідження. На цьому етапі увага зміщується від загальних теоретичних підходів до конкретного транзакційного датасету, на основі якого навчаються й оцінюються моделі машинного навчання. Важливо не лише описати, які саме дані використовуються, а й зрозуміти їхню структуру, якість, наявні викривлення та обмеження, оскільки всі ці чинники безпосередньо впливають на кінцеву ефективність антифродмодель.

Спочатку розглядається характеристика обраного набору транзакцій та результати розвідувального статистичного аналізу (EDA), що дозволяють виявити базові закономірності, аномалії, пропуски та співвідношення між легальними й шахрайськими операціями. Далі описується методика попередньої обробки даних: очищення від шумів і технічних записів, опрацювання пропущених значень, масштабування та нормалізація ознак відповідно до вимог моделей. Окремий акцент робиться на формуванні поведінкових ознак (Feature Engineering) — часових інтервалах, частотних характеристиках та агрегованих показниках, що відображають типову динаміку операцій клієнта. Завершальним елементом методики є реалізація балансування навчальної вибірки за допомогою методу SMOTE, спрямована на пом’якшення впливу критичного дисбалансу класів і підготовку даних до побудови гібридної антифродмодель, розробка та оцінка якої розглядається у наступному розділі.

2.1. Характеристика та розвідувальний статистичний аналіз (EDA) обраного датасету транзакцій

Практична частина дослідження ґрунтується на публічно доступному наборі даних транзакцій за банківськими картками, у якому інформацію про

клієнтів та реквізити було попередньо знеособлено, сформованому на основі реальних операцій платіжної системи.Цей датасет широко застосовується у наукових роботах та експериментах із побудови систем виявлення шахрайства, що робить результати дослідження порівнюваними з іншими підходами та підсилює їхню практичну значущість.

Загальний обсяг вибірки становить 284 807 записів, кожен рядок відповідає окремій транзакції. Для кожної операції доступно 31 змінну: Time, ознаки V1–V28, параметр Amount, а також цільова змінна Class, яка набуває значення 0 для коректних операцій і 1 для шахрайських. Усі ознаки, окрім Class, є числовими, пропущені значення відсутні, що значно полегшує базовий етап попередньої обробки даних. Основні характеристики датасету узагальнено в таблиці 2.1.

Таблиця 2.1Основні характеристики датасету транзакцій

Показник	Значення	Коментар
Загальна кількість транзакцій	284 807	Кількість рядків у вибірці (кожен рядок – окрема транзакція)
Кількість коректних транзакцій (Class = 0)	284 315 (≈ 99,83 %)	Транзакції без ознак шахрайства, клас більшості
Кількість шахрайських транзакцій (Class = 1)	492 (≈ 0,17 %)	Виявлені шахрайські операції, міноритарний клас
Частка шахрайських транзакцій	0,17 %	Ілюструє критичний дисбаланс класів
Кількість змінних загалом	31	30 вхідних ознак + 1 цільова змінна Class
Ознака Time – діапазон	0 – 172 792 (секунд)	Час відносно першої транзакції у датасеті; приблизно 2 доби спостереження
Ознака Time – середнє значення	94 813,86	Середній час відносно початку спостереження
Ознака Time – медіана	84 692,00	Половина транзакцій відбулася раніше, половина – пізніше цього

		моменту
Ознаки V1–V28	28 трансформованих числових ознак	Результат попередньої анонімізації/перетворень; не мають прямої бізнес-інтерпретації
Ознака Amount – мінімум	0,00	Мінімальна сума транзакції
Ознака Amount – максимум	25 691,16	Максимальна сума транзакції
Ознака Amount – середнє значення	88,35	Середній розмір платежу
Ознака Amount – медіана	22,00	Типовий (медіанний) розмір транзакції
Ознака Amount – стандартне відхилення	250,12	Висока варіативність сум, наявність великої кількості дрібних і поодиноких великих операцій
Цільова змінна Class	0 – коректна транзакція; 1 – шахрайська транзакція	Використовується як вихідна змінна в задачі бінарної класифікації

Особливість цього набору даних полягає в тому, що більшість ознак V1–V28 є результатом попередніх перетворень, які були виконані для збереження конфіденційності клієнтів та внутрішніх параметрів платіжної системи. У відкритому описі датасету зазначається, що ці змінні отримані за допомогою методів, подібних до компонентного аналізу, а отже вони не мають прямого інтерпретаційного зв’язку з конкретними бізнес-параметрами (тип мерчанта, географія, канал транзакції тощо). Таким чином, у межах цього дослідження акцент робиться не на буквальному змісті кожної з цих ознак, а на їхніх статистичних характеристиках, кореляціях та внеску у якість передбачення шахрайства моделями машинного навчання.

Змінна Time відображає час, що минув від моменту першої транзакції в наборі, і вимірюється в секундах. Значення Time варіюються від 0 до приблизно 172 792 секунд, що відповідає інтервалу близько двох діб спостереження. Розподіл цієї ознаки є нерівномірним: у певні проміжки часу щільність транзакцій помітно зростає, що може відображати пікові періоди активності користувачів (наприклад, робочі години або вечірні покупки). Уже на етапі

EDA це дає підстави припускати, що часовий контекст може відігравати важливу роль при формуванні поведінкових ознак — зокрема, інтервалів між операціями, інтенсивності транзакцій у певні часові вікна, тощо.

Параметр Amount характеризує суму транзакції. Його розподіл є типовим для платіжних даних: переважна більшість операцій здійснюється на невеликі суми, тоді як у правій частині розподілу спостерігається невелика кількість транзакцій із дуже великими значеннями. За описовими статистиками, середнє значення Amount становить близько 88,35, медіана — 22, мінімальне значення дорівнює 0,00, а максимальне перевищує 25 тис. Така асиметрія вказує на те, що поодинокі великі списання можуть суттєво впливати на масштаб ознаки, а отже потребують обережного врахування під час навчання моделей. Зокрема, це обґрунтовує використання процедур масштабування (standardization, robust scaling тощо), що дозволяють «вирівняти» вплив різних за абсолютним значенням ознак на процес оптимізації.

Ключовим аспектом EDA є аналіз цільової змінної Class та співвідношення між класами. Із 284 807 транзакцій лише 492 належать до шахрайських (Class = 1), тоді як 284 315 — до класу коректних (Class = 0). У відсотковому вираженні шахрайські операції становлять близько 0,17 % від усіх спостережень, а понад 99,8 % — це звичайні транзакції. Такий розподіл ілюструє вкрай сильний дисбаланс класів, який було детально охарактеризовано в теоретичній частині роботи і який виступає центральним викликом при побудові ефективної антифродмодель. Структуру класів доцільно показати за допомогою простої стовпчикової діаграми.

З рисунка 2.1 наочно видно, що шахрайські транзакції становлять лише мізерну частку від загального обсягу операцій. Така ситуація є типовою для реальних фінансових систем, але водночас серйозно ускладнює навчання моделей: алгоритм, орієнтований лише на мінімізацію загальної помилки, вчиться добре відтворювати поведінку класу більшості, майже не приділяючи уваги рідкісним, але критично важливим випадкам шахрайства. Саме тому на подальших етапах виникає об’єктивна потреба у застосуванні методів

балансування вибірки, зокрема SMOTE.

Рисунок 2.1 Розподіл транзакцій за класами (коректні та шахрайські операції)

Попередній аналіз параметра Amount окремо для кожного класу показує, що між коректними та шахрайськими операціями існують відмінності як у середніх значеннях, так і в характеристиках розподілу. Середнє значення суми для шахрайських транзакцій, як правило, вище, ніж для звичайних операцій, однак сама структура розподілу є більш складною: поряд із відносно великими списаннями зустрічаються й операції на незначні суми, які можуть виконувати роль тестових або використовуватись у схемах дроблення платежів. Це підтверджує тезу про те, що підозрілість транзакції не можна оцінювати лише за одним числовим параметром; необхідно враховувати сукупність ознак та їх взаємозв’язки.

Ознаки V1–V28, попри відсутність прямої бізнес-інтерпретації, демонструють низку корисних властивостей з точки зору моделювання. Аналіз їхніх розподілів показує, що більшість із них мають значення, сконцентровані навколо нуля, із відносно обмеженим діапазоном змін. Це спрощує роботу

алгоритмів, чутливих до масштабу ознак, і створює передумови для стабільнішого навчання моделей. У деяких ознак спостерігаються виражені витягнуті ділянки розподілу або поодинокі аномальні значення, що може свідчити про специфіку поведінки окремих груп клієнтів або про крайні випадки транзакційної активності. Такі спостереження надалі можуть бути враховані під час побудови моделей, а також при інтерпретації результатів Explainable AI-аналізу.

Кореляційний аналіз між ознаками V1–V28 виявляє здебільшого слабкі або помірні взаємозв'язки, що очікувано після застосування методів зниження вимірності. Водночас для окремих пар ознак кореляція є більш вираженою, що може відображати спільне джерело варіацій у вихідних даних. Більш практичний інтерес становить аналіз кореляції між цими ознаками та цільовою змінною Class: параметри, які демонструють більшу відмінність між коректними і шахрайськими транзакціями, відіграватимуть важливішу роль при навчанні моделей і можуть бути віднесені до найбільш інформативних. Надалі це відкриває можливості для поєднання автоматизованого відбору ознак із експертним аналізом їхньої важливості.

Додатковий напрям EDA пов'язаний із аналізом часових характеристик. Хоча вихідний датасет не містить явних ідентифікаторів клієнтів чи інформації про канали, поєднання ознаки Time із іншими параметрами дає змогу оцінювати, наприклад, інтенсивність транзакцій у певні періоди, частоту появи операцій із нетиповими сумами, можливі сплески активності, пов'язані з реалізацією шахрайських схем. Такі спостереження мають значення для подальшого етапу Feature Engineering, де з окремих транзакцій конструюються агреговані та поведінкові показники.

Узагальнюючи результати попереднього статистичного аналізу даних, можна виділити кілька ключових висновків, які визначають подальшу методику роботи з вибіркою. По-перше, датасет чітко демонструє критичний дисбаланс класів: шахрайські операції становлять менше відсотка від загального обсягу транзакцій, що робить неможливим використання простих підходів до навчання

моделей без спеціальних заходів балансування. По-друге, розподіли основних числових ознак (Amount, Time, трансформовані компоненти V1–V28) свідчать про доцільність масштабування даних та обережного поводження з екстремальними значеннями. По-третє, знеособлений характер ознак та особливості їхніх розподілів підкреслюють важливість побудови додаткових поведінкових характеристик, які краще відображають контекст транзакційної активності.

У сукупності ці спостереження визначають подальшу методику роботи з даними: передбачають ретельну попередню обробку транзакцій, побудову додаткових поведінкових ознак та застосування методів балансування вибірки (зокрема SMOTE), спрямованих на підвищення чутливості антифродмодель до рідкісних шахрайських подій.

2.2. Методика попередньої обробки даних: очищення від шумів, масштабування та нормалізація ознак

Ефективність антифродмодель значною мірою визначається не стільки вибором алгоритму, скільки якістю та коректністю підготовки вхідних даних. Для задачі виявлення шахрайства це особливо важливо, адже йдеться про рідкісні, але критично значущі події, на фоні яких домінують звичайні транзакції. Навіть потужні моделі машинного навчання не здатні продемонструвати стабільні результати, якщо навчальна вибірка містить технічні помилки, непродумані перетворення або ознаки з несумірними масштабами. Тому перед етапом моделювання було сформовано чітку послідовність кроків попередньої обробки, яка забезпечує цілісність даних, уніфікацію ознак та відсутність витоку інформації між навчальною й тестовою вибірками.

Вихідний датасет складається з 31 стовпчика: параметра Time, набору перетворених числових ознак V1–V28, суми транзакції Amount та цільової змінної Class. Усі змінні, окрім Class, мають числову природу, що спрощує побудову моделей і знімає потребу у додатковому кодуванні категоріальних

ознак. Водночас трансформований та знеособлений характер параметрів V1–V28 означає, що їхній прямий зміст не інтерпретується у термінах банківських продуктів, каналів чи типів клієнтів. Це робить особливо важливою коректну статистичну обробку та подальше конструювання поведінкових ознак, які краще відображають контекст транзакцій.

Першим кроком було здійснено перевірку цілісності вибірки. Було проаналізовано наявність пропущених значень, некоректних типів даних, дубльованих записів, а також очевидних помилок у діапазонах змінних (наприклад, від'ємні значення там, де вони неможливі з точки зору бізнес-логіки). На основі цього аналізу встановлено, що датасет не містить пропусків у ключових стовпчиках, а значення змінних знаходяться в очікуваних числових межах. Додатково було перевірено унікальність ідентифікаторів рядків, що дозволило зробити висновок про відсутність явних дублікатів транзакцій. Це дало можливість зберегти повний обсяг даних для подальшого моделювання, не вдаючись до видалення спостережень з технічних причин.

На наступному етапі було розмежовано матрицю ознак і цільову змінну. Усі стовпчики, окрім Class, були об'єднані у матрицю X, яка описує параметри транзакцій, тоді як бінарний стовпчик Class (0 — звичайна операція, 1 — шахрайська) формував вектор цільових значень y. Такий поділ відповідає постановці задачі бінарної класифікації та дозволяє однозначно визначити, що саме є входом для моделі, а що — бажаним результатом її роботи.

Важливим кроком стало формування навчальної й тестової вибірок. Датасет було розділено у пропорції 80/20, при цьому використовувалася стратифікація за ознакою Class. Стратифікований підхід забезпечує збереження приблизно однакового співвідношення звичайних і шахрайських транзакцій у кожній підвибірці, що особливо важливо за умов критичного дисбалансу класів. Це дозволяє отримати більш адекватну оцінку здатності моделей працювати з новими даними: тестова вибірка за своєю структурою краще відображає реальну ситуацію у транзакційних потоках банку, де шахрайські операції становлять лише незначну частку від загального обсягу.

Окремо було враховано питання уникнення витоку інформації (dataleakage). Усі подальші маніпуляції з балансуванням класів і масштабуванням ознак виконувалися лише на навчальній вибірці. Тестовий набір залишався недоторканим і використовувався виключно для фінальної оцінки якості моделей. Такий підхід гарантує, що моделі не пристосовуються до специфіки тестових даних ще на етапі підготовки, а отримані результати дійсно відображають здатність алгоритмів працювати з новими, раніше не баченими транзакціями.

Основні етапи попередньої обробки даних, їх зміст та особливості реалізації узагальнено в таблиці 2.2.

Таблиця 2.2 Основні етапи попередньої обробки даних

Етап	Зміст етапу	Особливості реалізації
1	Перевірка цілісності датасету	Аналіз пропусків, некоректних типів, дубльованих записів; підтверджено коректність і повноту даних
2	Відокремлення ознак і цільової змінної	Формування матриці ознак X (Time, V1–V28, Amount) та вектора цільової змінної y=Class
3	Розбиття на тренувальну і тестову вибірки	Поділ у пропорції 80/20 зі стратифікацією за класом для збереження дисбалансу та репрезентативності
4	Балансування тренувальної вибірки	Застосування SMOTE лише до навчальної вибірки для збільшення частки шахрайських операцій без зміни тестових даних
5	Навчання перетворення масштабування	Обчислення параметрів стандартизації (середнє та стандартне відхилення) за ознаками навчальної вибірки
6	Масштабування ознак	Застосування одного й того самого перетворення до навчальних і тестових даних для забезпечення порівнюваності ознак

Після узгодження структури вибірки наступним критичним аспектом стало масштабування ознак. У вихідному датасеті параметри Time і Amount мають свої природні діапазони, тоді як ознаки V1–V28 отримані в результаті попередніх перетворень і можуть мати різні середні значення та розмах. Якщо залишити змінні в такому вигляді, моделі, чутливі до масштабу (зокрема

багатошаровий персептрон), можуть фактично переорієнтовуватися на ознаки з найбільшими числовими діапазонами, ігноруючи внесок інших параметрів. Це призводить до нестабільності процесу навчання, сповільнює збіжність алгоритмів оптимізації та ускладнює інтерпретацію результатів.

Для розв'язання цієї проблеми було застосовано стандартизацію ознак за принципом z-score-нормалізації. Для кожної числової змінної перетворення виконується за формулою:

$$x_i' = \frac{x_i - \mu}{\sigma}$$

де x_i — вихідне значення ознаки, μ — середнє значення цієї ознаки в навчальній вибірці, σ — стандартне відхилення, а x_i' — стандартизоване значення.

За допомогою інструменту StandardScaler для кожної ознаки навчальної вибірки спочатку обчислювалися параметри μ та σ, після чого всі значення цієї ознаки переводилися у безрозмірну шкалу з порівнюваними діапазонами. Важливо, що параметри масштабу (середнє та стандартне відхилення) обчислювалися виключно на навчальній вибірці, а вже потім використовувалися для перетворення як навчальних, так і тестових даних. Такий підхід забезпечує узгодженість обробки та не допускає використання інформації з тестової частини на етапі попередньої підготовки.

Ще одним аспектом є поводження з потенційними шумами та аномальними значеннями. У транзакційних даних великі суми або нетипові поєднання параметрів не завжди свідчать про технічну помилку; нерідко саме такі спостереження несуть важливий сигнал про можливе шахрайство. Тому агресивні підходи до очищення даних, пов'язані з видаленням усіх викидів за формальними критеріями, у даному дослідженні не застосовувалися. Замість цього було обрано більш обережну стратегію: технічні аномалії відфільтровувалися на стадії базової перевірки, а нетипові, але можливі з точки зору бізнес-логіки операції зберігалися у вибірці, щоб моделі могли навчитися

розрізняти ризикову та коректну поведінку навіть у найбільш нетипових ситуаціях.

Загальну послідовність описаних етапів доцільно уявляти як єдиний конвеєр обробки даних, що починається з сирих транзакцій та завершується підготовленими ознаками, готовими до використання в моделях. Схематично цей процес може бути поданий у вигляді блок-схеми, де відображено основні стадії: завантаження початкового датасету, перевірка якості й цілісності, формування навчальної та тестової вибірок зі стратифікацією, балансування навчальних даних методом SMOTE, навчання та застосування скейлера, а також передавання підготовлених матриць ознак до модулів моделювання (рис. 2.2).

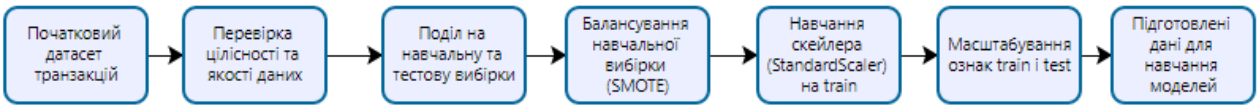


Рисунок 2.2 – Схема попередньої обробки даних для побудови антифродмодель

Узгоджена й формалізована методика попередньої обробки даних забезпечує відтворюваність експериментів, мінімізує ризик технічних спотворень результатів і створює надійне підґрунтя для подальших етапів — побудови поведінкових ознак та навчання моделей машинного навчання в задачі виявлення шахрайських транзакцій.

2.3. Feature Engineering: Розробка та обґрунтування поведінкових ознак(часові інтервали, частотні характеристики, агрегації)

Ефективність моделей виявлення шахрайських транзакцій значною мірою залежить не лише від вибору алгоритму (Random Forest, XGBoost, MLP тощо), а й від того, які саме ознаки подаються на вхід моделі. Базові параметри транзакції (час, сума, технічні атрибути авторизації) часто відображають лише

одноразовий стан операції й не дають повної картини поведінки клієнта у часі. Завдання етапу Feature Engineering полягає в тому, щоб перетворити ці базові поля на більш інформативні поведінкові характеристики, які дозволяють відрізнити типові сценарії використання карти від потенційно шахрайських.

На концептуальному рівні поведінкові ознаки зручно поділяти на три основні групи:

- часові характеристики (як, коли і з якою регулярністю клієнт зазвичай здійснює операції);
- частотні характеристики (як часто, з якою інтенсивністю та в яких «серіях» проводяться платежі);
- агреговані показники (узагальнений профіль поведінки за певний період або для окремих об'єктів – клієнта, мерчанта, каналу тощо).

У випадку часових ознак важливим є не лише сам факт моменту проведення транзакції, а й його позиція в добовому чи тижневому циклі. Для більшості клієнтів активність концентрується у певні години (наприклад, денний час у будні), тоді як нічні серії операцій або транзакції у нетипових часових інтервалах можуть бути маркером зловживань. Тому, окрім базового параметра Time, доцільно використовувати його перетворення: відносний час від початку спостереження, година доби, індикатори робочий/вихідний день, день/ніч тощо. У традиційних банківських системах на основі цих значень формують індивідуальний часовий профіль клієнта і порівнюють нові операції з його типовою поведінкою.

Частотні характеристики дозволяють описати інтенсивність і структуру платіжної активності. Багато шахрайських схем супроводжуються різким зростанням кількості транзакцій за короткий період, дробленням суми на низку дрібних платежів, повторними спробами списання коштів або тестовими транзакціями. Щоб зробити ці патерни помітними для моделей, використовують показники на кшталт кількості операцій за останні N хвилин/годин, сумарної або середньої суми в цьому вікні, мінімальної та максимальної суми, середнього інтервалу між послідовними транзакціями та ін.

Такі ознаки добре відображають короткочасні піки активності, характерні для зловмисників, але нетипові для звичайних клієнтів.

Агреговані ознаки описують накопичений досвід взаємодії із системою за довший період часу. Сюди можуть входити: тривалість обслуговування, типові діапазони сум, середній розмір платежу, домінуючі категорії торгових точок, географія операцій, частка відхилених або повернених транзакцій, стабільність чи, навпаки, різка зміна поведінки. Такі показники дозволяють моделі бачити не одну окрему операцію, а її контекст у межах ширшого профілю клієнта або мерчанта.

У контексті даного дослідження можливості формування повноцінного набору поведінкових ознак обмежені структурою обраного датасету credit card fraud detection, у якому:

- відсутні явні ідентифікатори клієнта чи торгової точки;
- більшість змінних (V1–V28) представлені у вже перетвореному вигляді (компоненти після попередньої трансформації, наприклад, PCA);
- наявні явні поля Time, Amount та бінарний клас Class.

Це означає, що класичні клієнтські профілі (на основі ID клієнта, мерчанта, карти) побудувати неможливо, однак є можливість працювати з доступними полями Time та Amount, а також використовувати V1–V28 як латентні носії інформації про структуру транзакцій. У цьому сенсі акцент робиться на:

- перетворенні Time (масштабування, приведення до більш зручної шкали, аналіз розподілу у часі);
- коректній обробці Amount з урахуванням асиметрії розподілу та наявності поодиноких великих значень;
- використанні компонент V1–V28 як вхідних ознак, що вже містять певну зашифровану поведінкову інформацію.

Для ілюстрації можливих груп поведінкових ознак використовується узагальнювальна таблиця 2.3, у якій приклади часових, частотних та агрегованих характеристик розподілено за групами з коротким поясненням

змісту та зазначенням, чи можуть вони бути реалізовані в межах обраного датасету.

Таблиця 2.3 Приклади поведінкових ознак для задачі виявлення шахрайських транзакцій

Група ознак	**Приклад ознаки**	**Короткий зміст**	**Можливість реалізації в обраному датасеті**
Часові	Година доби транзакції	Позиція операції у добовому циклі клієнта	Обмежено (Time → похідні ознаки)
Часові	Індикатор «ніч / день»	Відокремлення нічної активності від денної	Так (на основі Time)
Частотні	Кількість операцій за останню годину	Інтенсивність транзакцій у ковзному часовому вікні	Ні (бракує ID клієнта)
Частотні	Середній інтервал між операціями	Ритм використання картки	Ні (бракує прив’язки до клієнта)
Агреговані	Середня сума транзакцій клієнта	Типовий розмір платежів	Ні (немає ID клієнта)
Агреговані	Частка відхилених операцій	Схильність до ризикової поведінки або технічних збоїв	Ні (у датасеті немає статусів авторизації)
Використання V1–V28	Компоненти V1–V28	Приховані (латентні) характеристики транзакцій	Так (пряме використання як ознак)
Час + сума (комбіновані)	Нормована сума операції у часі	Врахування того, які суми є типовими для різних періодів	Частково (через Amount та Time)

Така структура ознак дає змогу зафіксувати, що повноцінний набір поведінкових характеристик для промислової антифродмодель був би значно ширшим, однак у межах цього дослідження використовується підмножина показників, які реально можна сформувати з наявного публічного датасету. Логіка формування поведінкових ознак проілюстрована на рисунку 2.3: змінні

Time, Amount та V1–V28 розглядаються як вихідні транзакційні дані, які після етапу Feature Engineering перетворюються на простір ознак, що подається на вхід моделі.

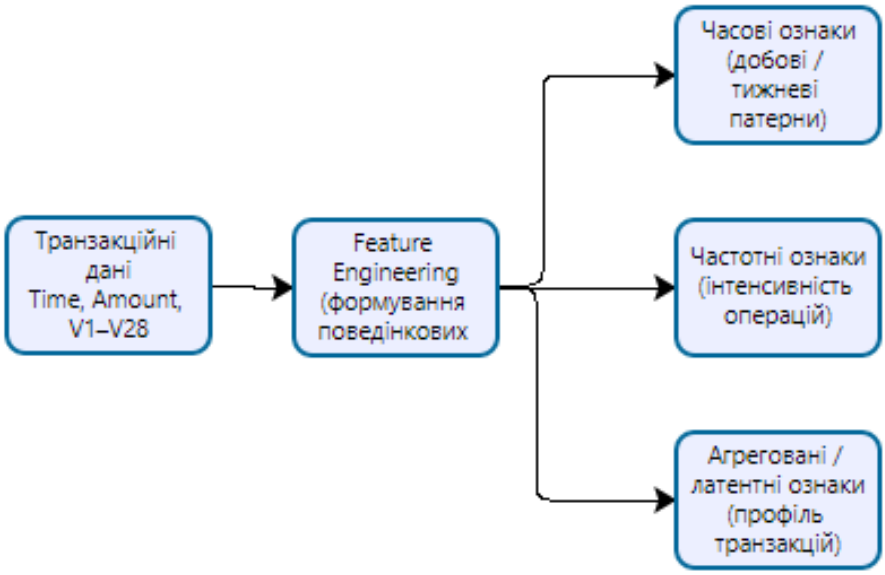


Рисунок 2.3 Концептуальна схема формування поведінкових ознак на основі сирих транзакційних даних

Окремо варто підкреслити, що розробка ознак відбувається в тісному зв’язку з проблемою дисбалансу класів. Поведінкові характеристики мають бути:

- достатньо стабільними для більшості коректних операцій;
- водночас чутливими до відхилень, властивих шахрайським сценаріям;
- не надто «крихкими» до шуму та випадкових аномалій.

Занадто агресивні або надто специфічні ознаки можуть призводити до ситуації, коли модель починає «запам’ятовувати» окремі випадки, замість того щоб виявляти загальні закономірності, що негативно позначається на здатності працювати з новими даними.

