Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Програмні компоненти для системи виявлення шахрайських банківських транзакцій

(тема)

Виконав:
здобувач 4 року навчання,
групи КІУКІ-21-1
Станіслав ШЕВЧЕНКО
(власне ім’я, прізвище)

Спеціальність
123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Комп’ютерна інженерія
(повна назва освітньої програми)

Керівник: доц. Олександр ШМАТКО
(посада, власне ім’я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім’я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Шевченку Станіславу Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Програмні компоненти для системи виявлення шахрайських банківських транзакцій

затверджена наказом по університету від " 26 " травня 2025 р. № 424 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 17 червня 2025 р.

3. Вхідні дані до роботи

1)Нормативні документи щодо виявлення шахрайських транзакцій;

2) Літературні джерела за темою дослідження;

4. Перелік питань, що потрібно опрацювати у роботі

1) аналіз проблеми та огляд існуючих рішень;

2) огляд методів та підходів до розробки систем виявлення шахрайських транзакцій

3) розробка опису програмних компонентів;

4) розробка програмних модулів;

5) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Презентація – 18 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз проблеми та огляд існуючих рішень	02.04.25-08.04.25	
2	Дослідження алгоритмів машинного навчання	09.04.25-16.04.25	
3	Вибір технологій та інструментів для розробки	17.04.25-22.04.25	
4	Розробка програмних модулів	23.04.25-06.05.25	
5	Запуск та тестування програмних модулів	07.05.25-23.05.25	
6	Оформлення матеріалів кваліфікаційної роботи	24.05.25-03.06.25	
7	Подання кваліфікаційної роботи керівникові	04.06.25-07.06.25	
	та її попередній захист		
8	Подання кваліфікаційної роботи на	08.06.25-12.06.25	
	рецензування		

Дата видачі завдання " 26 " травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Олександр ШМАТКО
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 60 с., 12 рис., 12 табл., 1 дод., 29 джерел.

ШАХРАЙСТВО, БАНКІВСЬКІ ТРАНЗАКЦІЇ, МАШИННЕ НАВЧАННЯ, ВИЯВЛЕННЯ АНОМАЛІЙ, КЛАСИФІКАЦІЯ, ФІНАНСОВА БЕЗПЕКА, ОБРОБКА ДАНИХ, АВТОМАТИЧНЕ ВИЯВЛЕННЯ, ПРОГРАМНІ КОМПОНЕНТИ.

У роботі розглянуто проєктування та розробку програмних компонентів для системи виявлення шахрайських банківських транзакцій на основі сучасних методів машинного навчання.

Об’єктом дослідження є автоматизовані інформаційні системи обробки фінансових операцій у банківській сфері, які потребують підвищеного рівня безпеки та контролю.

Предметом дослідження виступають методи аналізу транзакційної активності клієнтів та побудова моделей виявлення аномалій у фінансових даних.

Метою роботи є підвищення ефективності виявлення шахрайських дій шляхом створення програмних компонентів, здатних автоматично виявляти нетипову поведінку користувачів на основі історичних даних транзакцій.

Розроблені програмні компоненти можуть бути інтегровані у банківські системи моніторингу для автоматичного виявлення підозрілих транзакцій у режимі реального часу. Отримані результати підтверджують доцільність застосування технологій машинного навчання у фінансовій безпеці та демонструють високу ефективність запропонованого підходу.

ABSTRACT

Master’s thesis: 60 pages, 12 figures, 12 tables, 1 appendices, 29 sources.

FRAUD, BANK TRANSACTIONS, MACHINE LEARNING, ANOMALY DETECTION, CLASSIFICATION, FINANCIAL SECURITY, DATA PROCESSING, AUTOMATIC DETECTION, SOFTWARE COMPONENTS.

The paper considers the design and development of software components for the fraudulent banking transaction detection system based on modern machine learning methods.

The object of research is automated information systems for processing financial transactions in the banking sector, which require an increased level of security and control.

The subject of the research is Methods for analyzing customer transaction activity and building models for detecting anomalies in financial data.

The purpose of the thesis is to increase the effectiveness of detecting fraudulent activities by creating software components that can automatically detect atypical user behavior based on historical transaction data.

