

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
«КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ
ВАДИМА ГЕТЬМАНА»

навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
кафедра інформаційних систем в економіці

Освітньо-професійна програма

галузь знань	12 Інформаційні технології
спеціальність	122 Комп'ютерні науки
спеціалізація	Системи штучного інтелекту

форма навчання: денна

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПРОТИДІЇ ФІНАНСОВОМУ
ШАХРАЙСТВУ В ПЛАТІЖНИХ СЕРВІСАХ

здобувача _____ Проценка Богдана Петровича _____
(ПІБ, підпис)

Науковий керівник: д.е.н., професор, професор кафедри ІСЕ Мозгаллі О. П.
(науковий ступінь, учене звання, ПІБ)

Кваліфікаційна магістерська робота
допущена до захисту в
Екзаменаційній комісії з атестації
здобувачів вищої освіти

завідувач кафедри: к.е.н., доцент

_____ Тішков Б.О.

Київ 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА

«Системи штучного інтелекту»

галузь знань 12 Інформаційні технології
спеціальність 122 Комп'ютерні науки

ПОГОДЖЕНО:

Керівник проєктної групи
(гарант)
освітньо-професійної програми

_____ Васильєва Л.В.
« _____ » _____ 2024 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри інформаційних систем в
економіці

_____ Тішков Б.О.
« _____ » _____ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти Проценка Богдана Петровича
очної форми навчання

на підготовку кваліфікаційної магістерської роботи
на тему: **ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПРОТИДІЇ ФІНАНСОВОМУ**
ШАХРАЙСТВУ В ПЛАТІЖНИХ СЕРВІСАХ

Тему затверджено наказом ректора Університету від «07» березня 2025 р. №464-ст

Кваліфікаційна магістерська робота виконується на матеріалах наукових
публікацій та матеріалів з мережі інтернет

План кваліфікаційної магістерської роботи

Розділ I Теоретичний розділ. Дослідження та аналіз підходів до протидії
фінансовому шахрайству

Розділ II Аналітичний розділ. Характеристика проблеми та постановка задачі

Розділ III Конструктивний розділ. Розроблення проєктних рішень

Об'єкт дослідження: інформаційні процеси, що характеризують фінансові
операції та механізми виявлення шахрайських транзакцій у платіжних
системах

Предмет дослідження: методи виявлення та запобігання фінансовому шахрайству
та їх використання для створення безпечного середовища у платіжних
системах

Мета кваліфікаційної магістерської роботи: розробка інтелектуальної системи
моніторингу та протидії фінансовому шахрайству у платіжних сервісах

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:
У розділі I Дослідити основні загрози фінансового шахрайства та їхні
механізми. Проаналізувати сучасні підходи до виявлення підозрілих операцій.
Розробити концепцію, визначити функціонал, технології та інструменти, що
можуть бути використані для створення системи з їх використанням в

предметній області, що досліджується в кваліфікаційній магістерській роботі.

У розділі II На основі дослідження інформаційних процесів, що характеризують процеси моніторингу та аналізу транзакцій, визначити архітектуру майбутньої системи та її основні компоненти, обґрунтувати вибір та функції її основних компонентів. Сформулювати вимоги до системи, з урахуванням специфіки фінансових установ. Обґрунтувати вибір алгоритмів аналізу даних та технологій для розробки системи.

У розділі III Запроектувати архітектуру системи протидії шахрайству у платіжних сервісах із використанням алгоритмів машинного навчання. Обґрунтувати вибір моделей машинного навчання для аналізу транзакцій. Опис механізму оцінки ризиків фінансових операцій та автоматизованого блокування шахрайських транзакцій. Реалізація прототипу системи та перевірка її ефективності на тестових сценаріях.

Завдання підготував
науковий керівник



(підпис)

Мозгаллі Ольга Петрівна

(ініціали, прізвище)

«10» березня 2025 р.

Завдання одержав
здобувач



(підпис)

Проценко Богдан Петрович

(ініціали, прізвище)

«10» березня 2025 р.

РЕФЕРАТ

Кваліфікаційна магістерська робота містить 83 сторінок, 1 таблицю, 18 рисунків, список використаних джерел з 44 найменувань, 2 додатків.

«Система протидії фінансовому шахрайству в платіжних сервісах»

Об'єктом дослідження кваліфікаційної магістерської роботи є сфера фінансових технологій та платіжних сервісів.

Предметом дослідження є інтелектуальні системи моніторингу та аналізу транзакцій для протидії фінансовому шахрайству.

Мета і завдання дослідження. Основною метою кваліфікаційної магістерської роботи є вдосконалення процесу обробки та обміну інвойсами шляхом проектування інтелектуальної інформаційної системи.

Відповідно до поставленої мети визначені такі *завдання*:

- дослідити предметну область та виконати аналіз існуючих систем протидії фінансовому шахрайству;
- представити обґрунтування вибору підходів і технологій для створення системи виявлення підозрілих операцій;
- визначити структуру та представити характеристику інтелектуальної системи аналізу транзакцій;
- описати методи та алгоритми машинного навчання для аналізу фінансових операцій;
- спроектувати базу даних для захисту локальних потужностей системи виявлення шахрайства;
- реалізувати систему автоматичного виявлення шахрайських дій на основі аналізу транзакцій;
- розробити користувацький інтерфейс для моніторингу підозрілих операцій у платіжних сервісах;
- описати програмне та технічне забезпечення для розробленої системи;
- представити результати тестування та аналіз ефективності роботи системи на реальних або змодельованих даних.

Теоретична, методична та практична значущість отриманих результатів. У процесі дослідження було проаналізовано існуючі методи та технології виявлення фінансового шахрайства, обґрунтовано вибір підходів до розробки інтелектуальної системи моніторингу транзакцій, визначено її структуру та ключові алгоритми.

Рік виконання кваліфікаційної магістерської роботи – 2025.

Рік захисту роботи – 2025.

Ключові слова: фінансове шахрайство, платіжні сервіси, машинне навчання, аналіз транзакцій, виявлення аномалій, штучний інтелект, інформаційна безпека.

Summary

Qualification master's thesis contains 83 pages, 1 tables, 18 figures, a list of references from 44 titles, 2 applications.

« Financial fraud prevention system in payment services»

The object of study of the qualification master's work is the field of financial technologies and payment services.

The subject of the study is intelligent systems for monitoring and analyzing transactions to combat financial fraud.

Purpose and objectives of the study. The main purpose of the master's thesis is to improve the process of processing and exchanging invoices by designing an intelligent information system.

In accordance with this goal, the following *tasks* are defined:

- to research the subject area and analyze existing financial fraud prevention systems;
- to present the rationale for choosing approaches and technologies to create a system for detecting suspicious transactions;
- define the structure and present the characteristics of an intelligent transaction analysis system;
- describe machine learning methods and algorithms for analyzing financial transactions;
- design a database to protect the local capacity of the fraud detection system; implement a system for automatic fraud detection based on transaction analysis;
- to develop a user interface for monitoring suspicious transactions in payment services;
- describe the software and hardware for the developed system;
- to present the results of testing and analysis of the system's performance on real or simulated data.

Theoretical, methodological and practical significance of the results. In the course of the research, the existing methods and technologies for detecting financial fraud were analyzed, the choice of approaches to the development of an intelligent transaction monitoring system was substantiated, its structure and key algorithms were determined.

The year of completion of the master's thesis is 2025.

The year of defense is 2025.

Keywords: financial fraud, payment services, machine learning, transaction analysis, anomaly detection, artificial intelligence, information security.

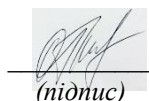
В і д г у к
про кваліфікаційну магістерську роботу
здобувача навчально-наукового інституту «Інститут інформаційних технологій в економіці»
освітньо-професійної програми «Системи штучного інтелекту»
Проценка Богдана Петровича

(прізвище, ім'я, по батькові)

на тему «Інтелектуальна система протидії фінансовому шахрайству в платіжних сервісах.»

- 1. Актуальність теми:** Тема дипломного проекту студента має високу актуальність, що зумовлено стрімким зростанням кількості онлайн-транзакцій у банківському та фінансовому секторах. В умовах цифровізації фінансових послуг проблема своєчасного виявлення шахрайських операцій стає пріоритетною для захисту інтересів користувачів і стабільності платіжних систем.
- 2. Позитивні риси кваліфікаційної магістерської роботи:** У роботі комплексно розглянуто технічні аспекти побудови системи виявлення шахрайства, проведено глибокий теоретичний аналіз предметної галузі, а також практично реалізовано функціональну модель з використанням методів інтелектуальних алгоритмів.
- 3. Наявність самостійних розробок автора:** Студентом самостійно реалізовано всі компоненти програмного забезпечення, включаючи навчання моделей класифікації, розробку веб-інтерфейсу, структуру бази даних MongoDB для авторизації користувачів. Значна частина технічних рішень — зокрема обробка CSV-даних, валідація введення, підключення моделі через API.
- 4. Цінність теоретичних висновків та практичних рекомендацій:** У ході виконання проекту обґрунтовано доцільність використання машинного навчання для задач виявлення шахрайства. Представлено аналіз наборів даних, описано методи попередньої обробки. Практичне значення розробки полягає в можливості адаптації створеної системи під умови банківських установ. Запропоноване рішення може бути доопрацьоване для впровадження у внутрішні фінансові системи з урахуванням політик безпеки.
- 5. Наявність недоліків:** У роботі доцільно було б розширити перелік протестованих моделей, включивши більш складні архітектури нейронних мереж. Крім того, не розглянуто порівняльну ефективність моделей з точки зору швидкодії.
- 6. Загальна оцінка кваліфікаційної магістерської роботи та її допущення до захисту перед ЕК:** Кваліфікаційна робота виконана на професійному рівні, у повній відповідності до вимог, з дотриманням логічної структури, технічної реалізації та наукової обґрунтованості. Робота заслуговує оцінки відмінно та підлягає допуску до захисту перед ЕК.

Науковий керівник _____ д.е.н., професор, професор кафедри ІСЕ Мозгаллі О. П.
(посада, учене звання, науковий ступінь)


(підпис)

Мозгаллі О. П.
(прізвище, ініціали)

« ____ » _____ 20__ р.

Рецензія
на кваліфікаційну магістерську роботу
здобувача вищої освіти
Проценка Богдана Петровича

(прізвище, ім'я, по батькові)

Тема: Інтелектуальна система протидії фінансовому шахрайству в платіжних сервісах.

Актуальність теми кваліфікаційної магістерської роботи і доцільність її розроблення: Сьогодні, в умовах щорічного збільшення об'єму фінансових операцій в електронній комерції, а також виявлення все нових методів фінансових злочинів, інструменти виявлення та запобігання шахрайства лише збільшують свою актуальність. Розробка системи виявлення фінансового шахрайства з використанням алгоритмів машинного навчання є своєчасною та важливою ініціативою, що відповідає сучасним викликам інформаційної безпеки в банківському секторі та сфері фінансових технологій.

Якість проведеного дослідження: Дослідження проведено якісно. Детально описана характеристика системи, наведена узагальнена методологія найпростіших систем виявлення шахрайства і запропонована власна вдосконалена модель. Автором систематизовано теоретичну базу щодо методів виявлення шахрайства, охарактеризовано використання алгоритмів машинного навчання у задачах класифікації транзакцій, приділено увагу проблемі дисбалансу даних.

Позитивні риси кваліфікаційної магістерської роботи: Проект вирізняється структурованістю, повнотою реалізації та практичною спрямованістю. Створено не лише класифікаційну модель, але й побудовано систему взаємодії з користувачем через веб-інтерфейс. Висвітлено питання безпеки даних, включаючи локальне збереження та авторизацію доступу до системи.

Зауваження: У проєкті переважно застосовано класичні алгоритми машинного навчання, без детального використання глибоких нейронних архітектур.

Практична значимість висновків і рекомендацій: Проект має високу прикладну цінність для впровадження у фінансових установах. Його локальне розгортання та ізоляція від зовнішніх систем дозволяють підвищити безпеку обробки даних. Результати дослідження та реалізації можуть бути використані для впровадження автономних, захищених рішень у банківських структурах з метою мінімізації ризиків фінансових втрат від шахрайства.

Маріупольський державний університет
к.тех.н., доцент, в.о. завідувача кафедри
системного аналізу та інформаційних технологій



М.П.

 Г.В. Мартинюк

підпис Мартинюк Г.В. засвідчує.
Насальник ВК  *Ольга Блогініна*
16.05.2025

ЗМІСТ

ВСТУП.....	3
I. ТЕОРЕТИЧНИЙ РОЗДІЛ. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ПРЕДМЕТНОЇ СШІ.....	5
1.1 Дослідження предметної області. Збір інформації та вивчення матеріалів з теми кваліфікаційної магістерської роботи	5
1.2 Аналіз існуючих СШІ й інтелектуальних систем предметної області	10
1.3 Постановка проблеми та формування задач	13
1.4 Обґрунтування вибору підходів і технологій для проектування та створення СШІ і їх компонентів.	15
РОЗДІЛ 2. АНАЛІТИЧНИЙ РОЗДІЛ. ХАРАКТЕРИСТИКА СШІ ТА ПОСТАНОВКА ЗАДАЧІ.....	19
2.1 Характеристика об'єкта дослідження	19
2.2 Структура і характеристика системи та її компонентів.	20
2.2.1 Алгоритм і техніка навчання	28
2.3 Методи, моделі і моделювання процесів і елементів складних систем	32
РОЗДІЛ 3. КОНСТРУКТИВНИЙ РОЗДІЛ. РОЗРОБЛЕННЯ ПРОЕКТНИХ РІШЕНЬ	40
3.1 Проектування бази знань для системи прийняття інтелектуальних рішень та управління	40
3.2 Розроблення користувацького інтерфейсу. Елементи та структура	51
3.3 Проектування забезпечувальних підсистем рекомендаційної системи та її реалізації.....	56
3.3.1 Інформаційне забезпечення.	56
3.3.2 Програмне забезпечення.	58
3.3.4 Організаційно-економічне забезпечення	60
ВИСНОВКИ	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	65
Додаток А	70
Додаток Б.....	76

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

AI – Artificial intelligence (Штучний інтелект)

AML – Anti-money laundering (Протидія відмиванню грошей)

API – Application programming interface (Інтерфейс прикладного програмування)

ATM – Automated teller machine (Банкомат)

BaaS – Banking as a service (Банківські послуги як сервіс)

BI – Business intelligence (Бізнес-аналітика)

CBDC – Central bank digital currency (Цифрова валюта центрального банку)

CCF – Credit card fraud (Шахрайство з кредитними картками)

CFT – Countering the financing of terrorism (Протидія фінансуванню тероризму)

CNP – Card-not-present (Транзакція без фізичної картки)

CVV – Card verification value (Код перевірки картки)

DDoS – Distributed denial of service (Розподілена атака на відмову в обслуговуванні)

E2EE – End-to-end encryption (Скрізне шифрування)

EFT – Electronic funds transfer (Електронний переказ коштів)

EMV – Europay, Mastercard, and Visa (Стандарт чіпованих карток)

FDS – Fraud detection system (Система виявлення шахрайства)

FF – Financial fraud (Фінансове шахрайство)

FRA – Fraud risk assessment (Оцінка ризику шахрайства)

FRTB – Fundamental review of the trading book (Фундаментальний огляд торгової книги)

IBAN – International bank account number (Міжнародний номер банківського рахунку)

IoT – Internet of things (Інтернет речей)

ISO – International Organization for Standardization (Міжнародна організація зі стандартизації)

KBA – Knowledge-based authentication (Автентифікація на основі знань)

KYC – Know your customer (Знай свого клієнта)

ML – Machine learning (Машинне навчання)

MFA – Multi-factor authentication (Багатофакторна автентифікація)

NACHA – National Automated Clearing House Association (Національна асоціація автоматизованих розрахункових палат)

NFC – Near field communication (Технологія безконтактної передачі даних)

NIST – National Institute of Standards and Technology (Національний інститут стандартів і технологій)

OTP – One-time password (Одноразовий пароль)

PCI DSS – Payment card industry data security standard (Стандарт безпеки даних індустрії платіжних карт)

PFM – Personal finance management (Управління персональними фінансами)

PII – Personally identifiable information (Персонально ідентифікована інформація)

PIN – Personal identification number (Персональний ідентифікаційний номер)

PSD2 – Payment Services Directive 2 (Директива ЄС про платіжні послуги 2)

PSP – Payment service provider (Постачальник платіжних послуг)

RBA – Risk-based approach (Підхід, заснований на оцінці ризиків)

SCA – Strong customer authentication (Посилена автентифікація клієнта)

SEPA – Single euro payments area (Єдина зона платежів у євро)

SIM swap – SIM swap fraud (Шахрайство із заміною SIM-карти)

SOC – Security operations center (Центр операційної безпеки)

SWIFT – Society for Worldwide Interbank Financial Telecommunication (Міжнародна міжбанківська система передачі фінансової інформації)

TFA – Two-factor authentication (Двофакторна автентифікація)

UPI – Unified Payments Interface (Єдиний платіжний інтерфейс)

ВСТУП

У 21-му столітті цифрова революція охопила всі сфери економіки та соціального життя, включаючи область фінансових послуг. Розповсюдження онлайн-платежів стало живим та розрахованим процесом під впливом як технологічних новацій, так і державних ініціатив та змін у споживчому поведінці.

Фундаментальні умови для впровадження онлайн-платежів було закладено ще на етапах розвитку електронною комерцією та поширення доступу до Інтернету в ранній стадії свого розвитку. Розробка безпечних платіжних систем і протоколів сприяла збільшенню довіри до електронних транзакцій в цей період часу. Головними методами безготівково оплати на цьому етапі стали кредитними та дебетовими картками, як і вершинна можливість для користувачів вчиняти покупки онлайн. Однак низький рівень розповсюдження Інтернету та обмежена цифрова грамотність призупиняли широке поширення цих інструментів.

Значну роль у розвитку онлайн-платежів відіграли державні ініціативи. Наприклад, в Індії була запущена кампанія “Цифрова Індія” з метою популяризації використання цифрових фінансових послуг серед населення краю. Дуже важливим етапом стала демонетизація у 2016 році, яка сприяла униканню готівкових розрахунків та прискорила перехід до цифрових транзакцій [1].

У 2016 році відбулася важлива подія - запровадження платформи Unified Payments Interface (UPI), яка дозволяла миттю пересилати кошти між банківськими рахунками за допомогою лише віртуального платіжного ідентифікатора або мобільного номера. На основі UPI з'явилися зручні мобільні додатки, наприклад Google Pay, PhonePe, Paytm та державна система BHIM. Спрощуючи доступ до електронних платежів для широкого кола користувачів [2].

Пандемія COVID-19 стала потужним чинником прискорення переходу до цифрових форм оплати. Умови карантину, необхідність дотримання соціального дистанціювання, а також зростання онлайн-покупок сприяли значному зростанню

попиту на онлайн-оплату. У цей період також активно розвивались цифрові банківські послуги [3].

Серед чинників, що впливають на поширення онлайн-платежів, варто виділити зручність, швидкість, безпеку транзакцій, довіру до платформи, а також соціальний вплив та рівень фінансової грамотності. Звичка до використання цифрових методів оплати формується внаслідок постійного використання, особливо в оточенні, де ці методи є нормою.

Попри динамічне зростання, впровадження онлайн-платежів супроводжується низкою викликів. Серед них – кіберзагрози, ризики шахрайства, недостатній захист персональних даних користувачів, а також цифрова нерівність, яка ускладнює доступ до електронних послуг у сільських і віддалених регіонах.

Майбутнє онлайн-платежів тісно пов'язане з новітніми технологіями, зокрема блокчейном, штучним інтелектом, біометричними системами та розвитком Інтернету речей. Інтеграція платіжних сервісів з платформами електронної комерції, соціальними мережами та іншими цифровими середовищами відкриває нові можливості для зростання.

Популяризація онлайн-платежів є результатом комплексного впливу інновацій, державної підтримки, соціальних трансформацій та глобальних викликів, таких як пандемія. Незважаючи на наявні труднощі, перспективи розвитку цифрової фінансової інфраструктури залишаються надзвичайно широкими. Подальша еволюція та адаптація платіжних систем мають забезпечити інклюзивний, безпечний і ефективний фінансовий простір майбутнього.

I. ТЕОРЕТИЧНИЙ РОЗДІЛ. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ПРЕДМЕТНОЇ СШІ

1.1 Дослідження предметної області. Збір інформації та вивчення матеріалів з теми кваліфікаційної магістерської роботи

Онлайн платежі – електронне переведення коштів через Інтернет. Онлайн платежі з'явилися як альтернатива традиційному розрахунку готівкою, картою або чеком з необхідністю обміну товаром, послугою чи грошима між покупцем та продавцем. Їх поява зумовлена розвитком технологій та впровадженням перших платіжних систем у банківській сфері задля миттєвих транзакцій [4].

З точки зору цілей, онлайн-платежі спрямовані на полегшення електронної комерції, підвищення швидкості транзакцій, зменшення витрат і покращення фінансової доступності, окрім того, цифрові платежі впливають на моделі витрат, що пов'язано з їхньою метою у стимулюванні економічної діяльності. Швидка циркуляція грошової маси в економіці трансформує модель споживчих витрат, впливаючи на економічну активність через прискорення грошової логістики. У сучасній економіці швидкість обігу грошей є чинником розвитку: оперативне переміщення коштів між економічними агентами стимулює функціонування економічної системи.

У контексті фінансової логістики, цифрові платежі оптимізують фінансові потоки, зменшуючи затримки в розрахунках та покращуючи управління ланцюгами постачання. Що є особливо важливо для малого та середнього бізнесу, де швидкість обігу коштів безпосередньо впливає на оборотний капітал та здатність до масштабування [22].

Дослідження Банку міжнародних розрахунків (BIS) показало, що збільшення використання цифрових платежів на 1% асоціюється з підвищенням темпів зростання ВВП на душу населення на 0,10% протягом двох років, а також зі зменшенням частки неформальної зайнятості на 0,06%. Ці дані свідчать про те, що

цифрові платежі не лише сприяють економічному зростанню, але й підтримують формалізацію економіки [5].

Згідно з дослідженням Moody's Analytics, зростання використання цифрових платежів сприяло збільшенню світового ВВП на \$245 мільярдів у період з 2015 по 2019 роки. Зокрема, кожне зростання на 1% у використанні платіжних карток призводило до додаткового споживання товарів і послуг на \$65–70 мільярдів щорічно, що створює позитивний цикл: підвищене споживання стимулює виробництво, зайнятість, доходи та подальше зростання ВВП [6].

Цифрові платежі виступають не тільки як зручний інструмент для споживачів, їх впровадження є універсальним для усіх типів розрахунків стимулюючи грошову логістику та споживчі витрати. Розглянемо основні типи онлайн розрахунків та їх призначення:

1) Карткові платежі (Card-Based Payments) - забезпечення швидкого та надійного способу оплати через банківські інститути. Кредитні та дебетові картки з вбудованими чипами або віртуальними номерами створені для підвищення безпеки транзакцій і запобігання шахрайству. Контактні платежі через NFC (Near Field Communication) забезпечують зручність завдяки безконтактному обміну даними, що пришвидшує процес оплати без необхідності фізичного введення даних картки [23].