Етап Feature Engineering виступає проміжною, проте визначальною ланкою між початковими транзакційними записами та модельними алгоритмами. На цьому кроці числові поля Time, Amount і латентні компоненти

V1–V28 набувають змістового наповнення у вигляді часових, частотних та агрегованих поведінкових ознак. Така трансформація формує більш інформативний простір ознак для навчання моделей машинного навчання у задачі виявлення шахрайських транзакцій і підвищує ймовірність коректного відокремлення рідкісних шахрайських подій від переважної більшості коректних операцій.

2.4. Алгоритмічна реалізація балансування навчальної вибірки методом SMOTE для підвищення чутливості моделі

Як показав попередній аналіз датасету, розподіл класів є різко нерівномірним: із 284 807 транзакцій лише 492 належать до шахрайських, тобто їх частка становить приблизно 0,17 %. Після поділу даних на навчальну та тестову частини у співвідношенні 80/20 зі стратифікацією за класом у навчальній вибірці залишилося 227 451 коректна операція та лише 394 шахрайські. За такого співвідношення більшість стандартних алгоритмів машинного навчання зосереджуються насамперед на правильній класифікації звичайних платежів, фактично ігноруючи рідкісні випадки шахрайства. Формально модель може демонструвати дуже високу загальну точність, але при цьому майже не виявляти потрібний клас.

Щоб зменшити вплив цього чинника, у роботі було застосовано штучне балансування навчальної вибірки методом SMOTE. Основна ідея полягає в тому, що нові приклади міноритарного класу не копіюються, а генеруються на основі вже наявних шахрайських транзакцій. Для кожного обраного елемента міноритарного класу знаходяться кілька найближчих до нього спостережень цього ж класу у просторі ознак. Далі між точками будується відрізок, і нове синтетичне спостереження розміщується в його середині або ближче до одного з кінців. Математично це можна подати у такому вигляді:

$$x_{new} = x_i + \lambda \cdot (x_j - x_i),$$

де x_j та x_i – два близькі приклади шахрайського класу, а λ— випадкове число в інтервалі [0;1]. У такий спосіб утворюється додаткова точка, що зберігає характерні риси обох вихідних спостережень, але не збігається з ними повністю.

У реалізації було використано модуль SMOTE з бібліотеки imblearn.over_sampling. Процес побудови збалансованої навчальної вибірки включав кілька послідовних кроків. Спочатку з повного датасету формувалися навчальна та тестова вибірки з фіксованою пропорцією 80/20 та стратифікацією за змінною Class, що забезпечувало збереження початкової частки шахрайських операцій у кожній підмножині даних. Далі SMOTE застосовувався лише до навчальної частини, тоді як тестові дані залишалися незмінними і відображали реальний дисбаланс класів. Це принципове обмеження: тестова вибірка використовується для незалежної оцінки якості моделей, тому будь-які перетворення, що змінюють її структуру, є недопустимими.

Наступним кроком було налаштування параметрів SMOTE. Було обрано стратегію балансування, за якої кількість синтетичних прикладів шахрайського класу збільшується до рівня класу більшості (співвідношення 1:1 у навчальній вибірці). Кількість найближчих сусідніх спостережень, що використовуються для побудови нових точок, встановлювалася на стандартному значенні k=5, а параметр random_state фіксувався для забезпечення відтворюваності експериментів. Згенеровані у такий спосіб транзакції зберігали структуру вихідних ознак, але розташовувалися між реальними шахрайськими прикладами, заповнюючи проріджені ділянки простору ознак.

Після застосування SMOTE було сформовано збалансовану навчальну вибірку, у якій обидва класи мають однакову кількість прикладів. Це дозволило моделям машинного навчання приділяти шахрайським операціям таку саму увагу, як і звичайним, що, своєю чергою, підвищує чутливість до рідкісних ризикових подій. Структуру навчальної вибірки до та після балансування подано в таблиці 2.4.

Таблиця 2.4 Розподіл класів у навчальній вибірці до та після застосування SMOTE

Стан вибірки	Клас 0 – коректні операції	Клас 1 – шахрайські операції
До балансування (train)	227 451	394
Після балансування (train*)	227 451	227 451

Ефект балансування полягає у суттєвій зміні співвідношення між кількістю коректних та шахрайських операцій у навчальній вибірці: якщо до застосування SMOTE шахрайський клас був представлений лише незначною кількістю спостережень, то після балансування його обсяг вирівнюється відносно класу коректних транзакцій. Візуальне порівняння початкового та збалансованого розподілів класів наведено на рисунку 2.4.

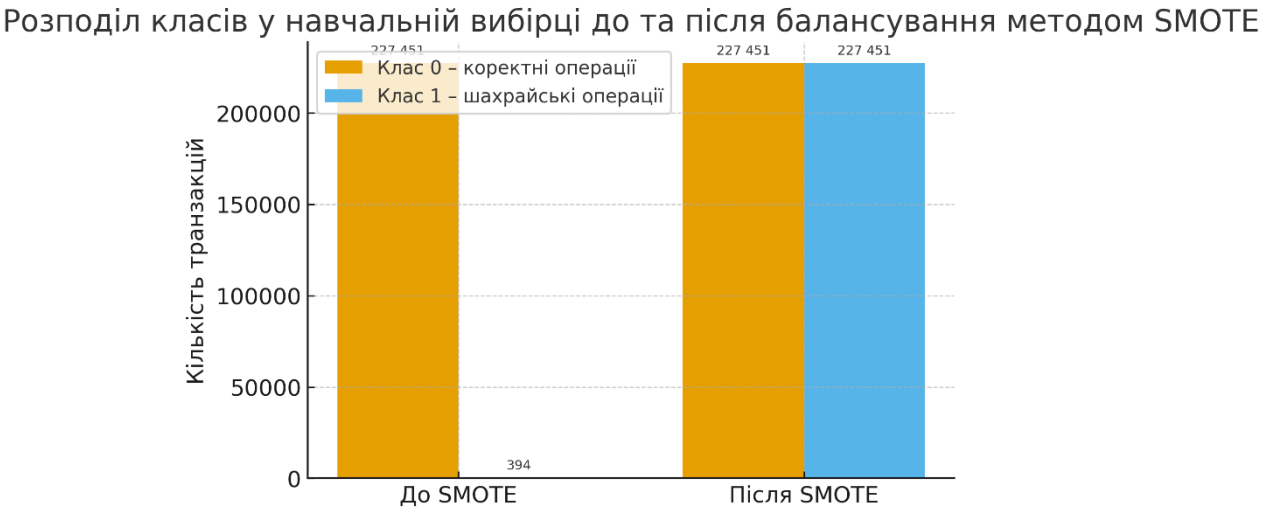


Рисунок 2.4 Розподіл класів у навчальній вибірці до та після балансування методом SMOTE

Спочатку формується навчальна вибірка, застосовується алгоритм балансування, і лише потім до збалансованих даних підбираються параметри стандартизації (середнє значення та стандартне відхилення для кожної ознаки). Далі ці параметри використовуються як для перетворення збалансованого train-набору, так і для початкової (небалансованої) тестової вибірки. Така послідовність дій запобігає ситуації, коли інформація про тестові дані впливає

на етап попередньої обробки, і забезпечує коректність подальшого порівняння моделей.

Застосування SMOTE не усуває повністю всі проблеми, пов’язані з дисбалансом класів, однак суттєво покращує представлення шахрайського класу в навчальних даних. У поєднанні з уважним вибором метрик оцінювання (Precision, Recall, F_1-score, ROC-AUC) та подальшою оптимізацією гіперпараметрів базових і ансамблевих моделей це створює підґрунтя для побудови більш чутливих і надійних антифрод-рішень, здатних працювати в умовах реальних транзакційних потоків, де шахрайські операції залишаються рідкісними, але надзвичайно важливими для виявлення.

Висновки до Розділу 2

У другому розділі було сформовано цілісну методику підготовки даних для побудови антифродмодель, орієнтованої на виявлення шахрайських транзакцій у середовищі цифрового банкінгу. На основі аналізу обраного публічного датасету встановлено його ключові особливості: великий загальний обсяг спостережень, різко виражений дисбаланс класів (≈0,17 % шахрайських операцій), відсутність пропусків та наявність трансформованих латентних компонент V1–V28. Це дозволило окреслити обмеження, з якими доводиться працювати моделі, та визначити вимоги до подальших етапів обробки даних.

Послідовність попередньої обробки охоплює перевірку цілісності вибірки, розмежування матриці ознак та цільової змінної, стратифікований поділ на навчальну і тестову частини, масштабування ознак за принципом z-score-нормалізації та обережне поводження з потенційними аномаліями. Окремо наголошено на запобіганні витоку інформації: всі перетворення, пов’язані з балансуванням класів і стандартизацією, виконуються виключно на навчальній вибірці, тоді як тестовий набір зберігає вихідну структуру даних і використовується лише для незалежної оцінки якості моделей.

На етапі Feature Engineering було обґрунтовано роль поведінкових ознак у задачі виявлення шахрайства. Попри те, що структура обраного датасету не дозволяє формувати повноцінні клієнтські профілі, показано, як за допомогою доступних змінних Time, Amount та латентних компонент V1–V28 можна конструювати часові, частотні та агреговані характеристики. Такі ознаки дають змогу більш повно відобразити динаміку транзакційної активності та посилити здатність моделей відрізняти типову поведінку від потенційно ризикових сценаріїв.

Окремий акцент зроблено на алгоритмічній реалізації балансування навчальної вибірки методом SMOTE. Показано, що генерування синтетичних представників міноритарного класу дає змогу суттєво покращити представлення шахрайських операцій у навчальних даних, не змінюючи при цьому структуру тестової вибірки. У результаті формується збалансований train-набір із рівною кількістю коректних і шахрайських транзакцій, що підвищує чутливість моделей до рідкісних подій і створює кращі умови для навчання алгоритмів бінарної класифікації.

Було визначено та формалізовано повний цикл підготовки даних — від початкового аналізу та очищення до побудови поведінкових ознак і балансування вибірки. Сформована методика забезпечує відтворюваність експериментів, зменшує ризик технічних викривлень і створює методологічно вивірену основу для подальшого етапу дослідження, пов’язаного з побудовою, навчанням і порівняльною оцінкою антифрод-моделей машинного навчання.

РОЗДІЛ 3. ПРОЕКТУВАННЯ, НАВЧАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ГІБРИДНОЇ AI-СИСТЕМИ

Проєктування, навчання та дослідження гібридної AI-системи виявлення шахрайства ґрунтується на підготовленому транзакційному датасеті, для якого вже виконано очищення, масштабування, формування поведінкових ознак та балансування класів. На основі цих даних будується набір моделей машинного навчання, здатних працювати в умовах критичного дисбалансу, високих вимог до чутливості до шахрайських операцій та збереження прийнятного рівня хибних спрацювань. У роботі послідовно досліджуються як окремі базові моделі (Random Forest, XGBoost, MLP), так і їхня гібридна комбінація у форматі стекінг-ансамблю, що дозволяє оцінити потенціал поєднання ансамблевих методів на основі дерев рішень і нейромережевих підходів.

Подальший виклад зосереджується на побудові та налаштуванні базових класифікаторів, розробці конфігурації стекінг-моделі, організації процесу навчання на збалансованій навчальній вибірці та оцінюванні якості рішень на тестових даних із реальним дисбалансом класів із використанням метрик Precision, Recall, F_1-score та ROC-AUC. Порівняння результатів окремих моделей і гібридної стекінг-схеми дає змогу обґрунтувати доцільність застосування досліджуваних підходів у реальних фінансових потоках та визначити, які з них є найбільш придатними для інтеграції в інфраструктуру антифрод-моніторингу.

3.1. Розробка архітектури гібридної моделі Stacking Ensemble: обґрунтування вибору базових класифікаторів та мета-моделі

У межах даного дослідження для задачі бінарної класифікації платіжних транзакцій було застосовано гібридний підхід у форматі Stacking Ensemble. Ідея стекінгу полягає в поєднанні кількох різнорідних моделей, кожна з яких формує власну оцінку ризику для однієї й тієї самої транзакції. Далі спеціальна мета-

модель навчається на цих оцінках і генерує фінальне рішення. Таким чином, замість пошуку єдиного «найкращого» алгоритму використовується композиція моделей, що спираються на різні підходи до побудови рішень і частково компенсують слабкі сторони одна одної.

У роботі послідовно досліджуються три алгоритми класифікації: Random Forest, XGBoost та MLP (Multilayer Perceptron). Спочатку кожна модель налаштовується й оцінюється окремо, після чого на основі найкращих конфігурацій будується гібридний ансамбль. При цьому у фінальній конфігурації стекінг-ансамблю як базові класифікатори першого рівня використовуються Random Forest та XGBoost, а компактний MLP виступає у ролі мета-моделі другого рівня. Окремий, більш «масивний» MLP розглядається також як самостійний базовий класифікатор, який порівнюється з ансамблем за якістю роботи.

Коротко охарактеризуємо обрані алгоритми.

- Random Forest. Модель забезпечує стійкість до шуму та локальних аномалій, добре працює з великою кількістю числових ознак (Time, Amount, V1–V28) і дозволяє оцінювати важливість окремих параметрів. У попередніх експериментах на збалансованій навчальній вибірці Random Forest показав стабільні значення основних метрик за помірної чутливості до налаштування гіперпараметрів, тому доцільно використовувати його як один із базових класифікаторів першого рівня.
- XGBoost. Реалізує градієнтний бустинг над деревами рішень і орієнтований на поетапне виправлення помилок попередніх дерев. Практичні результати засвідчили, що XGBoost здатний досягати високих значень ROC-AUC та F_1-score на збалансованих даних (після застосування SMOTE), однак потребує ретельного налаштування глибини дерев, швидкості навчання, кількості ітерацій та параметрів регуляризації. У стекінг-ансамблі XGBoost доповнює Random Forest за рахунок більшої чутливості до складних, тонких закономірностей у даних.

- MLP (Multilayer Perceptron). Використовується у двох режимах: як окрема нейромережева модель-класифікатор і як компактна мета-модель другого рівня у складі стекінгу. Завдяки послідовності нелінійних перетворень MLP моделює складні взаємозв'язки між сумою й часом транзакцій, а також латентними компонентами V1–V28. У практичній частині показано, що за коректного вибору архітектури (кількість нейронів, функції активації, параметри регуляризації) MLP досягає конкурентних значень метрик, але є більш вимогливим до якості попередньої обробки даних. Компактний варіант MLP доцільно використовувати як мета-модель, яка узагальнює ймовірнісні прогнози базових алгоритмів.

Схематичну структуру гібридної стекінг-моделі, реалізованої в дослідженні, наведено на рисунку 3.1.

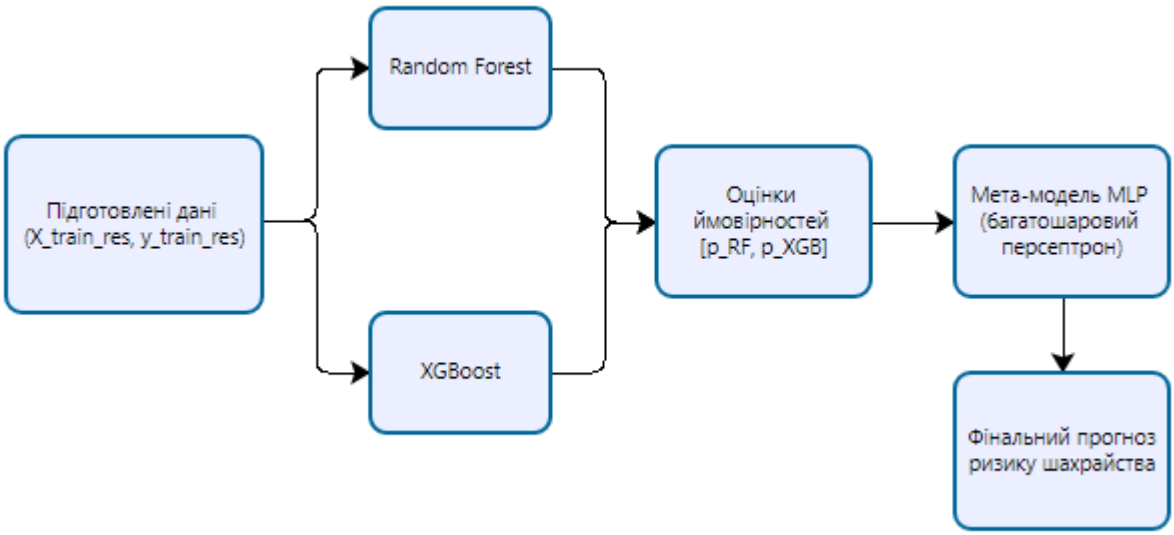


Рисунок 3.1 Архітектура гібридної моделі Stacking Ensemble для виявлення шахрайських транзакцій

На лівому боці схеми розміщено блок «Підготовлені дані», що відповідає матриці ознак та цільовій змінній після етапів, описаних у розділі 2: поділу на train/test із стратифікацією, балансування навчальної вибірки методом SMOTE та стандартизації ознак за принципом z-score. На вхід базових моделей стекінгу подається пара $(X_{\text{train\_res}}, y_{\text{train\_res}})$,де $X_{\text{train\_res}}$ – збалансовані (після SMOTE)

ознаки, а $y_{\text{train\_res}}$ – відповідні мітки класів (0 – легітимна транзакція, 1 – шахрайська).

У центральній частині схеми зображені два базові класифікатори першого рівня – Random Forest та XGBoost, які незалежно один від одного навчаються на збалансованій навчальній вибірці й для кожної транзакції формують ймовірнісні оцінки належності до шахрайського класу (Class = 1). Отримані значення можна позначити як p_{RF} та p_{XGB}.

. На практиці ці оцінки повертаються методами predict_proba() відповідних моделей.

Далі ймовірності p_{RF} і p_{XGB} групуються в нову матрицю ознак другого рівня$[p_{RF}, p_{XGB}]$, яка подається на вхід мета-моделі. У реалізації, що використовується в роботі, як мета-модель обрано компактний багатошаровий персептрон (MLP) із меншою кількістю нейронів, ніж у базового MLP-класифікатора. Завдання цієї моделі полягає в тому, щоб навчитися поєднувати «точки зору» базових алгоритмів у єдину фінальну оцінку ризику транзакції, підвищивши стійкість рішень до помилок окремих моделей.

Технічну реалізацію стекінгу виконано за допомогою класу StackingClassifier бібліотеки scikit-learn. У коді практичної частини базові оцінювачі задаються як пари «назва–модель», наприклад ("rf", rf_clf) та ("xgb", xgb_clf),а як final_estimator використовується компактний MLPClassifier із попередньо визначеною архітектурою. Параметр stack_method="predict_proba" забезпечує використання саме ймовірнісних прогнозів базових моделей як вхідних ознак другого рівня. Навчання мета-моделі відбувається на out-of-fold-прогнозах базових класифікаторів (через внутрішню крос-валідацію cv=5 у StackingClassifier), що зменшує ризик інформаційного витоку між рівнями та підвищує надійність оцінки.

Гіперпараметри кожної з моделей підібрано на основі серії експериментів, у яких Random Forest, XGBoost і MLP спочатку навчалися окремо на збалансованій навчальній вибірці. Для кожного алгоритму було протестовано кілька конфігурацій (кількість дерев, глибина, швидкість

навчання, розмір прихованих шарів тощо), після чого обрано робочі варіанти, що демонстрували прийнятний компроміс між показниками Precision, Recall, F_1-score та ROC-AUC на тестових даних. Саме ці налаштування використано при формуванні стекінг-ансамблю. Узагальнену інформацію про ключові гіперпараметри наведено в таблиці 3.1.

Таблиця 3.1 Основні гіперпараметри базових моделей та мета-моделі Stacking Ensemble

Модель	**Тип алгоритму**	**Ключові гіперпараметри (фрагмент)**	**Роль у стекінг-моделі**
Random Forest (базовий)	Ансамбль дерев рішень (bagging)	n_estimators = 200; min_samples_leaf = 2; n_jobs = -1; random_state = RANDOM_STATE	Базовий класифікатор першого рівня, стійкий до шуму, дає оцінку важливості ознак
XGBoost (базовий)	Градієнтний бустинг над деревами рішень	n_estimators = 300; learning_rate = 0.05; max_depth = 5; subsample = 0.8; colsample_bytree = 0.8; eval_metric = "logloss"; random_state = RANDOM_STATE; n_jobs = -1	Потужний базовий класифікатор першого рівня, орієнтований на виявлення тонких закономірностей у даних
MLP (базовий)	Багатошаровий персептрон (нейронна мережа)	hidden_layer_sizes = (64, 32); activation = "relu"; solver = "adam"; max_iter = 500; random_state = RANDOM_STATE	Окремий нелінійний класифікатор, що моделює складні взаємозв’язки між ознаками
MLP (meta_model)	Багатошаровий персептрон – мета-модель	hidden_layer_sizes = (32, 16); activation = "relu"; solver = "adam"; max_iter = 300; random_state = RANDOM_STATE	Класифікатор другого рівня, який поєднує ймовірнісні прогнози базових моделей у фінальну оцінку ризику
StackingClassifier	Стекінг-ансамбль (мультимодельна схема)	estimators=[("rf", rf_clf), ("xgb", xgb_clf)]; final_estimator = meta_model;	Інтегрує базові моделі RF і XGBoost та мета-модель MLP у

		stack_method="predict_proba"; cv=5; n_jobs=-1	єдину гібридну систему виявлення шахрайства

Після побудови архітектури стекінг-ансамблю було здійснено порівняння його роботи з окремими базовими моделями. У практичній частині для Random Forest, XGBoost, MLP та Stacking Ensemble побудовано ROC-криві на одній і тій самій тестовій вибірці з початковим дисбалансом класів. Відповідний графік наведено на рисунку 3.2, де для кожної моделі показано форму ROC-кривої та значення площі під кривою (AUC), отримане в ході експериментів.

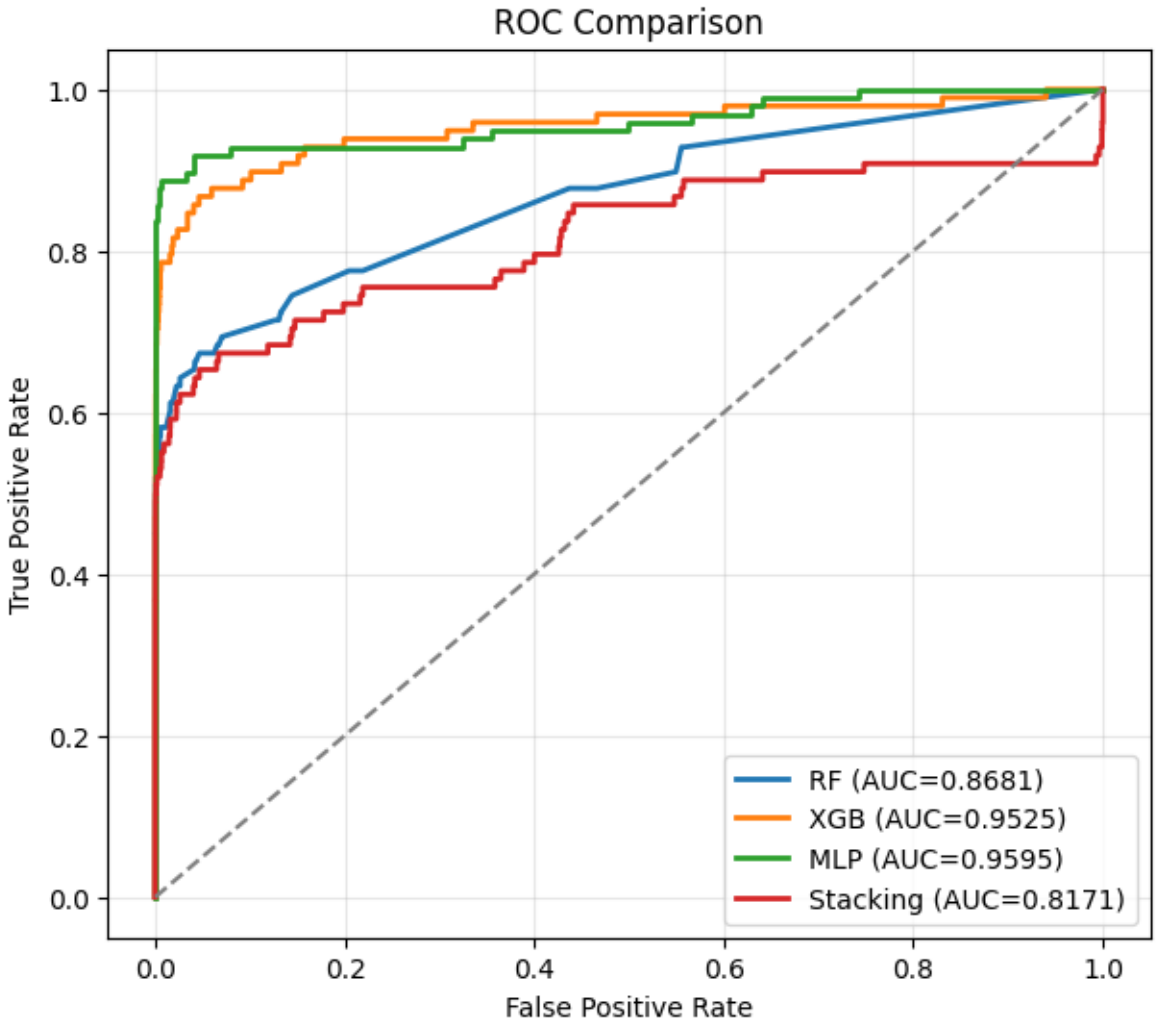


Рисунок 3.2 Порівняння ROC-кривих базових моделей та гібридної моделі Stacking Ensemble на тестовій вибірці

Аналіз рисунка 3.2 показує, що найвищі значення площі під ROC-кривою демонструють моделі MLP (AUC = 0,9595) та XGBoost (AUC = 0,9525), тоді як

Random Forest поступається їм за якістю (AUC = 0,8681), а стекінг-ансамбль у поточній конфігурації має найнижчий показник (AUC = 0,8171). Це свідчить про те, що окрема нейромережева модель та градієнтний бустинг краще узагальнюють структуру збалансованих даних після застосування SMOTE й ефективніше вловлюють нелінійні взаємозв'язки між ознаками. Водночас отримані результати для Stacking Ensemble вказують, що обрана комбінація базових моделей і параметрів мета-моделі не забезпечує очікуваного синергетичного ефекту: мета-класифікатор, імовірно, не повною мірою використовує інформацію з out-of-fold-прогнозів базових алгоритмів і частково переобучається на шумі. Це підкреслює чутливість стекінг-схем до вибору архітектури та гіперпараметрів і потребу в додатковій оптимізації конфігурації ансамблю.

Числові результати порівняння узагальнюються в таблиці 3.2, де для кожної моделі наведено значення Precision, Recall, F_1-score та ROC-AUC. Така форма подання дозволяє кількісно оцінити, як змінюється баланс між точністю та повнотою при переході від окремих класифікаторів до гібридної стекінг-моделі та які з підходів є найбільш доцільними для подальшої інтеграції в реальну систему моніторингу.