The developed software components can be integrated into banking monitoring systems to automatically detect suspicious transactions in real time. The results obtained confirm the feasibility of using machine learning technologies in financial security and demonstrate the high efficiency of the proposed approach.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AI – штучний інтелект (англ. Artificial Intelligence)

CSV – формат табличних даних зі значеннями, розділеними комами (англ. Comma-Separated Values)

F1-міра – зважене гармонійне середнє між точністю і повнотою (англ. F1-Score)

FN – хибно негативне спрацювання (англ. False Negative)

FP – хибно позитивне спрацювання (англ. False Positive)

ML – машинне навчання (англ. Machine Learning)

PPV – прецизійність, точність позитивних передбачень (англ. Positive Predictive Value / Precision)

ROC-AUC – площа під ROC-кривою (англ. Receiver Operating Characteristic – Area Under Curve)

TP – істинно позитивне спрацювання (англ. True Positive)

TPR – повнота виявлення (англ. True Positive Rate / Recall)

ВСТУП

У сучасному цифровому середовищі, де обсяги фінансових транзакцій постійно зростають, проблема виявлення шахрайських операцій набуває особливої актуальності. Шахрайство у банківському секторі не лише завдає прямих фінансових збитків, але й підриває довіру клієнтів до платіжних систем, банківських установ та цифрових фінансових сервісів загалом. Традиційні методи контролю та аудиту виявляються недостатньо ефективними в умовах високої динаміки та складності поведінкових шаблонів користувачів. У зв’язку з цим все більшої ваги набувають автоматизовані системи виявлення шахрайських транзакцій на основі інтелектуального аналізу даних.

Особливу роль у побудові таких систем відіграють алгоритми машинного навчання, які дозволяють здійснювати аналіз великих масивів транзакційної інформації та виявляти приховані аномалії, характерні для шахрайської поведінки. Їхнє застосування дозволяє не лише підвищити точність виявлення загроз, але й мінімізувати кількість хибних спрацювань, що є критично важливим для збереження клієнтського досвіду.

Об’єктом дослідження є автоматизовані інформаційні системи обробки фінансових операцій у банківській сфері, які потребують підвищеного рівня безпеки та контролю.

Предметом дослідження виступають методи аналізу транзакційної активності клієнтів та побудова моделей виявлення аномалій у фінансових даних.

Метою роботи є підвищення ефективності виявлення шахрайських дій шляхом створення програмних компонентів, здатних автоматично виявляти нетипову поведінку користувачів на основі історичних даних транзакцій.

Досягнення поставленої мети передбачає вирішення низки задач:

- аналіз існуючих підходів до виявлення фінансового шахрайства;

- вибір і обґрунтування відповідних алгоритмів класифікації;
- створення програмного модуля з інтеграцією моделі машинного навчання;
- проведення експериментальної перевірки розробленої системи на основі відкритих наборів даних.

Результати цієї роботи можуть бути впроваджені в реальні банківські IT-системи для підвищення рівня фінансової безпеки та автоматизації процесу моніторингу транзакцій.

1 ОГЛЯД ПРОБЛЕМИ ВИЯВЛЕННЯ ШАХРАЙСЬКИХ ТРАНЗАКЦІЙ

1.1 Огляд проблемної області

Упродовж останніх років розвиток систем штучного інтелекту (ШІ), призначених для виявлення шахрайських банківських операцій, привертає значну увагу дослідників і фахівців з інформаційної безпеки. Це пояснюється зростаючою кількістю фінансових правопорушень у банківській сфері, які призводять до серйозних фінансових втрат як для самих банків, так і для їхніх клієнтів. Системи, побудовані на основі штучного інтелекту, демонструють значний потенціал для ефективного виявлення та запобігання шахрайським транзакціям у режимі реального часу, що є суттєвою перевагою порівняно з традиційними методами виявлення шахрайства.

Задача виявлення шахрайських банківських транзакцій ґрунтується на використанні алгоритмів машинного навчання для аналізу великих обсягів даних із різноманітних джерел, зокрема записів транзакцій, інформації про клієнтів та логів мережевої активності. Такі алгоритми здатні виявляти приховані закономірності й аномалії у даних, що можуть свідчити про шахрайські дії, як-от несанкціонований доступ, нетипові шаблони транзакцій або підозрілу поведінку користувача.

Банківська транзакція – це будь-яка дія, пов’язана з банківським рахунком, що може відбуватись як в онлайн-, так і в офлайн-режимі між усіма залученими сторонами. Завершення транзакції відбувається шляхом подання письмового або електронного розпорядження банку за допомогою інтернет-банкінгу, каналів комунікації або платіжних інструментів [1]. Усі банківські транзакції поділяються на дві основні категорії: справжні та шахрайські. Шахрайськими вважаються ті транзакції, які не відповідають правилам фінансового обігу або не були санкціоновані клієнтом. Найпоширенішими типами банківського шахрайства є: підміна особи, перехоплення облікових

даних, незаконні перекази коштів, відмивання грошей та махінації з обліком.