2) Мобільні гаманці (Mobile Wallets) - виконання ролі електронного посередника між споживачем та платіжною інфраструктурою, дозволяючи зберігати банківські реквізити на мобільному пристрої. Сервіси на кшталт PayPal, Google Pay та Apple Pay спрощують транзакції через функцію "один клік", знижуючи час і зусилля, необхідні для оплати. Додатково, системи QR-кодів, дозволяють ініціювати платежі без необхідності фізичної карти або банкомата, що особливо ефективно в умовах високої чисельності населення та потреби в фінансовій інклюзії [24].

3) Банківські перекази (Bank Transfers) - онлайн-банкінг або мобільні додатки: monobank (АТ «Універсал Банк»), Ощад24, ПриватБанк та інші, призначені для прямого переміщення коштів між банківськими рахунками без

участі третьої сторони. Реалізуюючи високу швидкість обробки транзакцій, зниження витрат на обслуговування та зручність у B2B- і P2P-сегментах. У таких системах особлива увага приділяється дотриманню регіональних регуляторних норм та захисту даних користувачів [25].

4) Криптовалюти (Cryptocurrencies) - Bitcoin, Ethereum, Tether, USD Coin (USDC) функціонують як децентралізовані платіжні системи, що забезпечують безпосередні розрахунки між користувачами без участі фінансових посередників. Їхнє призначення полягає у зниженні транзакційних витрат [26].

5) Електронні гаманці для електронної комерції (E-Wallets for E-Commerce) – на кшталт Alipay та Amazon Pay, інтегровані безпосередньо з платформами електронної комерції, виконують функцію автоматизованої обробки платежів між покупцями й продавцями. Їх основне призначення — забезпечити безперервний, безпечний і масштабований платіжний досвід у межах онлайн-торгівлі, що особливо важливо в умовах глобалізованого ринку [27].

Теза про те, що онлайн платежі поступово витісняють класичні фізичні розрахунки картками та готівкою все частіше присутня в контексті банківської сфери та комерційної діяльності як базис. Те, що раніше було маркетинговою “фішкою” окремих банків та сервісів, сьогодні вже є традиційною послугою, яку не відокремлюють та не наголошують увагу.

Процес популяризації онлайн-платежів можна розглядати крізь хронологічну призму, розділяючись ключовими технологічними досягненнями, соціально-економічними факторами та конкретними подіями, такими як пандемія COVID-19.

Хронологія розвитку онлайн платежів:

- 1967: Запущено перший банкомат, що дозволив знімати готівку без участі касира.
- 1970-ті: Кредитні картки з магнітною стрічкою стали стандартом для фінансових операцій.
- 1994: Здійснено першу онлайн-покупку через NetMarket — компакт-диск з музикою Стінга, що ознаменувало початок електронної комерції.

- 1998: Запущено PayPal — новаторську платформу для онлайн-платежів, яка змінила цифрову торгівлю.
- 2000-ні: Мобільні та безконтактні технології.
- 2003: Представлено безконтактні кредитні картки з використанням радіочастотної ідентифікації (RFID).
- 2007: У Кенії запущено мобільну платіжну систему M-Pesa, що здійснила революцію у фінансовій доступності в країнах, що розвиваються [28].
- 2009: Представлено Bitcoin, що започаткував використання блокчейну як децентралізованої платіжної технології.
- 2014: Запуск Apple Pay, який популяризував мобільні гаманці на основі технології NFC.
- 2016: В Індії запущено систему UPI (Unified Payments Interface), що дозволила здійснювати платежі через QR-коди між різними банками [29].
- 2020–2022: Пандемія COVID-19 спричинила глобальне зростання цифрових платежів на 30% через гігієнічні та безпекові міркування.
- 2023: У Китаї (e-CNY) та Індії (e-рупія) розпочато пілотні проєкти цифрових валют центральних банків (CBDC), що стало кроком до державної цифрової фінансової системи

Серед усіх хронологічних періодів, ключовими можна вважати: період цифрової революції 1990-х років, бум фінтеху у 2010-х роках, а також фазу пандемічного прискорення у 2020–2022 роках.

Період 1990-х років, який позначився як "цифрова революція", є першим ключовим етапом. У цей час були закладені технологічні та інституційні основи сучасної електронної комерції. Впровадження першої онлайн-транзакції у 1994 році символізувало перехід до глобалізованої моделі споживання, а запуск платформи PayPal у 1998 році створив механізм безпечних онлайн-платежів без участі традиційних банківських інструментів. Цей період започаткував зміну парадигми у способі взаємодії споживача з фінансовими сервісами та сформував довіру до цифрових фінансових технологій.

Другим вагомим етапом у розвитку був бум фінансових технологій у 2010-х роках. Саме в цей період відбувся технологічний прорив, який суттєво трансформував швидкість, доступність і масштаб цифрових платежів. Важливу роль тут відіграло впровадження криптовалют (зокрема Bitcoin у 2009 році), а також мобільних платіжних систем на основі NFC — таких як Apple Pay у 2014 році. Найбільш революційним нововведенням стала система UPI в Індії (2016) [7], яка забезпечила універсальний, миттєвий та міжбанківський платіжний інтерфейс.

Третій період — роки пандемії COVID-19 (2020–2022), які спричинили безпрецедентне прискорення цифровізації фінансових процесів. Через обмеження фізичних контактів, уряди, бізнес і споживачі масово перейшли на безготівкові методи оплати. Згідно з емпіричними даними, в цей період відбулося зростання цифрових платежів на понад 30% у глобальному масштабі. Пандемія виступила каталізатором незворотної зміни поведінкових моделей споживачів та стала умовою для прискореного переходу до діджитал-економіки.

Так як глобальну пандемію COVID-19 можна вважати одним із найвагоміших чинників розвитку електронної комерції в останні роки, доречно було б розглянути вплив пандемії на світовий банкінг більш детально у кількісному вираженні на основі емпіричних досліджень і офіційних даних:

- У 2020 році загальні обсяги глобальної електронної комерції досягли \$26,7 трлн, що включає як B2B, так і B2C сегменти. Становлячи приблизно 30% світового ВВП того року [8].
- Частка онлайн-продажів у загальному обсязі роздрібної торгівлі зросла з 16% у 2019 році до 19% у 2020 році. У деяких країнах спостерігалось ще більше зростання: у Республіці Корея — з 20,8% до 25,9%, у Великій Британії — з 15,8% до 23,3%, у Китаї — з 20,7% до 24,9%, у США — з 11% до 14% [9].
- У другому кварталі 2020 року роздрібні онлайн-продажі в США досягли \$211,5 млрд, що на 31,8% більше порівняно з першим кварталом і на 44,5% більше порівняно з аналогічним періодом попереднього року. Частка електронної комерції в загальному обсязі роздрібних продажів зросла з 11,8% у першому кварталі до 16,1% у другому кварталі 2020 року [10].

Узагальнюючи інформацію з різних джерел, вплив глобальної пандемії COVID-19 призвів до зростання онлайн комерції приблизно на 30% у світі [16,17].

1.2 Аналіз існуючих СШІ й інтелектуальних систем предметної області

Інтеграція систем штучного інтелекту, інтелектуальних алгоритмів та машинного навчання як функцію безпеки у онлайн-комерції зробила революцію у виявленні шахрайства, захисті транзакцій та автентифікації користувачів.

Штучний інтелект дозволяє обробляти великі обсяги транзакційних даних у реальному часі, виявляючи аномалії та підозрілі патерни, які можуть свідчити про шахрайські дії. На відміну від традиційних систем, що базуються на фіксованих правилах, інтелектуальні системи здатні адаптуватися до нових типів шахрайства, використовуючи машинне навчання на нових даних. Забезпечується більш точне виявлення шахрайських транзакцій та зменшення кількості хибнопозитивних спрацювань.

Використання машинного навчання для оцінки ризиків транзакцій, присвоюючи кожній з них ризиковий бал на основі історичних даних та поведінкових патернів користувачів. Системи здатні виявляти та блокувати підозрілі транзакції до їх завершення, мінімізуючи фінансові втрати. Підвищення безпеки автентифікації користувачів застосовуючи біометричні дані, розпізнавання обличчя або відбитків пальців, а також аналіз поведінкових характеристик користувачів, таких як швидкість набору тексту, підтвердження дій та інші дані, які штучний інтелект може проаналізувати на основі минулих дій та параметрів дозволяє системам виявляти спроби несанкціонованого доступу навіть у випадках, коли зловмисник володіє обліковими даними користувача [30].

Серед найпопулярніших інтелектуальних систем захисту в онлайн-транзакціях можна виділити наступні:

- PayPal Fraud Detection AI

Система виявлення шахрайства від PayPal використовує алгоритми машинного навчання, що навчається на мільйонах щоденних транзакцій, система PayPal робить акцент на аналізі поведінкових шаблонів у реальному часі. Його перевага полягає в детальному виявленні аномалій на рівні транзакцій, позначенні порушень, таких як раптові перекази великої вартості або географічні невідповідності. Платформа інтегрується з платіжною екосистемою замкнутого циклу, забезпечуючи швидку перевірку ідентифікації користувача та історії покупок. Однак його залежність від історичних даних може відставати від нових векторів шахрайства, таких як атаки на синтетичну ідентифікацію (тип фінансового шахрайства, при якому зловмисники створюють нову фіктивну особу, комбінуючи справжні дані з вигаданими особистими даними).

Однією з провідних інновацій PayPal у сфері безпеки є впровадження графових методів виявлення шахрайства, таких як eFraudCom, які ефективно ідентифікують кооперативні шахрайські схеми. Ці алгоритми дозволяють виявляти пов'язані аномальні транзакції, які традиційні лінійні методи не розпізнають. Додатково, двофакторна автентифікація з використанням біометричних даних, зокрема розпізнавання обличчя, значно знижує ризики захоплення облікового запису, що є критично важливим в епоху зростаючої кількості соціальних інженерних атак.

Однак, попри високий ступінь технологічного оснащення, PayPal все ще значною мірою покладається на історичні дані для навчання своїх моделей, що обмежує здатність системи адаптуватися до нових типів шахрайства, особливо так званих "zero-day" атак (використання невідомих вразливостей в програмному забезпеченні, про які ще не знають навіть розробники і не існує оновлень для їх усунення. Зловмисники використовують ці "нульові дні" для отримання доступу до систем, крадіжки даних або встановлення шкідливого програмного забезпечення). Більше того, система потребує ручного втручання експертів у деяких випадках, що уповільнює оперативність реагування на нові або складні загрози, підвищуючи ризики втрат при атаках високої швидкості (HUANG et al., 2020).

- Visa Advanced Authorization (VAA)

Адаптивна архітектура штучного інтелекту Visa відмінно підходить для міжінституційної кореляції шахрайства, аналізуючи понад 500 атрибутів транзакцій із затримкою <1 мс. Система ARIC визначає законну поведінку клієнтів шляхом постійного навчання, особливо ефективна проти розподілених атак, таких як шахрайство тріангуляції. У 2023 році це запобігло збиткам на суму 40 мільярдів доларів США завдяки виявленню складних моделей шахрайства з кількома продавцями. Обмеження виникає в децентралізованих мережах, де фрагментація даних знижує точність моделі.

Visa VAA вирізняється впровадженням адаптивних штучно-інтелектуальних моделей, які постійно оновлюють профілі користувачів у реальному часі. Така динамічна побудова поведінкових патернів дозволяє істотно зменшити кількість хибнонегативних спрацьовувань і своєчасно виявляти нетипові операції. Інтеграція з технологією блокчейн також зміцнює систему, забезпечуючи незмінність аудиторських слідів, що критично важливо для регуляторного контролю та прозорості.

Разом з тим, одним із ключових обмежень Visa VAA є високі витрати на підтримку глобальної інфраструктури консенсусу, необхідної для децентралізованої перевірки даних, що робить систему менш ефективною у масштабуванні. Окрім цього, система має обмежену здатність до виявлення шахрайства між різними каналами (наприклад, між онлайн і офлайн транзакціями), що залишає певні вектори атаки не покритими.

- JPMorgan Chase's AI Fraud Detection

Зосереджуючись на гібридних моделях глибокого навчання, JPMorgan об'єднує неструктуровані дані (наприклад, взаємодію з клієнтами) із сигналами транзакцій. Його прогностична аналітика запобігає спробам захоплення облікового запису, виявляючи тонкі зміни поведінки до шахрайства. Система демонструє високу продуктивність у синтетичному виявленні ідентифікаційної інформації, але стикається з проблемами, що масштабуються в глобальному нормативному середовищі.

Система виявлення шахрайства JPMorgan використовує гібридні моделі, зокрема комбінацію автокодерів і XGBoost, які особливо ефективні при роботі з розбалансованими наборами даних, що є типовими для фінансових транзакцій. Завдяки впровадженню адверсаріального навчання, JPMorgan здатна протидіяти атакам з отруєнням даних, що підвищує стійкість моделі до спотвореної або навмисно модифікованої інформації.

Попри високий рівень технічної реалізації, JPMorgan дещо відстає у впровадженні графових підходів до виявлення шахрайських мереж, які продемонстрували ефективність у конкурентів, зокрема PayPal. Крім того, широке застосування поведінкового аналізу викликає побоювання щодо конфіденційності, оскільки збір і обробка детальних даних про користувача може порушувати принципи, особливо в юрисдикціях із жорстким регулюванням на зразок GDPR [31].

1.3 Постановка проблеми та формування задач

Зростаюча інтеграція фінансових операцій в онлайн-просторі у цифровому суспільстві зумовлює зростання електронної комерції та платіжних сервісів. Пандемія COVID-19, яка почалася у 2020 році, стала глобальним каталізатором цифровізації фінансової інфраструктури, спричинивши рекордне зростання частки онлайн-транзакцій та електронних платіжних систем у загальному обсязі економічної діяльності. Однак поряд із цим явищем спостерігається і паралельне зростання фінансового шахрайства, яке використовує нові вектори атак, зокрема соціальну інженерію, фальсифікацію особистостей та алгоритмічні маніпуляції.

Постановка проблеми створення і використання інтелектуальних інструментів у протидії фінансовому шахрайству полягає у тому, що виявлення фінансового шахрайства за допомогою інтелектуальних технологій наразі працює в рамках передового досвіду та дослідницьких методологій, що швидко

розвиваються, а не універсально сертифікованих стандартів до яких можна було б звернутися та побудувати захисні системи ґрунтуючись на них.

Кожна методика інтелектуальних інструментів має свої переваги та недоліки, розробляючи подібну систему, треба орієнтуватися на поставлені цілі та проблеми, які необхідно вирішити.

На основі відомих проблем систем захисту можна сформулювати необхідні для виконання задачі:

Перше завдання полягає в тому, щоб забезпечити систематичне збирання та попередню обробку даних із різних джерел, зокрема банківських баз даних, систем електронної комерції, транзакційних журналів, мережевої активності користувачів, а також відкритих чи захищених наборів даних. Ця задача включає у себе розв'язання проблем сумісності, структурної неоднорідності та приватності даних, а також забезпечення високої якості вхідної інформації для подальшого аналізу.

Друге завдання - розробка інтелектуальних алгоритмів, здатних ефективно аналізувати великі обсяги гетерогенних даних, виявляючи приховані закономірності, аномальні транзакційні шаблони, поведінкові відхилення та потенційні вектори фінансового шахрайства. У центрі задачі — створення моделей машинного навчання або гібридних підходів із інтелектуальними алгоритмами. Необхідно формалізувати критерії аномальності та шахрайських дій у змінному контексті, з урахуванням еволюції кіберзагроз.

Третє завдання - інтеграція розробленої системи в існуючі архітектури фінансових платформ або електронної комерції, з урахуванням питань сумісності з внутрішніми протоколами, правовими обмеженнями та нормами кібербезпеки. Завдання вимагає уніфікації інтерфейсів, забезпечення безперервного обміну даними між модулями в режимі реального часу, а також уникнення конфліктів між новими алгоритмами безпеки та вже діючими системами управління ризиками.

Четверте завдання - проведення повномасштабного тестування та валідації ефективності впроваджених рішень у реальному або наближеному до реального середовищі. Особливу увагу слід приділити точності виявлення шахрайських

транзакцій, показникам хибнопозитивних та хибнонегативних рішень, а також стійкості системи до спроб обману або обходу алгоритмів.

П'яте завдання - впровадження механізмів постійної експлуатації та технічного обслуговування системи протидії шахрайству, що включає моніторинг продуктивності, оновлення моделей машинного навчання, усунення виявлених помилок та адаптацію до змін у поведінці користувачів і шахрайських схем. Важливо також передбачити механізми автоматичного самонавчання та повторного тренування моделей з урахуванням накопичених нових даних.

Шосте завдання - розв'язання фундаментальної проблеми співвідношення між автономним прийняттям рішень системою та контролем з боку людини. Постає необхідність розмежування повноважень між автоматизованими модулями та операторами, зокрема у випадках критичних інцидентів або спірних транзакцій. Завдання включає формування чітких правил ескалації, визначення меж відповідальності та впровадження механізмів інтерпретованості рішень штучного інтелекту для забезпечення прозорості й довіри з боку користувачів та регуляторів.

1.4 Обґрунтування вибору підходів і технологій для проектування та створення СШ і їх компонентів.

В проектуванні використані наступні технології:

1) Фреймворки веб-розробки:

- Django (Python): високорівневий веб-фреймворк, який забезпечує швидку розробку та чисту архітектуру. Його інтеграція в проєкт дозволяє ефективно реалізувати адміністративні панелі управління, обробку запитів і безпечну аутентифікацію.

- Laravel (PHP): використовується для побудови надійних бекенд-компонентів системи, забезпечуючи масштабованість, підтримку RESTful API, високий рівень безпеки та зручну взаємодію з реляційними базами даних.

2) Інтерфейс користувача:

- Bootstrap (HTML, CSS, JavaScript): фреймворк для створення адаптивного та кросплатформного веб-інтерфейсу користувача, що дозволяє швидко впроваджувати дизайн із використанням готових стилів, компонентів і сіткової системи.

3) Мови програмування:

- Python: використовується як основна мова програмування для реалізації моделей машинного навчання, а також для серверної логіки системи.

- PHP: застосовується в частині побудови логіки взаємодії з базами даних та авторизаційних модулів (через Laravel).

- JavaScript: задіяна для реалізації інтерактивного фронтенду та зв'язку з сервером через AJAX/Fetch.

4) Алгоритми машинного навчання та аналітики:

- Логістична регресія: базовий класифікаційний алгоритм, який дозволяє розмежовувати шахрайські та звичайні транзакції за наявністю бінарної цільової змінної.

- Random Forest: ансамблевий метод навчання, ефективний при роботі з великими обсягами фінансових даних, здатний обробляти високовимірні простори.

- Gradient Boosting: метод послідовного навчання моделей, що дозволяє точно моделювати складні залежності, часто застосовується для виявлення прихованих шаблонів шахрайської активності.

- К-середніх (k-means): алгоритм кластеризації, що дозволяє виявляти нетипові транзакції без попередньої розмітки.

- DBSCAN: метод щільнісної кластеризації, який виявляє аномальні точки, що не належать до жодного з виявлених кластерів, ефективний у разі нерівномірного розподілу даних.

- Нейронні мережі: здатні моделювати складні взаємозв'язки між параметрами транзакцій, мають високу точність при достатньому обсязі тренувальних даних.

5) Системи управління даними:

- SQL: застосовується для побудови запитів до баз даних транзакцій, здійснення агрегацій, пошуку аномалій та реалізації збережених процедур у бекенді.

6) Підходи до виявлення шахрайства:

- На основі правил (rule-based): дозволяє виявляти стандартні порушення шляхом накладання логічних обмежень і порогових значень.

- Моделі на основі навчання: машинне навчання дозволяє автоматично виявляти нові шаблони, які не можна формалізувати за допомогою ручних правил.

7) Архітектурні підходи:

- Модульна архітектура: реалізована через окремі компоненти користувача, адміністратора та аналітичного модуля.

- Клієнт-серверна взаємодія: із використанням REST API для обміну даними між фронтендом і бекендом.

Методологія проектування базується на використанні об'єктно-орієнтованого підходу та модульної архітектури, що дозволяє гнучко масштабувати рішення, змінювати компоненти без порушення загальної логіки системи та забезпечує її супровід у реальному середовищі експлуатації.

Висновки до розділу 1

У першому розділі було проведено комплексне дослідження предметної області, аналіз підходів до створення інтелектуальної системи протидії фінансовому шахрайству. Дослідження предметної області, а саме електронної комерції, розпочався зі збору та аналізу інформації з теми кваліфікаційної роботи задля сформування цілісного уявлення про проблематику сучасної галузі, потреби і потенціал ринку та основні технології, які використовуються.

У сучасній сфері протидії фінансовому шахрайству не існує універсальних або еталонних технологічних рішень. Провідні світові фінансові компанії, зокрема PayPal, Visa та JPMorgan Chase, демонструють, що кожна організація створює унікальні підходи до виявлення та запобігання шахрайським транзакціям,

зважаючи на власну архітектуру, специфіку обслуговування клієнтів і масштаби операцій. Застосовувані інструменти включають машинне навчання, графові методи, гібридні моделі, блокчейн та біометричні системи автентифікації. Проте навіть найрозвиненіші технології мають свої обмеження. Вказуючи на те, що поточні рішення залишаються вразливими до нових, динамічно змінюваних форм шахрайства.

Основна мотивація розробки нових підходів у даній галузі полягає у постійному зростанні складності шахрайських схем та еволюції цифрового середовища, що супроводжується зростанням кількості онлайн-транзакцій, особливо після пандемії COVID-19. У міру того як шахрайство стає дедалі більш технологічно підготовленим і технологічно досконалим, виникає потреба у створенні інтелектуальних систем, здатних адаптуватися до невизначених загроз і забезпечувати не лише точність виявлення, а й оперативність реагування. У цьому контексті розробка нових моделей протидії шахрайству не є лише бажаною інновацією, а необхідною умовою забезпечення фінансової стабільності та довіри користувачів до цифрових сервісів.

РОЗДІЛ 2. АНАЛІТИЧНИЙ РОЗДІЛ. ХАРАКТЕРИСТИКА СШІ ТА ПОСТАНОВКА ЗАДАЧІ

2.1 Характеристика об'єкта дослідження

Об'єкт дослідження полягає у створенні та оптимізації системи виявлення шахрайства у платіжних системах. Ця система буде використовуватися для аналізу транзакційних даних користувачів задля виявлення підозрілої та злочинної діяльності як самих власників карток або рахунків, так і сторонніх осіб, які незаконно отримали доступ до конфіденційних даних з метою фінансового шахрайства.

Вона буде базуватися на аналізі великої кількості банківських даних, фінансової історії, поведінкових моделях користувачів. Використовуючи алгоритми машинного навчання та інтелектуальних алгоритмів, система буде виявляти підозрілі дії задля протидії фінансової злочинності аналізуючи вже існуючі дані фінансового обороту суб'єктів контролю.

Ключовими аспектами дослідження будуть розробка ефективних алгоритмів аналізу дій суб'єктів у платіжній системі, таких як фінансова та кредитна історія, історія транзакцій, поведінкові патерни поведінки суб'єктів.

Основні характеристики системи:

Використання інтелектуальних алгоритмів для аналізу даних і виявлення відхилень, які будуть класифікуватися як шахрайські дії або потенційно шахрайські дії.