Таблиця 3.2 Порівняння основних метрик якості для базових моделей та Stacking Ensemble на тестовій вибірці

Модель	**Precision**	**Recall**	**F_1-score**	**ROC-AUC**
Random Forest	0,6500	0,1327	0,2203	0,8681
XGBoost	0,8857	0,3163	0,4662	0,9525
MLP	0,7431	0,8265	0,7826	0,9595
Stacking Ensemble	0,5000	0,0204	0,0392	0,8171

Аналіз даних таблиці 3.2 показує, що найкращий компроміс між точністю та повнотою забезпечує модель MLP. Вона демонструє найвище значення ROC-AUC (0,9595) та суттєво кращий F_1-score (0,7826) порівняно з іншими підходами, що свідчить про здатність нейромережі коректно виявляти більшість шахрайських операцій (Recall = 0,8265) при збереженні прийнятного рівня хибних спрацьовувань (Precision = 0,7431). На рисунку 3.3 наведено ROC-

криву моделі MLP на тестовій вибірці, де високе значення площі під кривою підтверджує її здатність ефективно відокремлювати шахрайські транзакції від легітимних.

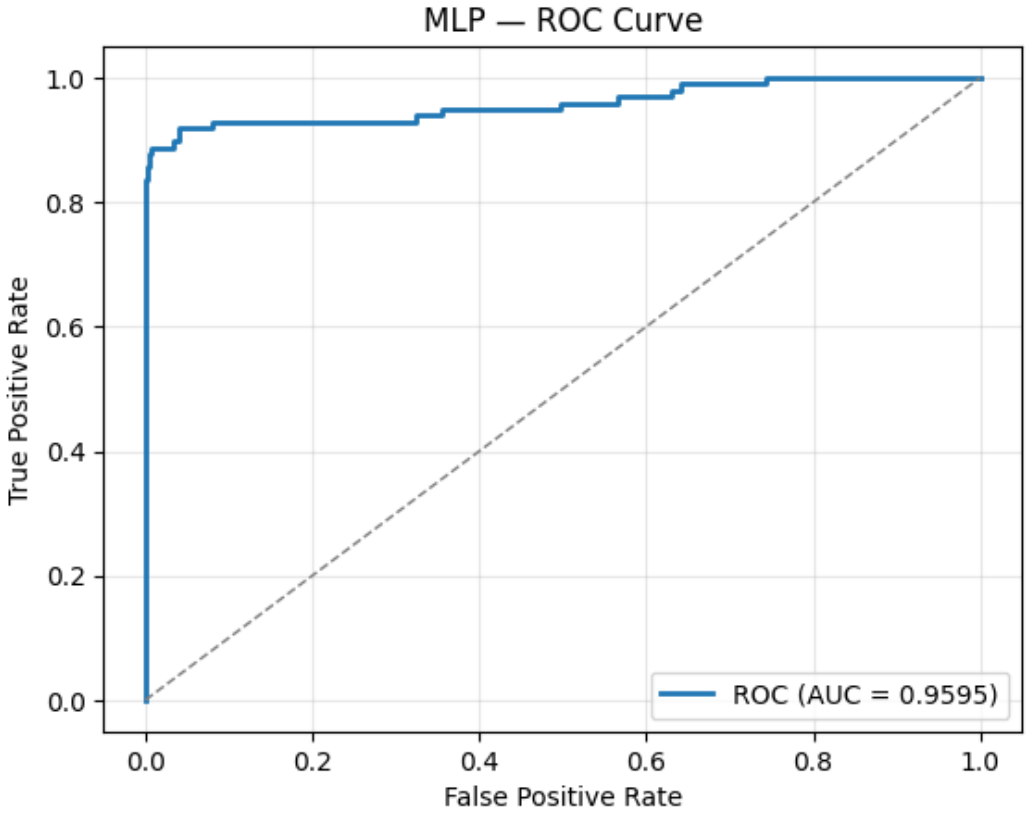


Рисунок 3.3 ROC-крива моделі MLP на тестовій вибірці

Крива Precision–Recall для цієї ж моделі подана на рисунку 3.4. Високе значення середньої точності (AP = 0,8331) та відносно полога ділянка кривої в області високих значень Recall свідчать про те, що модель зберігає достатньо високу точність навіть за умови пріоритизації виявлення максимальної кількості шахрайських операцій.

Модель XGBoost також показує високий рівень якості (ROC-AUC = 0,9525) і відзначається дуже високою точністю для шахрайського класу (Precision = 0,8857), однак має нижчу повноту (Recall = 0,3163), тобто виявляє меншу частку реальних шахрайських транзакцій. Random Forest, хоча й демонструє непогані результати за AUC (0,8681), суттєво поступається за F_1-score та Recall, що робить його менш придатним для задачі, де критичною є мінімізація пропущених випадків шахрайства.

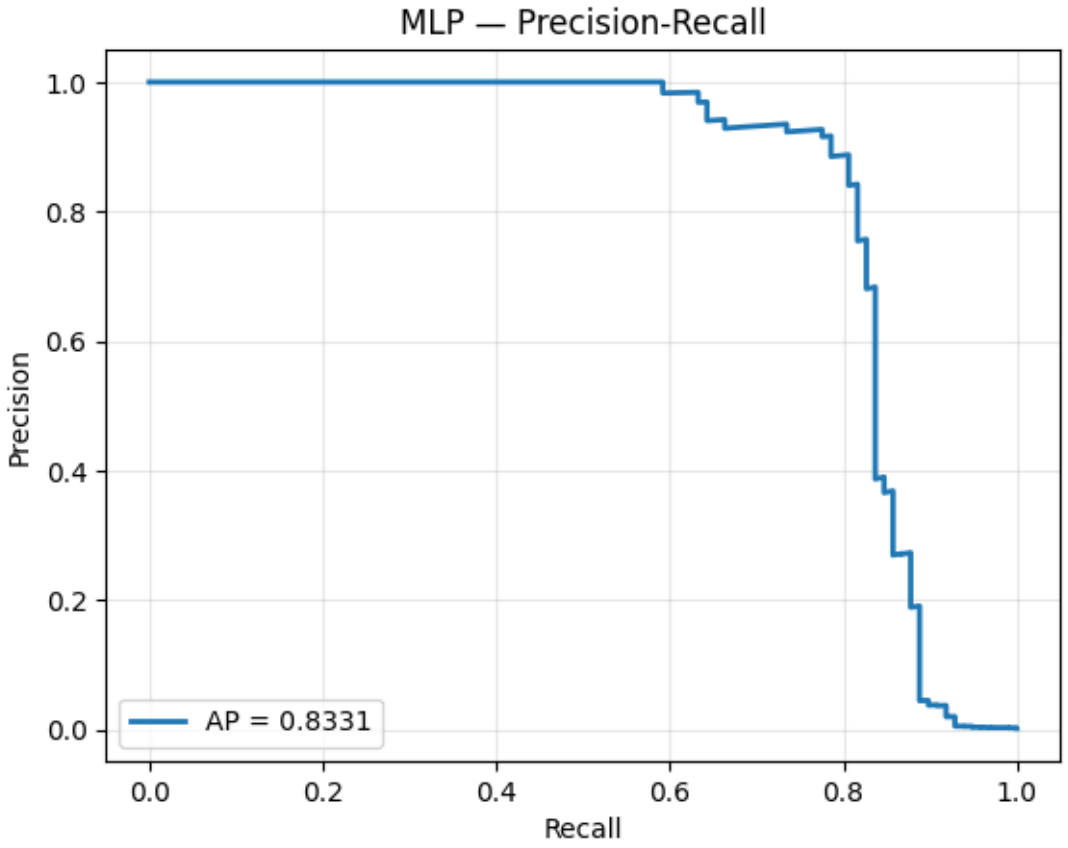


Рисунок 3.4 Крива Precision–Recall для моделі MLP на тестовій вибірці

Водночас гібридний стекінг-ансамбль у поточній конфігурації не забезпечив очікуваного покращення якості. Попри формально високу загальну точність класифікації, значення ROC-AUC (0,8171), а особливо дуже низькі показники Recall (0,0204) та F_1-score (0,0392) для шахрайського класу свідчать про те, що більшість аномальних транзакцій залишаються невиявленими. Це може бути зумовлено тим, що мета-модель не змогла ефективно узагальнити ймовірнісні прогнози базових алгоритмів, а також тим, що стандартний поріг прийняття рішення (0,5) виявився завищеним для даної конфігурації ансамблю. Таким чином, отримані результати демонструють, що добре налаштована окрема нейромережева модель (MLP) може перевершувати формальний стекінг-ансамбль і є більш доцільним кандидатом для інтеграції в реальну систему моніторингу платіжних транзакцій.

З огляду на прикладний характер роботи важливо також продемонструвати програмну реалізацію описаної архітектури. У лістингу 3.1

наведено фрагмент Python-коду, який демонструє створення та навчання стекінг-моделі: визначення базових класифікаторів, задання мета-моделі, навчання на збалансованій навчальній вибірці та отримання ймовірнісних прогнозів на тестових даних. Цей лістинг фіксує конкретні налаштування моделі, забезпечує відтворюваність експериментів і підкреслює практичну реалізованість запропонованої гібридної схеми для задачі виявлення шахрайських транзакцій.

Лістинг 3.1 Фрагмент програмної реалізації гібридної моделі Stacking Ensemble мовою Python

```
from sklearn.ensemble import RandomForestClassifier, StackingClassifier
from xgboost import XGBClassifier
from sklearn.neural_network import MLPClassifier

# Базові класифікатори першого рівня
rf_clf = RandomForestClassifier(
    n_estimators=200,
    min_samples_leaf=2,
    n_jobs=-1,
    random_state=RANDOM_STATE
)

xgb_clf = XGBClassifier(
    n_estimators=300,
    learning_rate=0.05,
    max_depth=5,
    subsample=0.8,
    colsample_bytree=0.8,
    use_label_encoder=False,
    eval_metric="logloss",
    random_state=RANDOM_STATE,
    n_jobs=-1
)

# Мета-модель (MLP) другого рівня
meta_model = MLPClassifier(
    hidden_layer_sizes=(32, 16),
    activation="relu",
```

```
    max_iter=300,
    random_state=RANDOM_STATE
)

# Побудова стекінг-ансамблю
stack_clf = StackingClassifier(
    estimators=[("rf", rf_clf), ("xgb", xgb_clf)],
    final_estimator=meta_model,
    stack_method="predict_proba",
    n_jobs=-1
)

# Навчання стекінг-моделі на збалансованій вибірці
stack_clf.fit(X_train_res, y_train_res)

# Отримання ймовірнісних прогнозів на тестових даних
y_prob_stack = stack_clf.predict_proba(X_test_scaled)[:, 1]
y_pred_stack = stack_clf.predict(X_test_scaled)
```

3.2. Процес навчання моделей та оптимізація гіперпараметрів (Grid Search CV) для максимізації метрик

Процес навчання моделей та налаштування їхніх гіперпараметрів у даному дослідженні побудовано таким чином, щоб, з одного боку, максимально використати інформацію, отриману після попередньої обробки даних (очищення, SMOTE, масштабування), а з іншого — забезпечити коректну оцінку якості алгоритмів у реалістичних умовах високого дисбалансу класів. Вихідний датасет банківських транзакцій було розділено на навчальну та тестову вибірки у співвідношенні 80/20 з використанням стратифікованого розбиття, що дозволило зберегти пропорцію між легітимними та шахрайськими операціями як у train-, так і в test-наборі. Надалі всі процедури балансування та налаштування моделей виконувалися лише на навчальній вибірці, тоді як тестова слугувала незалежною репрезентативною підмножиною для фінальної перевірки.

Для боротьби з дисбалансом класів на навчальній вибірці застосовано метод Synthetic Minority Oversampling Technique (SMOTE). У результаті його

використання було сформовано збалансований тренувальний набір $(X_{train\_res}, y_{train\_res})$, у якому кількість об'єктів шахрайського класу (Class = 1) штучно зрівнювалась із кількістю об'єктів легітимного класу (Class = 0). Такий підхід дозволяє базовим класифікаторам не ігнорувати рідкісні аномальні випадки й краще навчатися розпізнавати закономірності, характерні для шахрайських транзакцій. Тестова вибірка (X_{test}, y_{test}) при цьому залишалася незмінною, із вихідним дисбалансом, що є критично важливим для адекватного вимірювання Precision, Recall, F_1-score та ROC-AUC у прикладній постановці задачі.

Подальший етап передбачав масштабування числових ознак за допомогою StandardScaler. Скейлер навчався на збалансованих ознаках X_{train_res}, після чого використовувався для трансформації як тренувальної вибірки (отримуючи $X_{train\_res\_scaled}$), так і тестової (отримуючиX_{test_scaled}). Масштабовані ознаки були особливо важливими для нейромережевої моделі MLP, яка чутлива до масштабу вхідних даних і швидше та стабільніше збігається при навчанні на стандартизованих ознаках. У практичній реалізації базовий MLP навчався саме на $X_{train\_res\_scaled}$, тоді як Random Forest і XGBoost навчалися на незмасштабованих збалансованих ознаках X_{traine_res}, оскільки деревні алгоритми менш чутливі до масштабу числових полів. Для стекінг-ансамблю як вхідні дані першого рівня також використовувалася збалансована вибірка ($X_{train\_res}, y_{train\_res}$), що узгоджується зі схемою, описаною в розділі 3.1.

Оптимізація гіперпараметрів моделей виконувалась у кілька кроків і була організована за принципами, близькими до Grid Search з перехресною валідацією (Grid Search CV). На початковому етапі для кожного алгоритму (Random Forest, XGBoost, MLP) було визначено набір найбільш впливових гіперпараметрів, які суттєво впливають на складність моделі, її здатність до узагальнення та чутливість до дисбалансу. Далі для цих гіперпараметрів задавалася дискретна сітка значень (grid), що включає як більш консервативні, так і агресивні налаштування. На збалансованій навчальній вибірці

($X_{train\_res}, y_{train\_res}$) для кожної комбінації параметрів проводилася крос-валідація (k-fold), у ході якої оцінювалися F_1-score та ROC-AUC для шахрайського класу. Отримані результати порівнювалися між собою, а найкращі конфігурації фіксувалися й надалі використовувалися в експериментальному пайплайні.

Узагальнений фрагмент сітки гіперпараметрів, використаної при налаштуванні моделей, наведено в таблиці 3.3. Зокрема, для Random Forest варіювалися кількість дерев та мінімальний розмір листа, для XGBoost — кількість ітерацій бустингу, швидкість навчання, глибина дерев та частки підвибірки ознак/об'єктів, для MLP — розміри прихованих шарів і кількість ітерацій навчання.

Таблиця 3.3 Приклад сітки гіперпараметрів, використаної для налаштування моделей

Модель	**Гіперпараметри, що налаштовувалися**	**Приклад діапазону значень (фрагмент)**
Random Forest	n_estimators, min_samples_leaf, class_weight	n_estimators ∈ {100, 200, 300}; min_samples_leaf ∈ {1, 2, 4}; class_weight ∈ {"balanced", None}
XGBoost	n_estimators, learning_rate, max_depth, subsample, colsample_bytree	n_estimators ∈ {200, 300}; learning_rate ∈ {0,03; 0,05; 0,10}; max_depth ∈ {3, 5, 7}; subsample ∈ {0,8; 1,0}; colsample_bytree ∈ {0,8; 1,0}
MLP (базовий)	hidden_layer_sizes, activation, max_iter	hidden_layer_sizes ∈ {(32, 16), (64, 32)}; activation ∈ {"relu"}; max_iter ∈ {200, 500}
MLP (meta_model)	hidden_layer_sizes, max_iter	hidden_layer_sizes ∈ {(16, 8), (32, 16)}; max_iter ∈ {200, 300}

За результатами такого перебірного налаштування (grid-підходу з елементами перехресної валідації) було обрано конкретні значення гіперпараметрів, наведені в таблиці 3.1. Для Random Forest це конфігурація з n_estimators = 200 та min_samples_leaf = 2, яка забезпечила стійку роботу моделі без суттєвого перенавчання. Для XGBoost оптимальною виявилася

зв’язка n_estimators = 300, learning_rate = 0,05 та max_depth = 5 за subsample = 0,8 і colsample_bytree = 0,8, що дозволило моделі досягати високих значень ROC-AUC і F_1-score на збалансованій навчальній вибірці. Для базового MLP найкращі результати показала архітектура із двома прихованими шарами по 64 та 32 нейрони (hidden_layer_sizes = (64, 32)) з функцією активації ReLU та кількістю ітерацій max_iter = 500, що забезпечило стабільну збіжність алгоритму градієнтного спуску та високе значення F_1-score для шахрайського класу.

Після фіксації оптимальних гіперпараметрів було побудовано кінцевий стекінг-ансамбль, описаний у розділі 3.1: Random Forest та XGBoost використовувалися як базові класифікатори першого рівня, а компактний MLP із архітектурою (32, 16) — як мета-модель другого рівня. Навчання стекінг-моделі здійснювалося на збалансованій вибірці ($X_{train\_res}, y_{train\_res}$), причому всередині StackingClassifier застосовувалася внутрішня крос-валідація (cv = 5) для формування out-of-fold-прогнозів базових моделей, які потім подавалися на вхід мета-моделі. Такий підхід зменшує ризик інформаційного витоку між рівнями ансамблю та наближений до концепції Grid Search CV, коли і базові, і мета-моделі тестуються на валідаційних фолдах перед фінальним тренуванням на всіх доступних тренувальних даних.

Фінальна оцінка якості моделей проводилась на незбалансованій тестовій вибірці (X_{test}, y_{test}), що відображає реальну структуру даних банківських транзакцій. У програмній реалізації для цього використовувалися спеціальні допоміжні функції, які обчислювали й виводили основні показники якості: AUC-ROC, середню точність (Average Precision), Precision, Recall, F_1-score для кожного класу, а також формували текстовий звіт classification_report. Крім того, для кожної моделі будувалися ROC-криві, криві Precision–Recall та матриці невідповідностей (confusion matrix), що дозволяло візуально проаналізувати співвідношення між правильно класифікованими легітимними й шахрайськими транзакціями та структурою помилок (пропущені шахрайства, хибні тривоги тощо).

Отримані числові результати F_1-score, Precision, Recall та ROC-AUC для всіх чотирьох моделей (Random Forest, XGBoost, MLP, Stacking Ensemble) зведено в таблицю 3.2, а порівняльні ROC-криві наведено на рисунку 3.2. Саме ці значення лягають в основу подальшого аналізу й інтерпретації поведінки моделей у реальних умовах потокової обробки транзакцій, що буде детальніше розглянуто в наступному підрозділі.

3.3. Порівняльний аналіз ефективності моделей (RF, XGBoost, MLP vs Stacking) на тестовій вибірці

Після побудови та навчання всіх моделей – Random Forest, XGBoost, MLP та гібридного стекінг-ансамблю – було проведено порівняльний аналіз їхньої ефективності на тестовій вибірці, що зберігає вихідний дисбаланс класів. Такий підхід є принципово важливим, оскільки дозволяє оцінити, як моделі поводяться в умовах, наближених до реальної експлуатації системи моніторингу платіжних транзакцій, де шахрайські операції становлять лише невелику частку від загального обсягу даних.

Для кожної моделі було обчислено значення основних метрик якості: Precision, Recall, F_1-score та ROC-AUC. Ці показники дозволяють оцінити баланс між точністю виявлення шахрайських транзакцій та кількістю пропущених аномалій. Количні результати зведено в таблицю 3.2, а графічну інтерпретацію у вигляді ROC-кривих для всіх моделей наведено на рисунку 3.2. На основі цих даних здійснюється інтерпретація сильних і слабких сторін кожного з підходів.

Аналіз таблиці 3.2 показує, що найкращий компроміс між точністю (Precision) та повнотою (Recall) забезпечує модель MLP. Вона демонструє найвище значення ROC-AUC (0,9595) та суттєво кращий F_1-score (0,7826) порівняно з іншими підходами. Це свідчить про здатність нейромережі коректно виявляти більшість шахрайських операцій (Recall = 0,8265) при збереженні прийнятного рівня хибних спрацьовувань (Precision = 0,7431). У

контексті практичних антифрод-систем це означає, що MLP дозволяє значною мірою зменшити як ризики фінансових втрат від пропущених шахрайств, так і навантаження на службу підтримки, пов’язане з великою кількістю помилкових блокувань легітимних транзакцій.

Модель XGBoost також показує високий рівень якості (ROC-AUC = 0,9525) і відзначається дуже високою точністю для шахрайського класу (Precision = 0,8857), однак має нижчу повноту (Recall = 0,3163). Це означає, що XGBoost добре «довіряє» тільки найхарактернішим для шахрайства транзакціям, але пропускає значну частину менш виражених аномалій. Такий профіль роботи може бути прийнятним для сценаріїв, де критично важливо мінімізувати кількість помилкових спрацювань на легітимні операції (false positives), однак у задачах виявлення шахрайства зазвичай пріоритет надається максимальному виявленню підозрілих операцій, навіть ціною деякого зростання кількості таких помилкових тривог.

Random Forest демонструє доволі посередні результати в контексті шахрайського класу. Попри прийнятне значення ROC-AUC (0,8681), модель має низькі показники Recall (0,1327) та F_1-score (0,2203). Це означає, що значна кількість шахрайських транзакцій залишається невиявленою, що є критичним недоліком для домену фінансової безпеки. У зв’язку з цим Random Forest доцільно розглядати радше як базову референсну модель, яка демонструє мінімальний прийнятний рівень якості, але не є оптимальним кандидатом для впровадження без додаткових удосконалень.

Найбільш несподіваним результатом експериментів стало те, що гібридний стекінг-ансамбль у поточній конфігурації не забезпечив очікуваного покращення якості порівняно з окремими моделями. Попри формально прийнятне значення ROC-AUC (0,8171), значення Recall (0,0204) та F_1-score (0,0392) для шахрайського класу виявилися вкрай низькими. Це означає, що в більшості випадків шахрайські транзакції не розпізнаються моделлю як аномальні. Причинами таких результатів можуть бути: недостатньо вдала конфігурація мета-моделі, некоректний баланс внеску базових класифікаторів у

фінальне рішення або використання стандартного порога прийняття рішення (0,5), який виявився завищеним для даного ансамблю.

З метою наочності отримані результати доповнено узагальненою інтерпретацією роботи кожної моделі. У таблиці 3.4 наведено короткі характеристики їхньої поведінки з точки зору практичного застосування в системах моніторингу.

Таблиця 3.4 Узагальнена характеристика моделей за результатами експериментів

Модель	**ROC-AUC**	**F_1-score**	**Коротка характеристика роботи моделі**
Random Forest	0,8681	0,2203	Забезпечує прийнятну якість розмежування класів, але виявляє лише незначну частку шахрайських транзакцій; може використовуватись як базова референсна модель.
XGBoost	0,9525	0,4662	Демонструє високу точність для шахрайського класу та хороші значення ROC-AUC, однак має відносно низьку повноту й може пропускати частину аномальних операцій.
MLP	0,9595	0,7826	Забезпечує найкращий баланс між точністю та повнотою, виявляє більшість шахрайських транзакцій при прийнятному рівні хибних спрацьовувань; є пріоритетним кандидатом для практичного впровадження.
Stacking Ensemble	0,8171	0,0392	Не реалізував очікуваного синергетичного ефекту: має найнижчі значення Recall та F_1-score, пропускаючи більшість шахрайських транзакцій; потребує подальшої оптимізації архітектури й порогів прийняття рішень.

Таким чином, порівняльний аналіз підтверджує, що добре налаштована нейромережева модель MLP є найбільш придатною для задачі виявлення шахрайських транзакцій у фінансових установах серед розглянутих підходів. Моделі Random Forest та XGBoost доцільно розглядати як альтернативні або додаткові компоненти у складі розширених ансамблевих схем, однак у ролі основного інструмента моніторингу найбільш виправданим є використання MLP. Гібридний стекінг-ансамбль, попри теоретичні переваги, у поточній

конфігурації показав гірші результати, що вказує на потребу в більш глибокій оптимізації його архітектури та параметрів або перегляді самого підходу до побудови ансамблю.

3.4. Оцінка результатів за метриками Precision, Recall, F_1-score та візуалізація (ROC-AUC, Confusion Matrix)

Оцінювання ефективності розроблених моделей виявлення шахрайських транзакцій виконувалося на тестовій вибірці з початковим (реальним) дисбалансом класів, тобто без додаткового балансування методами на кшталт SMOTE. Такий підхід дозволяє перевірити, наскільки отримані моделі здатні узагальнювати знання, набуті на збалансованому тренувальному наборі, та коректно працювати в умовах, наближених до реальних потоків фінансових операцій.

Базовою групою метрик, що використовувалися для порівняння моделей Random Forest, XGBoost, MLP та Stacking Ensemble, стали Precision, Recall та F_1-score для шахрайського класу (Class = 1), а також інтегральний показник ROC-AUC. Саме ці метрики є більш інформативними для задачі виявлення шахрайства, ніж звичайна точність (accuracy), оскільки остання може бути штучно завищеною через домінування легітимних транзакцій у датасеті.

Метрика Precision (точність) для шахрайського класу показує, яка частка транзакцій, позначених моделлю як шахрайські, насправді є шахрайськими (тобто частку коректно виявлених позитивів серед усіх спрацювань моделі). Високе значення Precision означає низьку частку хибних тривог, що є важливим для зменшення навантаження на системи ручної перевірки та збереження лояльності клієнтів.

Метрика Recall (повнота, чутливість) відображає, яку частку реальних шахрайських операцій модель змогла виявити серед усіх дійсно шахрайських транзакцій. Для фінансових установ цей показник є критично важливим,

оскільки пропуск навіть невеликої кількості шахрайських операцій може призвести до значних збитків та репутаційних ризиків.

Метрика F_1-score є гармонійним середнім між Precision та Recall і дозволяє оцінити баланс між здатністю моделі не пропускати шахрайство і не генерувати надмірну кількість хибних спрацювань. Високе значення F_1-score свідчить про те, що модель досягає прийнятного компромісу між точністю та повнотою.

Для кращого розуміння ролі кожної з метрик у контексті задачі виявлення шахрайства доцільно узагальнити їхню інтерпретацію (табл. 3.5).

Таблиця 3.5 Загальна характеристика основних метрик оцінювання моделей

Метрика	**Формальний зміст**	**Інтерпретація в задачі виявлення шахрайства**
Precision	Частка коректно виявлених шахрайських транзакцій серед усіх транзакцій, позначених моделлю як шахрайські	Наскільки "довіряти" спрацюванню моделі; чим вище Precision, тим менше легітимних операцій помилково блокуються
Recall	Частка коректно виявлених шахрайських транзакцій серед усіх реально шахрайських операцій	Наскільки добре модель "бачить" більшість шахрайських операцій; низький Recall означає значну кількість пропущених шахрайств
F_1-score	Гармонійне середнє між Precision та Recall	Узагальнений показник балансу між виявленням шахрайства і кількістю хибних спрацювань
ROC-AUC	Площа під ROC-кривою (TPR vs FPR для різних порогів)	Інтегральна оцінка здатності моделі відокремлювати шахрайські транзакції від легітимних на всьому діапазоні порогів прийняття рішень

Крім табличних метрик, у роботі активно використовувалися візуальні інструменти оцінювання – ROC-криві та матриці неточностей (confusion matrix). ROC-криві для всіх розглянутих моделей наведено на рисунку 3.2 (порівняння Random Forest, XGBoost, MLP та Stacking Ensemble), а для моделі MLP окрема ROC-крива показана на рисунку 3.3. Високе значення площі під ROC-кривою (AUC = 0,9595) підтверджує здатність MLP ефективно

відокремлювати шахрайські транзакції від легітимних на широкому діапазоні порогів, що є важливим для подальшого практичного налаштування системи.