З огляду на ускладнення форм шахрайства, постає необхідність розробки нових засобів захисту. Серед п’яти основних методів попередження банківського шахрайства виділяють: штучний інтелект, біометричну ідентифікацію, використання консорціумних даних, стандартизацію високих технологій та машинне навчання [2]. Після початку пандемії COVID-19, а згодом і повномасштабної війни в Україні, випадки шахрайських банківських операцій почастішали. Це пов’язано із різким зростанням кількості онлайн-транзакцій та появою численних благодійних фондів, які шахраї використовують для обману користувачів. Тому виникає потреба у створенні надійних автоматизованих алгоритмів, які дозволяють виявляти операції, що несуть загрозу фінансам користувачів, порушують податкове або фінансове законодавство.

У цьому дослідженні було зроблено акцент на використанні алгоритмів машинного навчання як ефективного інструменту аналізу й класифікації онлайн-транзакцій. Основною метою стало створення моделей, здатних ідентифікувати шахрайські банківські операції, особливо в умовах пандемії COVID-19 та воєнного стану, коли онлайн-платежі стали переважаючими, а діяльність благодійних фондів — мішенню для шахраїв.

У межах реалізованого проєкту здійснено побудову кількох моделей машинного навчання на основі різних підходів, проведено їх оцінку з використанням кількісних метрик (точність, повнота, F1-міра) та візуалізацій. Результати аналізу дозволили зробити висновок щодо ефективності обраного підходу. Було застосовано техніки попередньої обробки даних, нормалізації та зменшення розмірності, що забезпечило підвищення точності моделі.

У завданні класифікації транзакцій використовувалась бінарна логіка, згідно з якою кожна операція позначається як справжня або шахрайська. Такий підхід дозволив ефективно реалізувати алгоритми класифікації, зокрема дерева рішень, логістичну регресію, K-ближчих сусідів, Random Forest та нейронні мережі.

Для забезпечення ефективного навчання моделі необхідно мати в розпорядженні повноцінну історичну базу даних. Дані про справжні транзакції, які ще не відмічені як шахрайські, часто шифруються для захисту конфіденційності клієнтів. Проте це не перешкоджає роботі алгоритмів машинного навчання, які здатні оперувати зашифрованими даними.

Запровадження технологій ШІ у сферу виявлення фінансового шахрайства супроводжується певними викликами, зокрема проблемою інтерпретованості моделей. Алгоритми, що використовуються, мають складну структуру, яка може бути важкою для розуміння та верифікації. Крім того, аналіз особистих даних викликає питання щодо дотримання принципів конфіденційності та захисту персональних даних. Ці виклики потребують ретельного розгляду для забезпечення етичного та безпечного використання технологій ШІ.

Узагальнюючи, можна зазначити, що розробка програмних рішень для виявлення шахрайських банківських операцій на основі штучного інтелекту є актуальним і перспективним напрямом, здатним суттєво покращити ефективність та точність виявлення фінансових загроз. Водночас важливо враховувати ризики, пов’язані з прозорістю алгоритмів, безпекою персональних даних та можливістю етичного використання таких рішень у банківській сфері.

1.2 Огляд публікацій за темою дослідження

Банківське шахрайство є однією з найбільш поширених і соціально небезпечних форм кіберзлочинності, що полягає в незаконному заволодінні коштами або активами банків, фінансових установ чи їхніх клієнтів. Згідно з визначенням [3], до банківського шахрайства відносяться будь-які навмисні дії, спрямовані на обман фінансової установи шляхом подання неправдивих даних, використання підроблених документів або отримання доступу до чужих рахунків. У сучасному цифровому середовищі, де основні банківські

операції все частіше відбуваються онлайн, проблема виявлення шахрайських транзакцій набула особливої актуальності (рисунок 1.1).