Наявність бази знань, сам набір даних який буде аналізуватися інтелектуальними алгоритмами для виявлення шахрайства. Сама база знань не обов'язково може бути виражена у програмному вигляді, бо використання традиційної програмної реалізації бази даних (наприклад, у вигляді реляційної СУБД) не завжди є оптимальним варіантом. Причиною цього є специфіка алгоритмів виявлення шахрайства, які часто орієнтовані не на типову обробку

структурованих записів, а на аналіз великих обсягів неструктурованого або слабо структурованого тексту. Такими даними можуть бути звіти, журнали транзакцій, службові записки, електронні листи або інші текстові документи, що містять контекстуальні ознаки підозрілої активності. Як висновок, зберігання даних у текстовому вигляді поза межами класичної БД може полегшити інтеграцію з інструментами машинного навчання, які часто очікують на вхід саме попередньо оброблені масиви тексту або дані у форматі, що не відповідає структурі таблиць

Доступ до особових даних користувачів та їх фінансової історії, такий доступ дозволяє системі отримувати повний профіль клієнта, включаючи транзакційну активність, шаблони витрат, історію користування фінансовими послугами та інші релевантні параметри. Наявність персоналізованих даних суттєво підвищує точність аналітичних моделей, які використовуються для виявлення нетипової поведінки або підозрілих змін у фінансовій активності.

Інтеграція системи з банківськими та небанківськими каналами грошового обороту, значна частина шахрайських операцій здійснюється з використанням різних фінансових платформ, включаючи електронні гаманці, криптовалютні сервіси, системи швидких платежів тощо. Тому забезпечення інтероперабельності з різними джерелами даних дозволяє не лише розширити охоплення аналітики, а й виявляти складні, багаторівневі шахрайські схеми, що виходять за межі окремої фінансової установи.

Можливості для масштабування задля обробки великих потоків фінансових даних.

2.2 Структура і характеристика системи та її компонентів.

Система виявлення шахрайства являє собою сукупність інструментів, технологій та процедур, що спрямовані на виявлення і запобігання шахрайським діям у різних прикладних сферах. До таких сфер належать фінансові операції,

онлайн-сервіси, електронна комерція, сфера охорони здоров'я та інші контексти, де можливе використання цифрових або фізичних ресурсів із шахрайською метою.

Основним завданням системи є мінімізація фінансових збитків, недопущення витоку конфіденційної інформації та забезпечення цілісності інформаційних і операційних процесів у межах цільової інфраструктури.

Частиною системи є база даних фінансової історії, яка містить інформацію користувачів. У реалізації системи виявлення фінансового шахрайства в межах дослідницького проєкту використовується не програмна база даних, а текстове представлення фінансових даних. Такий підхід зумовлений особливостями обробки вхідних даних, які мають вторинне походження та представлені у стандартизованому текстовому форматі. Джерелом даних є відкритий набір, розміщений на платформі Kaggle під назвою «Credit Card Fraud Detection», що містить анонімізовані транзакції, класифіковані як шахрайські або справжні [11, 32].

Credit Card Fraud Detection
▲ 12151
<> Code

[Data Card](#)
[Code \(5325\)](#)
[Discussion \(109\)](#)
[Suggestions \(0\)](#)

creditcard.csv (150.83 MB)							↓ [] >
Detail Compact Column							10 of 31 columns ▾
# Time	# V1	# V2	# V3	# V4	# V5	# V6	
0	-1.3598071336738	-0.0727811733098497	2.53634673796914	1.37815522427443	-0.338320769942518	0.462387777762292	
0	1.19185711131486	0.26615071205963	0.16648011335321	0.448154078460911	0.0600176492822243	-0.0823608088155687	
1	-1.35835406159823	-1.34016307473609	1.77320934263119	0.379779593034328	-0.503198133318193	1.80049938079263	
1	-0.966271711572087	-0.185226008082898	1.79299333957872	-0.863291275036453	-0.0103088796030823	1.24720316752486	
2	-1.15823309349523	0.877736754848451	1.548717846511	0.403033933955121	-0.407193377311653	0.0959214624684256	
2	-0.425965884412454	0.960523044882985	1.14110934232219	-0.168252079760302	0.42098688077219	-0.0297275516639742	
4	1.22965763450793	0.141003507049326	0.0453707735899449	1.20261273673594	0.191880988597645	0.272708122899098	

Рисунок 2.1 – Зразок набору даних кредитних карток

Джерело: загальнодоступний датасет з сайту *kaggle.com*

Набір даних був зібраний у межах співпраці між компанією Worldline та дослідницькою групою машинного навчання Вільного університету Брюсселя. Він охоплює транзакції європейських власників кредитних карток, здійснені у вересні 2013 року протягом двох днів. Загальна кількість транзакцій становить 284 807, з яких 492 випадки класифікуються як шахрайство. Таким чином, дані є суттєво незбалансованими, оскільки позитивний клас становить лише 0,172% від загальної кількості записів. Структура набору даних включає виключно числові вхідні змінні, отримані в результаті перетворення методом головних компонент (PCA), що унеможлиблює використання первинних атрибутів через обмеження конфіденційності. Єдиними змінними, що не були перетворені, є «Time» (час, що минув з моменту першої транзакції) та «Amount» (сума транзакції), а змінна «Class» виступає як мітка цільового класу (1 — шахрайство, 0 — справжня транзакція). Враховуючи анонімізований характер даних, відсутність структури, характерної для класичних програмних баз даних, та необхідність попередньої обробки у форматі, придатному для алгоритмів машинного навчання, текстове представлення даних у вигляді CSV-файлів є доцільним і ефективним. Додатково, система була доповнена іншими текстовими наборами даних, сформованими на основі функціональності звітування, що дозволяє демонструвати роботу системи у ширшому контексті використання.

Однією з основних проблем використання загальнодоступних наборів даних для задач виявлення фінансового шахрайства є суттєвий дисбаланс класів. У таких наборах кількість шахрайських транзакцій є значно меншою порівняно з кількістю легітимних операцій. Цей дисбаланс негативно впливає на ефективність алгоритмів машинного навчання, оскільки моделі схильні орієнтуватися на більшість ігноруючи рідкісні, але критично важливі випадки шахрайства [12].

Для компенсації дисбалансу використовуються спеціальні техніки, такі як надсемплінг, андерсемплінг та навчання з урахуванням вартості помилок. Наприклад, метод SMOTE (Synthetic Minority Over-sampling Technique) дозволяє генерувати синтетичні записи меншості, створюючи штучну рівновагу між

класами. Крім того, підхід чутливого до вартості навчання дозволяє змінювати ваги класифікації таким чином, щоб помилки виявлення шахрайських дій каралися сильніше, ніж помилки у класифікації легітимних операцій. У випадках, коли застосовується SMOTE для збалансування класів до співвідношення 50:50, моделі класифікації дійсно демонструють підвищену точність. Однак такі результати можуть бути штучно завищеними через змінену структуру даних, яка не відображає реальну ситуацію у фінансових системах, де шахрайство є рідкісним явищем [13].

Система виявлення шахрайства реалізована на основі підходу контрольованого навчання, що передбачає використання історичних даних для побудови прогностичної моделі. Основними етапами побудови такої системи є інженерія ознак, навчання моделі та її валідація. На початковому етапі використовується історичний набір даних, який після проходження етапу обробки ознак поділяється на навчальну вибірку, валідаційну вибірку та тестову вибірку. На основі навчального набору здійснюється побудова або тренування моделі. Після побудови модель тестується за допомогою тестової вибірки, що дозволяє оцінити її прогностичну здатність. Паралельно, результати тестування валідуються з використанням валідаційного набору, що забезпечує контроль за можливим перенавчанням. Після успішної побудови модель розгортається в робочому середовищі для обробки нових даних. Нові дані, подібно до історичних, також проходять етап інженерії ознак, після чого модель здійснює прогнозування на основі набутих знань. Таким чином, реалізується замкнений цикл функціонування системи виявлення шахрайства, який поєднує фазу дослідження та тестування із фазою продуктивного застосування [33].

Загальний вигляд пропонованої системи представлено на рисунку 2.2:

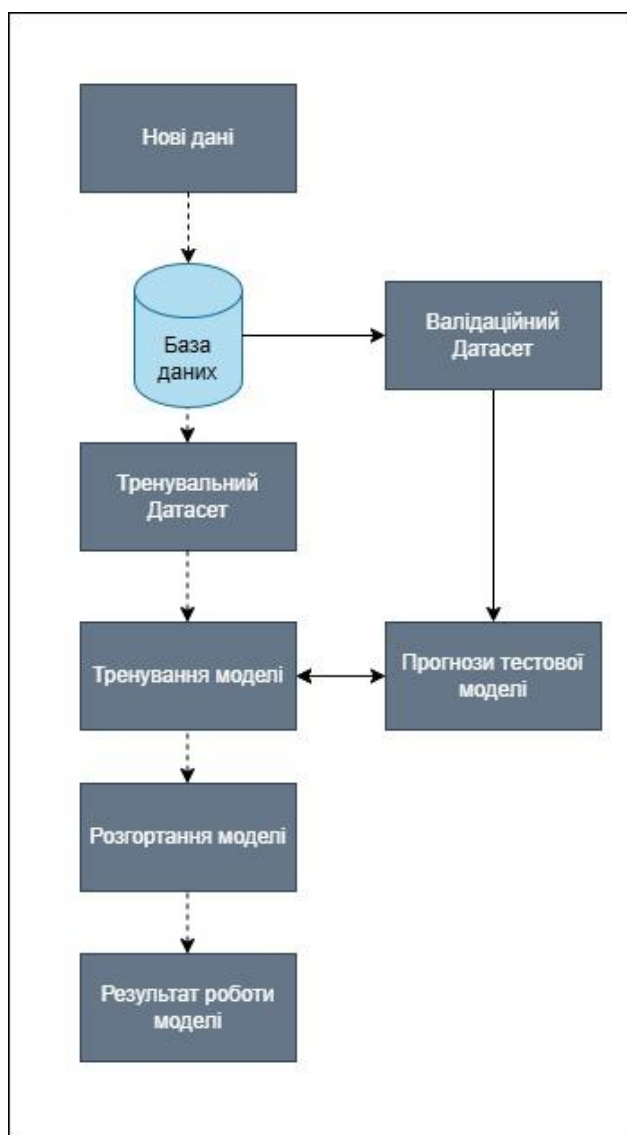


Рисунок 2.2 – Концептуальна схема системи виявлення шахрайства

Джерело: розроблено автором

Системи виявлення шахрайства з використанням машинного навчання базуються на побудові класифікаційної моделі, здатної автоматично визначати, чи є конкретна транзакція шахрайською. Процес розробки такої моделі включає кілька послідовних етапів, починаючи з підготовки даних і завершуючи оцінюванням ефективності моделі. Початковий етап передбачає збір історичних даних, що містять як легітимні, так і шахрайські транзакції. Далі виконується детальний аналіз даних, під час якого оцінюються характеристики кожного атрибуту. Особлива увага приділяється наявності пропущених значень, дублікатів, викидів і аномалій. Значення, які не мають статистично значущого впливу на класифікацію,

видаляються або трансформуються. Ці дії спрямовані на підвищення якості вхідних даних, що прямо впливає на точність моделі.

Після попередньої обробки здійснюється розділення набору даних на навчальну та тестову вибірки. Метою цього етапу є створення умов для об'єктивної перевірки узагальнюючої здатності моделі. На основі навчальних даних обрана модель навчається розпізнавати патерни, характерні для шахрайських транзакцій. У результаті формується класифікатор, здатний приймати нові приклади транзакцій на вхід та здійснювати бінарну класифікацію.

До навчального набору даних застосовуються різні алгоритми машинного навчання. Кожен з них реалізує власний підхід до пошуку межі розділення між класами “шахрайство” і “не шахрайство”. Найчастіше використовуються логістична регресія, дерева рішень, випадкові ліси, градієнтний бустинг, опорні вектори та штучні нейронні мережі. Після навчання моделі перевіряються на тестовому наборі даних. На основі отриманих результатів обчислюються ключові метрики ефективності, зокрема точність (accuracy), повнота (recall), специфічність (specificity), точність позитивного прогнозу (precision) та F1-міра. Ці показники дозволяють об'єктивно оцінити здатність моделі ідентифікувати шахрайські транзакції у нових, раніше не бачених даних.

Остаточний вибір моделі базується на порівнянні її продуктивності з іншими підходами. Оптимальним вважається алгоритм, що демонструє найбільшу збалансованість між рівнями виявлення шахрайства та мінімізацією хибнопозитивних спрацювань. У випадку високих вимог до швидкодії та масштабованості, додатково враховується складність моделі та її здатність до обробки великого обсягу даних у реальному часі.

Узагальнена схема найпростіших систем виявлення шахрайства наведена на рисунку 2.3:

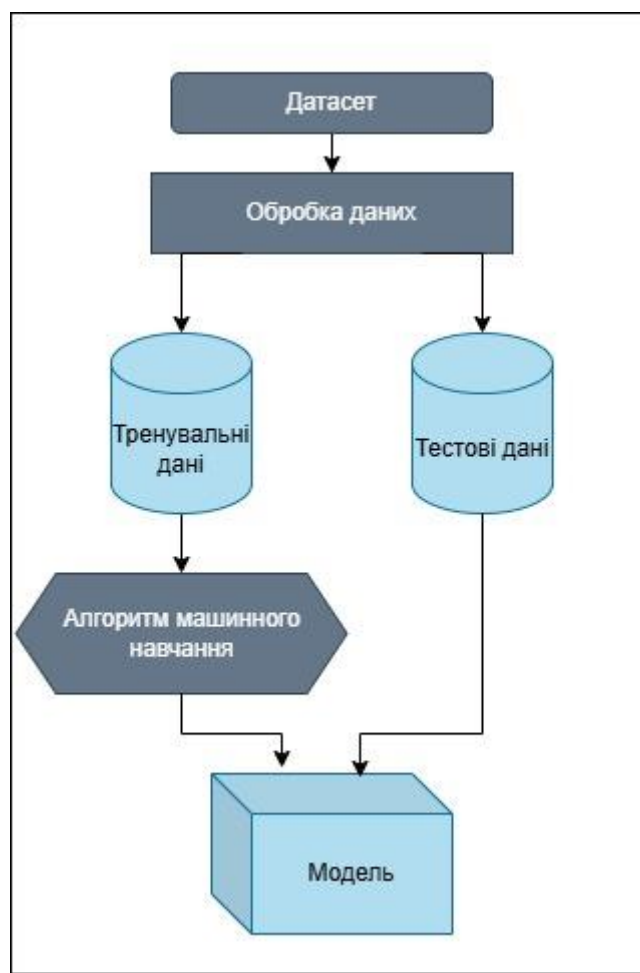


Рисунок 2.3 – Блок-схема системи виявлення шахрайства

Джерело: розроблено автором

Запропонована модель виявлення шахрайства характеризується низкою обмежень, що впливають на її практичну придатність. По-перше, система орієнтована виключно на один тип шахрайської активності — транзакції, подібні до операцій з кредитними картками. Така вузька спеціалізація знижує загальну адаптивність моделі до ширшого спектра фінансових аномалій. По-друге, реалізація моделі не передбачає зручного графічного інтерфейсу користувача. Усі вхідні дані подаються через термінал, що ускладнює доступ до системи для користувачів без технічної підготовки. По-третє, модель не має вбудованих механізмів для автоматичного збирання або генерації актуальних даних про шахрайські дії, що обмежує її здатність до оновлення знань про нові види шахрайства в режимі реального часу.

Формування класифікаційної моделі базується на використанні історичних даних, які включають як оброблені, так і необроблені транзакції. Необроблені дані підлягають попередній обробці, яка включає очищення, нормалізацію та трансформацію змінних. Ці дії є обов'язковими для забезпечення сумісності даних із машинними алгоритмами.

Наступним етапом є вибір відповідного алгоритму машинного навчання. У випадку задачі класифікації кредитних транзакцій доцільним виявляється застосування контрольованих методів, зокрема логістичної регресії, яка дозволяє будувати бінарну модель для передбачення настання шахрайства. Навчання моделі здійснюється на основі розмічених даних, а тестування забезпечує оцінку узагальнюючої здатності. Періодична перекалібровка моделі є необхідною для збереження її точності у динамічному середовищі.

Дані, що використовуються для виявлення шахрайських операцій, походять із банківських джерел, зокрема з ручної перевірки транзакцій, що були підозрілими у минулому. На підставі цих даних формуються правила з використанням встановлених порогових значень. Після цього до нових транзакцій застосовується набір критеріїв, на основі яких ухвалюється рішення щодо наявності або відсутності ознак шахрайства. У разі відповідності транзакції одному або кільком критеріям — вона позначається як потенційно шахрайська. Такий підхід дозволяє поєднувати алгоритмічне передбачення з експертною оцінкою.

Враховуючи усе вище зазначене, пропонується побудувати наступну систему виявлення шахрайства (рис. 2.4):

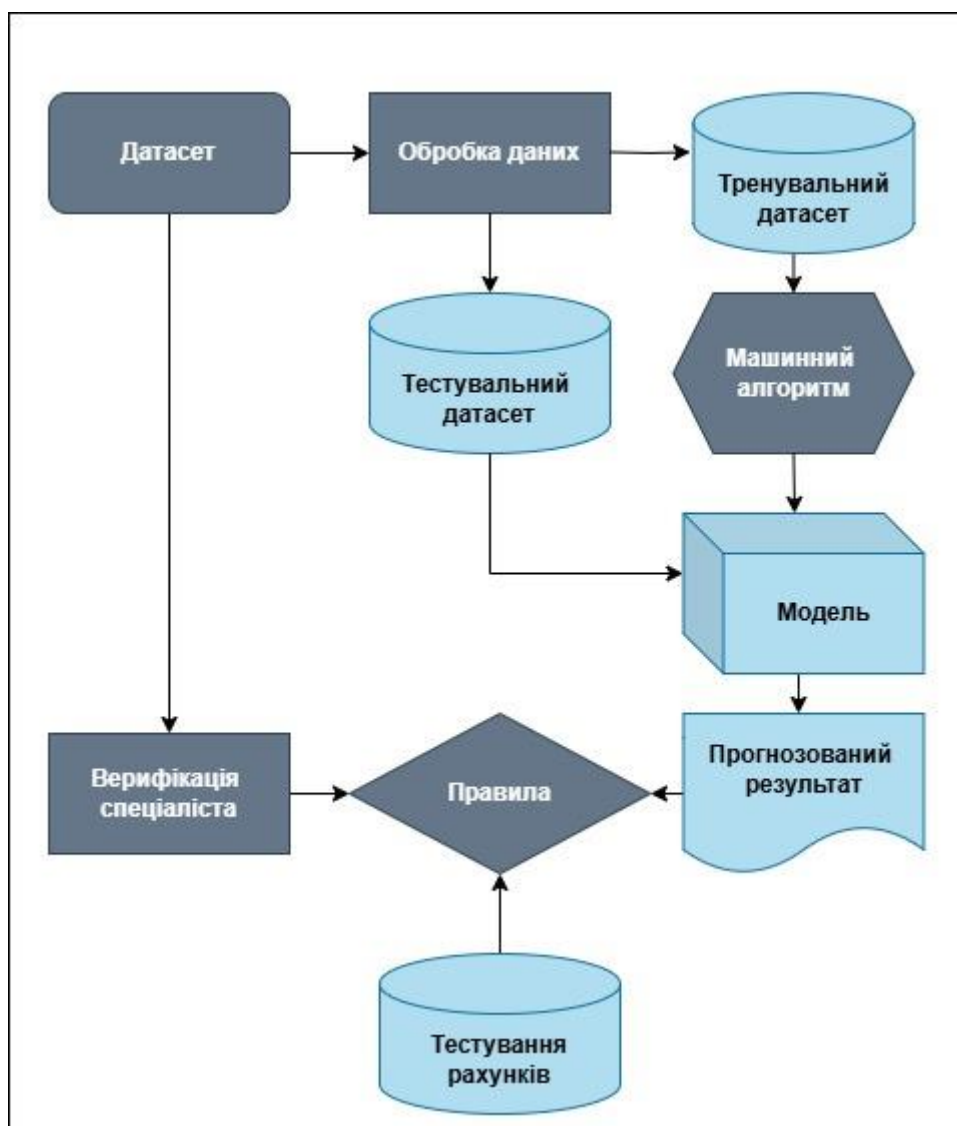


Рисунок 2.4 – Структурна схема запропонованої системи

Джерело: розроблено автором

2.2.1 Алгоритм і техніка навчання

Логістична регресія є статистичною моделлю, що використовується для моделювання ймовірності виникнення певної події, коли результуюча змінна є дихотомічною. У випадку виявлення шахрайських транзакцій ця модель дозволяє оцінити ймовірність того, що певна транзакція є шахрайською (або не є такою) на основі значень ряду незалежних змінних, що описують її характеристики. У моделі логістичної регресії залежна змінна Y набуває значення 1, якщо транзакція є

шахрайською, і значення 0, якщо вона є законною. Таким чином, модель будується з метою прогнозування ймовірності шахрайства на основі незалежних змінних, таких як сума транзакції, час доби, кількість транзакцій за короткий період, географічне розташування користувача, тощо [34].

Після розрахунку ймовірності належності транзакції до класу шахрайських, приймається порогове рішення. Якщо розраховане значення перевищує певне заздалегідь визначене порогове значення (наприклад, 0.5), транзакція класифікується як шахрайська (тобто $Y = 1$); у протилежному випадку — як не шахрайська (тобто $Y = 0$). Застосування логістичної регресії у даному контексті дозволяє не лише автоматизувати процес виявлення шахрайських транзакцій, а й забезпечує інтерпретованість моделі. Аналізуючи, які фактори найбільше впливають на ймовірність виникнення шахрайства, що є корисним при розробці додаткових систем контролю та профілактики фінансових зловживань.