Окремо проаналізовано криву Precision–Recall для моделі MLP (рис. 3.4), яка є особливо інформативною в умовах сильного дисбалансу класів. Високе значення середньої точності (AP = 0,8331) та форма кривої свідчать про те, що модель зберігає прийнятний рівень Precision навіть за високих значень Recall, тобто здатна виявляти значну частку шахрайських операцій без різкого зростання кількості хибних спрацювань.

Ще одним важливим інструментом інтерпретації результатів є матриця неточностей. Для кожної з моделей (Random Forest, XGBoost, MLP та Stacking Ensemble) у практичній частині було побудовано матриці неточностей на тестовій вибірці, які відображають кількість:

- TP (True Positives) – коректно виявлені шахрайські транзакції;
- FP (False Positives) – легітимні транзакції, помилково позначені як шахрайські;
- FN (False Negatives) – шахрайські транзакції, які модель не розпізнала;
- TN (True Negatives) – коректно класифіковані легітимні транзакції.

На рисунках 3.5–3.8 наведено матриці неточностей для моделей Random Forest, XGBoost, MLP та Stacking Ensemble відповідно.

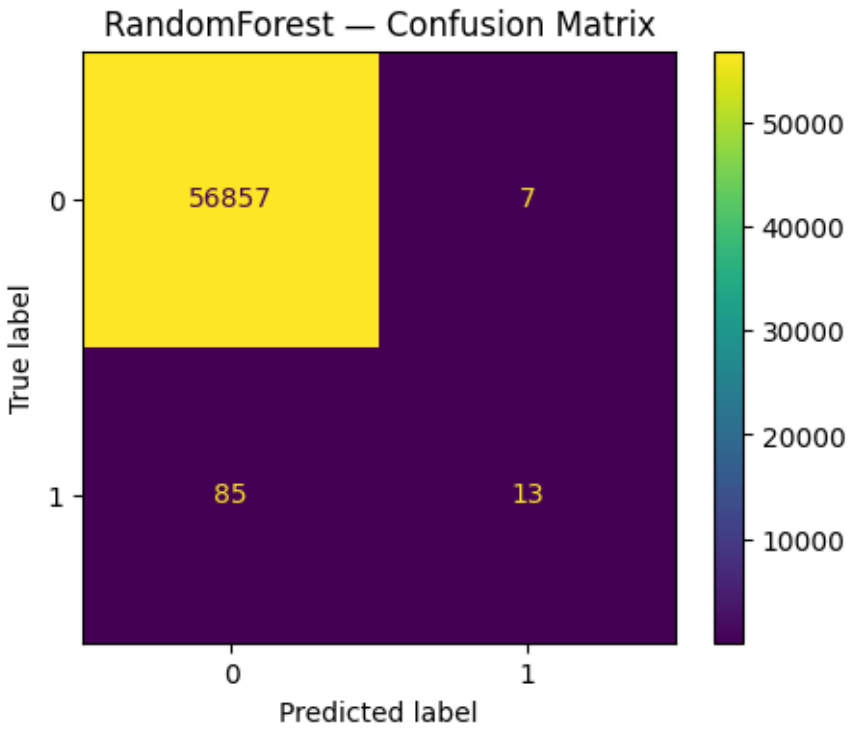


Рисунок 3.5 Матриця неточностей моделі Random Forest на тестовій вибірці

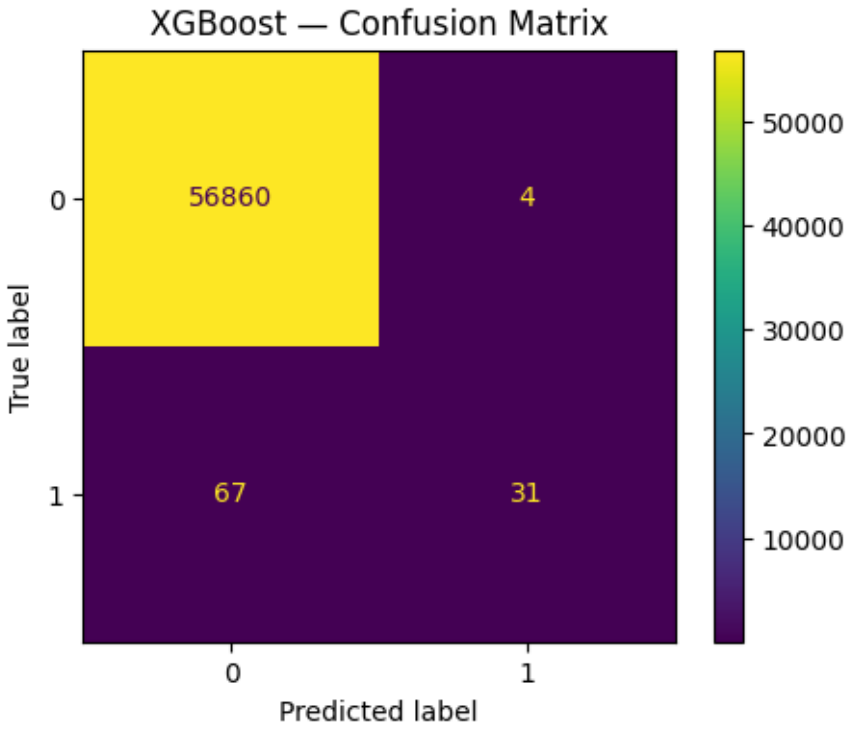


Рисунок 3.6 Матриця неточностей моделі XGBoost на тестовій вибірці

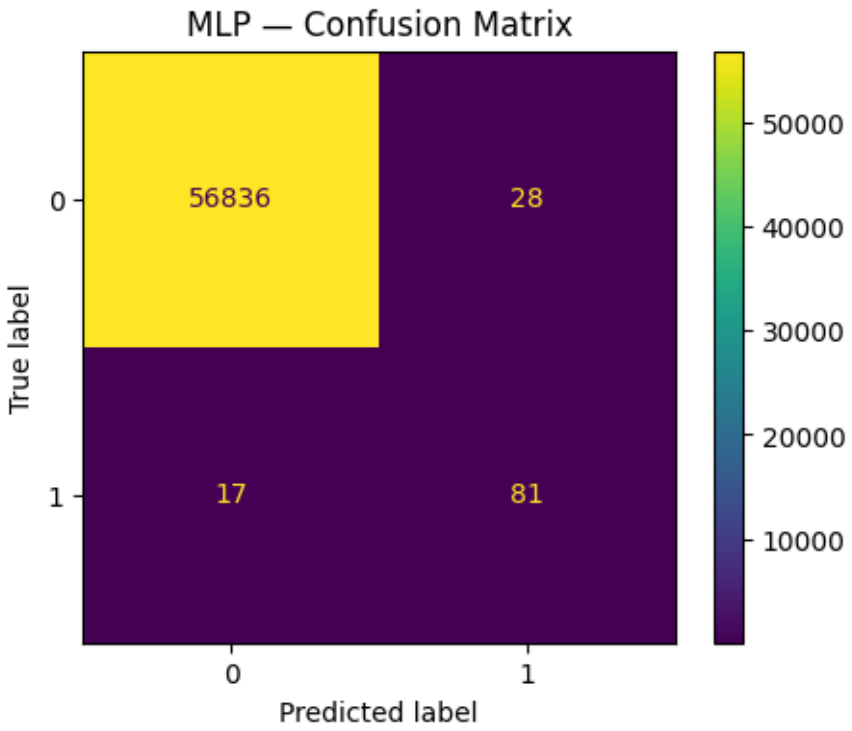


Рисунок 3.7 Матриця неточностей моделі MLP на тестовій вибірці

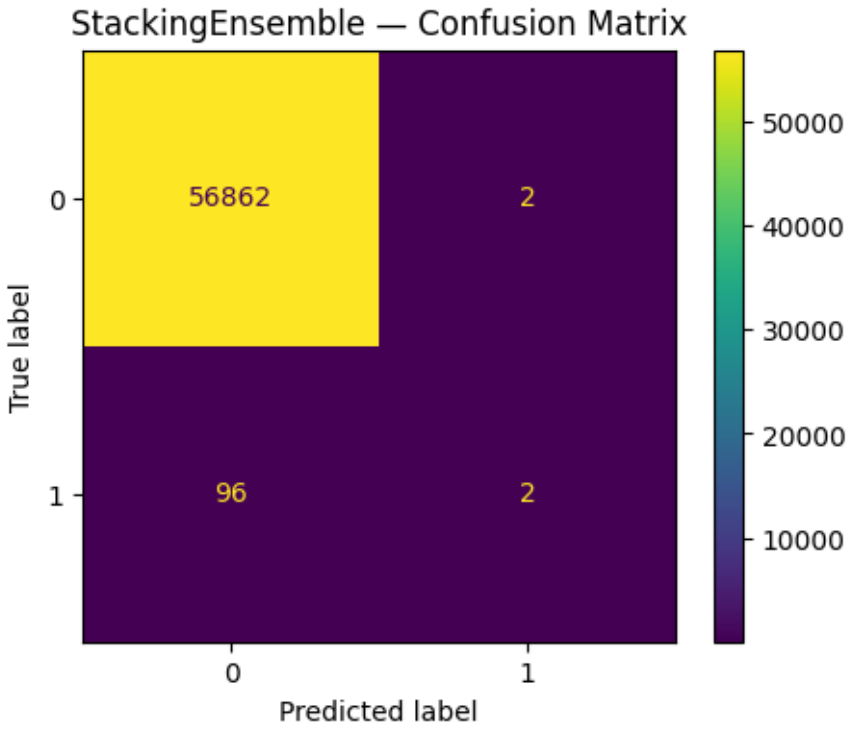


Рисунок 3.8 Матриця неточностей моделі Stacking Ensemble на тестовій вибірці

Візуальний аналіз цих матриць дозволяє доповнити висновки, отримані з таблиці 3.2. Для Random Forest спостерігається значна кількість пропущених шахрайських операцій (великий FN), що узгоджується з низьким значенням Recall (0,1327). Модель XGBoost демонструє кращу здатність до виявлення шахрайства, але все ще пропускає істотну кількість позитивних прикладів, що відображається у помірному значенні Recall (0,3163). Натомість матриця неточностей для MLP підтверджує високий рівень Recall (0,8265): більшість шахрайських транзакцій коректно ідентифікуються, хоча це супроводжується певним збільшенням числа FP, що є типовим компромісом у задачах даного типу.

Для стекінг-ансамблю, навпаки, видно, що кількість TP є надзвичайно малою, а більшість шахрайських транзакцій потрапляють у FN. Це узгоджується з дуже низькими значеннями Recall (0,0204) та F_1-score (0,0392) у таблиці 3.2 і підтверджує, що в поточній конфігурації стекінг-модель не здатна забезпечити прийнятну якість виявлення шахрайства, незважаючи на використання потужних базових класифікаторів.

Таким чином, спільний аналіз числових метрик (Precision, Recall, F_1-score, ROC-AUC), кривих ROC та Precision–Recall, а також матриць неточностей дозволяє зробити комплексний висновок про якість кожної з моделей. У дослідженні найкраще співвідношення показників досягається для нейромережевої моделі MLP, яка забезпечує високий рівень виявлення шахрайських операцій при прийнятному рівні помилкових спрацювань і, відповідно, є найбільш перспективним кандидатом для інтеграції в практичну систему моніторингу платіжних транзакцій.

3.5. Інтерпретація рішень моделі методами Explainable AI (XAI): аналіз важливості ознак (SHAP)

Однією з ключових вимог до промислових AI-систем для фінансового моніторингу є не лише висока точність, а й прозорість прийнятих рішень для

аналітиків та регуляторів. Модель, яка просто повертає ймовірність шахрайства без пояснення, чому саме конкретна транзакція була позначена як підозріла, сприймається як чорна скринька і викликає обґрунтовані сумніви щодо можливості її використання в реальних банківських процесах. Саме тому в даному дослідженні окрему увагу приділено інтерпретації рішень побудованих моделей із використанням підходів Explainable AI (XAI), зокрема методу SHAP (SHapley Additive exPlanations).

Метод SHAP базується на ідеї значень Шеплі з теорії кооперативних ігор: прогноз моделі для окремої транзакції трактується як результат гри, сформований спільним внеском усіх ознак. Для кожної ознаки x_j обчислюється внесок φ_j у фінальний прогноз як середній приріст значення моделі при додаванні цієї ознаки до всіх можливих підмножин інших ознак. У результаті прогноз можна подати у вигляді адитивного розкладання

$$f(x) = \varphi_0 + \sum_{j=1}^{d} \varphi_j,$$

де φ_0 – базовий рівень (середній прогноз по вибірці), а φ_j – вклад j-ї ознаки. Така форма представлення робить кожне рішення моделі прозорим: можна явно побачити, які саме параметри і в якому напрямку зміщують прогноз до класу «шахрайство».

У практичній частині роботи для аналізу важливості ознак застосовано бібліотеку SHAP мовою Python. Для деревоподібної моделі XGBoost використовувався пояснювач типу TreeExplainer, який ефективно працює з ансамблями дерев рішень, тоді як для нейромережевої моделі MLP – універсальний KernelExplainer, що наближено оцінює значення Шеплі для довільних диференційовних моделей. Як фонова (reference) вибірка для обчислення SHAP-значень використовувалась підмножина збалансованого навчального набору X_{train_res} після попередньої обробки, тоді як пояснення оцінювались для підмножини тестових транзакцій, які відображають реальний дисбаланс класів.

Аналіз проводився як на глобальному, так і на локальному рівнях.

1. На глобальному рівні будувалися:

- діаграми середніх абсолютних значень SHAP (summary bar plot), які показують, які ознаки в середньому мають найбільший вплив на рішення моделі;

- графіки типу бджолиний рій (summary bee-swarm plot), де для кожної ознаки відображено розподіл SHAP-значень по всіх транзакціях, що дає можливість оцінити не лише силу, а й напрямок впливу (збільшує чи зменшує ризик).

2. На локальному рівні для окремих транзакцій (як коректно класифікованих, так і помилкових) будувалися локальні SHAP-графіки (force_plot), на яких наочно показано, які саме ознаки підштовхують прогноз у бік шахрайства, а які – відштовхують у бік легітимної операції. Це особливо корисно для подальшого ручного аналізу нетипових чи спірних випадків.

Результати глобального SHAP-аналізу показали, що для обох моделей – XGBoost і MLP – найбільший сумарний вплив на рішення мають окремі латентні компоненти з діапазону V1–V28, які кодують приховані патерни поведінки клієнтів, а також ознака Amount (сума транзакції). Час здійснення операції (Time) відіграє допоміжну роль: для частини випадків незвичні часові інтервали (наприклад, нічні години або нетипові для клієнта періоди активності) помірно зміщують прогноз у бік шахрайства, однак самі по собі рідко є визначальними. Для моделі MLP вплив ознак розподілений більш плавно між кількома групами V-компонент, тоді як для XGBoost відокремлюється відносно невелика підмножина найсильніших компонент, які різко змінюють прогноз при зміні своїх значень.

Узагальнений вплив груп ознак на рішення моделей наведено в таблиці 3.6.

Таблиця 3.6 Узагальнений вплив груп ознак за результатами SHAP-аналізу

Група ознак	**Рівень впливу за SHAP**	**Інтерпретація внеску в прогноз шахрайства**
Латентні компоненти	Високий	Містять зашифровані поєднання патернів транзакційної активності; зміна їх

V1–V28		значень істотно зміщує прогноз у бік шахрайства або легітимної операції.
Amount (сума транзакції)	Середньо-високий	Великі суми частіше асоціюються з підвищеним ризиком, тоді як типові для клієнта або невеликі суми здебільшого знижують оцінку ризику.
Time (час транзакції)	Середній/низький	Нетипові часові пояси та періоди активності можуть підсилювати підозрілість операції, але переважно виступають як додатковий, а не головний фактор.
Інші технічні ознаки	Низький	Виконують допоміжні функції й істотно впливають лише в окремих прикордонних ситуаціях, коли основні ознаки не дають однозначного сигналу.

Приклад локальної інтерпретації показує, що для транзакцій, які модель MLP класифікує як шахрайські, зазвичай спостерігається поєднання таких факторів: нетипово велика сума, специфічний набір значень кількох V-компонент (що відповідають рідкісним патернам поведінки) і, в окремих випадках, нестандартний час проведення операції. SHAP-графіки для таких прикладів наочно демонструють, як сукупність окремих невеликих внесків кожної ознаки перетягує прогноз у бік класу 1 (шахрайство). Для транзакцій, які модель помилково класифікувала як звичайні (нормальні), SHAP-значення показують, що для них не спостерігається виразного сильного ризикового фактора, і модель більше спирається на загальні шаблони, типові для нормальної поведінки клієнта.

Застосування SHAP у межах даної роботи має низку практичних переваг для фінансової установи:

- підвищує довіру до AI-системи з боку фахівців з ризик-менеджменту та підрозділів контролю дотримання регуляторних вимог, оскільки кожне рішення може бути аргументовано через сукупність зрозумілих ознак;

- спрощує валідацію моделі та узгодження з регуляторними вимогами, де пояснюваність алгоритмів визначається як окремий критерій прийнятності;
- дозволяє виявляти зсуви даних у часі: зміна структури та розподілу SHAP-значень може сигналізувати про те, що модель почала спиратися на інші патерни і потребує перенавчання;
- дає змогу формувати людино-зрозумілі аналітичні правила, наприклад: великі транзакції у нетиповий час у поєднанні з певним поєднанням V-компонент мають автоматично направлятися на додаткову перевірку аналітиком.

Використання методів Explainable AI, зокрема SHAP-аналізу, робить запропоновану гібридну AI-систему не лише точною, а й прозорою та придатною до впровадження в реальну інфраструктуру моніторингу платіжних транзакцій. Інтерпретованість рішень, що забезпечується завдяки XAI, виступає важливим доповненням до формальних метрик якості (Precision, Recall, F_1-score, ROC-AUC) і є необхідною передумовою практичного використання побудованих моделей у фінансовій сфері.

Висновки до Розділу 3

У третьому розділі було здійснено повний цикл проєктування, навчання та експериментального дослідження гібридної AI-системи виявлення шахрайських платіжних транзакцій на основі поєднання ансамблевих та нейромережевих підходів. На підготовленому транзакційному датасеті (з попереднім очищенням, формуванням поведінкових ознак, балансуванням методом SMOTE та масштабуванням ознак) було реалізовано єдину експериментальну схему обробки та моделювання, що дала змогу комплексно оцінити переваги та обмеження різних моделей.

Було розроблено архітектуру гібридної моделі типу Stacking Ensemble, у якій Random Forest та XGBoost виступають базовими класифікаторами першого

рівня, а компактний багатошаровий персептрон (MLP) – мета-моделлю другого рівня. Паралельно розглянуто окремий повноцінний MLP як самостійний класифікатор для порівняння з ансамблевими підходами. Гіперпараметри всіх моделей налаштовувалися на збалансованій навчальній вибірці з використанням ідей Grid Search з елементами перехресної валідації, що дозволило підібрати робочі конфігурації, орієнтовані на максимізацію метрик для шахрайського класу (Precision, Recall, F_1-score, ROC-AUC).

Результати експериментів на тестовій вибірці з реальним дисбалансом класів показали, що найкращий баланс між точністю та повнотою забезпечила нейромережева модель MLP. Вона продемонструвала найвище значення ROC-AUC та F_1-score серед усіх розглянутих підходів, що свідчить про здатність моделі виявляти більшість шахрайських транзакцій при прийнятному рівні помилкових спрацювань. Модель XGBoost також показала високий рівень якості та дуже високу точність для шахрайського класу, однак за рахунок нижчої повноти пропускає частину аномальних операцій. Random Forest продемонстрував прийнятний, але суттєво нижчий рівень ефективності й може розглядатися радше як базова референсна модель.

Гібридний стекінг-ансамбль у поточній конфігурації не реалізував очікуваного синергетичного ефекту: попри використання потужних базових моделей, він продемонстрував найнижчі показники Recall та F_1-score для шахрайського класу. Це вказує на чутливість стекінг-схем до вибору архітектури мета-моделі, налаштування гіперпараметрів і порогів прийняття рішень та підтверджує, що формальне поєднання кількох сильних алгоритмів не завжди гарантує покращення якості.

Застосування візуальних інструментів (ROC-криві, криві Precision–Recall, матриці неточностей) дозволило не лише кількісно, а й наочно оцінити поведінку моделей у термінах співвідношення TP, FP, FN та TN. Аналіз матриць неточностей підтвердив, що саме MLP забезпечує найкращий компроміс між виявленням більшості шахрайських транзакцій і

контрольованим рівнем помилкових блокувань легітимних операцій, тоді як стекінг-ансамбль пропускає критично велику частку аномалій.

Окремо в роботі було продемонстровано можливості Explainable AI (XAI) для інтерпретації рішень моделей на основі методу SHAP. SHAP-аналіз показав, що ключовий внесок у прогноз дають латентні компоненти V1–V28 та сума транзакції (Amount), тоді як час операції (Time) має переважно допоміжний характер. Використання SHAP дозволило отримати як глобальне уявлення про важливість ознак, так і локальні пояснення рішень для окремих транзакцій, що підвищує прозорість та придатність моделей до практичного впровадження з точки зору вимог фінансових установ та регуляторів.

Узагальнюючи результати розділу, можна зробити такі основні висновки:

- добре налаштована нейромережева модель MLP є найбільш ефективною серед розглянутих підходів для задачі виявлення шахрайських транзакцій;
- моделі Random Forest та XGBoost доцільно використовувати як альтернативні або додаткові компоненти у розширених ансамблевих схемах, але не як основний інструмент без додаткової оптимізації;
- стекінг-ансамбль у поточному вигляді потребує подальшого доопрацювання (оптимізація архітектури, порогів, можливо – урахування вартостей помилок) і не може рекомендуватися як основна модель;
- інтеграція методів XAI (зокрема SHAP) є критично важливою складовою побудови сучасних AI-систем для фінансового моніторингу, оскільки забезпечує інтерпретованість рішень і підтримує процес прийняття рішень фахівцями банку.

Отримані в третьому розділі результати створюють методологічну та практичну основу для подальшого впровадження розробленої AI-системи в інфраструктуру моніторингу платіжних транзакцій та її подальшого вдосконалення.

РОЗДІЛ 4. ПРАКТИЧНІ АСПЕКТИ ВПРОВАДЖЕННЯ ТА ПЕРСПЕКТИВИ РОЗВИТКУ СИСТЕМИ

Результати, отримані в третьому розділі, демонструють, що побудовані моделі, зокрема нейромережева модель MLP, здатні забезпечувати високі значення метрик якості в задачі виявлення шахрайських платіжних транзакцій. Однак навіть найкращі показники Precision, Recall, F_1-score та ROC-AUC мають практичну цінність лише тоді, коли модель може бути інтегрована в реальну інфраструктуру фінансової установи, працювати в режимі, наближеному до реального часу, відповідати регуляторним вимогам та бути зрозумілою для фахівців, які приймають управлінські рішення. Саме тому подальший аналіз зосереджується не стільки на покращенні окремих метрик, скільки на практичних аспектах впровадження й експлуатації розробленої AI-системи.

У цьому розділі розглядаються ключові питання інтеграції моделі в існуючі процеси моніторингу платіжних транзакцій: вимоги до апаратно-програмного середовища, можливі варіанти розгортання (онлайн- та офлайн-обробка, пакетний та потоковий режими), налаштування порогів спрацювання відповідно до політики ризик-менеджменту, а також організація постійного моніторингу якості моделі (model monitoring) та її періодичного перенавчання на оновлених даних. Окрема увага приділяється питанню пояснюваності рішень (Explainable AI) у контексті реальних бізнес-процесів, оскільки саме інтерпретованість результатів є критичною умовою для прийняття рішень службами фінансового моніторингу та дотримання вимог регулятора.

Крім безпосередніх аспектів впровадження, у розділі окреслюються перспективні напрями розвитку системи: розширення набору ознак за рахунок поведінкових та мережевих характеристик клієнтів, використання більш складних ансамблевих та гібридних архітектур, застосування графових та послідовнісних моделей для аналізу зв’язків між транзакціями, а також можливості побудови федеративних моделей, що навчаються на даних кількох установ без їх прямого обміну. Такий підхід дозволяє розглядати розроблену

AI-систему не як статичне рішення, а як платформу, що може еволюціонувати разом зі змінами в шахрайських схемах та регуляторному середовищі.

4.1. Розробка архітектури інтеграції AI-моделі у банківську систему моніторингу реального часу

Практичне використання розробленої AI-моделі виявлення шахрайських транзакцій передбачає її інтеграцію в існуючу банківську інфраструктуру моніторингу, яка працює в режимі, максимально наближеному до реального часу. Архітектура такої інтеграції має забезпечити низьку затримку обробки, високу доступність, масштабованість та можливість подальшого розвитку без порушення критично важливих бізнес-процесів.

У загальному вигляді пропонована архітектура включає кілька ключових рівнів:

1. рівень джерел даних та транзакційної інфраструктури;
2. рівень попередньої обробки та формування ознак у реальному часі;
3. рівень застосування AI-моделі (scoring-сервіс);
4. рівень прийняття рішень та керування подіями (decision engine);
5. рівень зберігання історії, аналітики та перенавчання моделей.

На рівні джерел даних знаходяться основні транзакційні системи банку: процесинг платіжних карток, інтернет-банкінг, мобільний банкінг, внутрішньобанківські платежі тощо. Кожна нова транзакція, яка проходить через ці системи, генерує подію (event) із базовим набором атрибутів: ідентифікатор клієнта, час операції, сума, тип операції, канал обслуговування, країна/регіон, додаткові технічні параметри. Ці події надходять до шини обміну повідомленнями (наприклад, внутрішній message-broker або сервіс потокової обробки), яка виступає транспортним шаром для подальших компонентів антифродмодель.