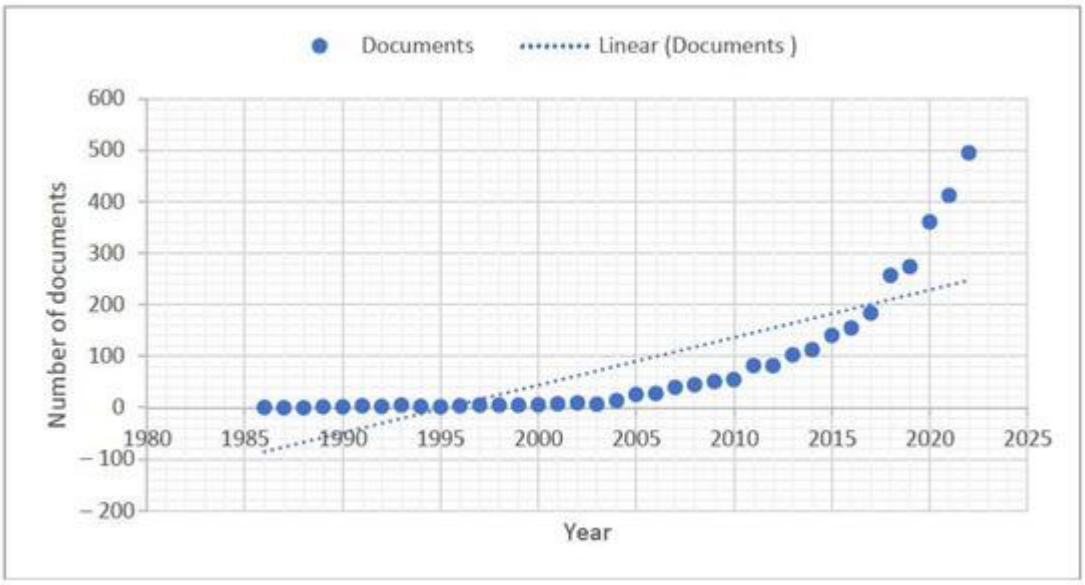

Рисунок 1.1 – Статиска публікаці за темою дослідження

З метою дослідження динаміки наукових розвідок у цій сфері було проаналізовано понад 3 000 публікацій у базі Scopus [4]. Як видно з візуалізацій, виконаних за допомогою інструмента VOSviewer (рисунок 1.2), інтерес до тематики стрімко зростав останніми роками. Тематичне картування продемонструвало три основні кластери досліджень, що охоплюють понад 94 тисячі зв’язків між публікаціями.

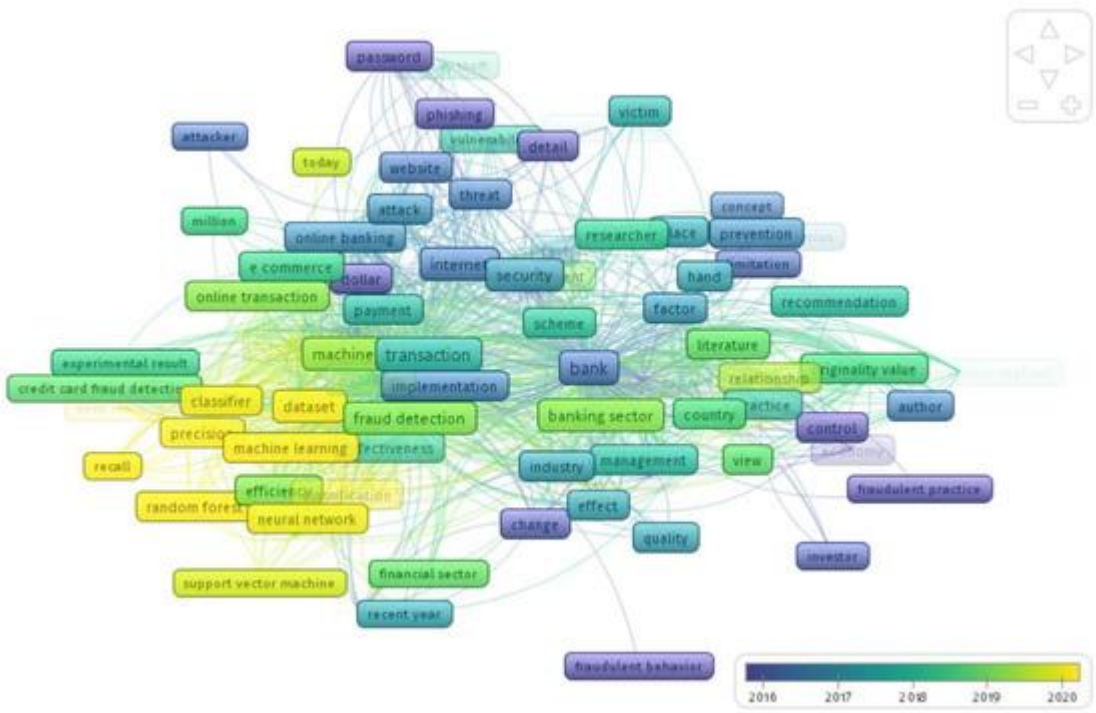

Рисунок 1.2 - Карта публікацій на основі статистики середньої кількості публікацій у Scopus за рік.