Зазначене вище, дає загальне представлення процесу виявлення шахрайства у системі, далі (рис. 2.5 та рис. 2.6) зображено візуальне представлення структури, компоненти та взаємозв'язки системи:

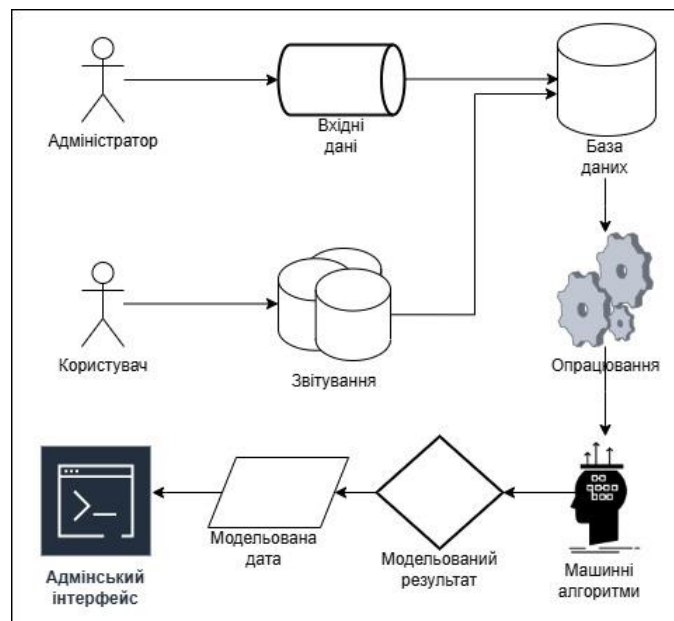


Рисунок 2.5 – Архітектура системи

Джерело: розроблено автором

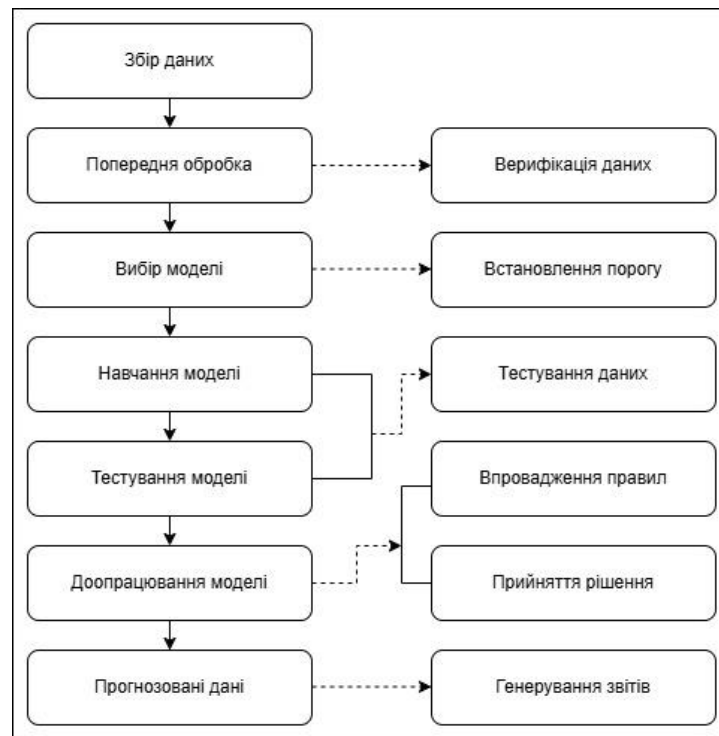


Рисунок 2.6 – Діаграма потоку даних

Джерело: розроблено автором

У контексті виявлення шахрайства машинне навчання та методи, засновані на правилах, спираються на наявність великих обсягів історичних даних. Ці дані можуть містити інформацію про минулі транзакції, внутрішні звіти, користувацькі поведінкові шаблони та інші джерела. Збір даних є критичним етапом, оскільки точність моделі значною мірою залежить від якості та репрезентативності вхідної інформації.

Після збору даних необхідно виконати попередню обробку. Цей етап включає виявлення та усунення пропущених значень, корекцію помилок форматування, стандартизацію одиниць виміру та трансформацію змінних. Нормалізація, кодування категоріальних змінних і зменшення вимірності можуть також бути застосовані для забезпечення сумісності даних із алгоритмами машинного навчання.

Модель або правила не залишаються статичними. Зі зміною поведінкових патернів або шахрайських технік виникає необхідність у повторному навчанні або оновленні бази правил. Така адаптація дозволяє підтримувати актуальність

системи в умовах динамічного середовища та підвищує її ефективність у довгостроковій перспективі.



Рисунок 2.7 – Блок-схема алгоритму дії системи

Джерело: розроблено автором

Блок-схема алгоритму дії системи (рис. 2.7) слугує графічним представленням для зображення та позначення послідовності подій та дій, які може виконати система.

Діаграма дій користувачів (рис. 2.8), описує тип взаємодії класу користувачів з системою, тобто адміністратори можуть виконувати різні дії (автентифікація, завантаження набору даних, прогнозування перегляду тощо), тоді як інші користувачі (люди) можуть лише повідомляти про банківські рахунки, пов'язані з шахрайством:

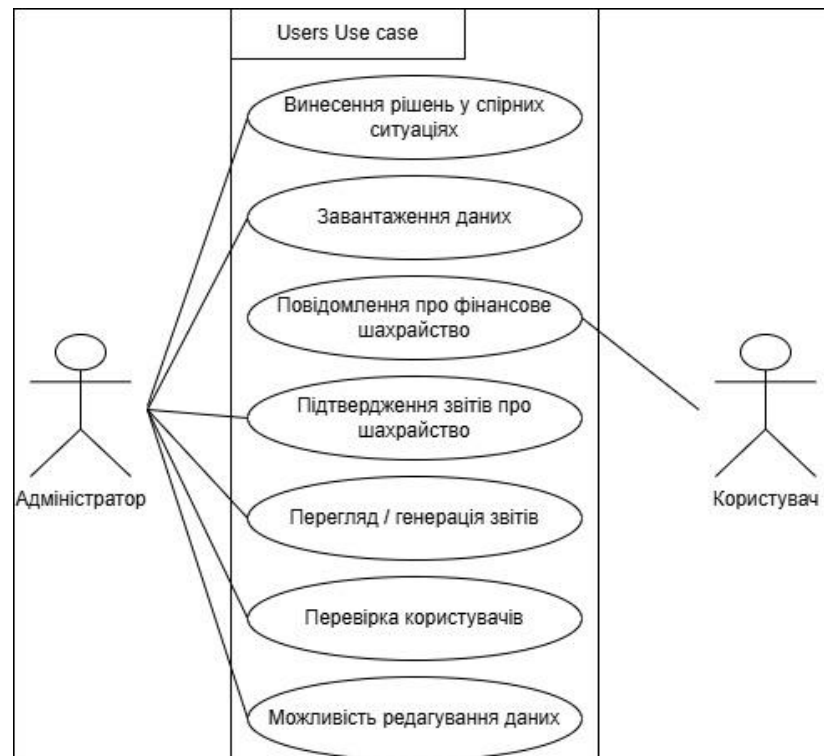


Рисунок 2.8 – Use-case діаграма дій користувачів

Джерело: розроблено автором

2.3 Методи, моделі і моделювання процесів і елементів складних систем

Системи виявлення шахрайства, попри значні досягнення в теорії та практиці, продовжують стикатися з низкою проблем, які потребують постійного наукового аналізу та вдосконалення. Однією з ключових перешкод є значна незбалансованість доступних наборів даних, у яких кількість легітимних транзакцій значно перевищує кількість шахрайських. Такий дисбаланс ускладнює процес навчання моделей, призводячи до упередженості на користь класу більшості, що знижує ефективність виявлення шахрайства. Усунення класової диспропорції залишається критично важливим завданням при розробці точних і надійних аналітичних систем.

Наукова спільнота зосереджує значну увагу на підвищенні точності виявлення шахрайства, забезпеченні адаптивності до змінюваних схем шахрайської поведінки та поліпшенні прозорості функціонування алгоритмів. У

цьому контексті досліджуються як традиційні методи, побудовані на експертних правилах, так і сучасні алгоритми машинного навчання, які здатні автоматично виявляти закономірності в транзакційних даних. Перевага новітніх підходів полягає у здатності масштабуватися до великих обсягів даних та пристосовуватися до нових загроз.

Система виявлення шахрайства — інтегрований комплекс інструментів, технологій та алгоритмічних рішень, спрямованих на ідентифікацію потенційно шахрайських дій у різних сферах застосування. До таких сфер належать фінансові транзакції, цифрові платформи, системи електронної комерції та охорони здоров'я. Головною метою системи є виявлення аномальних шаблонів поведінки, які можуть свідчити про шахрайські дії, з метою мінімізації фінансових збитків, збереження конфіденційності даних користувачів і підтримання цілісності функціонування інформаційних систем.

Ансамблеві методи на сьогодні є одним з ефективних підходів для вирішення задач виявлення шахрайства у фінансових транзакціях. Їх застосування ґрунтується на комбінації кількох моделей машинного навчання, що дозволяє підвищити точність прогнозування та забезпечити стійкість до змін у даних. Основна ідея полягає в тому, що окремі моделі, які самотійно можуть мати обмежену ефективність, у сукупності можуть доповнювати одна одну, компенсуючи індивідуальні недоліки. Фінансове шахрайство становить серйозну загрозу як для окремих користувачів, так і для компаній та інфраструктури фінансових систем загалом. У зв'язку з ускладненням методів шахрайської діяльності традиційні підходи часто виявляються недостатньо ефективними. У таких умовах ансамблеві методи забезпечують гнучкість і адаптивність, необхідні для ефективного виявлення нових і змінюваних схем шахрайства. Комбінування моделей дозволяє охопити ширший спектр ознак та патернів, властивих шахрайській активності [14].

Методологічно ансамблеві методи спрямовані на зменшення дисперсії, зміщення та ризику перенавчання моделей. Підвищуючи здатність алгоритмів до

узагальнення, що критично важливо при роботі з реальними фінансовими даними. Серед поширених підходів виділяють:

- Boosting — послідовне комбінування слабких моделей, де кожна наступна зосереджується на помилках попередніх. Така стратегія дозволяє поетапно підвищувати точність шляхом зменшення залишкових похибок.
- Bagging — створення множини моделей, зазвичай дерев рішень, які тренуються на різних випадкових підмножинах даних. Результати цих моделей комбінуються (наприклад, голосуванням), що знижує варіативність і підвищує стійкість прогнозу.
- Stacking — багаторівнева схема, де мета-навчальна модель поєднує виходи кількох базових моделей для формування фінального рішення. Враховуючи взаємодію між окремими підходами та оптимізувати їх комбінацію.

Методи глибокого навчання активно досліджуються та впроваджуються у сферу виявлення шахрайства з кредитними картками завдяки їхній здатності моделювати складні закономірності та взаємозв'язки в багатовимірних транзакційних даних. Традиційні системи, засновані на фіксованих правилах, демонструють обмежену гнучкість щодо змінних стратегій шахраїв. Класичні алгоритми машинного навчання, як-от логістична регресія або метод опорних векторів, не завжди ефективно виявляють нелінійні залежності, характерні для фінансових транзакцій. У цьому контексті глибокі нейронні мережі забезпечують розширені можливості аналізу [36].

Одним із напрямів, що демонструє перспективність у цій сфері, є використання рекурентних нейронних мереж (RNN). Цей тип архітектури підходить для обробки послідовностей, зокрема транзакційної активності клієнтів, де кожна операція розглядається як окремий етап у часовому ряді. У контексті виявлення шахрайства RNN дозволяють моделювати залежності між операціями, виявляючи потенційно підозрілі закономірності на основі попередніх дій [35].

Довга короткочасна пам'ять (LSTM): LSTM — особливий тип RNN, призначений для пом'якшення проблеми зникнення градієнта, дозволяючи їм

фіксувати довгострокові залежності в послідовних даних. Комірка LSTM складається зі стану комірки і кілька воріт:

$$i_t = \sigma(W_{\{xi\}}x_t + W_{\{hi\}}h_{\{t-1\}} + b_i) \quad (2.3)$$

$$f_t = \sigma(W_{\{xf\}}x_t + W_{\{hf\}}h_{\{t-1\}} + b_f) \quad (2.4)$$

$$o_t = \sigma(W_{\{xo\}}x_t + W_{\{ho\}}h_{\{t-1\}} + b_o) \quad (2.5)$$

$$C_t = f_t \odot C_{\{t-1\}} + i_t \odot \tanh(W_{\{xc\}}x_t + W_{\{hc\}}h_{\{t-1\}} + b_c) \quad (2.6)$$

$$h_t = o_t \odot \tanh(C_t) \quad (2.7)$$

де:

x_t - є введенням у час t (наприклад, функції транзакцій);

h_t - прихований стан у часі t ;

C_t - стан клітини в момент часі t ;

σ - сигмоїдна функція;

W - представляє вагові матриці;

b - представляє вектори зміщення;

$\odot$ - позначає поелементне множення. LSTM можуть вивчати такі шаблони, як звички витрачати, типові суми транзакцій і часто відвідувані місця, що дозволяє їм визначати відхилення, які можуть вказувати на шахрайство.

Поведінкове виявлення шахрайства з використанням алгоритмів машинного навчання – підхід, що дозволяє виявляти шахрайські дії шляхом аналізу поведінкових характеристик користувачів, транзакцій або інших об'єктів. На відміну від класичних систем на основі фіксованих правил, машинне навчання дає змогу виявляти складні закономірності в історичних даних, що сприяє більш гнучкому й точному виявленню відхилень, які можуть вказувати на шахрайство [37].

Процес поведінкового виявлення шахрайства включає кілька ключових етапів. Спочатку виконується збір даних, до яких можуть належати історії транзакцій, профілі користувачів, мережеві журнали активності та інші релевантні

джерела. Далі проводиться попередня обробка даних, що включає очищення, трансформацію та нормалізацію з метою забезпечення їх відповідності вимогам алгоритмів. Важливим кроком є створення нових ознак (feature engineering), наприклад: частота транзакцій, середній розмір транзакції або час, що минув з моменту останньої операції [44].

Вибір алгоритму машинного навчання залежить від характеру даних і конкретного сценарію виявлення шахрайства. Зазвичай використовувані алгоритми включають:

- Контрольоване навчання: ці алгоритми вимагають мічених даних (тобто даних, у яких виявлено шахрайські та нешахрайські випадки). Приклади:
- Логістична регресія: передбачає ймовірність шахрайської транзакції на основі вхідних характеристик. Формула логістичної регресії така:

$$P(fraud(y = 1|x)) = \frac{1}{(1+e^{(-z)})} \quad (2.8)$$

де:

$x = (x_1, x_2, x_3, \dots, x_n)$ – вектор вхідних ознак;

e – основа натурального логарифму;

$P(fraud(y = 1|x))$ – ймовірність позитивного класу (в даному випадку шахрайства)

- Древа рішень і випадкові ліси: дерева рішень поділяють дані на основі значень ознак для класифікації екземплярів. Випадкові ліси, метод ансамблю, поєднують кілька дерев рішень для підвищення точності та зменшення надмірного оснащення.
- Машини підтримки векторів (SVM): SVM знаходять оптимальну гіперплощину для відокремлення шахрайських екземплярів від нешахрайських у багатовимірному просторі.
- Нейронні мережі: складні моделі, здатні вивчати складні шаблони з даних. Вони складаються з взаємопов'язаних вузлів (нейронів), організованих у шари, які можуть автоматично вивчати ієрархічні представлення ознак.

- Неконтрольоване навчання: ці алгоритми працюють з даними без міток і спрямовані на виявлення аномалій без попередньої інформації про випадки шахрайства.

Алгоритми неконтрольованого навчання становлять важливий інструмент у системах виявлення шахрайських операцій з кредитними картками. Їх актуальність зумовлена динамічним характером шахрайства та обмеженим обсягом доступних мічених даних. Ці алгоритми орієнтовані на виявлення закономірностей та аномалій, що суттєво відрізняються від типових транзакцій, без необхідності попереднього маркування даних як шахрайських або легітимних [38].

Задача виявлення шахрайства характеризується низкою суттєвих труднощів. По-перше, транзакції-шахрайства становлять лише незначну частку всіх операцій, що призводить до суттєвого дисбалансу класів. Що ускладнює побудову моделей, здатних точно ідентифікувати рідкісні шахрайські випадки. По-друге, шахрайські схеми постійно змінюються, що робить неактуальними моделі, навчені на історичних даних. По-третє, марковані приклади шахрайства є обмеженими або недоступними, особливо в умовах появи нових видів шахрайства. По-четверте, системи повинні функціонувати в режимі реального часу, що потребує ефективних і масштабованих обчислювальних рішень. Нарешті, інтерпретованість рішень є критичною у багатьох прикладних контекстах, де важливо пояснити, чому певна транзакція була визначена як потенційно шахрайська.

У практиці виявлення шахрайства застосовуються різні методи неконтрольованого навчання. Однією з ключових груп таких алгоритмів є кластеризаційні методи. Зокрема, алгоритми K-means, DBSCAN і ієрархічна кластеризація використовуються для групування схожих транзакцій. Транзакції, що відхиляються від типових кластерів або утворюють власні невеликі групи, можуть трактуватися як потенційно шахрайські.

У випадку алгоритму K-means мета полягає у поділі n спостережень на k кластерів таким чином, щоб кожне спостереження належало до кластера з найближчим центроїдом. Оптимізаційна функція цього методу формулюється як

мінімізація суми квадратів відстаней між об'єктами та відповідними центрами кластерів (within-cluster sum of squares, WCSS):

$$J = \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2 \quad (2.9)$$

де:

k – кількість кластерів;

S_i – набір точок даних у кластері i ;

x – є точкою даних у S_i ;

μ_i – є центроїдом кластера i ;

$\|x - \mu_i\|$ – евклідова відстань між точками даних x і центроїда μ_i ;

Ізоляційний ліс: цей алгоритм ізолює аномалії шляхом випадкового розділення простору даних. Аномалії легше ізолювати, тому потрібно менше розділів. Оцінка аномалії визначається як:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \quad (2.10)$$

де:

$h(x)$ – довжина шляху для точки даних x ;

$E(h(x))$ – середня довжина шляху набору точок даних;

$c(n)$ – середня довжина шляху невеликого пошуку в бінарному дереві пошуку, і вона використовується для нормалізації довжини шляху;

n – кількість зовнішніх вузлів;

Автокодери – нейронні мережі навчені реконструювати свої вхідні дані. Шахрайські транзакції, оскільки вони відрізняються від звичайних транзакцій, можуть бути погано реконструйовані, що призводить до високих помилок реконструкції, які можна використовувати як бали аномалії.

Функція втрат для автокодуювальника часто являє собою середню квадратичну помилку (MSE) між входом і реконструйованим виходом:

$$MSE = \frac{1}{n} \sum_{i=1}^n ||x_i - \hat{x}_i||^2 \quad (2.11)$$

де:

x_i – є точкою вхідних даних.;

$\hat{x}_i$ – є реконструйованою точкою даних;

n – кількість точок даних;

Висновки до розділу 2

У цьому розділі здійснено ґрунтовний аналітико-дослідницький розгляд проблеми виявлення фінансового шахрайства з використанням систем штучного інтелекту. Об'єктом дослідження виступає система виявлення шахрайських транзакцій у платіжних системах, що функціонує на основі алгоритмів машинного навчання та інтелектуального аналізу даних. Було охарактеризовано архітектуру системи, її функціональні компоненти, а також обрано підхід до обробки вхідних даних на базі відкритих анонімізованих наборів транзакцій.

У розділі розглянуті методи контрольованого та неконтрольованого навчання, зокрема логістичну регресію, дерева рішень, випадкові ліси, SVM, глибокі нейронні мережі, автокодерери та кластеризаційні алгоритми (K-means, DBSCAN). Для вирішення проблеми дисбалансу класів, притаманної задачам шахрайства, використано методи надсемплінгу та чутливого до вартості навчання. Особливу увагу приділено поведінковим ознакам користувачів і моделюванню часових залежностей за допомогою RNN та LSTM.

Описано процес побудови аналітичної системи від етапу збору та очищення даних до формування прогностичних моделей і їх валідації. Запропонована концептуальна та структурна схема системи з відповідними алгоритмами функціонування. Розглянуто архітектурні та обчислювальні аспекти системи, а також визначено обмеження реалізованої моделі та напрями її вдосконалення.

РОЗДІЛ 3. КОНСТРУКТИВНИЙ РОЗДІЛ. РОЗРОБЛЕННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Проектування бази знань для системи прийняття інтелектуальних рішень та управління

У системах виявлення та запобігання фінансовому шахрайству поняття бази даних трансформується від класичної програмної реалізації до функціонального набору даних, структурованих відповідно до вимог алгоритмів машинного навчання. Такий підхід відображає зсув акценту з транзакційного зберігання інформації на її аналітичне опрацювання. Замість реалізації повноцінної реляційної або нереляційної бази даних у традиційному розумінні, система оперує структурованими вибірками, що формуються з даних, зібраних із транзакційних логів, клієнтських профілів, мережевої активності та інших джерел. Ці дані акумулюються з метою виявлення аномалій, класифікації подій або прогнозування ймовірності шахрайських дій.

Такі набори даних піддаються серії попередніх обробок, зокрема фільтрації, нормалізації, інженерії ознак, а також кодування змінних, що є необхідною умовою для забезпечення придатності даних до машинного аналізу. Під час цих трансформацій дані перестають функціонувати як база в термінах операційної підтримки, і натомість виконують роль вхідного середовища для аналітичного ядра системи. Фактично, роль бази даних у цьому контексті зводиться до тимчасового зберігання, трансформації та агрегування інформації, яка в подальшому використовується як навчальна або тестова вибірка. Таким чином, функціональне значення БД у системі полягає не в реалізації транзакційної логіки, а у забезпеченні високої якості даних для інтелектуального аналізу.

Датасети кредитних карток для тренування або ж безпосереднього виявлення шахрайства використовується структура даних із наявністю числових ознак, які є результатом попереднього опрацювання і знеособлення первинної

інформації. Типовий набір даних включає в себе набір вхідних параметрів, де кожен рядок репрезентує окрему транзакцію, а стовпці – її характеристики [39]. Загальна мета структурування таких даних полягає в тому, щоб забезпечити відповідність вимогам алгоритмів машинного навчання, які використовуються для виявлення аномалій або побудови моделей класифікації:

- Ознака Time представляє собою часову відмітку, що визначає кількість секунд, які минули з моменту початку збору даних до моменту виконання транзакції. Цей параметр дозволяє враховувати часові закономірності або залежності між транзакціями.
- Ознаки V1–V28 є результатом застосування методу головних компонент (PCA), що використовується для зменшення розмірності даних та забезпечення конфіденційності шляхом знеособлення первинних фінансових та особистих характеристик. Ці змінні репрезентують латентні фактори, які є комбінаціями оригінальних ознак, таких як місце транзакції, тип пристрою, географічне розташування, ідентифікатор продавця тощо. Вони використовуються як основні предиктори у побудові моделей машинного навчання.
- Параметр Amount визначає грошову суму транзакції. Цей параметр не піддається PCA-трансформації та зберігає реальне числове значення, що є важливим для оцінки масштабу можливого шахрайства.
- Цільова змінна Class є бінарною та використовується для визначення належності транзакції до певного класу. Значення 0 означає, що транзакція є легітимною, тоді як значення 1 вказує на ймовірність шахрайства. Саме ця ознака слугує основою для навчання моделей класифікації у підходах з контрольованим машинним навчанням.