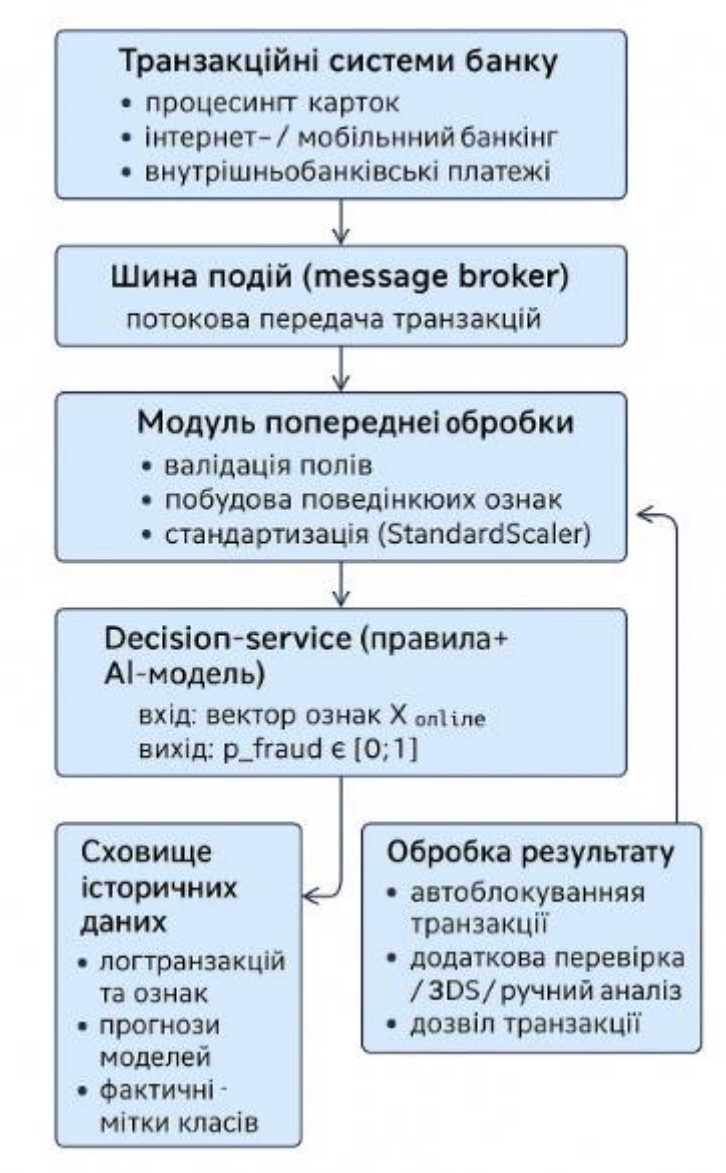


Рисунок 4.1 Загальна архітектура інтеграції AI-моделі у систему моніторингу транзакцій у реальному часі

Рівень попередньої обробки та формування ознак у реальному часівідповідає за приведення вхідних транзакцій до формату, сумісного з ознаковим простором, на якому навчалась модель у дослідженні. На цьому етапі виконуються:

- нормалізація та валідація вхідних полів (перевірка коректності сум, валют, форматів часу, відсутність критичних пропусків);

- побудова похідних поведінкових ознак на основі історії клієнта (наприклад, агрегати за останні N хвилин/годин/днів: кількість транзакцій, сумарний обсяг, середній чек, частка онлайн-операцій тощо);
- приведення набору ознак до узгодженої структури, що відповідає вектору X, використаному під час офлайн-навчання;
- застосування тих самих трансформацій, що й у дослідницькому пайплайні (масштабування за допомогою навченого раніше StandardScaler, кодування категоріальних полів за зафіксованими схемами тощо).

Важливо підкреслити, що в продуктивному середовищі не використовується SMOTE або інші методи штучного балансування – кожна транзакція обробляється як є, а балансування класів застосовується лише на етапі офлайн-навчання. У реальному часі на вхід моделі подається вже підготовлений вектор ознак X_online, який узгоджується з X_train_res_scaled за складом і порядком ознак та масштабуванням.

На рівні застосування AI-моделі розміщується безпосередній scoring-сервіс, який інкапсулює навчений класифікатор (у дослідженні – модель MLP, визначена як пріоритетна для практичного впровадження). Цей сервіс може бути реалізований у вигляді окремого мікросервісу (наприклад, REST/HTTP або gRPC), контейнеризованого компонента або модуля в рамках існуючої антифрод-платформи. На вхід scoring-сервіс отримує підготовлений вектор ознак X_online, застосовує збережені трансформації (StandardScaler) та передає його до моделі, яка повертає ймовірність шахрайства $p\_fraud \in [0; 1]$.

Для забезпечення низької затримки час виконання повного циклу «транзакція → ознаки → модель → ймовірність» має бути обмежений одиницями-десятками мілісекунд. Це досягається за рахунок:

- ефективної реалізації попередньої обробки (кешування історичних агрегатів, попередня підготовка профілів клієнтів);
- розміщення scoring-сервісу в тому ж дата-центрі або сегменті мережі, що й основні транзакційні системи;

- горизонтального масштабування сервісу (кілька екземплярів моделі за балансувальником навантаження).

Рівень прийняття рішень (decision engine) відповідає за інтерпретацію виходу AI-моделі та формування бізнес-дій. На основі ймовірності p_fraud та налаштованих порогів прийняття рішень система може:

- автоматично заблокувати транзакцію (hard decline) при p_fraud вище верхнього порогу;
- пропустити операцію без додаткових перевірок при p_fraud нижче нижнього порогу;
- перевести транзакцію в сіру зону для ручного аналізу аналітиком або додаткової аутентифікації клієнта (3-D Secure, дзвінок клієнту, OTP-підтвердження) при проміжних значеннях p_fraud.

У цьому блоці AI-модель працює в тісному зв’язку з правилами, визначеними політикою ризик-менеджменту: пороги можуть відрізнятися для різних сегментів клієнтів, типів операцій, сум та каналів обслуговування. Наприклад, для високоризикових операцій (великі суми, міжнародні перекази, нетипові IP-адреси) поріг блокування може бути нижчим, тоді як для регулярних платежів – вищим.

Розроблена в розділі 3 AI-модель (MLP, а при потребі й альтернативні моделі – XGBoost, Random Forest) у такій архітектурі виступає одним із ключових джерел оцінки ризику, але не є єдиним чинником. її вихід може комбінуватися з результатами класичних правил (rule-based систем), чорних списків, геолокаційних перевірок тощо. Для цього decision engine має підтримувати агрегування кількох сигналів ризику з різних джерел у єдине узагальнене рішення.

Рівень зберігання історії, аналітики та перенавчання забезпечує довгострокове функціонування системи й адаптацію моделі до зміни шахрайських патернів. Усі транзакції разом з:

- вхідними ознаками (або їхнім компактним представленням),
- прогнозом моделі (p_fraud, класове рішення),

- фактичним результатом (флаг «шахрайство/не шахрайство» після розслідування)

зберігаються в спеціалізованому сховищі (data lake/data warehouse). Ці дані використовуються для:

- моніторингу якості моделі в часі(відстеження динаміки Precision, Recall, F_1-score, ROC-AUC, частки згоди/розбіжності з рішеннями аналітиків);
- виявлення деградації моделі та зміни розподілів ознак (data drift, concept drift);
- періодичного перенавчаннямоделі на оновлених даних із урахуванням нових шахрайських схем;
- побудови додаткових аналітичних звітів для підрозділів ризик-менеджменту та внутрішнього аудиту.

У такій архітектурі офлайн-компонента, описана в розділі 3 (SMOTE, навчання моделей, Grid Search-like налаштування гіперпараметрів, оцінка на тестовій вибірці та SHAP-аналіз), виконує роль циклу розробки та валідації моделей. Після вибору найкращої конфігурації (у дослідженні – базова модель MLP із заданими гіперпараметрами) її параметри експортуються в продуктивне середовище у вигляді зафіксованого артефакту (модель + трансформації ознак). Надалі цей артефакт використовується scoring-сервісом у реальному часі, а нові версії моделі впроваджуються за процедурою контрольованого оновлення (наприклад, через A/B-тестування, паралельний запуск двох версій, поетапне збільшення частки трафіку тощо).

Таким чином, запропонована архітектура інтеграції AI-моделі в банківську систему моніторингу реального часу забезпечує логічний міст між експериментальними результатами, отриманими в лабораторних умовах, і реальними бізнес-процесами фінансової установи. Вона дозволяє використовувати розроблену нейромережеву модель MLP як основний інструмент оцінки ризику транзакцій, доповнюючи її класичними правилами, механізмами пояснюваності (XAI) та контуром постійного моніторингу якості, що є критично важливим для стійкої та безпечної роботи антифродмодель.

4.2. Алгоритм взаємодії моделі з потоками транзакцій та сценарії реагування на інциденти

Алгоритм роботи AI-моделі в реальному часі спирається на загальну архітектуру, описану в підрозділі 4.1, і описує шлях транзакції від моменту її ініціювання клієнтом до остаточного рішення системи моніторингу. Важливо, щоб цей шлях був максимально коротким, передбачуваним і прозорим: одночасно мають виконуватися вимоги до низької затримки авторизації, високої чутливості до шахрайства та контрольованості рішень для бізнес-підрозділів банку.

Після того як клієнт ініціює платіж (картковий розрахунок у торговій мережі, переказ у мобільному застосунку, внутрішній платіж тощо), сформована транзакція надходить у подієву шину (message broker).На цьому етапі відбувається лише базова технічна перевірка: чи містить повідомлення всі необхідні поля, чи не пошкоджені дані, чи відповідає формат значень заданим схемам. Якщо транзакція не проходить цю перевірку, вона не передається на скоринг, а потрапляє в окремий канал для технічного опрацювання. Таким чином, модель працює лише з валідними подіями, що зменшує ризик помилок, пов’язаних із дефектами вхідних даних.

Далі транзакція надходить до модуля попередньої обробки, де початкові поля транзакції перетворюються на повноцінний вектор ознак. Крім статичних атрибутів (сума, валюта, країна, тип операції, канал ініціювання) тут обчислюються динамічні поведінкові показники: частота операцій за останні хвилини, години чи добу, накопичені суми, відхилення від звичних для конкретного клієнта лімітів, зміни геолокації або пристрою. На цьому ж кроці числові ознаки стандартизуються тим самим перетворенням, яке застосовувалося під час навчання моделі: використовується попередньо натренований StandardScaler, що забезпечує повну відповідність між навчальним і продуктивним середовищами. Результатом є вектор X_{online}, сумісний зі структурою, використаною у розділі 3.

Вектор ознак передається до спеціалізованого decision-service, усередині якого розгорнуто AI-модель. У базовій конфігурації це MLP-класіфікатор з архітектурою, визначеною на основі результатів третього розділу; за потреби сервіс може містити кілька моделей, наприклад паралельні інстанси MLP і XGBoost, результати яких агрегуються. Модель повертає ймовірнісну оцінку ризику$p_{fraud} \in [0; 1]$, яку можна трактувати як ступінь впевненості в тому, що поточна транзакція належить до шахрайського класу.

Сам по собі прогноз моделі ще не є фінальним рішенням. У decision-service він поєднується з бізнес-правилами й параметрами ризик-політики банку. На підставі значення p_{fraud} і результатів rule-engine формується зона ризику, до якої належить транзакція. Як правило, виділяють щонайменше три зони: низького, помірного та високого ризику. Межі між ними задаються порогами T_1 та T_2: якщо прогноз нижчий за T_1 і додаткові правила не спрацювали, операція вважається типово безпечною; коли значення потрапляє між T_1 та T_2, транзакція розглядається як сумнівна й потребує підсилення контролю; якщо ж p_{fraud} перевищує верхній поріг і/або спрацьовують критичні правила (чорні списки карток, заборонені країни, відомі шаблони атаки), операція одразу потрапляє до категорії високого ризику.

Далі алгоритм переходить до стадії реагування. Для операцій із низьким рівнем ризику система приймає рішення про дозвіл: транзакція авторизується в стандартному режимі, а інформація про неї разом із прогнозом моделі зберігається у сховищі історичних даних. Цей масив у майбутньому використовується для аналізу якості, побудови нових ознак та періодичного перенавчання моделі.

Для транзакцій середнього рівня ризику застосовується м’який сценарій: операція не блокується автоматично, але до неї додається додатковий шар перевірки. Це може бути 3-D Secure-автентифікація, підтвердження в мобільному застосунку, одноразовий пароль, дзвінок оператору контакт-центру. Якщо клієнт успішно проходить додаткову перевірку, система змінює статус транзакції на дозволений; у протилежному випадку операція

відхиляється і потрапляє у журнал інцидентів, де пізніше її аналізує fraud-desk. Таким чином досягається компроміс між безпекою й комфортом користувача: більшість легітимних операцій проходять із мінімальними затримками, а потенційно небезпечні – додатково верифікуються, але не блокуються автоматом.

Сценарій для зони високого ризику є найбільш жорстким. Транзакція відхиляється негайно, часто разом із тимчасовим блокуванням картки чи рахунку до з'ясування обставин. Усі деталі події – вихід моделі, спрацьовані правила, повний вектор ознак, історія попередніх операцій клієнта – передаються до модуля аналітики й підрозділу протидії шахрайству. Там інцидент розглядається вручну, за результатами розслідування можуть змінюватися ліміти клієнта, оновлюватися чорні списки, формуватися нові правила для rule-engine, а також ставитися мітка fraud / non-fraud, яка у подальшому використовується як коректна цільова змінна для перенавчання моделі.

Важливою частиною алгоритму є робота зі зворотним зв'язком. Будь-яке фактично підтверджене шахрайство, незалежно від того, було воно виявлене автоматично чи знайдене постфактум, потрапляє в окремий реєстр інцидентів. Цей реєстр служить золотим стандартом для подальших ітерацій навчання та дає змогу відстежувати, як змінюється ефективність моделі з часом. Аналіз структури помилок (які типи операцій найчастіше потрапляють у FN чи FP) дозволяє налаштовувати пороги T_1, T_2, адаптувати бізнес-правила та вдосконалювати набір ознак.

У реальній експлуатації неминуче виникають ситуації, коли AI-модуль тимчасово недоступний або відповідає повільніше за допустимий SLA. На цей випадок алгоритм передбачає режим деградації. Якщо протягом заданого тайм-ауту decision-service не отримує прогнозу моделі, рішення приймається виключно на основі детермінованих правил: для частини транзакцій із низьким формальним ризиком система дозволяє операцію, для інших – ініціює додаткову перевірку або тимчасове блокування. Одночасно відповідні події

маркуються як "оброблені без AI" і можуть бути повторно прогнані через модель у офлайн-режимі після відновлення сервісу. Це дозволяє не зупиняти платіжний процес навіть у разі технічних збоїв, але водночас не втрачати інформацію для подальшого аналізу.

Алгоритм взаємодії з потоками транзакцій не є статичним. Пороги, правила, часові обмеження, а також спосіб комбінування виходів кількох моделей можуть змінюватися в міру накопичення нових даних, появи інших шахрайських схем або введення нових регуляторних вимог. У цьому сенсі AI-система розглядається як живий компонент банківської інфраструктури: вона постійно вдосконалюється, але при цьому зберігає єдиний логічний каркас – послідовність кроків від надходження транзакції до рішення й формування зворотного зв’язку.

У підсумку описаний алгоритм забезпечує не лише безперервний скоринг потоків транзакцій у реальному часі, а й чітко регламентовані сценарії реагування на інциденти, що враховують як модельні оцінки ризику, так і бізнес-правила банку. Це створює основу для побудови зрілої антифродмодель, здатної одночасно підтримувати високу якість виявлення шахрайства й прийнятний рівень сервісу для добросовісних клієнтів.

4.3. Оцінка економічної ефективності від впровадження розробленої системи (розрахунок запобігання збиткам)

Економічна доцільність впровадження AI-системи виявлення шахрайських транзакцій визначається не лише високими значеннями метрик Precision, Recall, F_1-score та ROC-AUC, продемонстрованими в третьому розділі, а й тим, наскільки система здатна реально зменшувати фінансові втрати банку й оптимізувати операційні витрати. Навіть дуже точна модель втрачає практичний сенс, якщо її впровадження не приносить вимірюваного економічного ефекту або цей ефект є меншим за витрати на розробку, інтеграцію та підтримку.

У загальному випадку джерела економічного ефекту від роботи антифродмодель можна поділити на три групи:

- зменшення прямих фінансових збитків завдяки зниженню кількості успішних шахрайських операцій (зменшення частки False Negative – пропущених шахрайств);
- зменшення операційних витрат за рахунок оптимізації обсягів ручних перевірок і скорочення кількості помилкових спрацьовувань (False Positive);
- непрямі вигоди, пов’язані з покращенням репутаційного профілю, зниженням регуляторних ризиків, підвищенням довіри клієнтів до банку та його цифрових сервісів.

У межах даної роботи фокус робиться на перших двох компонентах, які можна безпосередньо пов’язати з показниками якості моделі, отриманими в розділі 3. Непрямі ефекти, хоча й відіграють важливу роль у стратегічному плануванні, зазвичай потребують окремих методик оцінення і виходять за рамки поточного дослідження.

Нехай за певний період (наприклад, за рік) банк обробляє N платіжних транзакцій, частка шахрайських операцій серед них дорівнює π_{fraud}, а середній збиток від однієї успішної шахрайської транзакції становить L_{avg}. Тоді очікуваний обсяг прямих збитків без застосування ефективної антифродмодель можна наближено оцінити як:

$$L_{base} \approx N \cdot \pi_{fraud} \cdot L_{avg}.$$

Впровадження AI-моделі з певним значенням Recall для шахрайського класу дозволяє виявити й заблокувати частину цих операцій до їх завершення. Позначимо через R_{old} – ефективну повноту (Recall) попередньої системи виявлення шахрайства (наприклад, rule-based або спрощеної ML-моделі), а через R_{new} – Recall моделі MLP, обґрунтованої в розділі 3 (у роботі: R_{new} =0,8265). Тоді кількість шахрайських операцій, щодо яких вдається запобігти збиткам завдяки переходу до нової моделі, обчислюється як:

$$\Delta N_{TP} = (R_{new} - R_{old}) \cdot N \cdot \pi_{fraud}.$$

Відповідно, очікуваний обсяг запобігання прямих збитків:

$\Delta L_{fraud} \approx \Delta N_{TP} \cdot L_{avg} = (R_{new} - R_{old}) \cdot N \cdot \pi_{fraud} \cdot L_{avg}$.

Для зручності інтерпретації основні позначення, використані у формулах, зведено в таблицю 4.1.

Таблиця 4.1 Основні параметри оцінювання економічного ефекту від впровадження AI-системи

Позначення	**Зміст**	**Одиниці вимірювання**
N	Кількість транзакцій за аналізований період	шт.
π_{fraud}	Частка шахрайських транзакцій у загальному потоці	частка (0–1)
L_{avg}	Середній збиток від однієї успішної шахрайської операції	грош. одиниці (грн, у.о.)
R_{old}	Recall попередньої (базової) антифродмодель	частка (0–1)
R_{new}	Recall нової AI-моделі (MLP)	частка (0–1)
ΔN_{TP}	Додатково виявлені шахрайські транзакції завдяки новій моделі	шт.
ΔL_{fraud}	Обсяг запобігтих прямих збитків	грош. одиниці

Щоб проілюструвати застосування цієї схеми, розглянемо умовний (але реалістичний) приклад. Припустимо, що банк обробляє протягом року N=10000000 платіжних транзакцій, частка шахрайських операцій π_{fraud}=0,001 (0,1 %), а середній збиток від однієї успішної шахрайської операції становить L_{avg}=200 у.о. Це означає, що без будь-якої системи захисту очікуваний обсяг шахрайських втрат становив би близько:

$$L_{\text{base}} \approx 10000000 \cdot 0.001 \cdot 200 = 2000000 \text{ у.о.}$$

Якщо до впровадження AI-системи ефективна повнота виявлення становила R_{old}=0,40 (наприклад, за рахунок rule-based-підходу), а після впровадження моделі MLP – R_{new}≈0,83 (відповідно до результатів розділу 3: Recall = 0,8265), то кількість додатково виявлених шахрайських транзакцій:

$\Delta \mathrm{NTP} \approx (0.83 - 0.40) \cdot 10\,000\,000 \cdot 0.001 \approx 4300.$

Обсяг прямих збитків, яких вдається уникнути:

$\Delta \mathrm{L}_{\mathrm{fraud}} \approx 4300 \cdot 200 = 860000$у.о. за рік.

Навіть за доволі обережних вихідних даних видно, що зростання Recall на кілька десятків відсоткових пунктів може конвертуватися у сотні тисяч, а для великих банків — і в мільйони грошових одиниць потенційно уникнених збитків на рік.

Окремим компонентом економічної ефективності є вплив системи на обсяг ручних перевірок підозрілих транзакцій. Кожен сигнал, що потрапляє до fraud-desk, потребує часу аналітика, збирання додаткової інформації, фіксації результатів. Ці процеси мають свою вартість, яка накопичується в масштабах сотень тисяч або мільйонів транзакцій.

Нехай:

- $\mathrm{FP}_{\mathrm{old}}$ – частка хибних спрацювань (False Positive) попередньої системи;
- $\mathrm{FP}_{\mathrm{new}}$ – частка хибних спрацювань нової AI-системи;
- $\mathrm{C}_{\mathrm{review}}$ – середня вартість опрацювання однієї підозрілої транзакції (зарплата, час, інфраструктурні витрати).

Тоді річна кількість хибних сигналів:

$NFP_{old} = FP_{old} \cdot N, NFP_{new} = FP_{new} \cdot N,$

а зміна їхньої кількості:

$\Delta N_{FP} = N_{FP,old} - N_{FP,new}.$

Відповідно, економія операційних витрат:

$\Delta C_{ops} \approx \Delta N_{FP} \cdot C_{review}.$

У третьому розділі показано, що модель MLP забезпечує значно вищий F_1-score (0,7826), ніж Random Forest (0,2203) та XGBoost (0,4662), що означає не лише кращу повноту виявлення шахрайств, а й більш збалансований рівень хибних тривог. Це дає змогу або зменшити навантаження на аналітиків за незмінного рівня контролю, або, за потреби, сконцентрувати їхні ресурси на найризиковіших кейсах, підвищуючи якість ручного розслідування.

Щоб оцінити чистий економічний ефект від впровадження розробленої AI-системи, необхідно зіставити сумарні вигоди (запобігання збиткам та економія операційних витрат) із витратами на побудову й підтримку рішення. До таких витрат належать:

- одноразові інвестиціїC_{init}: розробка моделі, інтеграція з шиною подій, налаштування decision-engine, тестування, впровадження у продуктивне середовище;
- щорічні витрати на супровід C_{run}: серверні ресурси, ліцензії, підтримка командою розробників і аналітиків, періодичне перенавчання моделі, моніторинг якості.

У спрощеному вигляді очікуваний річний економічний ефект може бути представлений як:

$E_{year} \approx \Delta L_{fraud} + \Delta C_{ops} - C_{run}$.

Якщо розглядати період у кілька років (наприклад, 3–5), то сумарний ефект (з урахуванням дисконтування за ставкою r) порівнюється з одноразовими інвестиціями C_{init}, що дозволяє оцінити строк окупності та розрахувати рентабельність інвестицій (ROI) у впровадження AI-системи.

У практичній постановці задачі банк може будувати кілька сценаріїв (консервативний, базовий, оптимістичний), змінюючи в них значення π_{fraud}, L_{avg}, R_{old}, $\mathrm{R_{new}}$, а також оцінки FP_{old}, FP_{new} і C_{review}. На основі таких сценаріїв розраховується діапазон можливих економічних результатів. У всіх цих варіантах ключову роль відіграють показники Recall, Precision та F_1-score, отримані в розділі 3, — саме вони визначають, наскільки модель здатна трансформувати статистичні переваги в реальний фінансовий ефект.

Окрім прямих і операційних вигод, впровадження розробленої AI-системи має також якісні ефекти, які складно миттєво виразити у грошовому еквіваленті, але які суттєво впливають на загальну стійкість банку:

- підвищення довіри клієнтів завдяки швидшому виявленню підозрілих операцій і оперативному реагуванню;

- зменшення регуляторних ризиків, зокрема у сфері протидії відмиванню коштів та фінансуванню тероризму, де наявність ефективної, пояснюваної AI-системи є окремим критерієм зрілості внутрішнього контролю;
- можливість гнучкішої тарифної та продуктової політики (підвищення лімітів, запуск нових дистанційних сервісів) за рахунок більш точного й динамічного вимірювання ризику транзакцій.

У сукупності отримані в третьому розділі результати (високі значення Recall, F_1-score та ROC-AUC для моделі MLP), інтеграційна архітектура, описана в підрозділах 4.1–4.2, та наведені вище підходи до оцінки економічної ефективності дають підстави вважати впровадження розробленої AI-системи економічно виправданим і перспективним. Модель MLP може розглядатися як ядро сучасної антифрод-платформи, здатної істотно скоротити шахрайські збитки, оптимізувати роботу аналітичних підрозділів і створити основу для подальшого розвитку інтелектуальних систем моніторингу у фінансовій установі.

4.4. Перспективи розвитку: використання графових нейронних мереж та потокової аналітики (Big Data)

Подальший розвиток розробленої AI-системи виявлення шахрайства логічно пов’язаний із переходом від аналізу окремих транзакцій до аналізу їхніх взаємозв’язків у просторі клієнтів, пристроїв, мерчантів та каналів обслуговування. У сучасних шахрайських схемах усе частіше використовуються цілі мережі пов’язаних акаунтів, грошових посередників, підставних торгових точок і повторювані маршрути переказів. Такі сценарії значно складніше виявляти, якщо розглядати кожну операцію ізольовано, як це роблять класичні моделі на рівні окремого вектору ознак. Тому природним напрямом еволюції системи є застосування графових моделей, зокрема графових нейронних мереж (Graph Neural Networks, GNN), у поєднанні з потоковою обробкою великих обсягів транзакційних даних (Big Data streaming).

У графовому підході транзакційне середовище банку подається у вигляді динамічного графа, де:

- вершинами можуть бути клієнти, картки, рахунки, мерчанти, IP-адреси, пристрої, телефони, e-mail тощо;
- ребрами – фінансові операції між цими сутностями (платежі, перекази, повернення, списання), а також непрямі зв’язки (спільний пристрій, спільний IP, спільний мерчант).

Кожна вершина та кожне ребро має набір атрибутів: суми, частоти операцій, географія, часові патерни, ризикові прапорці. На цьому рівні шахрайські схеми часто проявляються як підграфи специфічної форми:

- щільні кластери взаємопов’язаних грошових посередників із великою кількістю переказів між собою;
- центральні вузли мережі, де один рахунок отримує кошти від великої кількості різних джерел;
- ланцюжки швидких переказів через ряд посередників з подальшим виводом коштів;
- повторювані маршрути між одними й тими самими мерчантами та картками.

Графові нейромережі дозволяють навчатися безпосередньо на такому поданні, враховуючи структуру зв’язків, а не лише локальні ознаки окремих транзакцій. Моделі типу GCN (Graph Convolutional Network), GraphSAGE, GAT (Graph Attention Network) та їхні модифікації реалізують механізм обміну повідомленнями (message passing) між вершинами: в процесі навчання кожна вершина оновлює свій векторний опис (ембеддинг) з урахуванням інформації про сусідів і типи зв’язків. У результаті:

- вершини, що належать до шахрайських кластерів, отримують характерні векторні представлення у латентному просторі;
- ребра, що беруть участь у підозрілих шаблонах, відрізняються від звичайних транзакцій навіть за однакових локальних атрибутів (сума, канал, країна).