Результати аналізу свідчать про те, що шахрайські банківські практики, зокрема фішинг, крадіжка особистих даних, відмивання коштів та несанкціонований доступ до рахунків, є глобальним викликом для фінансових установ і регуляторів. Наприклад, фішинг-атаки зазвичай реалізуються через підроблені електронні листи або вебсайти, які імітують справжні ресурси банку з метою отримання особистої інформації клієнтів. Ідентифікаційна крадіжка може мати довготривалі наслідки як фінансового, так і психологічного характеру. Ще однією формою шахрайства є відмивання коштів — спроба приховати походження нелегальних фінансових активів, що загрожує банку юридичною відповідальністю. Також поширеним є захоплення акаунтів — отримання несанкціонованого доступу до рахунку жертви з метою привласнення коштів.

У відповідь на ці виклики фінансові установи активно впроваджують передові системи виявлення шахрайства, двофакторну автентифікацію, аналіз поведінки клієнтів і машинне навчання. Проте загрозу становить постійна

еволюція способів шахрайства, що вимагає постійного оновлення захисних механізмів.

У нещодавніх наукових працях [13–29] запропоновано низку ефективних підходів до виявлення шахрайських операцій на основі машинного навчання. Зокрема, дослідження [13] пропонує алгоритм, що використовує генетичний алгоритм для вибору ознак, після чого застосовуються класифікатори: Random Forest, нейронна мережа, дерево рішень, логістична регресія та наївна баєсівська мережа. У роботі [14] наголошується на усуненні упередженості в даних шляхом фільтрації найінформативніших змінних, після чого дані обробляються за допомогою таких алгоритмів, як SVM, градієнтне підсилення, лінійна регресія. Попри 76% точність, основна перевага роботи — у запропонованій системі попередньої обробки даних.

У дослідженні [15] було використано багатошарову перцептронну мережу (MLP) для виявлення шахрайства. Автори оцінили модель за точністю, специфічністю, чутливістю, F1-мірою тощо. У роботі [16] запропоновано нетиповий підхід: сегментацію користувачів (нові та існуючі) та застосування CatBoost і глибокої нейронної мережі до кожної з груп. Це дозволило досягти AUC 0.97 для CatBoost та 0.84 для DNN. У [17] розглянуто гібридну модель, що поєднує LSTM-мережу та AdaBoost. Результати моделі засвідчили дуже високу чутливість (0.996) і специфічність (0.998).

Дослідження [18] порівнює ефективність логістичної регресії, SVM, нейронної мережі та Random Forest, де найкращий результат показала нейронна мережа з F1-мірою 0.91. У роботі [19] акцент зроблено на LSTM-моделях, що враховують послідовність транзакцій. Метод дозволяє зменшити кількість помилкових спрацювань, що є важливою особливістю. У [20] використано різні моделі, серед яких найвищу точність показав Random Forest (97.58%).

Крім того, у роботі [21] запропоновано узагальнити метрику редагування рядків, що має застосування у виявленні шахрайства. У [22]

продемонстровано переваги попередньої обробки даних у прогнозуванні показників повітря. У [23] представлено модель для безпечної аутентифікації учасників з використанням логістичної регресії. У [25] метод надлишкового повторного вибіркового оброблення дозволив досягти точності 0.99. У [26] акцент зроблено на адаптацію системи до реального часу. Робота [28] демонструє модель кібернападу на фінансові установи за допомогою баєсівських мереж. У [29] показано зв'язок між рівнем управління знаннями в банках і впровадженням захисних стратегій.

Загалом, огляд літератури свідчить про високу практичну цінність застосування алгоритмів машинного навчання для виявлення шахрайських банківських операцій. Попри різні підходи до розв'язання проблеми, ключовим залишається правильне формування вибірки, обробка ознак та вибір алгоритму класифікації. Це дозволяє забезпечити високу точність виявлення та мінімізацію помилкових спрацювань, що є критичними чинниками у банківській сфер.

1.3 Постановка задачі досліджень

У контексті стрімкого розвитку цифрових фінансових послуг, збільшення обсягу онлайн-транзакцій та еволюції методів шахрайства постає нагальна потреба у створенні ефективних механізмів виявлення та запобігання шахрайським банківським операціям. Враховуючи це, основною метою даного дослідження є розробка, навчання та оцінювання моделей машинного навчання, здатних з високою точністю класифікувати банківські транзакції як справжні або шахрайські.