Первинні дані, що містять персональну або фінансову інформацію користувачів, не можуть бути використані у відкритому вигляді через високий ризик витоку конфіденційної інформації. Таким чином, забезпечення захисту персональних даних користувачів та відповідність нормам законодавства щодо обробки персональної інформації є ключовими чинниками для попередньої трансформації наборів даних. Для досягнення анонімності часто застосовується

метод головних компонент (Principal Component Analysis, PCA), що дозволяє знизити розмірність початкового простору ознак і замінити оригінальні змінні похідними латентними компонентами. У результаті дані представлені у вигляді змінних V1–V28, які не мають прямої інтерпретації, але зберігають основну інформацію про структуру та варіації у транзакціях. Цей підхід дозволяє захистити чутливі поля, такі як номер картки, ім'я власника або геолокаційні дані, оскільки вони не присутні в явному вигляді, загальний вигляд таких анонімізованих даних представлено на рисунку 3.1:

	A
1	Time,"V1","V2","V3","V4","V5","V6","V7","V8","V9","V10","V11","V12","V13","V14","V15","V16","V17","V18","V19","V20","V21","V22","V23","V24","V25","V26","V27","V28","Amount","Class"
2	0,-1.3598071336738,-0.0727811733098497,2.53634673796914,1.37815522427443,-0.338320769942518,0.462387777762292,0.239598554061257,0.0986979012610507,0.363786969611213,0.0907941719789316,-0.551599533260813,-0.6:
3	0,1.1918571131486,0.26615071205963,0.16648011335321,0.448154078460911,0.0600176492822243,-0.0823608088155687,-0.078802983323113,0.0851016549148104,-0.255425128109186,-0.166974414040614,1.61272666105479,1.0
4	1,-1.35835406159823,-1.34016307473609,1.77320934263119,0.37977593034328,-0.503198133318193,1.80049938079263,0.791460956450422,0.247675786588991,-1.51465432260583,0.207642865216696,0.624501459424895,0.066083
5	1,-0.966271711572087,-0.185226008082898,1.79299333957872,-0.863291275036453,-0.0103088796030823,1.24720316752486,0.23760893977178,0.377435874652262,-1.38702406270197,-0.0549519224713749,-0.226487263835401,0.1
6	2,-1.15823309349523,0.877736754848451,1.548717846511,0.403033933955121,-0.407193377311653,0.0959214624684256,0.592940745385545,-0.270532677192282,0.817739308235294,0.753074431976354,-0.822842877946633,0.5381
7	2,-0.425965884412454,0.96052304488285,1.14110934232219,-0.168252079760302,0.42098688077219,-0.0297275516639742,0.476200948720027,0.260314333074874,-0.56867137571251,-0.371407196834471,1.34126198001957,0.3591
8	4,1.22965763450793,0.141003507049326,0.0453707735899449,1.20261273673594,0.191880988597645,0.272708122899098,-0.00515900288250983,0.0812129398830894,0.464959994783886,-0.0992543211289237,-1.41690724314928,-0.2
9	7,-0.644269442348146,1.41796354547385,1.0743803763556,-0.492199018495015,0.948934094764157,0.428118462833089,1.12063135838353,-3.80786423873589,0.615374730667027,1.24937617815176,-0.619467796121913,0.2914743
10	7,-0.89428608220282,0.286157196276544,-0.113192212729871,-0.271526130088604,2.6695986595986,3.72181806112751,0.370145127676916,0.851084443200905,-0.392047586798604,-0.410430432848439,-0.705116586646536,-0.110
11	9,-0.33826175242575,1.11959337641566,1.04436655157316,-0.222187276738296,0.49936080649727,-0.24676110061991,0.651583206489972,0.0695385865186387,-0.736727316364109,-0.366845639206541,1.01761446783262,0.836381
12	10,1.44904378114715,-1.17633882535966,0.913859832832795,-1.37566665499943,-1.97138316545323,-0.62915213889734,-1.4232356010359,0.0484558879088564,-1.72040839292037,1.626659505834133,1.1996439495421,-0.67143977
13	10,0.38497821518095,0.616109459176472,-0.874299702595052,-0.0940186259679115,2.92458437888177,3.31702716826156,0.470454671805879,0.53824722837695,-0.558894612428441,0.30975539423728,-0.259115563735702,-0.326:
14	10,1.249998742053,-1.22163680921816,0.383930151282291,-1.23489868766892,-1.48541947377961,-0.753230164566149,-0.689404975426345,-0.227487272519552,-0.29401057344842,1.32372927445937,0.227666231237246,-0.24268:
15	11,1.0693735878819,0.287722129331455,0.828612726634281,2.71252042961718,-0.178398016248009,0.337543730282968,-0.0967168617359562,0.115981735546597,-0.221082566236194,0.4602304440301678,-0.773656930526689,0.32
16	12,-0.7518457659339,-0.327770756658658,1.64175016056605,1.76747274389883,-1.36588446465306,0.80759646826532,-0.422911389711497,-1.90710747624096,0.755712908314791,1.1510869876677,0.844555470974377,0.7929439:
17	12,-0.752417042956605,0.345485415344747,2.05732291276727,-1.46864329840046,-1.1583936804082,-0.0778498291166733,-0.608581418236123,0.00360348436201849,-0.436166983515744,0.747730827192802,-0.793980602837221,-
18	12,1.10321543528383,-0.0402962145973447,1.2673320885949,1.28909146962552,-0.735997163604068,0.288069162976262,-0.58605786337461,0.189379713679593,0.782332891785191,-0.267975066537173,-0.450311279515466,0.93:
19	13,-0.436905071360625,0.918966212909322,0.92459077438817,-0.727219053596792,0.915678718106307,-0.127867352079254,0.707641607333935,0.0879623545672504,-0.66527135413364,-0.73797823596458,0.32409781346169,0.2:
20	14,-5.40125766315825,-5.45014783420644,1.18630463143652,1.73623880012095,3.04910587764025,-1.76340557365201,-1.55973769907953,0.160841747266769,1.23308974041888,0.345172827050629,0.917229867699146,0.97011671
21	15,1.4929359769862,-1.02934573189487,0.45479473374366,-1.43802587991702,-1.55543410136344,-0.720961147043557,-1.08066413038614,-0.0531271179483221,-1.9786815953872,1.63807603690446,1.07754241162743,-0.6320465:
22	16,0.69488475607337,-1.36181910308009,1.02922103956032,0.834159299216716,-1.19120879445965,1.30910881872952,-0.878585911450457,0.4452901278385,-0.446195831557423,0.568520735086962,1.01915061274695,1.2983287
23	17,0.96249609914852,0.32846102605212,-0.17147905415064,2.10920406774016,1.12956557126894,1.6960376856836,0.107711607311367,0.521502163844302,-1.19131110205143,0.724396314588903,1.6903299205808,0.4067735756:
24	18,1.16661638244228,0.502120087854101,-0.0673003143663533,2.26156923949128,0.428804194630708,0.0894735167274599,0.241146579907281,0.138081705243702,-0.989162359259052,0.922174967079328,0.744785789437272,-0.5
25	18,0.24749112778665,0.27765627353681,1.18547084217971,-0.0926025498576041,-1.314239397897076,-0.150115997622665,-0.946364950111676,-1.61793505064675,1.54407140197844,-0.8298806074647,-0.583199526903095,0.52
26	22,-1.94652513121534,-0.044900504418194,-0.405570068378956,-0.101305733702394,2.94196769950545,2.95505339674562,-0.0630631473635638,0.855546309018146,0.0499669002270836,0.573742508037695,-0.0812565147381817,-
27	22,-2.0742946726229,-0.121481799450951,1.32202063048967,0.410007514171835,0.295197545759436,-0.95953722984438,0.543985491287656,-0.104626728092018,0.475664017945495,0.149450615348245,-0.856566363963256,-0.180
28	23,1.17328461017079,0.353497876646456,0.28390506526532,1.1335633178795,-0.172577181842493,-0.916053706932115,0.36902484539083,-0.327260242196767,-0.24665102789456,-0.0461393022689538,-0.143418527309124,0.979:
29	23,1.32270726911234,-0.17404083293642,0.434555031250987,0.576037652384661,-0.836758045983797,-0.831083411483829,-0.264904960791055,-0.220981942667704,-0.171424617272052,0.868558547570527,-0.641506289794406,-0.

Рисунок 3.1 – CSV-представлений датасет

Джерело: загальнодоступний датасет [11]

Системи виявлення шахрайства, що ґрунтуються на алгоритмах машинного навчання, функціонують з анонімізованими числовими даними шляхом математичного аналізу векторних представлень транзакцій. Незалежно від того, що ці числові значення є непридатними для безпосереднього людського тлумачення, вони несуть структуровану інформацію про приховані зв'язки між параметрами транзакцій, які можуть бути критично важливими для виявлення аномальної поведінки.

Такі значення, як 0, -1.3598071, -0.0727811 у датасетах (рис. 3.1), представляють координати у зменшеному векторному просторі, отриманому, наприклад, через перетворення головних компонент (PCA). Алгоритми не намагаються "розшифрувати" ці числа у буквальному сенсі; замість цього вони

використовують їх у якості вхідних ознак для навчання класифікаційних або кластеризаційних моделей. Ці ознаки дозволяють алгоритмам формувати гіперплощини, правила або класифікаційні дерева, що розділяють простір транзакцій на зони "нормальної" та "аномальної" активності.

Ключовим елементом є не семантичне розуміння кожного числа, а їхня статистична значущість та кореляційна структура. Під час навчання модель визначає комбінації чисел, які найчастіше зустрічаються у нормальних або шахрайських транзакціях, і використовує це знання для подальшого прийняття рішень у тестових або нових даних. Тобто, обчислення базується на виявленні відхилень у багатовимірному числовому просторі.

Організація збору і передання первинної інформації.

Побудова датасетів, призначених для виявлення фінансового шахрайства, є складним багаторівневим процесом, що вимагає суворого дотримання принципів інформаційної безпеки, конфіденційності та юридичної відповідності. Дані, які використовуються в таких задачах, зазвичай походять з різних джерел у фінансовому секторі, зокрема з банківських систем, систем електронної комерції, платіжних процесорів та інших інституцій, що мають справу з транзакційною діяльністю користувачів. Первинні записи транзакцій містять велику кількість чутливої інформації: особисті дані користувачів, реквізити рахунків, суми операцій, геолокаційні мітки, часові відмітки та інші метадані [40].

Етап збору даних є ключовим для забезпечення достовірності і повноти транзакційної інформації, яка має відповідати реальним операціям, незалежно від того, були вони законними чи шахрайськими. Цей процес часто включає автоматизоване вилучення інформації з баз даних платіжних систем та журналів подій, після чого проводиться попереднє фільтрування з метою усунення дублювань, збоїв введення або системних помилок. Після цього дані агрегуються у структуровані записи, що представляють кожну транзакцію як окремий об'єкт з чітко визначеними ознаками.

Одразу після етапу збору ініціюється процес анонімізації даних, що має критичне значення для подальшої роботи з ними. Оскільки фінансові транзакції

безпосередньо пов'язані з особистістю клієнта, повний доступ до первинних необроблених даних може створити загрозу витоку конфіденційної інформації. З міркувань безпеки, навіть у межах однієї організації доступ до вихідної інформації мають виключно уповноважені працівники, а обробка відбувається у захищеному середовищі з відповідними обмеженнями прав доступу. Анонімізація реалізується шляхом вилучення або шифрування персональних атрибутів, таких як імена, номери рахунків, адреси, IP-адреси тощо, а також шляхом перетворення числових змінних у менш ідентифіковувану форму (наприклад, через головні компоненти або нормалізацію).

Після анонімізації дані піддаються первинному кодуванню, класифікації та формуванню ознак (feature engineering). На цьому етапі виконується трансформація необроблених атрибутів у форму, зручну для використання алгоритмами машинного навчання. Наприклад, часові ознаки можуть бути перетворені у відносні значення або категорії; географічні дані — замінені на кластери або відстані; а деякі числові ознаки — нормалізовані в межах стандартного діапазону. Також вводиться цільова змінна (target), яка у випадку виявлення шахрайства зазвичай має бінарне значення: 0 — транзакція не є шахрайською, 1 — транзакція визнана шахрайською.

Передача інформації з технічної точки зору відбувається у формалізованому цифровому форматі через захищені канали. Найчастіше використовуються внутрішні приватні сервери або шифровані тунелі VPN, а також стандарти безпечної передачі даних (наприклад, TLS, HTTPS, SSH). Застосовуються також механізми цифрового підпису та контроль цілісності пакетів, що дозволяє уникати спотворень або підміни інформації під час передачі [41].

Публічний доступ до таких датасетів є надзвичайно обмеженим. Навіть якщо дані проходять процедуру анонімізації та спрощення, вони рідко потрапляють у відкритий простір через високі ризики несанкціонованого відновлення чутливої інформації або використання даних у зловмисних цілях. У багатьох випадках фінансові установи, які володіють такими даними, обмежують їх використання внутрішніми дослідницькими командами або надають доступ до

них лише в рамках контрольованих партнерських проєктів, зобов'язуючи учасників дотримуватись політик збереження конфіденційності і не передавати інформацію третім сторонам [42].

Задля забезпечення інформаційної безпеки та збереження комерційної та клієнтської таємниці банки та інші фінансові установи дотримуються суворої політики щодо обробки та зберігання даних, пов'язаних із фінансовими операціями. Одним із ключових аспектів цієї політики є відмова від передачі навіть анонімізованих транзакційних даних зовнішнім стороннім організаціям або підрядникам, що здійснюють аутсорсингову діяльність у сфері аналітики та машинного навчання. Такий підхід є стратегічно виваженим і обґрунтованим з точки зору як юридичних, так і практичних міркувань [43].

Використання зовнішніх підрядників збільшує ризики витоків даних і неузгодженості в дотриманні регуляторних вимог. Основним ризиком при передачі навіть знеособлених даних є збереження потенційної можливості їх деанонімізації, особливо з урахуванням сучасних технологій обробки даних та обчислювальних потужностей, які дозволяють поєднувати множинні джерела інформації для відновлення особистості клієнта або структури фінансових зв'язків. Створюючи загрозу витоку інформації, яка може бути використана не лише для нанесення репутаційної шкоди установі, але і для зловмисних дій щодо клієнтів або самого банку. Зважаючи на ці ризики, фінансові організації реалізують політику кадрової замкненості, в межах якої формуються внутрішні команди з аналітики, обробки даних, розробки моделей машинного навчання та інформаційної безпеки [15].

Замість аутсорсингу, банки схиляються до формування власних аналітичних відділів або центрів обробки даних, що функціонують як інтегрована частина загальної IT-інфраструктури організації. Відбір фахівців до таких підрозділів відбувається з урахуванням не лише професійної кваліфікації, а й високих етичних стандартів, а також здатності до роботи в умовах суворого режиму конфіденційності. Працівники таких підрозділів проходять регулярні перевірки, укладають договори про нерозголошення та можуть бути піддані технічному моніторингу в процесі роботи із закритими даними. Установа зберігає повний

контроль над усіма етапами життєвого циклу даних: від збору до моделювання та реалізації аналітичних рішень. Відмова від аутсорсингу дозволяє знизити ризики, перетворюючи їх на конкурентні переваги:

По-перше, він дозволяє суттєво знизити ризики витоку інформації, оскільки усі дії з даними виконуються в рамках єдиного ізольованого технічного середовища.

По-друге, він сприяє гнучкості у розробці та адаптації алгоритмів до специфіки банківської діяльності, оскільки внутрішні команди краще ознайомлені зі структурою бізнес-процесів установи.

По-третє, спрощує реалізацію юридичних вимог та процедур аудиту, адже всі дані і пов'язані з ними дії зберігаються у внутрішніх реєстрах та логах систем.

Довготривала взаємодія з внутрішніми спеціалістами дозволяє фінансовим установам накопичувати стратегічний досвід, що є неможливим у випадку тимчасової взаємодії з зовнішніми підрядниками. Фахівці, які тривалий час працюють у внутрішніх аналітичних підрозділах, краще розуміють контекстні особливості поведінки клієнтів, політику банку щодо оцінки ризиків, юридичні рамки обробки даних та вимоги до звітності. Знання у свою чергу дозволяє підвищити якість аналітики, точність моделей виявлення шахрайства, а також сприяє оперативному реагуванню на нові типи загроз.

Розробляючи систему з виявлення фінансового шахрайства, реалізованого з використанням локальних обчислювальних потужностей, MongoDB може бути інтегрована як основний інструмент для локального зберігання та управління обліковими записами працівників, що мають доступ до функціональних компонентів системи. У даній архітектурі MongoDB виконує функцію бази даних авторизації, в якій зберігаються відомості про користувачів, їхні ролі, рівні доступу, історію входу та контрольні атрибути.

Доцільність використання MongoDB у якості механізму контролю доступу базується на її здатності обробляти документоорієнтовані структури даних, що дозволяє створювати гнучкі схеми представлення інформації про користувачів, які можуть бути легко модифіковані без потреби в структурних змінах бази.

Наприклад, обліковий запис може містити поля для збереження зашифрованих паролів, токенів сесії, параметрів автентифікації, журналів дій, часових міток входу та виходу із системи. Повністю ізолюючи дані авторизації в межах локального середовища, що є критично важливим для зменшення ймовірності зовнішнього несанкціонованого доступу. Крім зберігання облікових записів, MongoDB може використовуватись як бекенд для реалізації механізмів автентифікації та авторизації у сервісі. Під час спроби входу користувача, запит надсилається до бази MongoDB, де перевіряються відповідність введених облікових даних та права доступу до конкретних компонентів системи. Реалізуючи багаторівневу модель контролю доступу, що визначає, які модулі або етапи аналітичного конвеєра може переглядати або редагувати конкретний працівник.

Для обслуговування та аудиту таких процесів може застосовуватись MongoDB Compass як інструмент локального адміністрування. За його допомогою адміністратор має змогу переглядати структуру користувацьких документів, контролювати активність, виявляти аномалії в логах доступу або своєчасно блокувати облікові записи у випадку виявлення підозрілої поведінки. Всі операції відбуваються у рамках локального сегменту без потреби підключення до зовнішніх серверів, що відповідає вимогам щодо збереження конфіденційності персональних даних персоналу.

Застосування MongoDB як локальної системи зберігання та контролю доступу забезпечує централізований, ізольований та гнучко масштабований механізм управління користувачами в межах сервісу виявлення шахрайства, зводячи до мінімуму ризики порушення цілісності та безпеки критично важливих операційних процесів.

Головна ідея використання MongoDB – захист внутрішніх обчислювальних ресурсів і даних від стороннього або несанкціонованого доступу. Несанкціонований доступ може мати вигляд як:

- *Несанкціонований локальний доступ до сервісу* - у разі фізичного або віддаленого доступу до внутрішньої мережі можливе втручання в роботу сервісу

через прямий запуск компонентів, маніпуляції з конфігурацією або спроби перехоплення даних.

MongoDB, реалізуючи авторизаційні механізми на основі ролей, забезпечує контроль над тим, які користувачі мають право на доступ до конкретних частин системи, що зменшує ймовірність виконання критичних дій сторонніми особами.

- *Внутрішні загрози з боку персоналу* - працівники, які мають доступ до системи, можуть навмисно або помилково спровокувати витік інформації, зміну моделі, некоректне навчання алгоритму або видалення даних.

Зберігання всіх дій у логах MongoDB, а також обмеження прав на виконання конкретних операцій дозволяє ідентифікувати джерело деструктивної активності та локалізувати шкоду.

- *Витік або модифікація параметрів навчальних даних* - навіть анонімізовані дані, що використовуються для навчання моделей, можуть бути піддані викривленню, якщо сторонні особи отримують доступ до файлової системи або сховища.

MongoDB дозволяє зашифрувати збережені документи, використовувати контроль доступу до колекцій і вести журнал змін, що унеможливорює несанкціоновану модифікацію даних.

- *Атаки через локальні API або інтерфейси обслуговування* - локальні інтерфейси, що надають доступ до сервісних функцій, можуть бути використані для ініціації атаки або втручання у процес обробки.

Використання MongoDB для збереження ключів доступу, токенів авторизації та параметрів сесії дозволяє централізовано контролювати всі точки входу до системи та оперативно відкликати доступ у разі виявлення загрози.

- *Відсутність цілісності в обліку змін і дій користувачів* - системи без чітко реалізованого аудиту змін не здатні виявити, коли і ким були внесені критичні зміни.

MongoDB надає можливість створення незалежного аудиторського журналу з хронологією дій користувачів, що дозволяє формувати повний слід змін для подальшого аналізу інцидентів.

- Зовнішнє сканування портів і підбір облікових даних - у разі відкритого порту MongoDB без автентифікації можливе сканування злоумисниками локальної мережі з метою доступу до бази.

Реалізація захищеного з'єднання (наприклад, через TLS) та автентифікації на основі облікових записів у MongoDB дозволяє заблокувати спроби автоматизованого доступу або брутфорсу (перебір всіх можливих варіантів ключів доступу до даних)

Приклад створення захисту локальних потужностей через налаштування та створення користувачів у середовищі MongoDB показано на рисунках 3.2 – 3.4:

Першим користувачем, який повинен бути створений, є користувач із роллю root. Цей користувач має повний адміністративний доступ до всіх баз даних і функціоналу MongoDB (рис. 3.2). Створення цього користувача є критичним для встановлення початкового контролю над системою, оскільки тільки root може ініціалізувати налаштування безпеки, створювати або видаляти інші облікові записи, змінювати конфігурації доступу та запускати адміністративні команди. Без такого користувача подальше управління системою було б неможливим або неконтрольованим.

```
MongoDB Enterprise > use admin
switched to db admin
MongoDB Enterprise > db.createUser({user: "Protsenko", pwd: "12345zxc", roles: ["root"]})
Successfully added user: { "user" : "Protsenko", "roles" : [ "root" ] }
MongoDB Enterprise > _
```

Рисунок 3.2 – Користувач із роллю root.

Джерело: Розроблено автором

Другим обов'язковим користувачем є той, хто має роль userAdmin або userAdminAnyDatabase (рис. 3.3). Користувач з такою роллю має повноваження створювати, змінювати або видаляти облікові записи в межах однієї або всіх баз даних відповідно. Його основна функція полягає в адмініструванні доступу користувачів, без необхідності надання повного root-доступу. Даючи гнучке делегування прав адміністрування, обмежуючи коло осіб, які можуть впливати на глобальну конфігурацію сервера MongoDB.

```

MongoDB Enterprise > db.createUser({user: "UserAdmin", pwd: "admin12345", roles: [{db: "admin", role: "userAdmin"}]})
Successfully added user: {
  "user" : "UserAdmin",
  "roles" : [
    {
      "db" : "admin",
      "role" : "userAdmin"
    }
  ]
}
MongoDB Enterprise >

```

Рисунок 3.3 – Користувач з правами userAdmin або userAdminAnyDatabase.

Джерело: Розроблено автором

Останній обов’язковий користувач — користувач, який має виключне право доступу до персональної бази даних (рис. 3.4), створеної у попередніх етапах роботи. Такий обліковий запис обмежено лише конкретною базою даних, що мінімізує ризики витоку або модифікації критичних даних іншими користувачами. Цей підхід дозволяє реалізувати принцип найменших привілеїв, відповідно до якого кожен користувач має доступ тільки до тих ресурсів, які необхідні для виконання його функцій.

```

MongoDB Enterprise > db.createUser({user: "dbOwner", pwd: "own12345", roles: [{db: "admin", role: "dbOwner"}]})
Successfully added user: {
  "user" : "dbOwner",
  "roles" : [
    {
      "db" : "admin",
      "role" : "dbOwner"
    }
  ]
}
MongoDB Enterprise >

```

Рисунок 3.4 – Користувач, який має виключне право доступу до БД.

Джерело: Розроблено автором

Політика іменування баз даних у системі управління MongoDB має на меті забезпечення структурної впорядкованості та уникнення неоднозначностей під час зберігання, обробки та доступу до інформації. У цьому контексті використовується уніфікований стиль іменування, що передбачає застосування нижнього регістру для всіх елементів структури – імен баз даних, колекцій та полів. Для покращення читабельності й однозначної інтерпретації складених назв використовується роздільник у вигляді дефіса («-»).

У процесі реалізації політики безпеки було створено три ключові ролі користувачів, кожна з яких має чітко визначений обсяг повноважень та функціональних обов’язків. Перший користувач із роллю root має максимальні повноваження доступу до всіх баз даних системи. Його облікові дані — логін

Protsenko та пароль 12345zxc. Ця роль призначена для розробника бази даних, який здійснює повне адміністрування, включаючи конфігурацію, зміну ролей, керування доступом, резервне копіювання та інші критичні адміністративні функції.