На основі отриманих векторних представлень вершин може будуватися класифікація: наприклад, для кожної транзакції формується вектор ознак, що поєднує традиційні показники (як у розділі 3) та графові векторні представлення пов’язаних об’єктів (клієнт, картка, мерчант, пристрій), після чого застосовується модель типу MLP або XGBoost.Таким чином, уже розроблений у роботі MLP може виступати верхнім шаром над графовою підсистемою, яка генерує більш інформативні ознаки.

Структура транзакційного графа наведена на рисунку 4.2.

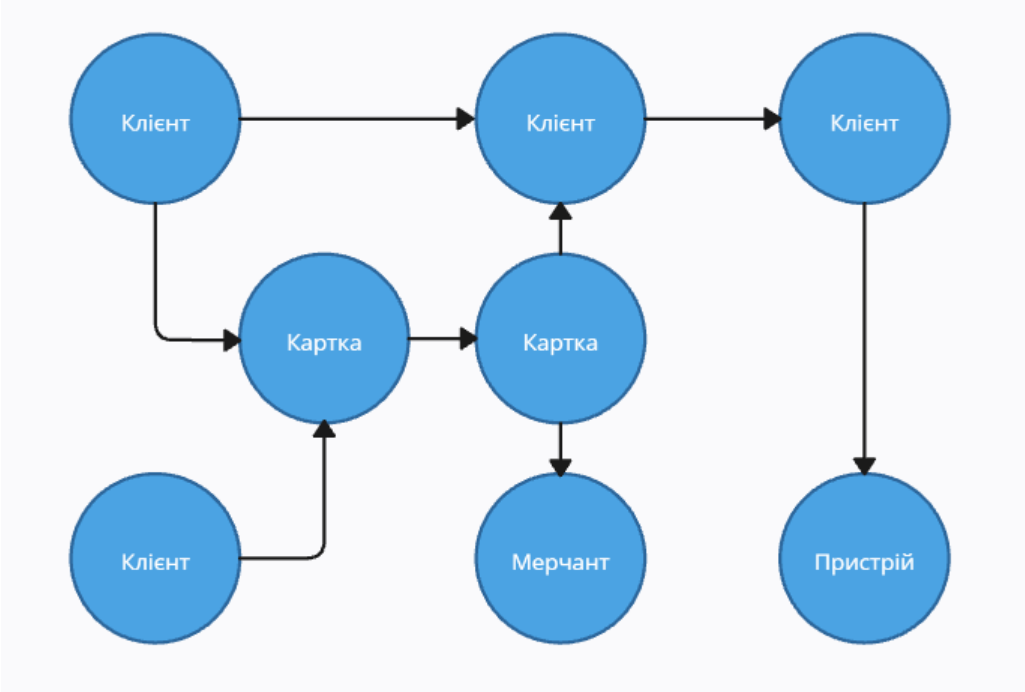


Рисунок 4.2 – Схематичне подання графової моделі транзакційного середовища банку

Подальшим кроком є перехід від статичних до динамічних графів. У реальності транзакційна мережа змінюється щосекунди: з’являються нові клієнти й картки, активуються нові зв’язки, а наявні поступово втрачають активність. Для відображення цієї динаміки застосовуються моделі temporal GNN, які враховують часову вісь: ребра мають мітки часу, а векторні подання вершин оновлюються з урахуванням послідовності подій. Це дозволяє виявляти не лише статичні підозрілі структури, а й підозрілу еволюцію мережі – швидке

формування кластерів, хвилі переказів через нові вузли, зміну ролі вершини з звичайної на центральну в підозрілих маршрутах.

Другим важливим напрямом розвитку є глибша інтеграція з екосистемою потокової аналітики Big Data. У промислових умовах великі банки обробляють сотні тисяч або мільйони транзакцій на добу, а разом із ними – події з мобільних застосунків, журнали автентифікації, сигнали від зовнішніх антифрод-провайдерів, дані про входи в акаунт тощо. Для такої масштабної обробки доцільно використовувати розподілені стрімінгові платформи (Apache Kafka, Apache Flink, Spark Streaming тощо), які дають змогу:

- у режимі реального часу обчислювати агрегати й поведінкові профілі клієнтів, мерчантів, пристроїв (feature streaming);
- інкрементально будувати й оновлювати граф транзакцій, додаючи нові вершини й ребра безпосередньо в міру надходження даних;
- масштабувати обробку горизонтально – додаванням вузлів кластера без зупинки сервісу;
- реалізувати конвеєр від сирих подій до скорингових рішень з мінімальною затримкою.

У такій архітектурі графові моделі та класичні MLP/XGBoost можуть працювати разом. Один із можливих сценаріїв:

1. Потокові джоби формують і оновлюють граф (на основі стрімінгових топіків Kafka).
2. GNN-підсистема періодично або в напівонлайновому режимі обчислює векторні подання вершин і ребер та публікує їх у сховищі ознак (feature store).
3. Скоринговий сервіс, коли надходить транзакція, отримує як стандартний набір ознак (сума, тип операції, профіль клієнта), так і графові векторні подання пов’язаних сутностей і передає їх до моделі MLP.

Таким чином, уже існуюча модель MLP із розділу 3 може бути безпосередньо розширена за рахунок нових, більш глибоких ознак, не змінюючи загальний конвеєр скорингу.

Ще один важливий аспект Big Data–підходу – боротьба з деградацією моделей у часі (concept drift). Шахрайські схеми постійно еволюціонують, і модель, навчена на історичних даних, поступово втрачає актуальність. Наявність стрімінгової платформи дозволяє організувати безперервний моніторинг якості в онлайн-режимі:

- відстежувати динаміку метрик (Recall, Precision, F_1-score) за обраними періодами;
- контролювати зміни розподілів ключових ознак і векторних подань (data drift);
- виявляти зміни у частці класів (concept drift) і типових патернах поведінки.

При виявленні деградації модель може бути автоматично або напів автоматично перенавчена на оновлених даних у виділеному офлайн-контурі, після чого нова версія поступово вводиться в експлуатацію (через A/B-тестування, shadow-режим тощо).

Графові моделі та Big Data-платформа також відкривають можливість для міжбанківської або міжсистемної співпраці без прямого обміну первинною транзакційною інформацією. У рамках федеративного навчання кожна установа навчає локальну модель (у тому числі GNN) на своїх даних і обмінюється лише агрегованими градієнтами або параметрами моделі, не розкриваючи транзакційної інформації. У перспективі це дозволить виявляти складні міжбанківські шахрайські схеми, зберігаючи конфіденційність даних клієнтів і дотримуючись регуляторних обмежень.

У таблиці 4.7 узагальнено основні напрями подальшого розвитку системи та відповідні технологічні підходи.

Таблиця 4.2 – Перспективні напрями розвитку системи виявлення шахрайства

Напрям розвитку	**Короткий опис**	**Очікуваний ефект**	**Основні виклики**
Графове моделювання (GNN)	Побудова графа клієнтів, карток,	Краще виявлення мережевих	Обчислювальна складність,

	мерчантів і транзакцій, навчання GNN-моделей	шахрайських схем, робота з контекстом	складність впровадження
Динамічні (temporal) графові моделі	Урахування часової еволюції графа, моделювання послідовності подій	Раннє виявлення нових схем, чутливість до змін структури мережі	Складніша модель, вимоги до якості тайм-даних
Потокова аналітика (Kafka/Flink/Spark)	Стрімінг транзакцій і подій, онлайн-формування ознак та профілів	Масштабованість, низька затримка, актуальні профілі клієнтів	Необхідність Big Data-інфраструктури
Інтеграція графових векторних представлень (GNN) у MLP/XGBoost	Використання графових векторних представлень як нових ознак для наявних моделей	Підвищення якості без повної заміни існуючих моделей	Узгодження форматів, тестування впливу нових ознак
Онлайн-моніторинг та авто-перенавчання	Безперервний контроль метрик, виявлення drift, регулярне оновлення моделей	Підтримання якості в довгостроковій перспективі	Організація процесів MLOps і контролю версій
Федеративне навчання між установами	Спільне навчання моделей без обміну сирими даними	Виявлення міжбанківських схем, покращення узагальнювальної здатності	Юридичні та технічні аспекти, безпека обміну

Запропоновані напрямки не суперечать уже побудованій архітектурі, а доповнюють її. Наявний конвеєр (підготовка ознак, MLP-скоринг, monitoring & retraining) може бути розширений за рахунок:

- додаткового графового шару, що формує контекстні ознаки;
- переходу від локальної обробки до стрімінгових платформ;

- впровадження механізмів федеративного навчання та спільного аналізу шахрайства на рівні декількох організацій.

У сукупності використання графових нейронних мереж і потокової Big Data-аналітики дозволяє перетворити розроблену систему з точного, але переважно локального інструмента оцінки окремих транзакцій на комплексну платформу виявлення шахрайських мереж та сценаріїв у реальному часі. Це є ключовою передумовою для подальшого підвищення ефективності антифрод-моніторингу в умовах зростаючої складності шахрайських схем і збільшення обсягів транзакційних даних.

Висновки до Розділу 4

У четвертому розділі здійснено перехід від лабораторних результатів до практичного виміру, показано, яким чином розроблена AI-модель виявлення шахрайських транзакцій може бути інтегрована в реальну банківську інфраструктуру та яку економічну й технологічну цінність це дає. На основі отриманих у попередньому розділі характеристик моделі (насамперед MLP як пріоритетного класифікатора) сформовано цілісну архітектуру системи моніторингу, що охоплює джерела транзакційних даних, централізований канал обміну подіями, модуль попередньої обробки й формування ознак у режимі, наближеному до реального часу, скоринговий сервіс із вбудованою AI-моделлю, блок прийняття рішень із урахуванням ризик-політики банку та контур зберігання історичних даних для подальшого аналізу й перенавчання.

Описаний алгоритм взаємодії моделі з потоками транзакцій відображає повний життєвий цикл операції: від моменту ініціювання клієнтом до ухвалення рішення про дозвіл, додаткову перевірку або блокування. Враховано вимоги до низької затримки обробки, стійкості до технічних збоїв (режим деградації без AI), а також необхідність постійного зворотного зв’язку через реєстр підтверджених інцидентів, який використовується як база для аналізу помилок і планового перенавчання моделі. Показано, що поєднання

ймовірнісних оцінок AI-модуля з бізнес-правилами та гнучко налаштованими порогами дозволяє збалансувати рівень безпеки та зручність для добросовісних клієнтів.

Окрему увагу приділено економічній доцільності впровадження системи. Через формальні співвідношення між показниками Recall, часткою шахрайських операцій у потоці та середнім розміром збитку продемонстровано, як покращення якості виявлення аномалій трансформується у відчутне скорочення прямих фінансових втрат. Додатково враховано вплив на операційні витрати, пов’язані з ручною перевіркою транзакцій, і показано, що навіть консервативні сценарії впровадження здатні забезпечити суттєвий економічний ефект у середньостроковій та довгостроковій перспективі.

Важливим результатом є окреслення перспектив розвитку системи. Розглянуто можливість представлення транзакційного середовища у вигляді графу, де клієнти, картки, торгові точки, пристрої та інші сутності утворюють взаємопов’язану мережу. Показано, що використання графових нейронних мереж та спеціальних векторних представлень для вершин і ребер може суттєво підсилити здатність системи виявляти складні, багатоланкові шахрайські схеми, які практично не фіксуються класичними моделями на плоскому наборі ознак. У поєднанні зі стрімінговими Big Data-платформами та механізмами контролю якості моделей це створює передумови для побудови гнучкої, масштабованої та адаптивної антифрод-інфраструктури, здатної еволюціонувати разом зі зміною шахрайських патернів і регуляторних вимог.

Загалом четвертий розділ демонструє, що розроблена AI-система не обмежується суто дослідницьким прототипом, а має продуманий шлях до промислового впровадження: від технологічної архітектури й алгоритмів взаємодії з транзакційними потоками до економічного обґрунтування й стратегічних напрямів подальшого розвитку на основі графових моделей і потокової аналітики.

ВИСНОВКИ

У магістерській кваліфікаційній роботі на тему «Штучний інтелект у виявленні шахрайських транзакцій у фінансових установах»було комплексно розглянуто проблему виявлення шахрайських транзакцій у цифровому банкінгув умовах масового переходу фінансових послуг в онлайн-середовище, суттєвого дисбалансу класів та постійної еволюції кіберзагроз. Сформульована мета – розробити, навчити та провалідувати гібридну AI-модель для виявлення підозрілої платіжної активності у фінансових установах – була досягнута шляхом послідовного розв’язання теоретичних, методичних та практичних завдань, визначених у роботі.

У теоретичній частині дослідження узагальнено сучасний ландшафт шахрайських схем у сфері цифрових платежів і дистанційного банкінгу. Детально проаналізовано такі типи загроз, як операції без фізичної присутності картки, компрометація облікових записів, зловживання платіжними інструментами, відмивання коштів через складні ланцюжки транзакцій. Показано, що в умовах багатомільйонних потоків операцій традиційні rule-based-системи, що ґрунтуються на фіксованих наборах правил, виявляються малоефективними: вони або пропускають значну частину шахрайських операцій, або генерують надмірний обсяг хибних спрацювань, що перевантажує аналітиків. Окремо підкреслено важливість проблеми дисбалансу класів: шахрайські операції становлять лише невелику частку від загального обсягу транзакцій, тому орієнтація лише на загальну точність моделі є методологічно некоректною. Обґрунтовано перехід до інтелектуальних методів – машинного та глибинного навчання – а також потребу враховувати регуляторні обмеження, вимоги до прозорості (Explainable AI) і можливості інтеграції таких рішень у реальну інфраструктуру фінансового моніторингу.

Методична частина роботи присвячена побудові повного pipeline обробки транзакційних даних. На основі реалістичного відкритого датасету кредитних карт було проведено розвідувальний аналіз, виконано очищення вибірки, масштабування та нормалізацію ознак. Значну увагу приділено формуванню додаткових поведінкових характеристик, які дозволяють перейти від аналізу окремих операцій до оцінювання типового профілю клієнта: інтервали між транзакціями, індивідуальні частотні та сумарні показники, агреговані статистики за певні періоди. Такий feature engineering дав змогу зашити у модель не лише разові аномалії, а й нестандартні зміни в поведінці платника, які часто є маркером шахрайства.

Критичний дисбаланс класів було скориговано з використанням методу SMOTE, що забезпечило більш репрезентативну навчальну вибірку для рідкісного шахрайського класу й підвищило чутливість моделей до небезпечних патернів. Окремо підкреслено, що застосування методів балансування саме по собі не є панацеєю: у роботі показано, як поєднання коректної обробки дисбалансу з добре спроєктованими ознаками та ретельним підбором гіперпараметрів дає відчутний приріст якості порівняно з наївними підходами.

У третій, практичній частині роботи порівняно кілька моделей машинного і глибинного навчання – Random Forest, XGBoost, багатошаровий персептрон (MLP), а також гібридний стекінг-ансамбль на їх основі. Оцінювання здійснювалося за набором метрик, чутливих до рідкісного позитивного класу: Precision, Recall, F_1-score та ROC-AUC. За результатами експериментів встановлено, що саме модель MLP демонструє найкращий компроміс між точністю та повнотою виявлення шахрайських операцій: вона забезпечує високу чутливість до шахрайства при збереженні прийнятного рівня хибних спрацювань, а також найвищу площу під ROC-кривою серед розглянутих моделей. Random Forest показав стабільні, але менш виражені результати, тоді як XGBoost продемонстрував хорошу якість у сценаріях, де

критичне значення має мінімізація пропусків шахрайських транзакцій навіть ціною деякого зростання хибних тривог.

Найбільш показовим є те, що гібридний стекінг-ансамбль у поточній конфігурації не досяг очікуваної переваги над кращою базовою моделлю. Незважаючи на використання потужних складників, ансамбль виявився надто чутливим до налаштуування гіперпараметрів, вибору метамоделі та порогу прийняття рішень. Це дозволило зробити важливий практичний висновок: у задачах виявлення шахрайства не завжди найбільш складна архітектура є найефективнішою; простіше, але добре налаштоване рішення часто виявляється більш надійним та зручним для впровадження.

Для підвищення прозорості прийняття рішень проведено аналіз із використанням інструментів Explainable AI, насамперед SHAP-значень. Це дало змогу кількісно оцінити внесок окремих ознак та їх груп у формування остаточного прогнозу моделі. Було показано, що найбільший вплив мають латентні компоненти, отримані внаслідок попередніх перетворень, сума транзакції, а також низка поведінкових характеристик, пов’язаних із частотою операцій та відхиленнями від нормального для клієнта режиму витрат. Такий аналіз не лише підсилює довіру до моделі з боку експертів і регуляторів, але й дає підказки для подальшого доопрацювання як самого набору ознак, так і бізнес-правил, що працюють разом з AI-моделлю.

Практичну значущість результатів підтверджено за рахунок розроблення концепції інтеграції моделі в інфраструктуру банківського моніторингу. У роботі запропоновано можливу архітектуру взаємодії моделі з потоками транзакцій, описано варіанти впровадження в режимі, наближеному до реального часу, узгодження з наявними процесами фінансового моніторингу та етапи взаємодії з аналітиками. На основі економічних розрахунків показано, що впровадження обраної моделі дозволяє потенційно зменшити прямі втрати від шахрайства й оптимізувати витрати на ручну перевірку інцидентів, знижуючи як кількість пропусків справжнього шахрайства, так і надлишкових блокувань

легітимних операцій. Це, своєю чергою, позитивно впливає на рівень довіри клієнтів до банку та зменшує репутаційні й регуляторні ризики.

Узагальнюючи результати, можна стверджувати, що поставлена в роботі мета досягнута повністю. Розроблено і досліджено гібридний підхід до виявлення шахрайських транзакцій, який поєднує сучасні методи попередньої обробки даних, корекцію дисбалансу класів, використання моделей машинного й глибинного навчання та інструментів Explainable AI. Сформовано завершену методику роботи з транзакційними даними – від збору й очищення до побудови, оцінювання та інтерпретації моделей, а також запропоновано практичні рекомендації щодо її інтеграції у реальні антифродмодель.

Наукова новизна одержаних результатів полягає в адаптації та поєднанні сучасних AI-підходів для задачі виявлення шахрайства в умовах значного дисбалансу класів, а також у застосуванні методів пояснюваного штучного інтелекту для глибокої інтерпретації поведінкових фінансових ознак. Практична цінність роботи полягає в можливості безпосереднього використання запропонованих рішень і підходів у фінансових установах для підвищення ефективності систем моніторингу й загального рівня фінансової кібербезпеки.

Перспективними напрямами подальших досліджень є застосування графових нейронних мереж для моделювання складних зв’язків між клієнтами, пристроями й транзакціями, розширення підходу на потокову обробку даних у режимі реального часу, а також регулярне донавчання моделей з урахуванням зміни тактик шахрайства. Окрему увагу доцільно приділити розвитку інтерпретованих моделей та дружніх до аналітиків інтерфейсів пояснення рішень AI. У сукупності це дозволить зробити системи фінансового моніторингу більш чутливими до сучасних кіберзагроз та стійкими до їх подальшої еволюції.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України «Про платіжні послуги» : Закон від 30.06.2021 р. № 1591-IX. Відомості Верховної Ради України. 2021.
2. Закон України «Про захист інформації в інформаційно-комунікаційних системах» : Закон від 05.07.1994 р. № 80/94-ВР. Відомості Верховної Ради України. 1994. № 31. Ст. 286.
3. Про заходи кіберзахисту в банківському секторі України : Постанова Правління Національного банку України від 28.09.2020 р. № 95. URL: https://zakon.rada.gov.ua/laws/show/v0095500-20 (дата звернення: 01.12.2025).
4. Зарніцин Д. В. Штучний інтелект у виявленні шахрайських транзакцій у фінансових установах. Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу : тези доп. III Всеукр. наук.-техн. конф., м. Київ, 2025. Секція «Штучний інтелект, машинне навчання у побуті і промисловості».
5. Зарніцин Д. В. Штучний інтелект у виявленні шахрайських транзакцій у фінансових установах. Експериментальні та теоретичні дослідження в контексті сучасної науки : тези доп. X Всеукр. студент. конф., м. Дніпро, 2026. Секція «Системний аналіз, моделювання та оптимізація».
6. Савчук І. Основи кібербезпеки : практикум. Львів : ЛНУ, 2022. 215 с.
7. Холованов М. Інтелектуальний аналіз даних. Київ : КНЕУ, 2021. 312 с.
8. Шматков О. Методи машинного навчання в економіці. Київ : КНЕУ, 2023. 280 с.
9. Abdallah A., Maarof M., Zainal A. Fraud detection system: A survey. Journal of Network and Computer Applications. 2016. Vol. 68. P. 90–113.

10. Aggarwal C. C. Data Mining: The Textbook. Springer, 2015. 734 p.

11. Bahnsen A., Aouada D., Ottersten B. Example-Dependent Cost-Sensitive Logistic Regression for Credit Card Fraud Detection. Machine Learning and Knowledge Discovery in Databases. 2014. P. 135–136.

12. Bishop C. M. Pattern Recognition and Machine Learning. Springer, 2006. 738 p.

13. Breiman L. Random Forests. Machine Learning. 2001. Vol. 45, No. 1. P. 5–32.

14. Carcillo F. et al. SCARFF: a Scalable FRamework for streaming Fraud detection with Spark. Information Fusion. 2018. Vol. 41. P. 182–194.

15. Chawla N. V., Bowyer K. W., Hall L. O., Kegelmeyer W. P. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research. 2002. Vol. 16. P. 321–357.

16. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016. P. 785–794.

17. Chollet F. Deep Learning with Python. 2nd ed. Manning Publications, 2021. 504 p.

18. Dal Pozzolo A. et al. Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy. IEEE Transactions on Neural Networks and Learning Systems. 2018. Vol. 29, No. 8. P. 3784–3797.

19. European Payments Council. Annual Fraud Report 2024. URL: https://www.europeanpaymentscouncil.eu (accessed: 01.12.2025).

20. FraudLabs Pro. Global Online Fraud Trends 2024. URL: https://www.fraudlabspro.com (accessed: 01.12.2025).

21. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 3rd ed. O'Reilly Media, 2022. 850 p.

22. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. 800 p.

23. Goyal P., Ferrara E. Graph embedding techniques, applications, and performance: A survey. Knowledge-Based Systems. 2018. Vol. 151. P. 78–94.

24. He H., Garcia E. A. Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering. 2009. Vol. 21, No. 9. P. 1263–1284.

25. IBM Security. X-Force Threat Intelligence Index 2024. URL: https://www.ibm.com/security (accessed: 01.12.2025).

26. ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection — Information security management systems — Requirements. ISO, 2022.

27. James G., Witten D., Hastie T., Tibshirani R. An Introduction to Statistical Learning: with Applications in R. 2nd ed. Springer, 2021. 607 p.

28. Jurgovsky J. et al. Sequence classification for credit-card fraud detection. Expert Systems with Applications. 2018. Vol. 100. P. 234–245.

29. Kaggle. Credit Card Fraud Detection Dataset. URL: https://www.kaggle.com/mlg-ulb/creditcardfraud (accessed: 01.12.2025).

30. Keras API Reference. URL: https://keras.io/ (accessed: 01.12.2025).

31. Kuhn M., Johnson K. Applied Predictive Modeling. Springer, 2013. 600 p.

32. Li J. et al. Survey on deep learning for anomaly detection. Journal of Big Data. 2019.

33. Marchal S., Jiang X., State R., Engel T. A Big Data Architecture for Large Scale Security Monitoring. IEEE International Congress on Big Data. 2014.

34. Mastercard. SafetyNet Behavioral Fraud Detection. 2023. URL: https://www.mastercard.com (accessed: 01.12.2025).

35. Murphy K. P. Probabilistic Machine Learning: An Introduction. MIT Press, 2022.

36. Pandas Library Reference. URL: https://pandas.pydata.org/ (accessed: 01.12.2025).

37. PCI Security Standards Council. PCI DSS v4.0: Payment Card Industry Data Security Standard. 2022.

38. Provost F., Fawcett T. Data Science for Business. O'Reilly Media, 2013. 414 p.

39. Python Software Foundation. URL: https://www.python.org/ (accessed: 01.12.2025).

40. Russell S., Norvig P. Artificial Intelligence: A Modern Approach. 4th ed. Pearson, 2020. 1166 p.

41. Salesforce. Financial Data Anomaly Detection — Technical Report. 2023.

42. Scikit-learn documentation. URL: https://scikit-learn.org/ (accessed: 01.12.2025).

43. TensorFlow documentation. URL: https://www.tensorflow.org/ (accessed: 01.12.2025).

44. VISA. AI Fraud Prevention Solutions — White Paper. 2022. URL: https://usa.visa.com (accessed: 01.12.2025).

45. Whitrow C. et al. Transaction aggregation as a strategy for credit card fraud detection. Data Mining and Knowledge Discovery. 2009. Vol. 18. P. 30–55.

46. World Bank. Global Findex Database 2021: Financial Inclusion, Digital Payments, and Resilience in the Age of COVID-19. 2022.

47. XGBoost Documentation. URL: https://xgboost.readthedocs.io/ (accessed: 01.12.2025).

48. Yin C., Zhao X. A Survey on Deep Learning for Fraud Detection. IEEE Access. 2021.

ДОДАТОК А

Програмний модуль побудови моделей виявлення шахрайських транзакцій (all_parts_pipeline.py)

У цьому додатку наведено програмний код повного однофайлового пайплайну all_parts_pipeline.py, який реалізує:

- завантаження та попередню обробку даних датасету кредитних карткових транзакцій;
- балансування вибірки із застосуванням методу SMOTE;
- масштабування ознак за допомогою StandardScaler;
- навчання моделей Random Forest, XGBoost, MLP та стекінг-ансамблю (StackingClassifier);
- обчислення основних метрик якості (AUC-ROC, Average Precision, Precision, Recall, F1, Accuracy);
- побудову та збереження графіків ROC-кривих, кривих Precision–Recall і матриць неточностей для кожної моделі.