Для досягнення поставленої мети необхідно розв'язати низку взаємопов'язаних задач:

- провести огляд сучасного стану проблеми шахрайства в банківській сфері та методів її автоматизованого виявлення за допомогою алгоритмів машинного навчання. На основі аналізу літератури обґрунтувати доцільність

використання конкретних моделей для виявлення шахрайства;

- сформувати вибірку для навчання моделей на основі одного або кількох відкритих датасетів (наприклад, кредитні картки або банківські транзакції), яка містить мічені приклади справжніх і шахрайських транзакцій. Виконати попередню обробку даних, включно з очищенням, нормалізацією та вибором інформативних ознак;

- розробити та реалізувати кілька моделей машинного навчання, зокрема методи класифікації, такі як логістична регресія, дерева рішень, Random Forest, SVM, нейронні мережі, зокрема багатошаровий перцептрон та рекурентні мережі;

- порівняти ефективність моделей за метриками точності (accuracy), повноти (recall), точності позитивного прогнозу (precision) та F1-міри. Особливу увагу приділити мінімізації хибнопозитивних і хибнонегативних спрацювань, що мають вирішальне значення в системах фінансової безпеки;

- побудувати узагальнену модель із найкращими характеристиками для інтеграції в систему виявлення шахрайства, яка може працювати в реальному часі або в умовах потокового аналізу транзакцій;

- провести експериментальну перевірку ефективності запропонованого підходу на тестовій вибірці та порівняти отримані результати з аналогічними працями в літературі.

Розв’язання вищезазначених задач дозволить створити надійну систему підтримки прийняття рішень у сфері фінансової безпеки, орієнтовану на своєчасне виявлення шахрайських транзакцій із високим рівнем точності, мінімальними затримками та з урахуванням сучасних викликів, таких як зміщені або незбалансовані дані, а також постійно змінювані сценарії атак.

2 ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ ШАХРАЙСЬКИХ ТРАНЗАКЦІЙ

2.1 Машинне навчання в задачах виявлення шахрайства

Для досягнення поставленої в дослідженні мети — розробки ефективного механізму виявлення шахрайських банківських транзакцій — було використано алгоритми класифікації. Такі алгоритми належать до категорії методів машинного навчання з учителем, оскільки вони вивчають відповідність між вхідними змінними (ознаками) та дискретним виходом, що представляє клас об’єкта. Особливістю алгоритмів класифікації є те, що їх вихід є категоріальним, а не числовим, тобто вони повертають не безперервні значення, а дискретні категорії.

Алгоритми класифікації використовують навчальні вибірки, в яких кожне спостереження має набір ознак та відповідну мітку класу. Завдяки аналізу таких прикладів моделі здатні передбачати класи нових об’єктів, які раніше не зустрічалися. Ця властивість робить алгоритми класифікації надзвичайно корисними в задачах виявлення шахрайства, де необхідно ідентифікувати транзакції як «чесні» або «шахрайські».

У даному дослідженні як базовий було обрано відкритий набір даних Credit Card Fraud Detection, розміщений на платформі Kaggle. Цей набір даних широко використовується в академічних і прикладних дослідженнях для вирішення задач виявлення шахрайства з використанням методів машинного навчання. Платформа також забезпечує інтеграцію з інструментами візуалізації, включно з графіком кривої ROC (Receiver Operating Characteristic) та метрикою AUC (Area Under Curve), що дозволяє об’єктивно порівнювати ефективність різних моделей. Програмна реалізація була здійснена у вигляді Jupyter-ноутбука безпосередньо на платформі Kaggle, що дозволяє зручно запускати всі кроки дослідження натисканням кнопки «Run all» або

використовуючи середовище локального виконання, яке підтримує Jupyter-ноутбуки.

Рисунок 2.1 - Загальна схема реалізації запропонованого програмного рішення

На рисунку 2.1 наведено загальну схему реалізації запропонованого програмного рішення. Вона охоплює всі етапи — від вибору датасету до виведення оптимальної моделі. Алгоритм роботи програмного рішення включає кілька ключових фаз:

Спочатку відбувається вибір відповідного набору даних (у цьому випадку — з Kaggle), після чого дані завантажуються до програми за допомогою бібліотек Pandas або NumPy. Далі відбувається поділ даних на навчальну та тестову вибірки з використанням функцій бібліотеки Scikit-learn.

Особливу увагу було приділено попередній обробці даних. Зокрема, було проведено стандартизацію ознак за допомогою StandardScaler, щоб забезпечити однаковий масштаб ознак та покращити збіжність моделей. Крім

того, було застосовано метод випадкового недосемплювання (random undersampling) для вирівнювання класів у навчальній вибірці, що дозволяє компенсувати дисбаланс між кількістю шахрайських та не шахрайських транзакцій.