Другий користувач має роль dbOwner та логін dbOwner, пароль own12345. Цей користувач має повний контроль лише над базою даних з назвою personal. Надання ролі власника обмежує адміністративні дії межами однієї БД, дозволяючи виконувати операції створення, редагування, видалення колекцій, керування індексами та збереженням даних у зазначеній базі. Ізолюючи доступ та підвищує рівень безпеки за рахунок обмеження впливу на інші бази даних.

Третій користувач виконує роль userAdmin, має логін UserAdmin та пароль admin12345. Цей користувач здійснює адміністративне керування обліковими записами. Його повноваження включають створення, редагування та видалення користувачів, призначення ролей, а також визначення або зміни прав доступу. Він не має доступу до змісту даних, проте відповідає за керування політикою автентифікації та авторизації користувачів, що забезпечує структурований та контрольований процес розмежування прав.

Параметри адміністрування доступу описані в таблиці 3.1:

Таблиця 3.1 – Параметри адміністрування доступу

№ з/п	Роль	Логін	Пароль	Доступна БД	Коментар
1	root	Protsenko	12345zxc	усі	Ця роль буде використовуватись розробником БД для управління всіма процесами та адміністрування
2	dbOwner	dbOwner	own12345	personal	Користувач має доступ до БД «personal» на правах її власника із відповідним набором привілеїв
3	userAdmin	UserAdmin	admin12345	admin	Виконання адміністративних завдань. Може керувати користувачами та ролями, створювати користувачів, надавати або відкликати ролі від користувачів, а також створювати або змінювати митні ролі.

Джерело: Розроблено автором

3.2 Розроблення користувацького інтерфейсу. Елементи та структура

Процес перевірки даних може бути реалізований за допомогою веб-сайту. Для прикладу було створено простий сайт, з текстовим полем для вводу транзакційних даних у текстове поле.

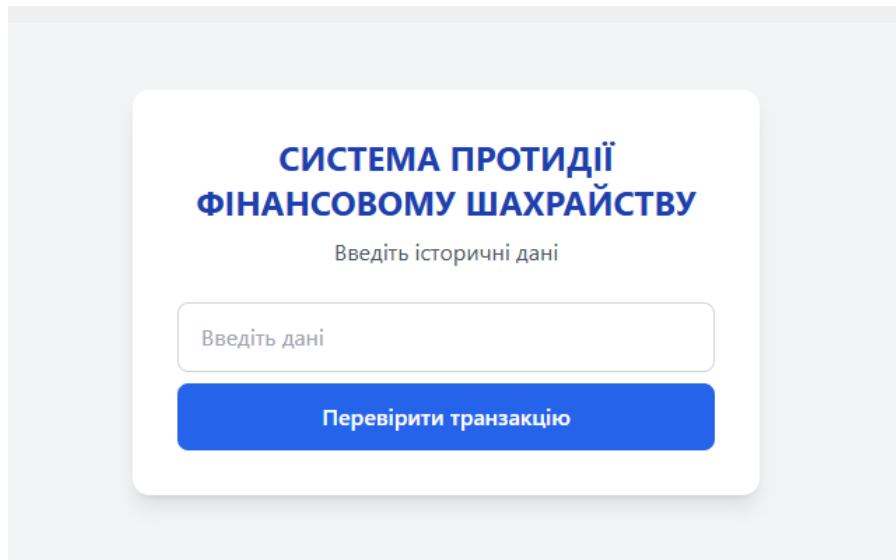


Рисунок 3.5 – Головне меню сайту перевірки транзакцій

Джерело: Розроблено автором

Наступним кроком роботи з веб-сайтом буде введення самих транзакційних даних у поле вводу.

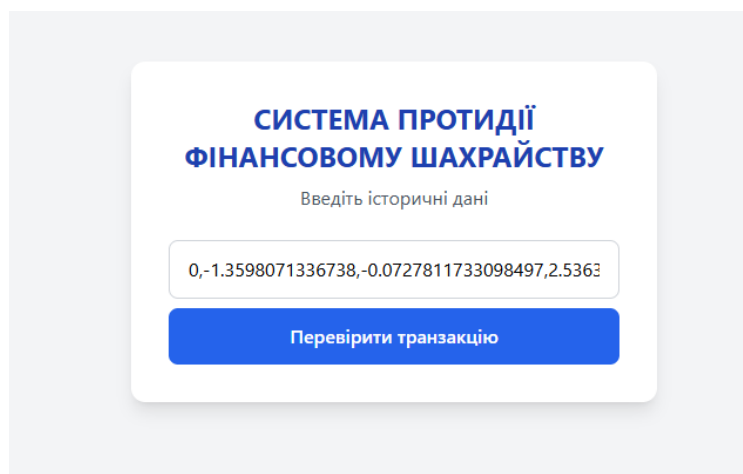
Вводити дані необхідно з набору даних, в нашому випадку CSV-файлу. Просто беремо якийсь із рядків з даними, копіюємо та вставляємо у поле вводу.

Дані для вводу мають наступний вигляд:

0,-1.3598071336738,-
 0.0727811733098497,2.53634673796914,1.37815522427443,-
 0.338320769942518,0.462387777762292,0.239598554061257,0.0986979012610507,0.
 363786969611213,0.0907941719789316,-0.551599533260813,-0.617800855762348,-
 0.991389847235408,-0.311169353699879,1.46817697209427,-
 0.470400525259478,0.207971241929242,0.0257905801985591,0.403992960255733,0.
 251412098239705,-0.018306777944153,0.277837575558899,-

0.110473910188767,0.0669280749146731,0.128539358273528,-
0.189114843888824,0.133558376740387,-0.0210530534538215,149.62,"0"

Після чого натискаємо перевірити дані (рис. 3.6):



**СИСТЕМА ПРОТИДІЇ
ФІНАНСОВОМУ ШАХРАЙСТВУ**

Введіть історичні дані

0,-1.3598071336738,-0.0727811733098497,2.5363

Перевірити транзакцію

Рисунок 3.6 – Введення транзакційних даних

Джерело: Розроблено автором

Після ініціалізації процесу перевірки шляхом натискання відповідної кнопки на інтерфейсі веб-сайту, система обробляє введені транзакційні дані за допомогою попередньо навчених інтелектуальних алгоритмів машинного навчання.

У випадку, якщо транзакція не містить аномальних показників, що потенційно можуть свідчити про шахрайську активність, система класифікує її як легітимну. Результатом такого аналізу є виведення повідомлення про низький рівень ризику та підтвердження достовірності транзакції, що свідчить про відсутність ознак фінансового шахрайства (рис. 3.7):

**СИСТЕМА ПРОТИДІЇ
ФІНАНСОВОМУ ШАХРАЙСТВУ**

Введіть історичні дані

0,-1.3598071336738,-0.0727811733098497,2.5363

Перевірити транзакцію

Транзакція легітимна
Ймовірність шахрайства: низька

Рівень довіри до транзакції

Рисунок 3.7 – Випадок підтвердження легітимності транзакції

Джерело: Розроблено автором

У протилежному випадку, якщо виявлено аномальні значення або сукупність показників, характерних для відомих патернів шахрайських дій, система класифікує транзакцію як потенційно нелігітимну. У такому випадку генерується попередження про можливе фінансове шахрайство, що супроводжується відповідним виводом у інтерфейсі користувача (рис. 3.8):

**СИСТЕМА ПРОТИДІЇ
ФІНАНСОВОМУ ШАХРАЙСТВУ**

Введіть історичні дані

0,1.19185711131486,0.26615071205963,0.166480

Перевірити транзакцію

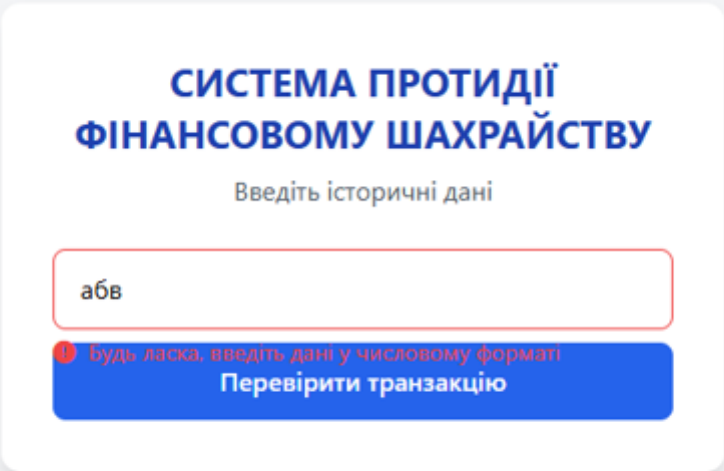
Транзакція нелігітимна
Можливе шахрайство

Рівень довіри до транзакції

Рисунок 3.8 – Випадок із виявленням підозрілої транзакції

Джерело: Розроблено автором

У системи передбачено механізм валідації формату введеної інформації. Зокрема, система орієнтована на обробку виключно числових даних, що відповідає структурі вхідних ознак, необхідних для коректної роботи алгоритмів машинного навчання. У випадку, коли користувач вводить символи, які не відповідають встановленому числовому формату, таких як літери, спеціальні символи або інші неліквідні елементи, система активує попередження з повідомленням про некоректність вводу (рис. 3.9):



**СИСТЕМА ПРОТИДІЇ
ФІНАНСОВОМУ ШАХРАЙСТВУ**

Введіть історичні дані

абв

❗ Будь ласка, введіть дані у числовому форматі

Перевірити транзакцію

Рисунок 3.9 – Демонстрація внесення даних у невстановленому форматі

Джерело: Розроблено автором

3.3 Проєктування забезпечувальних підсистем рекомендаційної системи та її реалізації

3.3.1 Інформаційне забезпечення.

Інформаційне забезпечення системи включає сукупність класифікованих і формалізованих даних, які необхідні для реалізації її функціонального призначення. В рамках розробленої системи запобігання фінансового шахрайству та видачі результатів аналізу транзакційних даних інформаційне забезпечення охоплює наступні основні компоненти: вхідну, проміжну та вихідну інформацію:

Основним джерелом даних виступає публічний датасет транзакцій з кредитних карток, в якому кожна транзакція представлена сукупністю числових значень, включаючи змінні V1–V28, отримані внаслідок перетворення головних компонент (PCA), а також змінні Time, Amount та Class. Змінна Class є міткою цільового класу, де «1» позначає виявлене шахрайство, а «0» — справжню транзакцію. Вхідні дані мають строго структурований вигляд, що відповідає вимогам алгоритмів машинного навчання до формату ознак. Всі дані анонімізовані з метою забезпечення захисту персональної інформації користувачів.

Вхідна інформація в системі формується шляхом передачі одного або кількох рядків CSV-файлу через спеціально розроблений веб-інтерфейс. У процесі перевірки система отримує числовий вектор, що описує конкретну транзакцію, після чого виконується обчислення ймовірності шахрайської активності за допомогою попередньо натренованої моделі машинного навчання. У разі порушення вхідного формату (наприклад, введення текстових символів замість чисел) система активує механізм валідації і виводить відповідне попередження користувачу, що гарантує коректність обробки та захист від зловмисного вводу.

Дані, що використовуються для тренування та тестування моделей у системах виявлення фінансового шахрайства, здебільшого формуються на основі публічно доступних наборів, розміщених на спеціалізованих онлайн-платформах. Однією з найбільш відомих платформ є ресурс [kaggle.com](https://www.kaggle.com/), який функціонує як репозиторій дослідницьких матеріалів, алгоритмів, навчальних наборів даних і

супутнього аналітичного коду. На цій платформі дослідники та практики публікують власні роботи, які можуть бути використані іншими розробниками за умови дотримання умов використання. Датасети, що розміщуються на подібних платформах, є загальнодоступними, однак їхнє використання в академічному або науково-дослідницькому контексті потребує обов'язкового цитування першоджерела. Використання таких ресурсів дозволяє пришвидшити процес прототипування моделей машинного навчання, оскільки надає вже структуровані та часто попередньо оброблені дані, які придатні для безпосереднього навчання моделей. У візуальному представленні структури використаного набору даних (рис. 3.10) демонструється приклад CSV-файлу з платформи Kaggle, який був використаний у процесі навчання моделі класифікації транзакцій. Такий формат зручний для обробки, інтеграції у середовище машинного навчання та подальшого використання в сервісах, орієнтованих на виявлення аномальної або шахрайської активності.

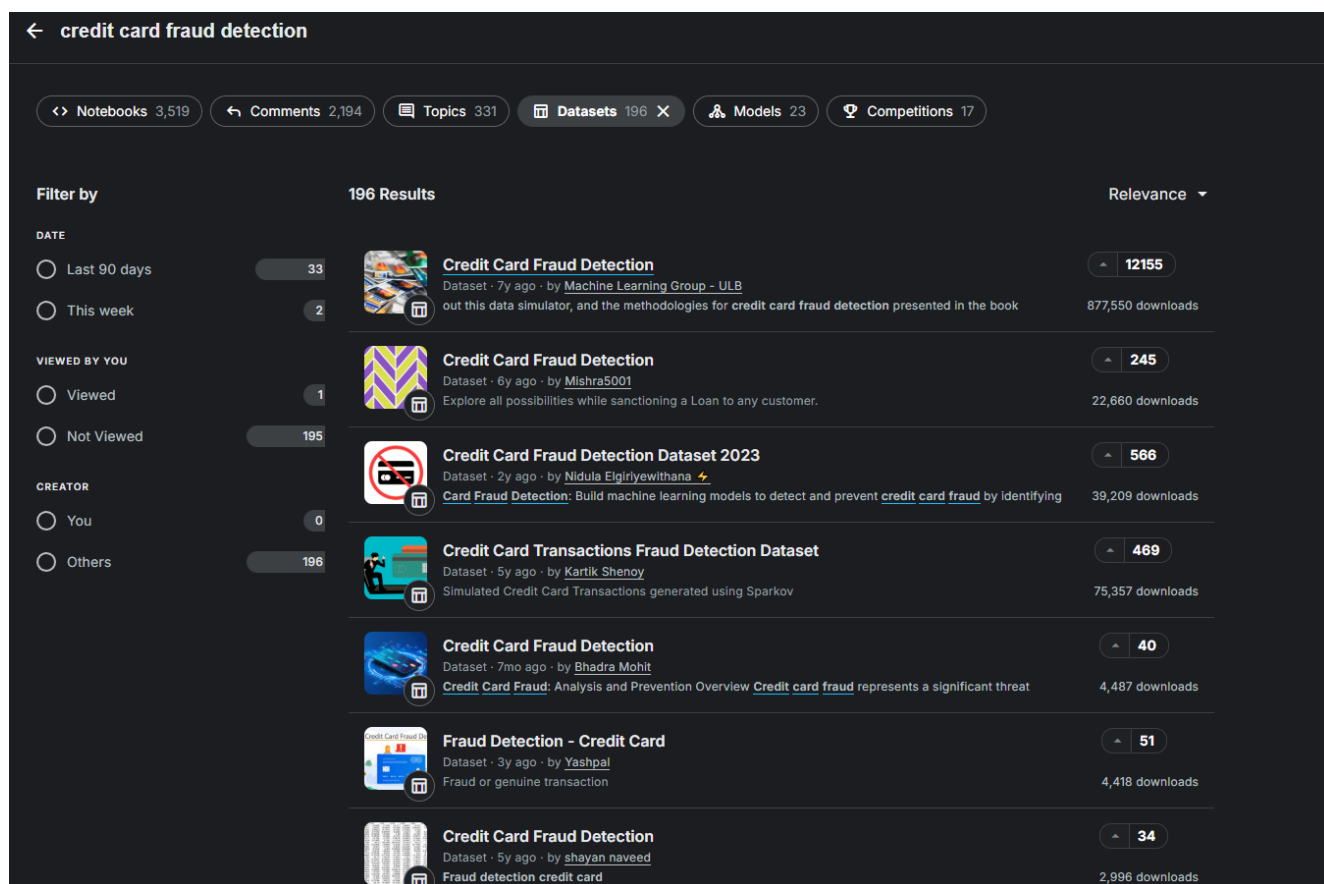


Рисунок 3.10 – Перелік загальнодоступних датасетів із сайту kaggle.com

Джерело: Загальнодоступні датасети з мережі Інтернет

Проміжна інформація формується в результаті обчислень всередині системи і включає в себе внутрішні оцінки ризику, класифікаційні мітки, метрики точності, а також логування користувацьких запитів для подальшого аудиту. Вся ця інформація зберігається тимчасово або постійно у відповідних локальних сховищах.

Вихідна інформація системи — результат класифікації транзакції: повідомлення про те, що операція є або не є шахрайською, з відповідною оцінкою ризику. Ця інформація виводиться в інтерфейс користувача для оперативного реагування та прийняття рішень.

3.3.2 Програмне забезпечення.

Програмне забезпечення системи виявлення фінансового шахрайства включає комплекс інструментів, фреймворків, мов програмування та алгоритмів машинного навчання, необхідних для реалізації процесів збирання, обробки, аналізу та класифікації транзакційних даних.

Початковий етап розробки системи передбачає створення і тестування моделей машинного навчання у середовищі Jupyter Notebook, яке використовується як зручне інтерактивне середовище для прототипування, візуалізації результатів та виконання поетапного аналізу транзакційних даних. Моделі навчаються на основі CSV-даних, попередньо підданих нормалізації та перетворенню ознак [11].

Після завершення тренування моделі експортуються та вбудовуються у веб-серверне середовище, реалізоване за допомогою фреймворку Flask, що дозволяє сформувати REST API для взаємодії з клієнтським інтерфейсом. Розгортання сервісу відбувається локально на локальних обчислювальних потужностях, що забезпечує автономність роботи системи та знижує ризики, пов'язані з передачею конфіденційних даних через публічні інтерфейси.

Альтернативою локальному розгортанню системи виявлення фінансового шахрайства є використання хмарних або віддалених платформ, зокрема Streamlit у

поєднанні з Ngrok або Gitpod. Такий підхід дозволяє реалізувати повноцінну взаємодію користувача із системою без необхідності розгортання серверного середовища на локальній машині.

Streamlit — високорівневий фреймворк для швидкої побудови веб-інтерфейсів на Python. Його використання дозволяє інтегрувати модель машинного навчання в інтуїтивно зрозумілий інтерфейс із мінімальними зусиллями. Завдяки автоматизованій генерації елементів управління, Streamlit забезпечує прямий доступ до вхідних даних, запуску класифікації, виводу результатів та побудови графіків, що робить його зручним для демонстрацій, дослідницьких прототипів і навчальних цілей [18].

Для публічного доступу до застосунку, розгорнутого локально, застосовується Ngrok — інструмент, що створює безпечний тунель між локальним середовищем і глобальною мережею. Ngrok дозволяє опублікувати локальний веб-сервер без необхідності мати статичну IP-адресу або змінювати налаштування маршрутизатора. У поєднанні з Streamlit забезпечуючи тимчасовий, але повноцінний веб-доступ до системи виявлення шахрайства з будь-якого пристрою, підключеного до Інтернету [19].

Ще одним варіантом є використання Gitpod — веб-орієнтованого середовища розробки, яке надає віртуальну машину з попередньо встановленими інструментами розробника. Gitpod дозволяє розгорнути код безпосередньо з репозиторію, запускати Jupyter Notebook, Flask або Streamlit-сервери, а також працювати із зовнішніми API. Даючи змогу протестувати або розгорнути систему без необхідності локального встановлення компонентів, що підвищує портативність і зменшує вимоги до клієнтських пристроїв. Але для розгортання проекту на сервісі Gitpod необхідна прив'язка до самого проекту, який необхідно викласти у середовищі GitHub [20].

Незалежно від варіанту розгортання, система реалізує веб-інтерфейс, що дозволяє користувачеві взаємодіяти із сервісом через веб-браузер. Основні компоненти веб-інтерфейсу реалізовані з використанням HTML/CSS. HTML

використовується для розмітки структури сторінки, створення текстових полів для введення транзакційних даних, кнопок управління та повідомлень. CSS використовується для стилізації елементів інтерфейсу, забезпечуючи візуальну доступність і зручність користування системою. Процес перевірки даних реалізовано таким чином: користувач вставляє числові значення однієї транзакції з CSV-файлу у відповідне текстове поле веб-сторінки. Натискання кнопки надсилає ці дані на сервер, де вони обробляються попередньо навченою моделлю. У разі виявлення допустимих значень, система виконує класифікацію і відображає результат — легітимна або потенційно шахрайська транзакція. Якщо ж дані мають некоректний формат (наприклад, містять символи, літери або пробіли), активується механізм валідації, і користувач отримує повідомлення про помилку (Розділ 3.2).

3.3.4 Організаційно-економічне забезпечення

Подібний проект з виявлення фінансового шахрайства має потенціал у сфері безпеки банківських та фінансових транзакцій. У сучасних умовах фінансові установи зацікавлені в інтеграції автоматизованих систем виявлення аномалій, але часто обмежують доступ до своїх внутрішніх даних і не передають критичні функції стороннім підрядникам. Тому проекти, реалізовані як демонстраційні рішення з використанням відкритих, знеособлених даних, є цінними в контексті моделювання реальних сценаріїв і можуть бути адаптовані для внутрішнього використання в конкретній установі.

Корисність системи підвищується завдяки дотриманню вимог внутрішньої політики безпеки банку за рахунок розгортання її локально без перенесення даних у хмарні сервіси. Таке рішення може успішно інтегруватися у внутрішню ІТ-інфраструктуру, зберігаючи контроль над усіма етапами обробки фінансової інформації, якщо модифікувати модель під реальні дані конкретної організації. Даючи можливість внутрішнього використання або додаткової комерціалізації без залучення зовнішніх постачальників послуг.

Створення функціонального прототипу з компетенціями в аналітиці, розробці моделей машинного навчання, проектуванні інформаційних систем та безпеці є одним з найбільш реалістичних способів реалізації такого проекту, враховуючи обмеження доступу до внутрішніх банківських систем. В рамках галузевих конкурсів, стажування або цільового працевлаштування у фінансовому секторі такий прототип можна показати роботодавцю, учаснику технічної співбесіди або хакатону.

Досвід побудови систем виявлення шахрайства на основі відкритих даних свідчить про наявність у кандидата прикладних навичок аналізу транзакційних даних, формалізації бізнес-задач, реалізації алгоритмів класифікації та створення безпечного середовища для роботи з чутливою інформацією. Претендуючи на роботу в аналітичних підрозділах, підрозділах ризиків або інформаційної безпеки банківських установ, ви отримуєте конкурентну перевагу.

Висновки до розділу 3

Основні проєктні ідеї, застосовані при створенні демонстраційної системи виявлення фінансового шахрайства, були детально висвітлені в третій частині. Підкреслюючи реальні умови роботи у фінансовому середовищі, технічні, функціональні та організаційні елементи розробки демонструють цілісність та повноту підходу.

Перш за все, реалізована за допомогою MongoDB, створена модель бази даних призначена лише для того, щоб запропонувати локальний механізм аутентифікації та авторизації користувачів системи. Хоча вона не використовується безпосередньо для виявлення шахрайства, вона є досить важливою для захисту доступу до локальних обчислювальних потужностей. База даних реалізує багаторівневі ролі доступу, зашифроване зберігання облікових записів та контроль дій користувачів. Мінімізуючи ризики несанкціонованого втручання, витоку даних та порушення цілісності навчальних наборів. Таке архітектурне рішення дозволяє реалізувати концепцію ізольованого середовища, в

межах якого забезпечується повний контроль над доступом до компонентів системи.