А.1. Лістинг програмного коду all_parts_pipeline.py

```
# all_parts_pipeline.py
# Повний однофайловий пайплайн для дипломної роботи
# Включає: завантаження даних, SMOTE, масштабування, RF, XGB, MLP, Stacking
# Графіки: ROC, Precision-Recall, Confusion Matrix (збережені в /figures)

import os
import warnings
warnings.filterwarnings("ignore")
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from collections import Counter

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import (
    roc_curve, auc, precision_recall_curve,
    confusion_matrix, ConfusionMatrixDisplay,
    roc_auc_score, average_precision_score,
    classification_report
)

from imblearn.over_sampling import SMOTE

from sklearn.ensemble import RandomForestClassifier, StackingClassifier
from xgboost import XGBClassifier
from sklearn.neural_network import MLPClassifier

# ----------------------------
# Параметри / шляхи
# ----------------------------
DATA_FILES = ["creditcard (1).csv", "creditcard.csv"]  # спробуємо ці імена
FIG_DIR = "figures"
RANDOM_STATE = 42
TEST_SIZE = 0.2
```

```
os.makedirs(FIG_DIR, exist_ok=True)


# ---------------------------

# Завантаження даних

# ---------------------------

df = None

for fn in DATA_FILES:

    if os.path.exists(fn):

        df = pd.read_csv(fn)

        print(f"Завантажено: {fn}")

        break


if df is None:

    raise FileNotFoundError(

        f"Не знайдено файл(ів): {DATA_FILES}. Помістіть CSV у робочу теку."

    )


print("Розмір датасету:", df.shape)

print(df.head())


# ---------------------------

# Підготовка ознак і цілі

# ---------------------------

# Припускаємо стандартну структуру датасету creditcardfraud: Time, V1..V28,
Amount, Class

if "Class" not in df.columns:

    raise ValueError("Очікував колонку 'Class' у датасеті.")


X = df.drop(columns=["Class"])
```

```
y = df["Class"]


print("\nРозподіл класів до SMOTE (усі дані):")
print(y.value_counts())


# ---------------------------
# Розбиття на train/test (SMOTE тільки на train)
# ---------------------------
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=TEST_SIZE, random_state=RANDOM_STATE, stratify=y
)


print("\nРозміри після розбиття:")
print("X_train:", X_train.shape, "X_test:", X_test.shape)
print("y_train value counts (before SMOTE):")
print(y_train.value_counts())


# ---------------------------
# SMOTE на тренувальному наборі
# ---------------------------
sm = SMOTE(random_state=RANDOM_STATE)
X_train_res, y_train_res = sm.fit_resample(X_train, y_train)


print("\nПісля SMOTE (train):")
print(Counter(y_train_res))


# ---------------------------
# Масштабування (StandardScaler)
# ---------------------------
```

```
scaler = StandardScaler()
# Навчаємо скейлер тільки на X_train_res, потім трансформуємо train/test
X_train_res_scaled = scaler.fit_transform(X_train_res)
X_test_scaled = scaler.transform(X_test)



# ----------------------------
# Допоміжні функції
# ----------------------------
def evaluate_model(name, model, X_test, y_test, save_prefix=None):
    """
    Обчислює основні метрики, друкує classification_report і повертає
    прогнозні ймовірності для позитивного класу.
    Також зберігає текстовий звіт у файл, якщо save_prefix заданий.
    """
    y_prob = model.predict_proba(X_test)[:, 1]
    y_pred = model.predict(X_test)

    roc_auc = roc_auc_score(y_test, y_prob)
    ap = average_precision_score(y_test, y_prob)
    report = classification_report(y_test, y_pred, digits=4)

    print(f"\n--- {name} ---")
    print(f"AUC-ROC: {roc_auc:.4f}")
    print(f"Average Precision (AP): {ap:.4f}")
    print(report)

    if save_prefix:
        with open(os.path.join(FIG_DIR, f"{save_prefix}_report.txt"), "w") as f:
```

```
            f.write(f"AUC-ROC: {roc_auc:.6f}\n")
            f.write(f"Average Precision (AP): {ap:.6f}\n\n")
            f.write(report)

    return y_prob


def plot_and_save_roc_pr_cm(y_test, y_prob, y_pred, title_prefix):
    # ROC
    fpr, tpr, _ = roc_curve(y_test, y_prob)
    roc_auc = auc(fpr, tpr)
    plt.figure()
    plt.plot(fpr, tpr, lw=2, label=f"ROC (AUC = {roc_auc:.4f})")
    plt.plot([0, 1], [0, 1], linestyle="--", lw=1, color="gray")
    plt.xlabel("False Positive Rate")
    plt.ylabel("True Positive Rate")
    plt.title(f"{title_prefix} — ROC Curve")
    plt.legend(loc="lower right")
    plt.grid(alpha=0.3)
    fn = os.path.join(FIG_DIR, f"{title_prefix}_roc.png")
    plt.savefig(fn, bbox_inches="tight", dpi=150)
    plt.show()

    # Precision-Recall
    precision, recall, _ = precision_recall_curve(y_test, y_prob)
    ap = average_precision_score(y_test, y_prob)
    plt.figure()
    plt.plot(recall, precision, lw=2, label=f"AP = {ap:.4f}")
    plt.xlabel("Recall")
```

```
    plt.ylabel("Precision")
    plt.title(f"{title_prefix} — Precision-Recall")
    plt.legend(loc="lower left")
    plt.grid(alpha=0.3)
    fn = os.path.join(FIG_DIR, f"{title_prefix}_pr.png")
    plt.savefig(fn, bbox_inches="tight", dpi=150)
    plt.show()

    # Confusion Matrix
    cm = confusion_matrix(y_test, y_pred)
    disp = ConfusionMatrixDisplay(cm)
    fig, ax = plt.subplots(figsize=(5, 4))
    disp.plot(ax=ax)
    plt.title(f"{title_prefix} — Confusion Matrix")
    fn = os.path.join(FIG_DIR, f"{title_prefix}_cm.png")
    plt.savefig(fn, bbox_inches="tight", dpi=150)
    plt.show()



# ---------------------------
# ЧАСТИНА 2: RANDOM FOREST
# ---------------------------
print("\n=== ЧАСТИНА 2: RANDOM FOREST ===")
rf = RandomForestClassifier(
    n_estimators=200,
    min_samples_leaf=2,
    n_jobs=-1,
    random_state=RANDOM_STATE
)
```

```
rf.fit(X_train_res, y_train_res)



rf_prob = evaluate_model("RandomForest", rf, X_test_scaled, y_test,
save_prefix="rf")

rf_pred = rf.predict(X_test_scaled)

plot_and_save_roc_pr_cm(y_test, rf_prob, rf_pred, "RandomForest")



# ---------------------------

# ЧАСТИНА 3: XGBOOST

# ---------------------------

print("\n=== ЧАСТИНА 3: XGBOOST ===")

xgb = XGBClassifier(

    n_estimators=300,

    learning_rate=0.05,

    max_depth=5,

    subsample=0.8,

    colsample_bytree=0.8,

    eval_metric="logloss",

    use_label_encoder=False,

    random_state=RANDOM_STATE,

    n_jobs=-1

)

xgb.fit(X_train_res, y_train_res)



xgb_prob = evaluate_model("XGBoost", xgb, X_test_scaled, y_test,
save_prefix="xgb")

xgb_pred = xgb.predict(X_test_scaled)

plot_and_save_roc_pr_cm(y_test, xgb_prob, xgb_pred, "XGBoost")



# ---------------------------
```

```
# ЧАСТИНА 4: MLP НЕЙРОМЕРЕЖА
# ----------------------------
print("\n=== ЧАСТИНА 4: MLP НЕЙРОМЕРЕЖА ===")
mlp = MLPClassifier(
    hidden_layer_sizes=(64, 32),
    activation="relu",
    max_iter=500,
    random_state=RANDOM_STATE
)
mlp.fit(X_train_res_scaled, y_train_res)

mlp_prob = evaluate_model("MLP", mlp, X_test_scaled, y_test, save_prefix="mlp")
mlp_pred = mlp.predict(X_test_scaled)
plot_and_save_roc_pr_cm(y_test, mlp_prob, mlp_pred, "MLP")

# ----------------------------
# ЧАСТИНА 5: STACKING (RF + XGB + MLP)
# ----------------------------
print("\n=== ЧАСТИНА 5: STACKING ENSEMBLE ===")
stack_clf = StackingClassifier(
    estimators=[
        (
            "rf",
            RandomForestClassifier(
                n_estimators=200,
                min_samples_leaf=2,
                random_state=RANDOM_STATE,
                n_jobs=-1,
            ),
```

```
        ),
        (
            "xgb",
            XGBClassifier(
                n_estimators=300,
                learning_rate=0.05,
                max_depth=5,
                subsample=0.8,
                colsample_bytree=0.8,
                use_label_encoder=False,
                eval_metric="logloss",
                random_state=RANDOM_STATE,
                n_jobs=-1,
            ),
        ),
    ],
    final_estimator=MLPClassifier(
        hidden_layer_sizes=(32, 16),
        activation="relu",
        max_iter=300,
        random_state=RANDOM_STATE,
    ),
    stack_method="predict_proba",
    n_jobs=-1,
)

stack_clf.fit(X_train_res, y_train_res)

stack_prob = evaluate_model(
```

```
    "StackingEnsemble", stack_clf, X_test_scaled, y_test, save_prefix="stacking"
)
stack_pred = stack_clf.predict(X_test_scaled)
plot_and_save_roc_pr_cm(y_test, stack_prob, stack_pred, "StackingEnsemble")

# ---------------------------
# ПОРІВНЯЛЬНІ ГРАФІКИ (ROC ВСІХ МОДЕЛЕЙ)
# ---------------------------
print("\n=== ПОРІВНЯННЯ: ROC ВСІХ МОДЕЛЕЙ ===")
plt.figure(figsize=(7, 6))

fpr, tpr, _ = roc_curve(y_test, rf_prob)
plt.plot(fpr, tpr, lw=2, label=f"RF (AUC={auc(fpr, tpr):.4f})")

fpr, tpr, _ = roc_curve(y_test, xgb_prob)
plt.plot(fpr, tpr, lw=2, label=f"XGB (AUC={auc(fpr, tpr):.4f})")

fpr, tpr, _ = roc_curve(y_test, mlp_prob)
plt.plot(fpr, tpr, lw=2, label=f"MLP (AUC={auc(fpr, tpr):.4f})")

fpr, tpr, _ = roc_curve(y_test, stack_prob)
plt.plot(fpr, tpr, lw=2, label=f"Stacking (AUC={auc(fpr, tpr):.4f})")

plt.plot([0, 1], [0, 1], linestyle="--", color="gray")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Comparison")
plt.legend(loc="lower right")
plt.grid(alpha=0.3)
```

```
plt.savefig(os.path.join(FIG_DIR, "comparison_roc.png"), bbox_inches="tight",
dpi=150)

plt.show()



print("\nГотово! Всі графіки збережені у папці:", FIG_DIR)
```

А.2. Результати роботи програмного модуля

Після виконання модуля на датасеті creditcard.csv було отримано такі загальні характеристики даних:

- Розмір датасету: 284 807 рядків, 31 стовпець.
- Розподіл класів до балансування (усі дані):
    1. Клас 0 (нормальні транзакції): 284 315
    2. Клас 1 (шахрайські транзакції): 492
- Розміри після поділу на train/test:
    1. X_train: 227 845 спостережень, 30 ознак
    2. X_test: 56 962 спостереження, 30 ознак
- Розподіл y_train до SMOTE:
    1. Клас 0: 227 451
    2. Клас 1: 394
- Після застосування SMOTE до тренувальної вибірки:
    1. Обидва класи збалансовані: 0: 227 451, 1: 227 451.

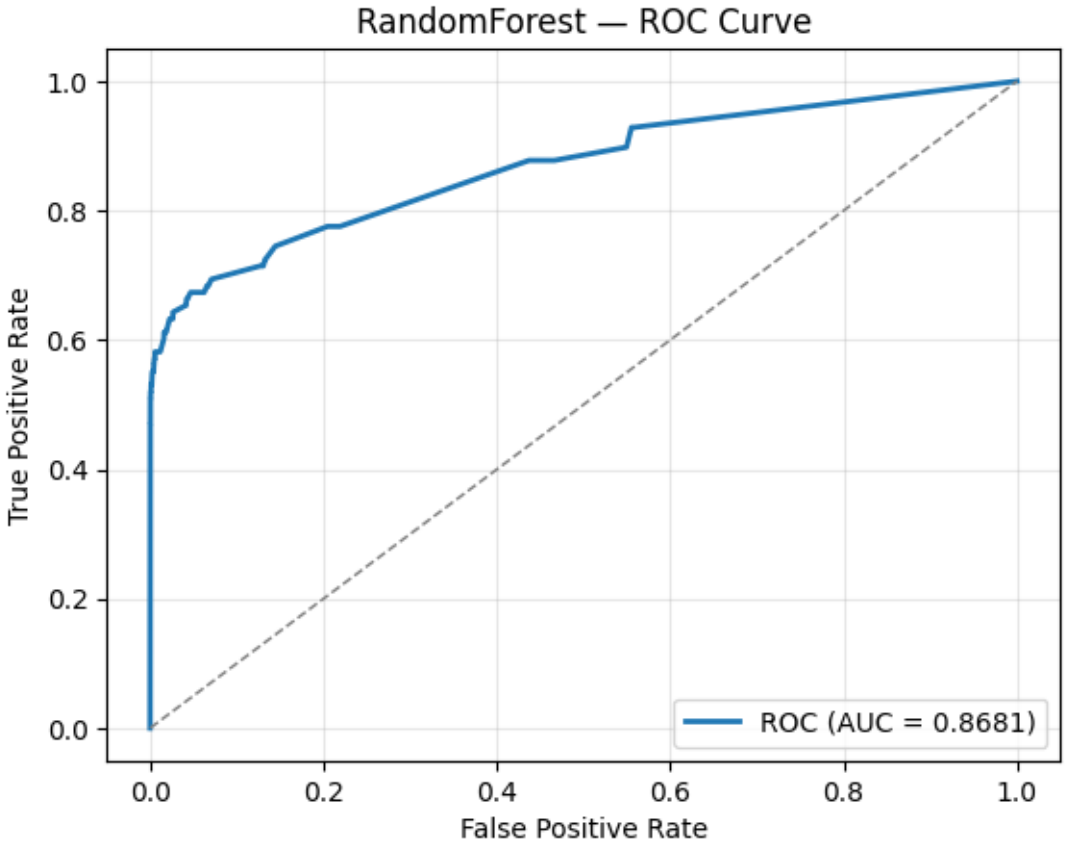
Основні підсумкові метрики якості для позитивного класу (шахрайські транзакції, Class = 1) наведено в таблиці.

Модель	**AUC-ROC**	**AP**	**Precision (клас 1)**	**Recall (клас 1)**	**F1-score (клас 1)**	**Accuracy**
Random Forest	0.8681	0.4296	0.6500	0.1327	0.2203	0.9984
XGBoost	0.9525	0.6156	0.8857	0.3163	0.4662	0.9988
MLP-мережа	0.9595	0.8331	0.7431	0.8265	0.7826	0.9992
Stacking Ensemble	0.8171	0.3790	0.5000	0.0204	0.0392	0.9983

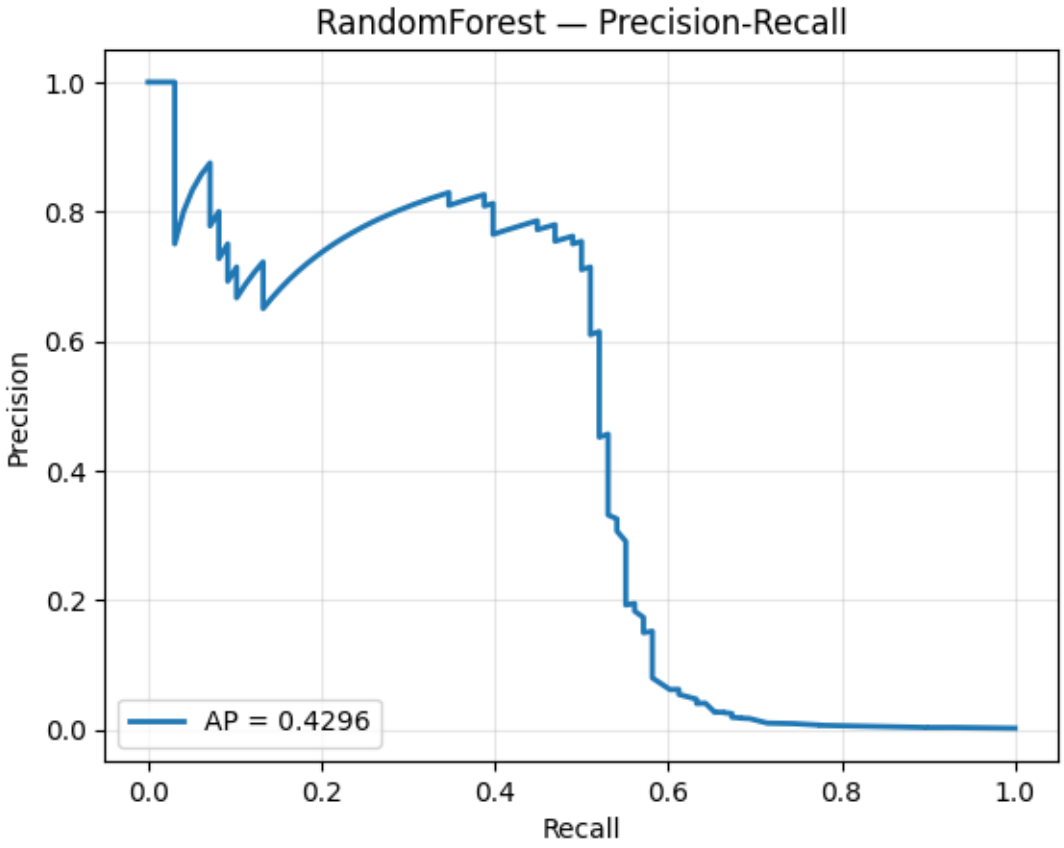
На основі отриманих результатів найкращу якість класифікації шахрайських операцій продемонструвала нейронна мережа MLP, яка забезпечила високе значення AUC-ROC (= 0.9595), дуже високе значення Average Precision (= 0.8331), а також найкращий баланс між повнотою (Recall = 0.8265) та точністю (Precision = 0.7431) для позитивного класу.

А.3. Графічні результати

Під час виконання програмного модуля були побудовані графічні результати, які ілюструють якість роботи окремих моделей та їх порівняння. На рисунках А.1–А.12 наведено ROC-криві, криві Precision–Recall та матриці неточностей для моделей Random Forest, XGBoost, MLP та стекінг-ансамблю. На рисунку А.13 подано порівняльну ROC-криву всіх розглянутих моделей на тестовій вибірці.

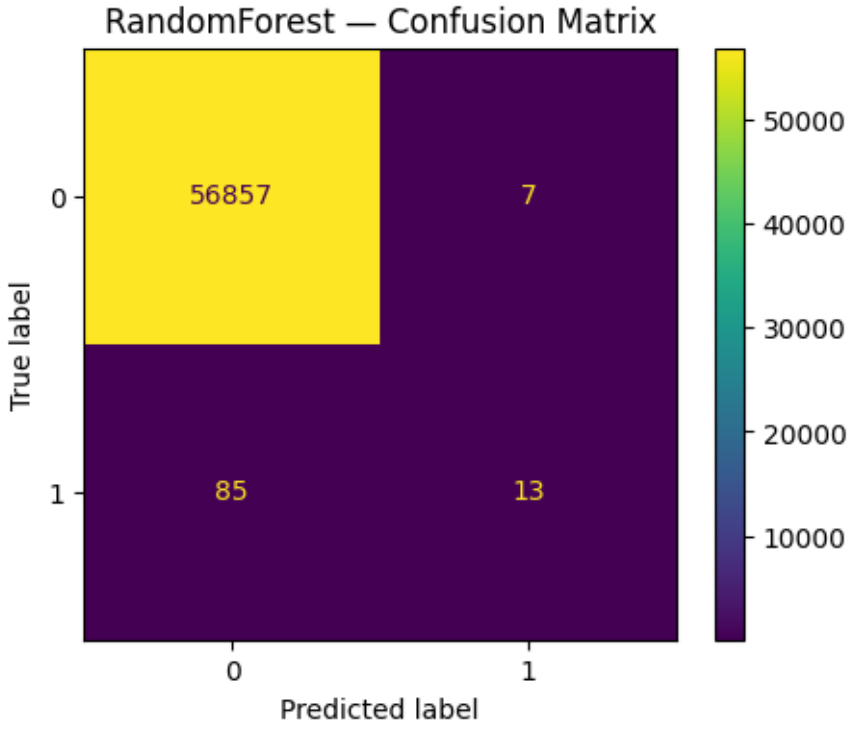


RandomForest — ROC Curve:
Рисунок А.1 – ROC-крива моделі Random Forest на тестовій вибірці (AUC = 0,8681).0,8681).



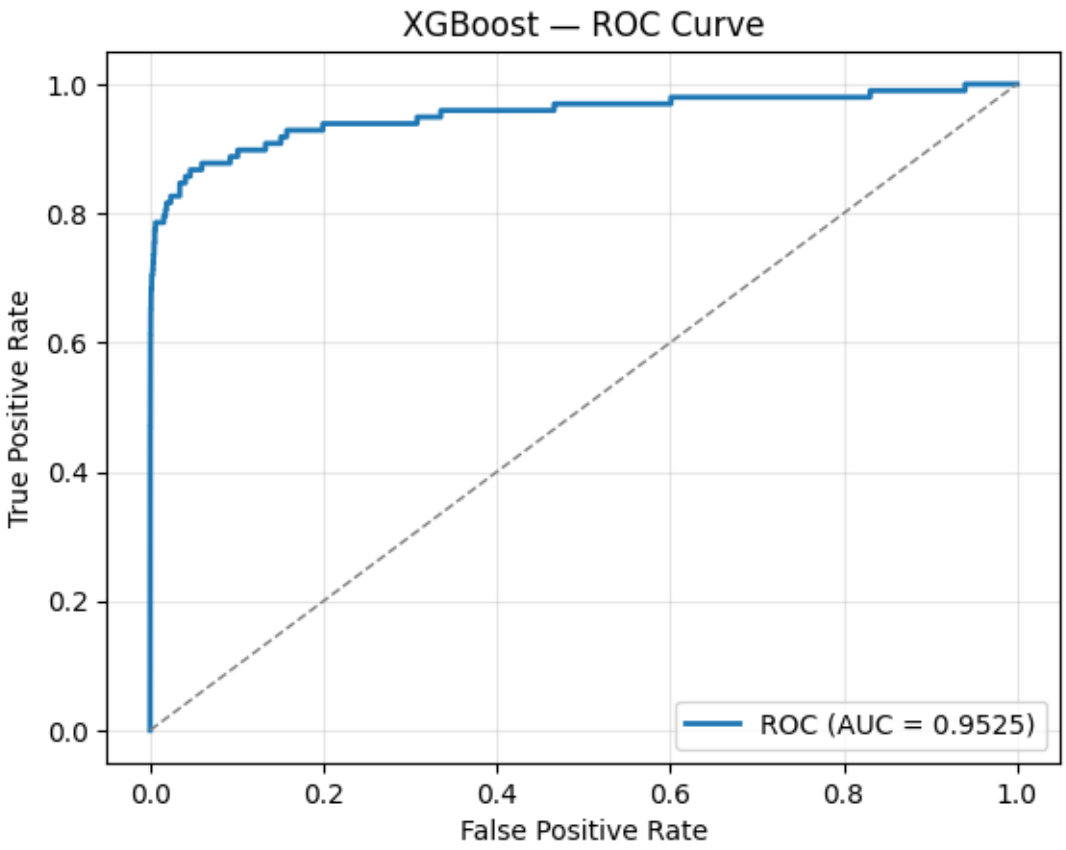
RandomForest — Precision-Recall:

Рисунок А.2 – Крива Precision–Recall для моделі Random Forest на тестовій вибірці (AP = 0,4296).



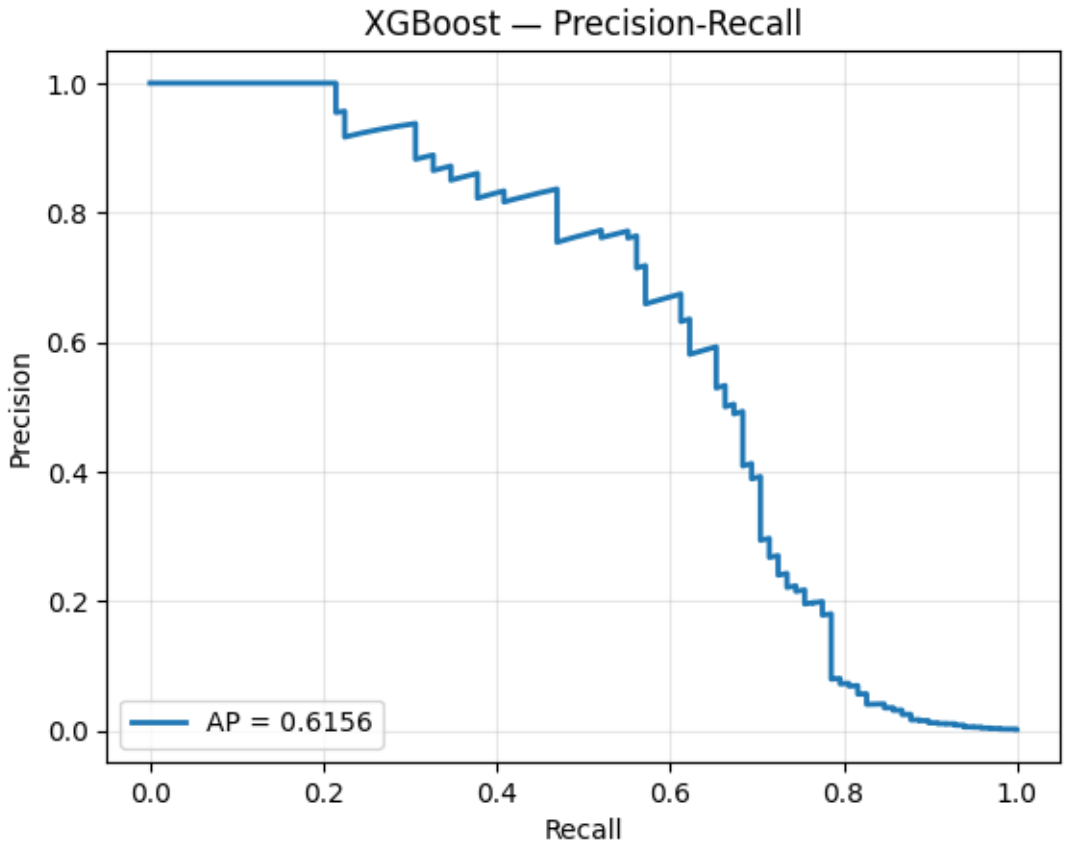
RandomForest — Confusion Matrix:

Рисунок А.3 – Матриця неточностей моделі Random Forest на тестовій вибірці.