Після підготовки даних відбувається вибір кандидатних моделей машинного навчання. Для кожної моделі здійснюється процес налаштування гіперпараметрів методом перехресної валідації (cross-validation). Це дозволяє забезпечити максимальну точність та узагальнюваність обраної моделі. Далі кожна з моделей оцінюється на тестовій вибірці з використанням відповідної метрики ефективності (ACC, PPV, TPR тощо).

Після тестування обирається модель з найкращими характеристиками, результати якої зберігаються для подальшого використання у продуктивному середовищі або в додаткових дослідженнях.

Описаний підхід реалізує повний цикл машинного навчання: від попередньої обробки даних до навчання та оцінки ефективності моделі. Запропоноване рішення ґрунтується на трьох ключових складових: алгоритмах машинного навчання, методах попередньої обробки даних та метриках оцінювання. Його структурованість і модульність дозволяють легко адаптувати систему до інших задач виявлення аномалій, що робить її перспективною для впровадження в реальних банківських системах.

2.2 Вибрані алгоритми машинного навчання

Для реалізації моделі виявлення шахрайських транзакцій було обрано сім алгоритмів машинного навчання: випадковий ліс (Random Forest), метод k-найближчих сусідів (k-Nearest Neighbors), логістична регресія (Logistic Regression), стохастичний градієнтний спуск (Stochastic Gradient Descent Classifier), дерево рішень (Decision Tree), наївний байєсівський класифікатор (Naïve Bayes) та метод опорних векторів (Support Vector Machine). Кожен з цих методів має свої переваги та обмеження, які зумовили їх використання у

контексті завдань класифікації.

Дерево рішень є однією з базових технік класифікації, яка відтворює структуру у вигляді графа з розгалуженнями, подібного до блок-схеми. Внутрішні вузли дерева відповідають перевірці значення атрибутів, гілки — варіантам результату цієї перевірки, а листові вузли — кінцевим класам. Процес навчання дерева базується на рекурсивному поділі вихідної множини прикладів на підмножини, поки розділення дає виграш у точності. Перевагами методу є його інтерпретованість, здатність працювати з великими наборами даних, відсутність потреби в попередньому налаштуванні параметрів та досить висока точність.

Випадковий ліс — це ансамблевий метод, який базується на комбінації множини дерев рішень. Алгоритм формує набір незалежних дерев, кожне з яких будується на основі випадкової підмножини даних. Остаточне рішення приймається за принципом голосування. Такий підхід дозволяє зменшити ризик переобучення, покращити загальну продуктивність класифікатора та оцінити важливість окремих ознак. Метод ефективно працює як для класифікації, так і для регресії, демонструючи високу точність.

Логістична регресія є поширеним статистичним методом, який використовується для вирішення задач бінарної класифікації. Основна ідея полягає у визначенні ймовірності належності спостереження до певного класу за допомогою логістичної функції, що трансформує значення ознак у діапазон [0;1]. Метод широко використовується у задачах передбачення на основі незалежних змінних, зокрема у сфері виявлення шахрайства.

Метод опорних векторів (SVM) відноситься до наглядуваного навчання і застосовується як для класифікації, так і для регресії. Його мета — побудувати гіперплощину у багатовимірному просторі ознак, яка максимально відокремлює об’єкти різних класів. Алгоритм використовує крайні точки — так звані опорні вектори — що мають найбільший вплив на побудову межі. SVM є ефективним методом, особливо у випадках з великим числом ознак та малим числом спостережень.

Метод k-найближчих сусідів (KNN) належить до непараметричних алгоритмів наглядуваного навчання. Принцип його дії полягає у класифікації нових прикладів на основі більшості класів k найближчих до нього точок навчальної вибірки. KNN характеризується простотою реалізації та високою точністю при належній нормалізації даних. У випадку багатокласової класифікації рішення приймається на основі голосування з порогом не менше 25%.

Класифікатор стохастичного градієнтного спуску (SGD) є швидким та ефективним методом адаптації лінійних моделей до великомасштабних задач. Його особливістю є здатність працювати з розрідженими вхідними даними та забезпечення обчислювальної ефективності. Саме тому SGD часто використовується в обробці природної мови та класифікації текстів. Метод дозволяє ефективно вирішувати задачі логістичної регресії або SVM, особливо коли кількість прикладів і ознак перевищує сотні тисяч.