По-друге, розроблено простий веб-інтерфейс користувача, який дозволяє вводити структуровані транзакційні дані для взаємодії з системою. Інтерфейс пропонує спосіб перевірки формату введених даних, сповіщення користувача про будь-яку підозрілу активність та результат перевірки транзакції. Гарантуючи зручність використання системи в дослідницьких або демонстраційних цілях і дозволяє проводити швидку первинну перевірку даних без використання складних інструментів.

По-третє, було розроблено та впроваджено інформаційне, програмне та організаційне забезпечення - всі важливі підсистеми системи штучного інтелекту. Анонімізовані набори даних для навчання моделей машинного навчання є частиною інформаційної підсистеми. Фреймворки Python Flask та Streamlit слугують основою для реалізації програмної підсистеми, що дозволяє проводити інтелектуальний аналіз локально. Ізоляція системи від зовнішніх джерел, відмова від аутсорсингу та дотримання внутрішньої політики безпеки, яка відповідає практикам фінансового сектору, стали можливими завдяки організаційній підтримці.

У завершення можна сказати, що проєктні рішення відповідають базовим потребам системи із виявлення шахрайства, який здатен задовольнити очікування та забезпечити виконання поставлених цілей.

ВИСНОВКИ

Оскільки фінансові процеси стають дедалі більш діджиталізованими, розробка веб-систем виявлення шахрайства набуває все більшого значення. Шахрайство з боку третіх осіб стає все більш імовірним внаслідок активного переходу споживачів до онлайн-платежів, електронної комерції та мобільного банкінгу. Сучасні фінансові транзакції часто відбуваються без фізичної присутності клієнта, що відкриває нові можливості для злочинної діяльності, особливо на етапах обробки та передачі даних.

У таких ситуаціях розробка інтелектуальних систем, здатних автоматично аналізувати транзакції та виявляти аномалії, є важливим аспектом інформаційної безпеки. Використання веб-інтерфейсів для реалізації таких систем дозволяє гарантувати їх зручність використання, ефективність взаємодії з користувачем і сумісність з поточною інфраструктурою організації. Веб-метод полегшує масштабування рішень і застосування їх на відкритих платформах для оцінки ризиків, а також у внутрішніх банківських системах.

Забезпечити належний рівень безпеки при збереженні фінансової та персональної інформації можна, розробивши систему з локальним розгортанням та ізольованим операційним середовищем. Інтегруючи механізми машинного навчання у веб-інтерфейс, транзакції можна обробляти та аналізувати автоматично, зменшуючи потребу у людському втручанні та пришвидшуючи виявлення потенційно шахрайської діяльності.

У першому розділі було проведено ґрунтовний аналіз теми електронної комерції з акцентом на методах створення складних систем запобігання фінансовому шахрайству. Надзвичайно важливо створювати гнучкі та адаптивні моделі, які враховують динаміку цифрового середовища та зростання кількості онлайн-транзакцій, особливо в умовах пандемії COVID-19, оскільки аналіз прикладів топових компаній підтвердив відсутність уніфікованих рішень.

У другому розділі розроблено структуру системи виявлення шахрайських транзакцій на основі машинного навчання, а також обговорено концепції обробки даних, стратегії навчання та вирішення проблеми дисбалансу класів. Розглянуто

алгоритмічні методи, які дозволяють враховувати часові залежності та поведінкові особливості користувачів - обидва ці фактори є критично важливими для підвищення точності виявлення аномалій, - а також технічні аспекти побудови моделі.

У третьому розділі було реалізовано важливі проектні рішення, такі як побудова інформаційної, програмної та організаційної інфраструктури системи, розробка веб-інтерфейсу для введення та обробки транзакційних даних, а також створення моделі бази даних авторизації в MongoDB. Доступ на основі ролей, анонімізація даних, ізоляція системи та локальне розгортання гарантують дотримання правил безпеки та гнучкість у реагуванні на обставини фінансової установи.

Відокремлюючи конфіденційні дані від зовнішніх мереж і зменшуючи можливість витоку інформації, встановлення локальної системи виявлення фінансового шахрайства забезпечує вищий рівень безпеки обробки конфіденційних даних. Фінансові установи можуть зберегти повний контроль над кожним етапом життєвого циклу даних, від збору до аналізу, відмовившись від використання хмарних рішень та аутсорсингу. Важливо саме для галузей, що обробляють конфіденційні дані клієнтів. Локальне розгортання полегшує дотримання внутрішніх політик безпеки, зокрема тих, що регулюють обробку та зберігання персональних даних, а також аудит.

Створення такої системи в рамках виконаної роботи було розумним і вдалим кроком, оскільки дозволило реалізувати повноцінну демонстраційну модель, а також врахувати реальні обмеження, з якими стикається фінансова індустрія. Веб-інтерфейс введення даних, локальна база даних авторизації та інструменти машинного навчання для аналізу транзакцій були серед технічних рішень, використаних для створення комплексної, функціонально завершеної системи, яка може бути адаптована до конкретних обставин. Таким чином, проєкт робить внесок у розвиток методів автономного та безпечного виявлення шахрайських дій у цифровій фінансовій інфраструктурі, що має як дослідницьке, так і практичне значення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Pahwa, A., Raj, T. Electronic Payments: Transformation in Payment Style and its Growth in India [Електронний ресурс]. – 2022. – Режим доступу: <https://rrjournals.com/index.php/rrjm/article/view/316>.
2. Chalise, A.K. Engineering and Advancement of Digitization through Unified Payment Interface [Електронний ресурс]. – 2023. – Режим доступу: <https://journaljerr.com/index.php/JERR/article/view/845>.
3. Coskun, M., Saygili, E., Karahan, M.O. Exploring Online Payment System Adoption Factors in the Age of COVID-19—Evidence from the Turkish Banking Industry [Електронний ресурс]. – Режим доступу: <https://www.mdpi.com/2227-7072/10/2/39>.
4. Ставерська, Т.О., Літвінова, Ю.І. Еволюція платіжних систем: інновації на шляху до цифрового майбутнього [Електронний ресурс]. – 2024. – Режим доступу: <https://economyandsociety.in.ua/index.php/journal/article/download/3625/3555>.
5. Aguilar, A., Frost, J., Guerra, R., Kamin, S., Tombini, A. Digital payments, informality and economic growth [Електронний ресурс]. – 2024. – Режим доступу: <https://www.bis.org/publ/work1196.htm>.
6. Zandi, M., Singh, A. The Impact of Payment Cards on Economic Growth [Електронний ресурс]. – 2021. – Режим доступу: <https://usa.visa.com/content/dam/VCOM/regional/na/us/visa-everywhere/documents/the-impact-of-payment-cards-on-economic-growth.pdf>.
7. Jaiswal, B., Singh, A. Money in the Digital Age: Exploring the Dynamics of the Digital Economy and Online Payment System Advancements [Електронний ресурс]. – 2023. – Режим доступу: <https://www.ijfmr.com/papers/2023/6/10962.pdf>.
8. Tudor, C. Integrated Framework to Assess the Extent of the Pandemic Impact on the Size and Structure of the E-Commerce Retail Sales Sector and Forecast Retail Trade E-Commerce [Електронний ресурс]. – 2022. – Режим доступу: <https://www.mdpi.com/2079-9292/11/19/3194>.

9. Global e-commerce jumps to \$26.7 trillion, fuelled by COVID-19 [Электронный ресурс]. – 2021. – Режим доступа: <https://news.un.org/en/story/2021/05/1091182>.
10. Perez, S. COVID-19 pandemic accelerated shift to e-commerce by 5 years, new report says [Электронный ресурс]. – 2020. – Режим доступа: <https://www.weforum.org/stories/2020/08/covid19-pandemic-social-shift-ecommerce-report>.
11. Credit Card Fraud Detection Dataset [Электронный ресурс]. – Режим доступа: <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data>.
12. Isangediok M., Gajamannage K. Fraud Detection Using Optimized Machine Learning Tools Under Imbalance Classes. 2022 [Электронный ресурс]. – Режим доступа: <https://arxiv.org/pdf/2209.01642>.
13. Seshagiri S., Prema K.V. Efficient Handling of Data Imbalance in Health Insurance Fraud Detection Using Meta-Reinforcement Learning. 2025 [Электронный ресурс]. – Режим доступа: <https://ieeexplore.ieee.org/document/10858064>.
14. Černevičienė J., Kabašinskas A. Explainable artificial intelligence (XAI) in finance: a systematic literature review. 2024 [Электронный ресурс]. – Режим доступа: <https://link.springer.com/article/10.1007/s10462-024-10854-8>.
15. McCahery J.A., de Roode A. Governance of Financial Services Outsourcing: Managing Misconduct and Third-Party Risks. 2018 [Электронный ресурс]. – Режим доступа: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3241196.
16. Jonker N., van der Cruisen C., Bijlsma M., Bolt W. Pandemic payment patterns [Электронный ресурс]. – Режим доступа: <https://www.sciencedirect.com/science/article/pii/S0378426622001893>.
17. Wani B.A., MohamadAli N.A.B.T. COVID-19 Sparked the E-Commerce Revolution; Some Benefited, While Others Left in Cold. 2022 [Электронный ресурс]. – Режим доступа: <https://aircconline.com/ijcsit/V14N3/14322ijcsit09.pdf>.
18. Streamlit [Электронный ресурс]. – Режим доступа: <https://streamlit.io/>.

19. Ngrok.com [Електронний ресурс]. – Режим доступу: <https://dashboard.ngrok.com/>.
20. Gitpod.io [Електронний ресурс]. – Режим доступу: <https://gitpod.io/>.
21. Credit card fraud detection. Kaggle.com [Електронний ресурс]. – Режим доступу: <https://www.kaggle.com/search?q=credit+card+fraud+detection+in%3Adatasets>.
22. Костецький В.В., Бутов А.М. Управління фінансовими потоками підприємств з позиції фінансової логістики. 2018 [Електронний ресурс]. – Режим доступу: <http://dspace.wunu.edu.ua/handle/316497/41953>.
23. Pasupathinathan V., Vijayakrishnan L., et al. Formal analysis of card-based payment systems in mobile devices // Proceedings of the 2006 Australasian workshops on Grid computing and e-research – Volume 54. – 2006. – P. 213–220 [Електронний ресурс]. – Режим доступу: <http://vkrishnan.com.s3.amazonaws.com/resources/publications/2006AISWNS.pdf>.
24. Aydin G., Burnaz S. Adoption of mobile payment systems: a study on mobile wallets // Journal of Business Economics and Finance. – 2016. – Vol. 5, No. 1. – P. 73–92 [Електронний ресурс]. – Режим доступу: <https://dergipark.org.tr/en/pub/jbef/article/360219>.
25. Калита О.В., Єркес О.Є. Діджитал-тренди банківських процесів в умовах пандемії COVID-19 // Фінансова система України як складова інституційних перетворень економіки: тези доп. всеукр. наук.-практ. конф. (Київ, 14–15 квітня 2021 р.). – Київ: КНТЕУ, 2021. – 147 с. [Електронний ресурс].
26. Куцевол М.А., Шевченко-Наумова О.А. Поняття та економічна природа криптовалюти [Електронний ресурс] // Вісник Київського національного економічного університету. – 2015. – № 8080. – С. 79–85. – Режим доступу: <http://ir.kneu.edu.ua>.
27. Subaramaniam K., et al. The impact of E-Wallets for current generation // Journal of Advanced Research in Dynamical and Control Systems. – 2020. – Vol. 12, No. 1. – P. 751–759 [Електронний ресурс].

28. Mbiti I., Weil D.N. Mobile banking: The impact of M-Pesa in Kenya // African successes. Volume III: Modernization and development. – University of Chicago Press, 2015. – P. 247–293 [Електронний ресурс].
29. Shahid M., et al. Exploring the determinants of adoption of Unified Payment Interface (UPI) in India: A study based on diffusion of innovation theory // Digital Business. – 2022. – Vol. 2, No. 2. – Article ID 100040 [Електронний ресурс].
30. Здолбіцька Н., Жигаревич О. Сучасні способи автентифікації та захист на основі токенів // Прикладні проблеми комп'ютерних наук, безпеки та математики. – 2024. – № 3. – С. 4–11 [Електронний ресурс].
31. Kumari B., Kaur J., Swami S. System dynamics approach for adoption of artificial intelligence in finance // Advances in Systems Engineering: Select Proceedings of NSC 2019. – Singapore: Springer, 2021. – P. 555–575 [Електронний ресурс].
32. Mitlöhner J., et al. Characteristics of open data CSV files // 2016 2nd International Conference on Open and Big Data (OBD). – IEEE, 2016. – P. 72–79 [Електронний ресурс].
33. Bao W., Lianju N., Yue K. Integration of unsupervised and supervised machine learning algorithms for credit risk assessment // Expert Systems with Applications. – 2019. – Vol. 128. – P. 301–315 [Електронний ресурс]. – Режим доступу: <https://doi.org/10.1016/j.eswa.2019.02.033>.
34. Tynymbayev A., Baisholanova K., Khikmetov A., Mambetov S. Application of logistic regression to detect an attack on financial monitoring // Bulletin of the National Engineering Academy of the Republic of Kazakhstan. – 2024. – Vol. 93, No. 3. – P. 245–255 [Електронний ресурс]. – Режим доступу: <https://doi.org/10.47533/2024.1606-146x.64>.
35. Садома Ю.В. Застосування альтернативних методів для моделювання і прогнозування фінансових процесів у банківській сфері. – 2021.
36. Onyeoma C.F., Rafiq H., Jeremiah D., Ta V.T., Usman M. Credit Card Fraud Detection Using Deep Neural Network with Shapley Additive Explanations // 2024 International Conference on Frontiers of Information Technology (FIT). – 2024. –

Р. 1–6 [Електронний ресурс]. – Режим доступу:

<https://doi.org/10.1109/fit63703.2024.10838456>.

37. Josyula H.P. Fraud Detection in Fintech Leveraging Machine Learning and Behavioral Analytics. – 2023.

38. Dornadula V.N., Geetha S. Credit card fraud detection using machine learning algorithms // *Procedia Computer Science*. – 2019. – Vol. 165. – P. 631–641 [Електронний ресурс].

39. Bansal A., Garg H. An efficient techniques for fraudulent detection in credit card dataset: a comprehensive study // *IOP Conference Series: Materials Science and Engineering*. – 2021. – P. 012181 [Електронний ресурс].

40. Bhattacharyya S., et al. Data mining for credit card fraud: A comparative study // *Decision Support Systems*. – 2011. – Vol. 50, No. 3. – P. 602–613 [Електронний ресурс].

41. Snader J.C. VPNs Illustrated: Tunnels, VPNs, and IPsec. – Addison-Wesley Professional, 2015.

42. Byun J.-W., et al. Secure anonymization for incremental datasets // *Secure Data Management: Third VLDB Workshop, SDM 2006, Seoul, Korea, September 10–11, 2006. Proceedings*. – Springer Berlin Heidelberg, 2006. – P. 48–63 [Електронний ресурс].

43. Barako D.G., Gatere P.K. Outsourcing practices of the Kenyan Banking sector // *African Journal of Accounting, Economics, Finance and Banking Research*. – 2008. – Vol. 2, No. 2.

44. Проценко Б. П. Використання штучного інтелекту для обробки великих даних: інноваційні рішення та виклики. Режим доступу: <https://drive.google.com/file/d/1A9VLK0F3ICtKkcX-5-od6rI3YgWeeTPf/view>, 2024, 171-175.

ДОДАТКИ

Додаток А

```

import io import pickle
import pandas as pd
from django.shortcuts import render, redirect, get_object_or_404 from django.http.response import HttpResponseRedirect, HttpResponse, JsonResponse
from django.contrib.auth import get_user_model, authenticate from django.contrib.auth import authenticate, login, logout from django.contrib import messages
from .models import DataFileUpload import requests
from django.http import JsonResponse
from sklearn.linear_model import LogisticRegression from sklearn.model_selection import train_test_split from sklearn.preprocessing import StandardScaler from sklearn.metrics import precision_score

def base(request):
    return render(request, 'homeApp/landing_page.html')

def upload_credit_data(request):
    return render(request, 'homeApp/upload_credit_data.html')
def prediction_button(request, id):
    return render(request, 'homeApp/fraud_detection.html', {'id': id})

def reports(request): all_data_files_objs=DataFileUpload.objects.all()
    return render(request, 'homeApp/reports.html', {'all_files': all_data_files_objs})
def enter_form_data_manually(request):
    return render(request, 'homeApp/enter_form_data_manually.html')

def predict_data_manually(request):
    return render(request, 'homeApp/predict_data_manually.html')

def add_files_single(request, id):
    return render(request, 'homeApp/add_files_single.html', {'id': id})
def predict_csv_single(request, id): if request.method == 'POST':
    try:
        obj = DataFileUpload.objects.get(id=id)
        # Deserialize the stored model model_binary = obj.trained_model_data loaded_model = pickle.loads(model_binary)

        # Deserialize X_test and y_test X_test = pickle.loads(obj.x_test_data) y_test = pickle.loads(obj.y_test_data)
        # Predict on X_test
        y_pred = loaded_model.predict(X_test)
        # Calculate precision
        precision = precision_score(y_test, y_pred)
        dataframe = pd.read_csv(request.FILES['actual_file_name']) prediction = loaded_model.predict(dataFrame)

        status = "Non-Fraudulent Transaction" if prediction[0] == 1:
        status = "Fraudulent Transaction"

        context = { 'status': status,
                    'precision': f'{precision:.2%}', # display it as a percentage 'data': dataframe.iloc[0]
                    }
        return render(request, 'homeApp/predict_csv_single.html', context) except:
        messages.warning(request, "Invalid/wrong format. Please upload File.") return redirect(f'/add_files_single/{id}')

def add_files_multi(request, id):

```

Лістинг views.py

```

return render(request, 'homeApp/add_files_multi.html', {'id': id})
def predict_csv_multi(request, id): if request.method == 'POST':
    try:
        obj = DataFileUpload.objects.get(id=id)
        # Deserialize the stored model model_binary = obj.trained_model_data loaded_model = pickle.loads(model_binary)

        # Deserialize X_test and y_test X_test = pickle.loads(obj.x_test_data) y_test = pickle.loads(obj.y_test_data)
        # Predict on X_test
        y_pred = loaded_model.predict(X_test)

        # Calculate precision
        precision = precision_score(y_test, y_pred)
        dataframe = pd.read_csv(request.FILES['actual_file_name']) predictions = loaded_model.predict(dataFrame)

        statuses = ["Fraudulent Transaction" if pred == 1 else "Non-Fraudulent Transaction" for pred in predictions]
        combined_data = []
        for record, status in zip(dataFrame.to_dict(orient='records'), statuses): combined_data.append({
            'record': record, 'status': status
        })
        context = {
            'precision': f'{precision:.2%}', # Precision as a percentage 'combined_data': combined_data,
        }
        return render(request, 'homeApp/predict_csv_multi.html', context) except:
        messages.warning(request, "Invalid/wrong format. Please upload File.")
        return redirect(f'/add_files_multi/{id}')

def account_details(request):
    return render(request, 'homeApp/account_details.html') def change_password(request):
    return render(request, 'homeApp/change_password.html') def analysis(request, id):
    obj = DataFileUpload.objects.get(id=id) df = pd.read_csv(obj.actual_file.path)
    # Empty DataFrame Columns
    empty_columns = len(df.columns[df.isnull().all()]).tolist()

    # Data Shape data_shape = df.shape
    # Unique Target Values
    # Assuming 'Class' is your target column unique_targets = df['Class'].unique().tolist()
    # Percentages
    percent_no_problem = 100 * (df[df['Class'] == 0].shape[0] / df.shape[0]) percent_problem = 100 * (df[df['Class'] == 1].shape[0] / df.shape[0])

    # Null values check
    has_null = df.isnull().any().any()
    # Transactions
    fraud_transactions = df[df['Class'] == 1] normal_transactions = df[df['Class'] == 0]

    fraud_shape = fraud_transactions.shape normal_shape = normal_transactions.shape
    # Return or pass these to your template as context context = {
        'data_shape': data_shape, 'unique_targets': unique_targets,
        'percent_no_problem': round(percent_no_problem, 3),
        'percent_problem': round(percent_problem, 3), 'has_null': has_null,
        'fraud_shape': fraud_shape, 'normal_shape': normal_shape
    },

```

Лістинг views.py

```

return render(request, 'homeApp/analysis.html', context) def view_data(request, id):
obj = DataFileUpload.objects.get(id=id) df = pd.read_csv(obj.actual_file.path) columns = df.columns.tolist()
return render(request, 'homeApp/view_data.html', {'id': id, 'columns': columns})
def delete_data(request, id): obj=DataFileUpload.objects.get(id=id) obj.delete()
messages.success(request, "File Deleted successfully", extra_tags = 'alert alert-success alert-dismissible show')
return HttpResponseRedirect('/reports') def upload_data(request):
if request.method == 'POST':
data_file_name = request.POST.get('data_file_name') try:
actual_file = request.FILES['actual_file_name'] data = pd.read_csv(actual_file)
# Assuming last column is the target variable and others are features X = data.iloc[:, :-1].values
y = data.iloc[:, -1].values
# Splitting the dataset into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Feature scaling
sc = StandardScaler()
X_train = sc.fit_transform(X_train) X_test = sc.transform(X_test)
# Train the Logistic Regression model
classifier = LogisticRegression(random_state=0) classifier.fit(X_train, y_train)
# Serialize the model
serialized_model = pickle.dumps(classifier) serialized_x_test = pickle.dumps(X_test) serialized_y_test = pickle.dumps(y_test)

description = request.POST.get('description')
DataFileUpload.objects.create( file_name=data_file_name, actual_file=actual_file, description=description, trained_model_data=serialized_model, x_test_data=serialized_x_test, y_test_data=serialized_y_test
)

messages.success(request, "File Uploaded successfully", extra_tags = 'alert alert-success alert-dismissible show')
return HttpResponseRedirect('/reports')

File.")
except:
messages.warning(request, "Invalid/wrong format. Please upload return redirect('/upload_credit_data')

def retrieve_data_by_id(request, id):
obj = DataFileUpload.objects.get(id=id) df = pd.read_csv(obj.actual_file.path)
# Receive parameters from DataTables on the frontend draw = int(request.GET.get('draw', 1))
start = int(request.GET.get('start', 0)) length = int(request.GET.get('length', 10))
# Paginate the data from the CSV using start and length paginated_df = df.iloc[start:start+length].reset_index() paginated_df['index'] = paginated_df['index'] + 1 + start
# Convert the paginated data to a list of lists data = paginated_df.values.tolist()
# Return a JSON response suitable for DataTables return JsonResponse({
'draw': draw, 'recordsTotal': len(df),
'recordsFiltered': len(df), # In case you add server-side filtering later on