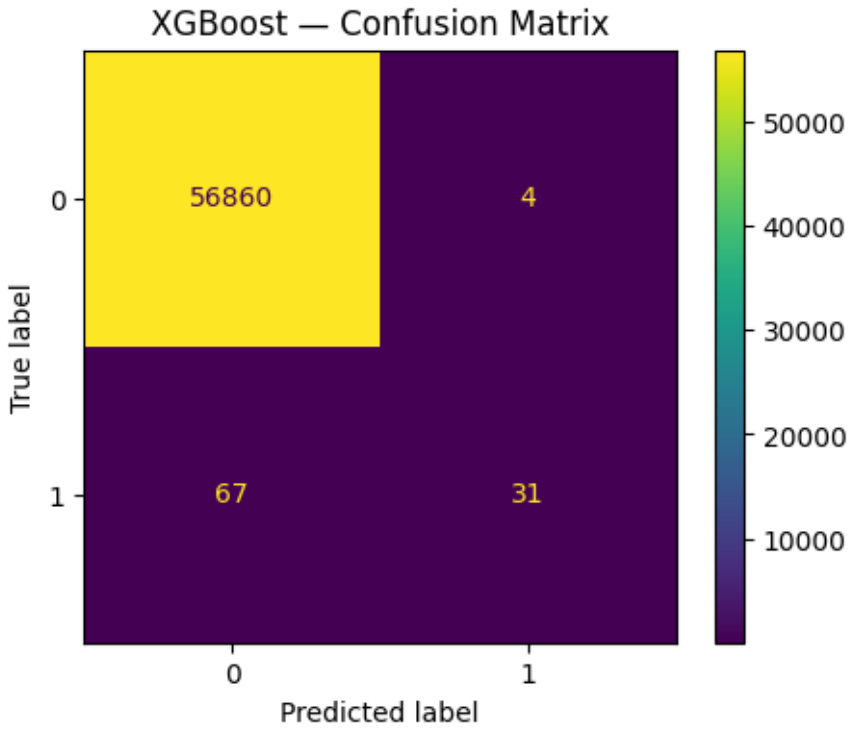


XGBoost — ROC Curve:

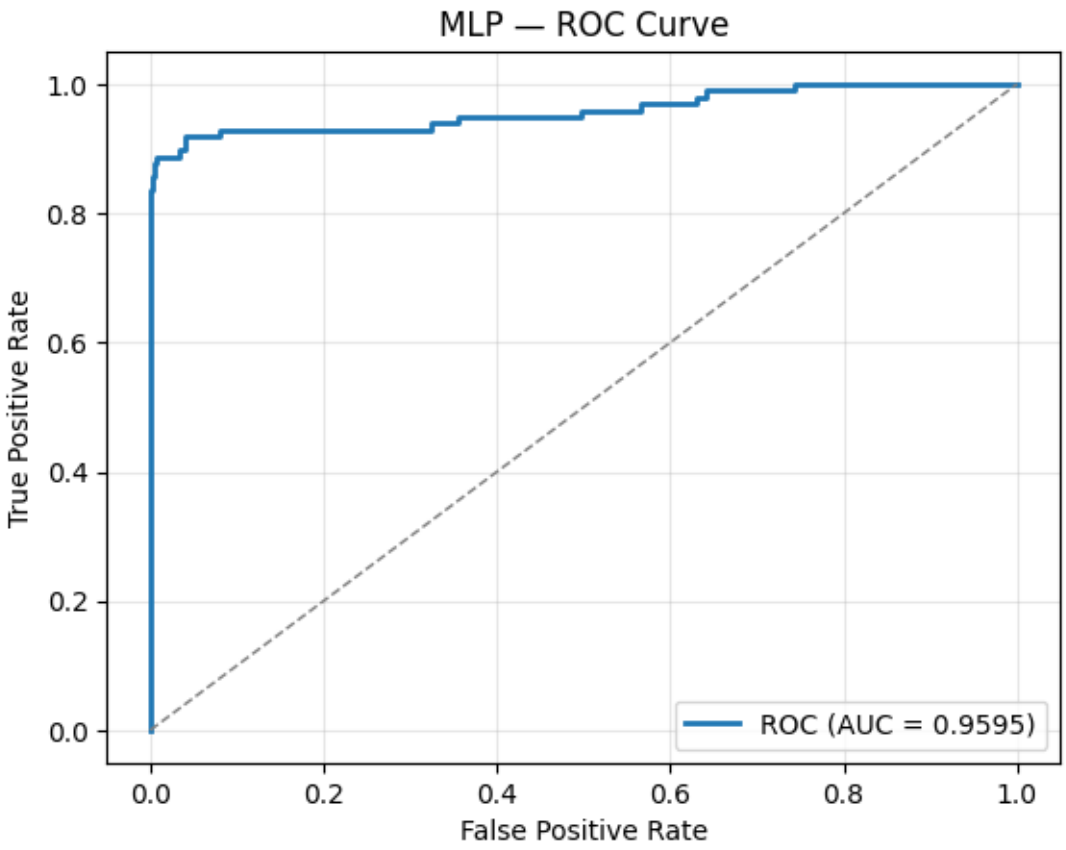
Рисунок А.4 – ROC-крива моделі XGBoost на тестовій вибірці (AUC = 0,9525).



XGBoost — Precision-Recall:
Рисунок А.5 – Крива Precision–Recall для моделі XGBoost на тестовій вибірці (AP = 0,6156)

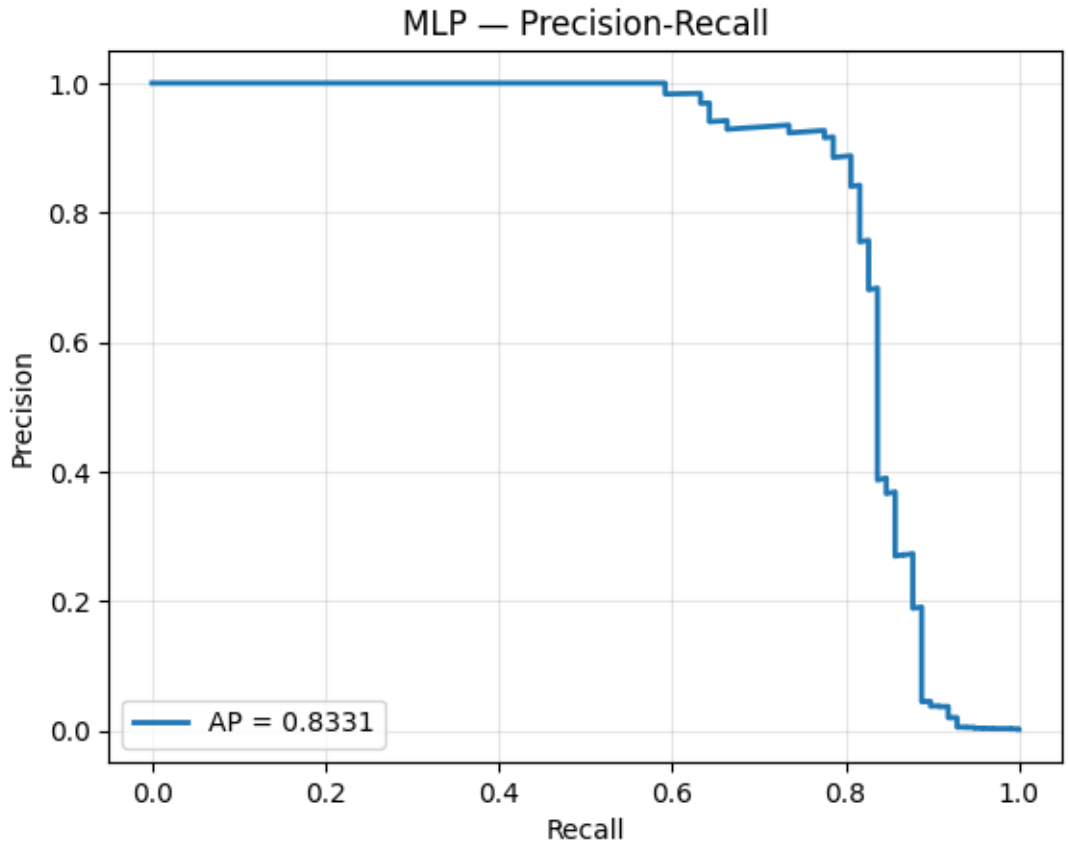


XGBoost — Confusion Matrix:
Рисунок А.6 – Матриця неточностей моделі XGBoost на тестовій вибірці.

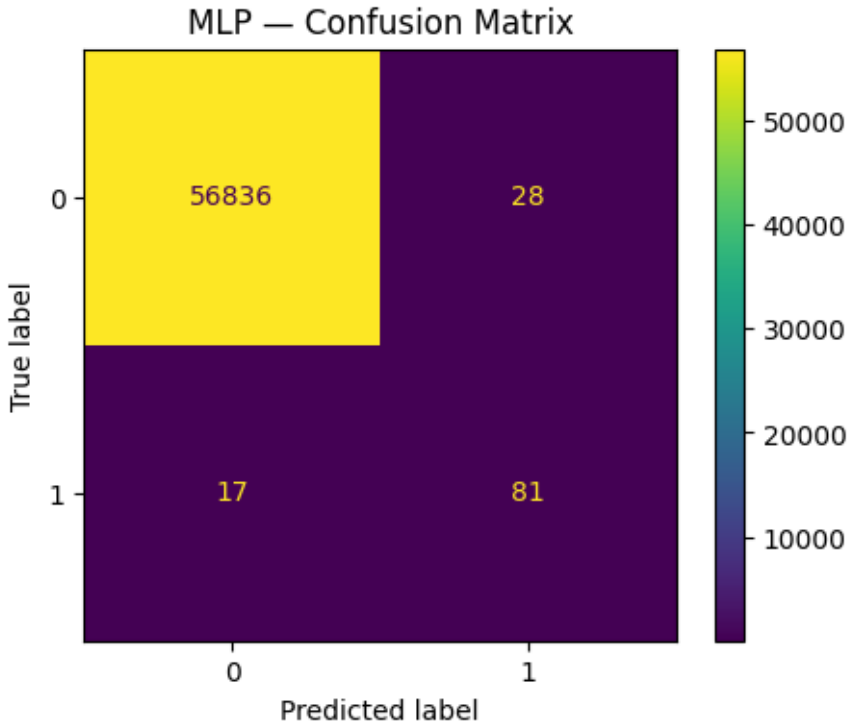


MLP — ROC Curve:

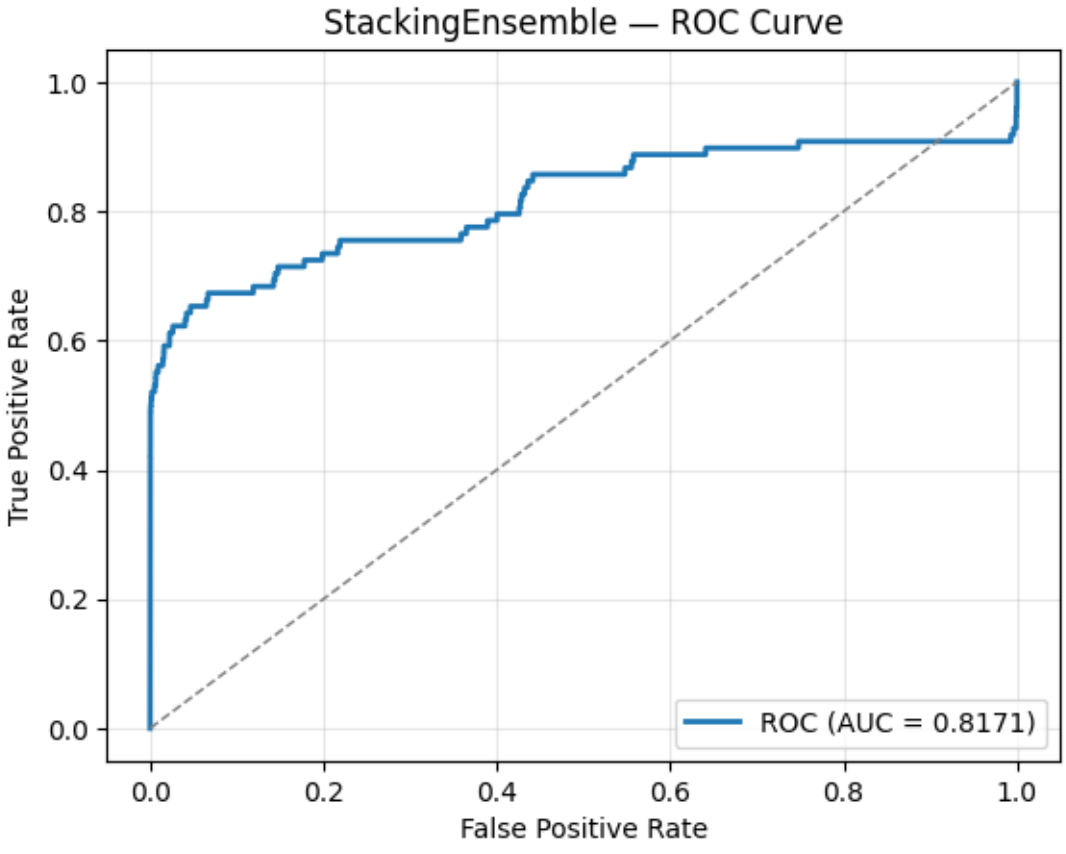
Рисунок А.7 – ROC-крива моделі MLP на тестовій вибірці (AUC = 0,9595).



MLP — Precision-Recall:
Рисунок А.8 – Крива Precision–Recall для моделі MLP на тестовій вибірці (AP = 0,8331).

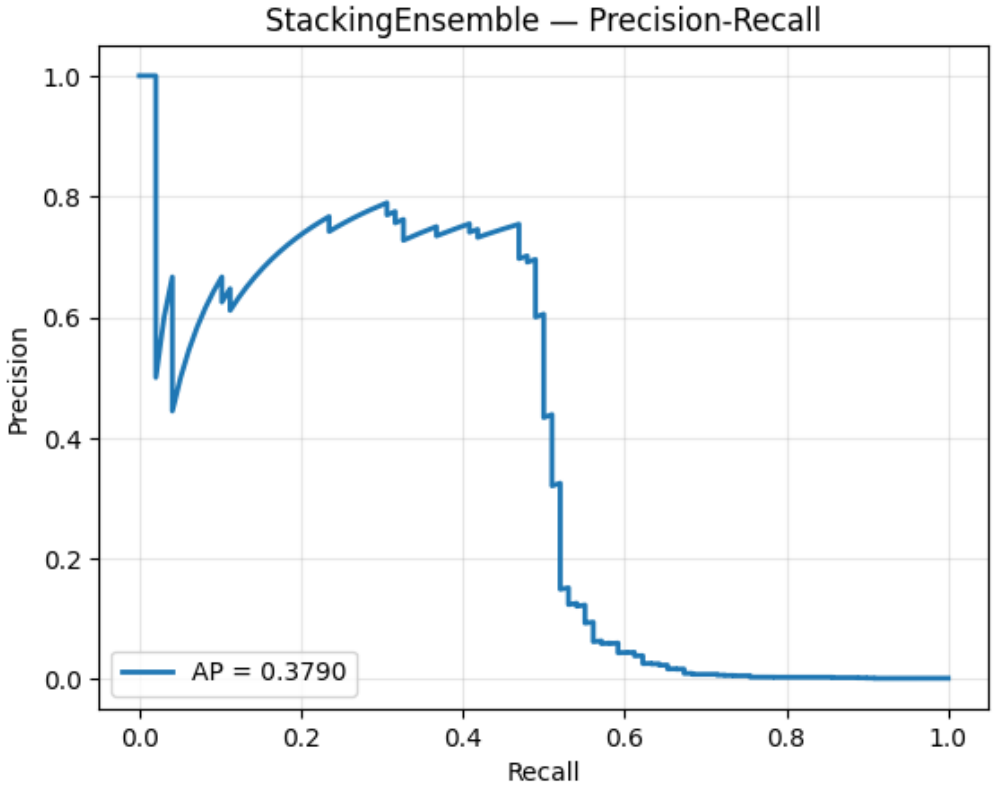


MLP — Confusion Matrix:
Рисунок А.9 – Матриця неточностей моделі MLP на тестовій вибірці.



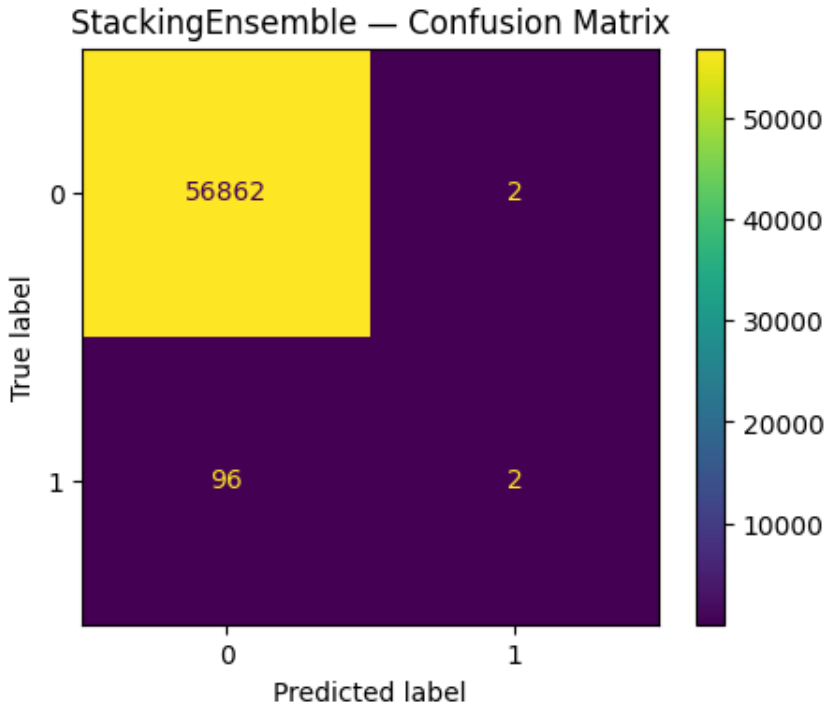
StackingEnsemble — ROC Curve:

Рисунок А.10 – ROC-крива стекінг-ансамблю на тестовій вибірці (AUC = 0,8171).

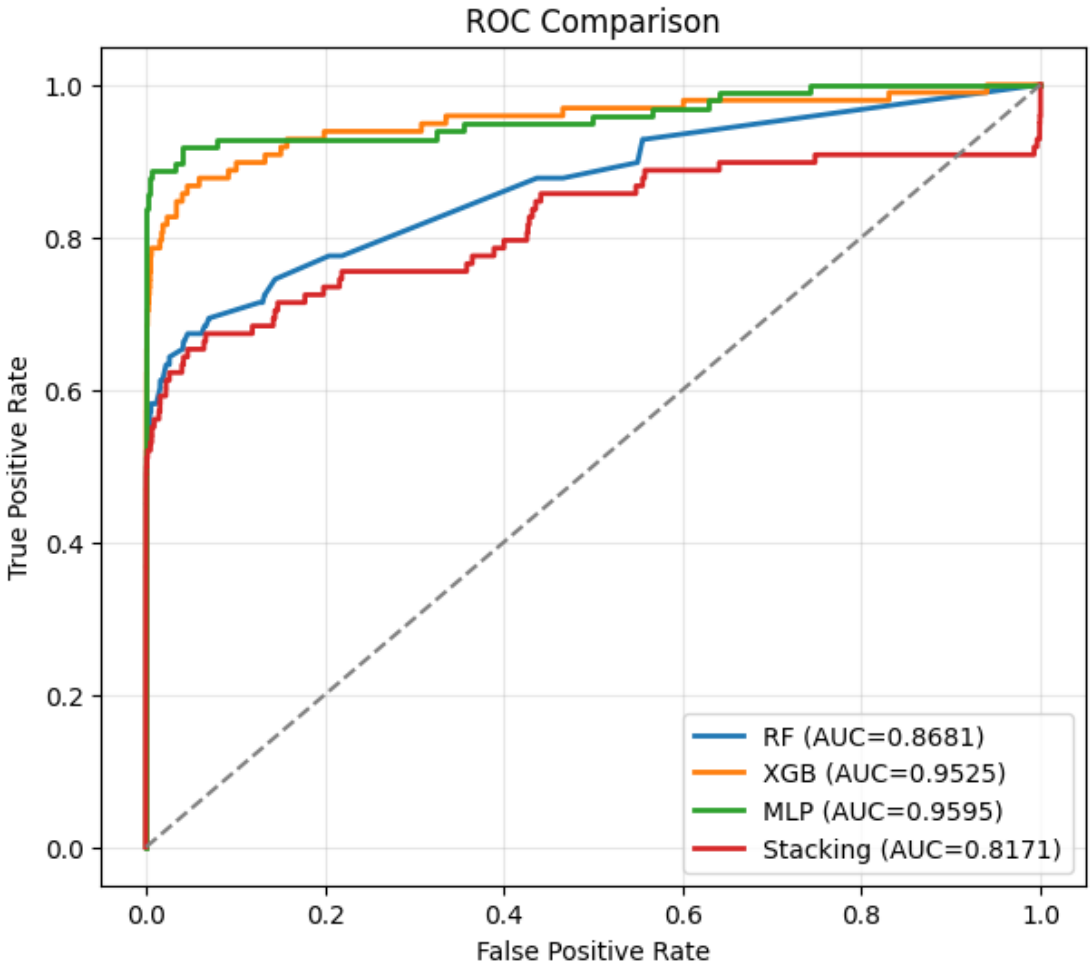


StackingEnsemble — Precision-Recall:

Рисунок А.11 – Крива Precision–Recall для стекінг-ансамблю на тестовій вибірці (AP = 0,3790).



StackingEnsemble — Confusion Matrix:
Рисунок А.12 – Матриця неточностей стекінг-ансамблю на тестовій вибірці.



ROC Comparison:
Рисунок А.13 – Порівняльні ROC-криві моделей Random Forest, XGBoost, MLP та стекінг-ансамблю на тестовій вибірці.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

«Штучний інтелект у виявленні шахрайських транзакцій у фінансових установах»

Виконав студент групи САДМ-61
ЗАРНІЦИН Денис Володимирович

Керівник кваліфікаційної роботи:
КУЗМІЧ Михайло Юрійович,
доцент кафедри ІСТ

Київ – 2025

Слайд 1

2

Мета магістерської роботи:

Розроблення, навчання та валідація гібридного підходу на основі методів штучного інтелекту для автоматизованого виявлення шахрайських транзакцій у фінансових установах в умовах суттєвого дисбалансу вибірки.

Об’єкт дослідження:

Процес виявлення шахрайських транзакцій у фінансових системах із застосуванням інтелектуальних методів аналізу даних.

Предмет дослідження:

Алгоритми машинного та глибинного навчання (ансамблеві моделі та штучні нейронні мережі), а також методи оброблення незбалансованих фінансових даних у задачах класифікації транзакцій.

Слайд 2

3

Актуальність:

Слайд 3

4

Основні завдання дослідження:

- Провести аналітичний огляд сучасних кіберзагроз і шахрайських схем у цифровому банкінгу.
- Дослідити еволюцію систем фінансового моніторингу та проаналізувати можливості AI-методів (Random Forest, XGBoost, MLP).
- Розробити методику підготовки та балансування транзакційних даних (очищення, нормалізація, feature engineering, SMOTE).
- Побудувати, навчити і налаштувати набір моделей (RF, XGBoost, MLP) та гібридну стекінг-модель.
- Провести експериментальну оцінку якості моделей (Precision, Recall, F_1-score, ROC-AUC).
- Розробити концепцію інтеграції моделі у банківську систему моніторингу та оцінити економічну ефективність.

Слайд 4

Слайд 5

Сладй 6

Слайд 7

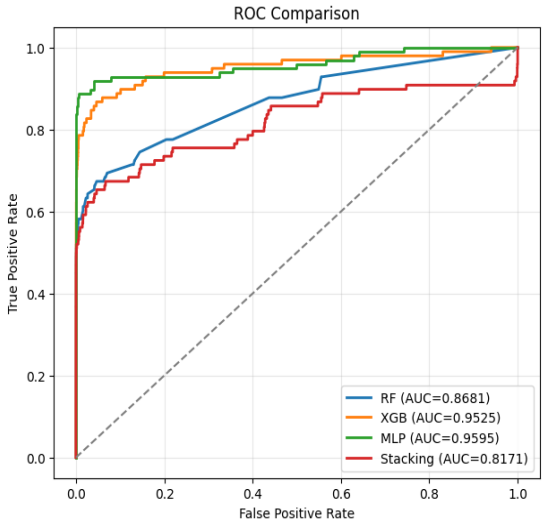
Слайд 8

9

Основні результати моделювання

Найкращі результати показала модель MLP (AUC = 0.9595, Precision ≈ 0.74, Recall ≈ 0.83, F_1-score ≈ 0.78).

1. Модель MLP має найвищий ROC-AUC та найкращий баланс показників Precision / Recall.
2. Поточний стекінг-ансамбль не перевищив найкращу базову модель, але залишається перспективним для подальшої оптимізації.
3. Застосування SMOTE підвищило чутливість усіх моделей до шахрайських транзакцій.



Сладй 9

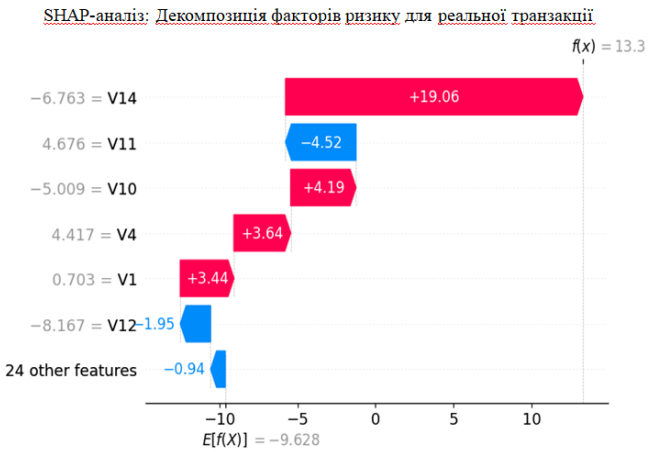
10

Explainable AI (SHAP-аналіз)

1. Мета інтеграції SHAP:
- Інтерпретація «Чорної скриньки» (MLP).
- Забезпечення прозорості згідно з **AI Act**.

2. Ключові маркери ризику:
- Аномальна поведінка (**V14**, **V4**).
- Часові інтервали між операціями.

3. Практична цінність:
- Математичне обгрунтування блокувань.
- Зниження помилкових спрацювань (False Positives).



Слайд 10

11

Висновки

Слайд 11

12

АПРОБАЦІЯ РЕЗУЛЬТАТІВ

Основні положення магістерської роботи пройшли апробацію на науковопрактичних конференціях всеукраїнського рівня.

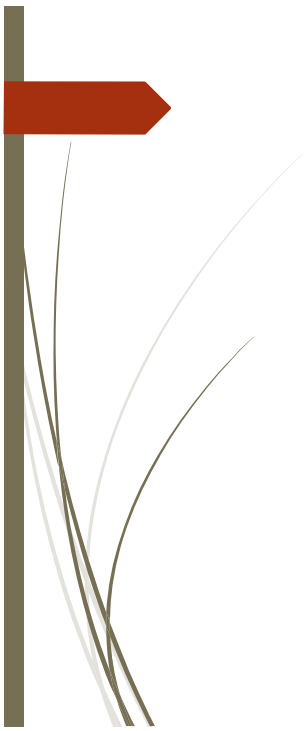
ІІІ Всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу»
(м. Київ, 2025 р.)

- Секція: «Штучний інтелект, машинне навчання у побуті і промисловості»
- Тема доповіді: «Штучний інтелект у виявленні шахрайських транзакцій у фінансових установах»
- Автор: Зарнішин Д. В.

X Всеукраїнська студентська наукова конференція «Експериментальні та теоретичні дослідження в контексті сучасної науки» (м. Дніпро, 2026 р.)

- Секція: «Системний аналіз, моделювання та оптимізація»
- Тези доповіді:
  Зарнішин Д. В. Штучний інтелект у виявленні шахрайських транзакцій у фінансових установах

Слайд 12



Дякую за увагу!

Слайд 13