```

Лістинг views.py

```

def userLogout(request): try:
del request.session['username'] except:
pass logout(request)
return HttpResponseRedirect('/')

def login2(request): data = {}
if request.method == "POST":
username = request.POST['username'] password = request.POST['password']
user = authenticate(request, username=username, password=password) print(user)
if user:
login(request, user)
return HttpResponseRedirect('/')

else:
data['error'] = "Username or Password is incorrect" res = render(request, 'homeApp/login.html', data) return res
else:
return render(request, 'homeApp/login.html', data)

#started here:
def about(request):
return render(request, 'homeApp/about.html')
def dashboard(request):
return render(request, 'homeApp/dashboard.html')

def verify_account_details(request):
return render(request, 'homeApp/verify_account_details.html')

# this makes the api call
def submit_verify_account_details(request):
url = "http://127.0.0.1:8080/api/v1/fraud/store"

payload = {
"account_no": request.POST['account_no'], "phone": request.POST['phone'],
"email": request.POST['email'], "description": request.POST['description'],
}
files = {
"proof": (request.FILES['proof'].name, request.FILES['proof']),
}
headers = { "Authorization": "Bearer
sk_test_b753e53c845f53c2923fdf3d818b9e0c5ca36fbb"
}
response = requests.post(url, data=payload, files=files, headers=headers) if response.status_code == 200:
return JsonResponse({"message": "API call succeeded"}) # Return a JSON response
else:
return JsonResponse({"error": "API call failed"}, status=response.status_code)
def confirm_account_fraud(request):
return render(request, 'homeApp/confirm_account_fraud.html')

def bank(request):
return render(request, 'homeApp/bank.html')
def approve(request):
return render(request, 'homeApp/approve.html')

```

Лістунів views.py

```

<?php
namespace App\Http\Controllers; use App\Mail\NotifyUser;
use App\Models\Account;
use App\Models\Report;

use App\Models\Threshold; use App\Models\Proof;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Mail;

class ReportController extends Controller
{
    //
    public function store(Request $request){
        // Acn no of the fraudster, Phone no of the fraudster(optional), Email of the reporter, Fraud Description, Proof (which would be an image)
        $acn = Account::where("account_no",$request->account_no)->first();
        $bank = Threshold::first(); if($acn){
            $acn->threshold += 1;
            if($acn->threshold >= $bank->threshold){
                $acn->status = 1;
            }
            $acn->save();
        }
        else{
            $acn = new Account();
            $acn->account_no = $request->account_no;
            $acn->threshold = 1;
            $acn->status = 0;
            $acn->save();
        }
        $report = new Report();
        $report->account_no = $request->account_no;
        $report->phone = $request->phone; //optional
        $report->email = $request->email;
        $report->description = $request->description;
        $report->bank_id = $request->bank;
        $uploadedFiles = [];
        $rand = $this->generateRandomAlphanumeric();
        $fileData = $request->file('proof'); foreach ($fileData as $file) {
            $fileName = time() . '_' . $file->getClientOriginalName();
            $filePath = $file->store('uploads', 'public');
            $proof = new Proof();
            $proof->name = $fileName;
            $proof->rand = $rand;

            $proof->save();
            // $uploadedFiles[] = $fileName;
        }
        $report->proof = $rand;
        $report->save();
        $emailAddress = $request->email; // Use appropriate logic to retrieve the email address
        if (isset($emailAddress)) {
            // Send email
            Mail::to($emailAddress)->send(new NotifyUser());
        } else {

```

Лістинг ReportController.php

```

return "failed";
}
return "success";
}
public function checkAccount(Request $request){
    $data = Account::where("account_no",$request->account_no)->first();
    $bank =Threshold::first(); if(empty($data) || empty($bank)){
        return "success";
    }
    if(($data->threshold >= $bank->threshold) || $data->status == 1){
        $data->status = 1;
        $data->save(); return "failed";
    }
    // return response(['threshold'=>$data->threshold,'data' => $data], 200); return "success";
}
public function setBankThreshold(Request $request){
    $bank =Threshold::where("name","LIKE","%".$request->name."%")->first();
    if(isset($bank)){
        $bank->name = $request->name;
        $bank->threshold = $request->threshold;
        $bank->save(); return "success";
    }
    $bank = new Threshold();

    $bank->name = $request->name;
    $bank->threshold = $request->threshold;
    $bank->save(); return "success";
}
public function downloadExcel()
{
    $acns = Account::all();
    $data = [
        ['account number','time','class'],
    ];
    foreach($acns as $acn){
        array_push($data,[$acn->account_no,$acn->created_at,$acn->status]);
    }
    $fileName = 'dataset.csv';

    $headers = array(
        "Content-type" => "text/csv",
        "Content-Disposition" => "attachment; filename=$fileName", "Pragma" => "no-cache",
        "Cache-Control" => "must-revalidate, post-check=0, pre-check=0", "Expires" => "0"
    );
    $handle = fopen('php://output', 'w'); foreach ($data as $row) {
        fputcsv($handle, $row);
    }
    return response()->stream( function () use ($handle) {
        fclose($handle);
    }, 200,
    $headers
    );
}
}

```

Лістинг ReportController.php

```

$characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZLMNOPQRSTUVWXYZ';
$randomString = '';

for ($i = 0; $i < $length; $i++) {
    $randomString .= $characters[rand(0, strlen($characters) - 1)];
}
return $randomString;
}

public function banks(){
    $banks = Threshold::all();
    return response()->json($banks);
}

public function getReportsForBank(Request $request){
    // return var_dump($request->bank);
    $data = Report::where("bank_id",$request->bank)->get(); return response()->json($data);
}

public function approve(Request $request){
    //get the report by acn and bank_code/id
    $report = Report::find($request->id); if($report->status == 1){
        return "failed";
    }
    $report->status = 1;
    $report->save();
    $data = Account::where("account_no",$report->account_no)
    ->first();
    //retrieve the bank information
    $bank =Threshold::find($request->bank);
    if(($data->threshold >= $bank->threshold) || $data->status != 1){
        $data->status = 1;
        $data->save();
    }
    // return response(['threshold'=>$data->threshold,'data' => $data], 200); return "success";
}
}

```

Лістинг ReportController.php



Дата звіту5/17/2025

Дата редагування---



Звіт не був оцінений

Звіт подібності

метадані

Назва організації

Kyiv National Economic University named after Vadym Hetman KNEU

Заголовок

Інтелектуальна система протидії фінансовому шахрайству в платіжних сервісах

Автор

Науковий керівник / Експерт

Проценко Богдан ПетровичМозгаллі О.П.

Підрозділ

кафедра інформаційних систем в економіці

Обсяг знайдених подібностей

2.75%

2.75%

КП 1

25

Довжина фрази для коефіцієнта подібності 2

0.78%

0.78%

КП 2

12424

Кількість слів

0.34%

0.34%

КЦ

100613

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про **МОЖЛИВІ** маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		1
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		15

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копію тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Копію тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://ir.kneu.edu.ua/bitstreams/8d504208-0af8-41fc-9856-893cdc8340ef/download	45 0.38 %
2	https://ir.kneu.edu.ua/bitstreams/8ccdb208-8499-47f8-b15b-5d42b9b14ce2/download	28 0.21 %
3	https://ir.kneu.edu.ua/bitstreams/8ccdb208-8499-47f8-b15b-5d42b9b14ce2/download	28 0.21 %
4	https://ir.kneu.edu.ua/bitstreams/8d504208-0af8-41fc-9856-893cdc8340ef/download	24 0.19 %

*Проценко Б.П., студент
ДВНЗ «Київський національний економічний
університет імені Вадима Гетьмана»
bohdan.protsen@gmail.com*

ЗАСТОСУВАННЯ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ В СФЕРІ ЦИФРОВИХ ФІНАНСІВ

Актуальність проблеми фінансового шахрайства у сфері цифрових технологій невинно зростає в умовах глобальної диджиталізації. За даними 2023 року, загальні світові збитки від фінансового шахрайства сягнули 485,6 мільярда доларів США, з яких 80% були спричинені саме шахрайством з платежами [1]. Найбільші втрати пов'язані з інвестиційними шахрайствами (4,6 мільярда доларів), а також з компрометацією бізнес-електронної пошти (6,7 мільярда доларів). Водночас споживачі масово повідомляють про втрати, пов'язані з онлайн-покупками та комунікацією через електронну пошту, що свідчить про трансформацію класичних схем обману у бік цифрових каналів.

У 2024 році спостерігається подальше загострення ситуації — зафіксовано понад мільйон випадків крадіжки особистих даних та майже пів мільйона випадків шахрайства з кредитними картками. Зростання випадків використання штучних ідентифікаційних даних та інструментів штучного інтелекту у шахрайських схемах робить цей вид злочинів дедалі складнішим для виявлення та попередження. Шахрайство вже не обмежується окремими інцидентами, а набуває системного характеру, що становить серйозну загрозу для фінансової безпеки як окремих споживачів, так і корпоративного сектору. Така динаміка підтверджує необхідність формування стійких систем цифрової безпеки, вдосконалення нормативної бази та підвищення фінансової грамотності громадян [2].

Посилаючись на концепцію шахрайства, викладену в вказівках Американського інституту сертифікованих бухгалтерів, це незаконна, навмисна дія, в якій є жертва (той, хто втрачає фінансовий ресурс) і жертва (хтось), хто отримує фінансовий ресурс від жертви). Таким чином, запропонована класифікація включає корпоративне шахрайство та/або шахрайство в організаціях, враховуючи, що метою є незаконне привласнення капітальних ресурсів юридичної чи фізичної особи: готівки, банківських рахунків, позик, облігацій, акцій, нерухомості та дорогоцінних металів тощо [3].

Фінансове шахрайство залишається системною загрозою, що впливає на всі сектори економіки, підриваючи довіру до фінансової системи та спричиняючи суттєві матеріальні втрати. Сучасні форми шахрайства охоплюють широкий спектр — від маніпуляцій із фінансовою звітністю до внутрішніх зловживань і кіберзлочинів. Зростаюча складність і масштаби цих загроз вказують на обмежену ефективність традиційних методів виявлення шахрайства, що спонукає до активного впровадження інтелектуальних аналітичних інструментів.

Методи машинного навчання набули особливої популярності в контексті виявлення аномалій та закономірностей у великих обсягах фінансових даних. Найпоширенішими підходами залишаються контрольовані моделі, зокрема в завданнях виявлення шахрайства з кредитними картками, страхуванням та у звітності. Проте проблема надмірної кількості хибнопозитивних результатів, обмежена інтерпретованість моделей, а також виклики, пов'язані з конфіденційністю та якістю даних, залишаються невирішеними й вимагають подальшого наукового осмислення [4].

У сучасних фінансових дослідженнях значна увага приділяється аналізу шахрайства з кредитними картками, страхового шахрайства та внутрішніх фінансових порушень. Використання методів машинного навчання (ML) дозволяє ефективно виявляти та запобігати таким видам шахрайства, аналізуючи поведінкові та транзакційні дані клієнтів, використання інтелектуальних технологій у цифрових платежах розділяється за такими основними напрямками:

1. Шахрайство з кредитними картками. Дослідження в цій сфері зосереджені на розробці моделей ML для аналізу транзакцій власників кредитних карток з метою ідентифікації підозрілих операцій. Використовуються різні алгоритми, такі як випадкові ліси (Random Forest), логістична регресія, метод опорних векторів (SVM) та градієнтний бустинг (XGBoost). Зокрема, дослідження показали, що XGBoost демонструє високу точність у виявленні шахрайських транзакцій [5].

2. Страхове шахрайство поділяється на:

- Медичне страхове шахрайство: включає підробку документів, фальшиві медичні рецепти та неправомірне виставлення рахунків [6].

- Автомобільне страхове шахрайство: передбачає змови між страхувальниками та ремонтними майстернями для отримання неправомірних виплат [7].

Для виявлення таких схем страхові компанії впроваджують надійні моделі ML, які аналізують аномальні патерни в поданих претензіях. Наприклад, використання блокчейн-технологій у поєднанні з гібридними класифікаторами, такими як поєднання випадкових лісів та SVM, сприяє підвищенню точності виявлення шахрайства.

3. Внутрішні порушення включають:

- Шахрайство з фінансовою звітністю: маніпуляції з фінансовими показниками компаній, що можуть впливати на ціни акцій та вводити в оману інвесторів.

- Відмивання грошей: процес приховування незаконно отриманих доходів через складні фінансові операції.

- Податкове шахрайство: ухилення від сплати податків шляхом надання неправдивої інформації податковим органам.

Сучасні дослідження у сфері виявлення фінансового шахрайства за допомогою методів машинного навчання демонструють як певний поступ у точності моделей, так і суттєві обмеження у доступності даних та різноманітності досліджуваних кейсів. Зокрема, літературний огляд вказує на домінування банківського сектору у фокусі уваги дослідників, особливо у зв'язку з шахрайством із кредитними картками та страхуванням. Такий вузький вектор досліджень обмежує застосовність результатів до інших галузей, де можуть проявлятися більш складні або нетипові моделі шахрайської поведінки, так при дослідженні загальнодоступних датасетів у мережі Інтернет, переважна більшість була орієнтована саме на банківську сферу [8][9].

Аналіз наявних наборів даних засвідчує переважання застарілої та обмеженої інформації, що ускладнює тестування новітніх моделей на актуальних прикладах. Дані, які часто походять із фондових бірж або закритих корпоративних джерел, мають недостатнє охоплення сучасних сценаріїв шахрайства, а також рідко включають повноцінну контекстну інформацію, яка б дозволила глибше інтерпретувати поведінкові патерни правопорушень. Водночас спостерігається нестача показників ефективності неконтрольованих алгоритмів, що унеможливує об'єктивну оцінку їх здатності до виявлення аномалій у фінансових потоках.

Тенденції розвитку галузі вказують на потребу в застосуванні більш прогресивних архітектур, таких як глибокі нейронні мережі або навчання з підкріпленням. Ці методи дозволяють підвищити адаптивність моделей до змінних стратегій шахрайства та підсилити здатність виявляти приховані аномалії у даних у режимі реального часу. Перехід від

традиційних алгоритмів до таких підходів потенційно змінює підхід до виявлення шахрайства, роблячи системи більш гнучкими та здатними до самонавчання.

У перспективі нагальною потребою залишається створення відкритих, актуалізованих та контекстуалізованих наборів даних, які б відображали ширший спектр фінансових злочинів у різних секторах економіки. Такий крок дозволить би не лише підвищити загальну достовірність експериментальних висновків, а й розширити межі практичного застосування моделей виявлення шахрайства за межами класичних фінансових інституцій.

Список використаних джерел

1. Global Financial Crime Report 2024. Visual Capitalist. [Електронний ресурс]: <https://elements.visualcapitalist.com/wp-content/uploads/2024/04/1711973384569.pdf> (дата звернення: 06.04.2025).
2. Identity Theft and Credit Card Fraud Statistics for 2025 / Author: Jack Caporal. The Motley Fool. [Електронний ресурс]: <https://www.fool.com/money/research/identity-theft-credit-card-fraud-statistics/> (дата звернення: 06.04.2025).
3. Statement on Auditing Standards No. 99: Consideration of Fraud. Wikipedia – The Free Encyclopedia. [Електронний ресурс]: https://en.wikipedia.org/wiki/Statement_on_Auditing_Standards_No._99:_Consideration_of_Fraud (дата звернення: 06.04.2025).
4. Visualisation of main machine learning models used for financial fraud detection. Nature – Humanities and Social Sciences Communications. [Електронний ресурс]: <https://www.nature.com/articles/s41599-024-03606-0/figures/7> (дата звернення: 06.04.2025).
5. Credit Card Fraud Detection using XGBoost Classifier with a Threshold Value / Authors: Arun Kumar, Ayushi Maurya, Shiv Prakash. ResearchGate. [Електронний ресурс]: https://www.researchgate.net/publication/361230475_Credit_Card_Fraud_Detection_using_XGBoost_Classifier_with_a_Threshold_Value (дата звернення: 06.04.2025).
6. Blockchain-Based Fraud Detection System for Healthcare Insurance Claims / Authors: Hopewell Ncube, Sibusisiwe Dube, Kudakwashe Maguraushe. ResearchGate. [Електронний ресурс]: https://www.researchgate.net/publication/384224523_Blockchain-Based_Fraud_Detection_System_for_Healthcare_Insurance_Claims (дата звернення: 06.04.2025).
7. Automobile Insurance Fraud Detection Using Data Mining: A Systematic Literature Review / Authors: Mohammed El-Hajj, Dipti Kapoor Sarmah, Gilian Schrijver. ResearchGate. [Електронний ресурс]: https://www.researchgate.net/publication/378178387_Automobile_Insurance_Fraud_Detection_Using_Data_Mining_A_Systematic_Literature_Review (дата звернення: 06.04.2025).
8. Detection of financial fraud by applying ML techniques: a RSL / Author: Ludivia Hernandez Aros (Universidad Cooperativa de Colombia). Harvard Dataverse. [Електронний ресурс]: <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/CM8NVY> (дата звернення: 06.04.2025).
9. Fraud detection datasets on Kaggle.com. Kaggle. [Електронний ресурс]: <https://www.kaggle.com/search?q=fraud+detection+in%3Adatasets> (дата звернення: 06.04.2025).

Науковий керівник: Мозгалі О. П., д.е.н., професор.

УДК 004.8:004.6:001.895(043.2)

Проценко Богдан

/ м. Київ /

ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ОБРОБКИ ВЕЛИКИХ ДАНИХ: ІННОВАЦІЙНІ РІШЕННЯ ТА ВИКЛИКИ

У сучасному світі кількість інформації, що генерується, зростає в геометричній прогресії. Ми щоденно стикаємося з великими обсягами даних, які можуть містити безцінні знання, приховані закономірності, прогнози та можливості. Для компаній, дослідників, медичних працівників і урядових установ доступ до цих даних створює потенціал для прийняття більш обґрунтованих рішень і поліпшення обслуговування клієнтів. Проте через масштаби та складність цих даних традиційні методи обробки вже не можуть впоратися з новими викликами. Саме тут на допомогу приходять технології

171

штучного інтелекту, які відкривають нові можливості для аналізу й обробки великих даних.

Сучасні інноваційні рішення базуються на методах штучного інтелекту, зокрема, на глибинному навчанні, нейронних мережах, машинному навчанні та автоматизованих системах виявлення аномалій.

Глибинне навчання використовує багатосарові нейронні мережі, що здатні обробляти та аналізувати складні дані з високою точністю. Наприклад, у медицині за допомогою глибинного навчання проводиться аналіз медичних зображень для діагностики захворювань. Цей процес дає змогу зменшити час на діагностику, підвищити її точність та мінімізувати людський фактор, що є критично важливим у ситуаціях, коли мова йде про рятування життя.

Методи машинного навчання дозволяють обробляти величезні обсяги даних у реальному часі. Такі рішення є надзвичайно важливими у фінансовому секторі, де автоматизоване виявлення шахрайства дозволяє відразу реагувати на підозрілі транзакції. За допомогою алгоритмів, що навчаються на основі історичних даних, система здатна миттєво аналізувати нові дані та повідомляти про аномальні дії.

Автоматизовані алгоритми виявлення аномалій знаходять своє застосування у багатьох сферах, від кібербезпеки до контролю якості на виробництві. Ці алгоритми дозволяють швидко і точно визначати патерни, що виходять за рамки норми. Наприклад, на виробництві використання ШІ для контролю якості допомагає виявляти дефекти на початкових етапах виробництва, що мінімізує втрати і підвищує ефективність.

Таблиця 1.

Приклади застосування ШІ для обробки великих даних

Метод ШІ	Сфера застосування	Опис
Глибинне навчання	Медицина	Аналіз медичних зображень для діагностики захворювань
Машинне навчання	Фінанси	Виявлення шахрайства у фінансових операціях
Нейронні мережі	Ритейл	Персоналізація рекомендацій для користувачів

172

Попри значні переваги, використання ШІ для обробки великих даних пов'язане з кількома викликами, які вимагають подальшого дослідження і розвитку технологій. Дозвольте виділити три ключові проблеми, з якими стикаються сьогодні фахівці.

Великі обсяги даних потребують потужних обчислювальних ресурсів для швидкої і точної обробки. Зазвичай, алгоритми ШІ потребують великих обчислювальних потужностей і сховищ для роботи з даними, особливо коли йдеться про глибинне навчання. Через це підвищується потреба в інноваційних архітектурах, таких як розподілені обчислення та хмарні сервіси, що забезпечують масштабованість систем і дозволяють їм обробляти великі масиви даних ефективно.

Використання персональних даних є важливою частиною процесів навчання ШІ, однак це створює загрозу конфіденційності та безпеки інформації. Питання етики використання особистих даних стає актуальним у зв'язку з тим, що персональні дані, такі як медичні записи або фінансова інформація, можуть бути використані без згоди користувача. Це призводить до необхідності розробки нормативних актів та стандартів конфіденційності для захисту прав користувачів.

Сучасні методи глибинного навчання та нейронних мереж працюють за принципом «чорної скриньки», що означає, що вони часто можуть видати точний результат без можливості пояснення процесу, який до цього призвів. Це стає серйозною проблемою у сферах, де пояснюваність результату є ключовою, наприклад, у медицині чи праві. Таким чином, інтерпретація рішень ШІ є важливою галуззю досліджень, спрямованою на підвищення прозорості та зрозумілості моделей.

Щоб забезпечити подальший розвиток систем обробки даних, необхідно звернути увагу на перспективні напрями розвитку технологій ШІ.

Гібридні моделі, що поєднують різні підходи штучного інтелекту, дозволяють створювати більш гнучкі та ефективні рішення. Наприклад,

поєднання алгоритмів глибинного навчання та машинного навчання дозволяє отримати більш точні результати і знизити залежність від людського фактору. Це особливо важливо для складних завдань, таких як прогнозування захворювань або аналіз складних ринкових умов.

Таблиця 2.

Основні виклики використання ШІ у великих даних

Виклик	Причина	Приклад
Масштабованість	Великі обсяги даних	Фінансові транзакції, що потребують обробки у реальному часі
Конфіденційність	Використання персональних даних	Медичні системи діагностики
Інтерпретація	Складність моделей	Глибинні нейронні мережі для аналізу генетичних даних

Важливим аспектом є управління даними та їхня якість. Впровадження таких інноваційних рішень, як розподілені бази даних, значно підвищує ефективність роботи з даними та знижує витрати на їх обробку. Оскільки обсяги даних продовжують зростати, ефективне зберігання та передача інформації стають критично важливими.

Для обробки великих даних потрібні значні обчислювальні ресурси, що, в свою чергу, споживають багато енергії. Тому оптимізація енергоспоживання є одним із ключових завдань, що стоїть перед фахівцями у сфері штучного інтелекту. Деякі компанії вже розробляють методи для зниження енергоспоживання при обробці даних, такі як оптимізація використання серверів та вдосконалення алгоритмів.

Отже, використання штучного інтелекту для обробки великих даних відкриває безмежні можливості для досліджень, бізнесу та суспільства в цілому. Проте, для досягнення повного потенціалу таких технологій необхідно вирішувати низку викликів, пов'язаних із масштабованістю, конфіденційністю та інтерпретацією результатів. Майбутнє штучного інтелекту залежить від ефективної реалізації нових інноваційних підходів, вдосконалення алгоритмів та систем управління даними, а також дотримання етичних норм і забезпечення безпеки користувачів.

Література

1. Нестеров В. Інтеграція технологій штучного інтелекту в інженерію даних: виклики та перспективи у сучасному інформаційному середовищі. Вісник Черкаського державного технологічного університету. Том 28, №4, 2023. URL: <https://bulletin-chstu.com.ua/uk/journals/tom-28-4-2023/integratsiya-tekhnologiy-shtuchnogo-intelektu-v-inzheneriyu-danikh-vikliki-ta-perspektivi-u-suchasnomu-informatsiynomu-seredovishchi>.
2. Тардіф А. Вплив штучного інтелекту на інновації: ключові висновки зі звіту Innovation Barometer 2025. Unite.AI. URL: <https://bit.ly/4fdWzQT>.