Наївний байєсівський класифікатор ґрунтується на теоремі Байєса і передбачає, що усі ознаки є умовно незалежними. Незважаючи на це спрощення, метод демонструє високу точність на практиці та часто використовується у класифікаційних задачах з великим числом ознак, наприклад, у задачах фільтрації спаму чи класифікації текстів. Формула Байєса, що лежить в основі алгоритму, має вигляд:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)},$$

де $P(A|B)$ — апостеріорна ймовірність події A за умови, що відбулась подія B;

$P(B|A)$ — ймовірність B за умови A;

$P(A)$ та $P(B)$ — апріорні ймовірності подій.

Усі зазначені алгоритми були реалізовані, протестовані та оцінені з метою порівняння їх точності в задачі виявлення шахрайських банківських

транзакцій. Їх вибір базувався на здатності ефективно класифікувати дані, адаптуватися до високовимірних просторових структур та забезпечувати інтерпретованість отриманих результатів.

2.3 Формування функціональних та нефункціональних вимог до системи обміну конфіденційною інформацією

У системі виявлення шахрайських банківських транзакцій визначено низку ключових учасників (акторів), кожен із яких виконує визначену роль у процесі функціонування інформаційної системи (рисунок 2.2). Центральним джерелом ініціації транзакцій виступає клієнтська програма, яка є інтерфейсом користувача для здійснення фінансових операцій. Ця програма надсилає дані транзакції, включаючи суми, рахунки відправника та одержувача, до системи для подальшого аналізу. Вся введена інформація надходить до модуля виявлення шахрайських транзакцій, який реалізує алгоритми машинного навчання, натреновані на великій кількості історичних прикладів.

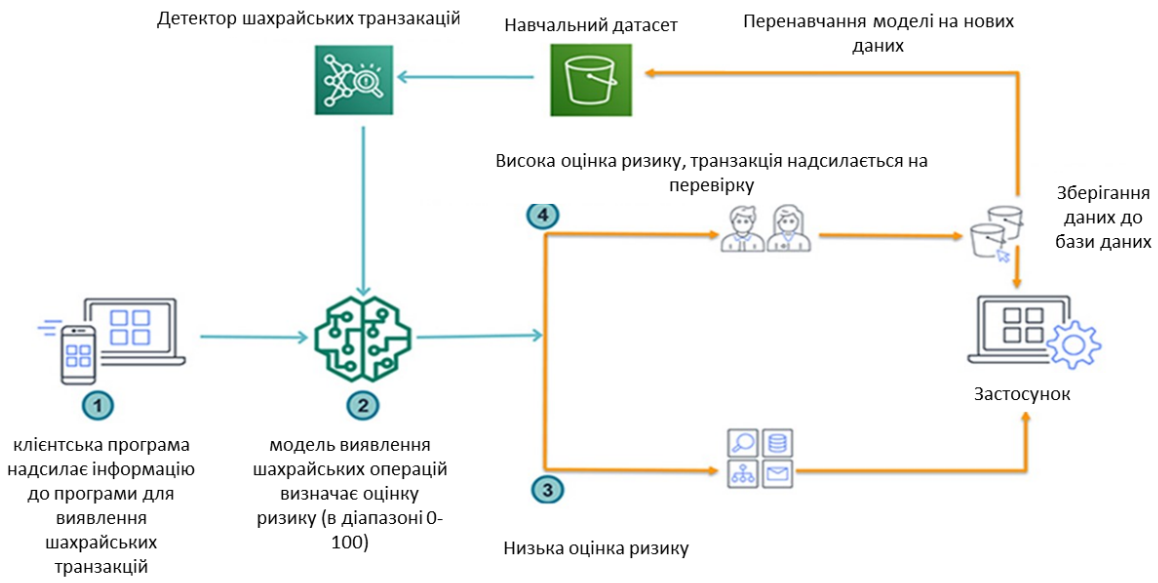

Рисунок 2.2 – Функціональна модель системи

Цей модуль виконує автоматичну оцінку рівня ризику кожної транзакції та класифікує її як таку, що має високий або низький рівень ймовірності шахрайства. У разі, якщо транзакція отримує високий ризиковий бал, вона автоматично передається до модуля ручної перевірки, де аналітики або оператори служби безпеки здійснюють її додаткову перевірку та ухвалюють остаточне рішення щодо подальшого опрацювання. У випадку підтвердження підозри транзакція може бути заблокована, а також відповідні дані зберігаються в централізованій базі даних для подальшого аналізу та перенавчання моделі.

Суттєвим елементом системи є база даних, що забезпечує збереження інформації про всі транзакції, їх оцінки ризику, результати експертної перевірки, а також формування навчального вибіркового набору для подальшого вдосконалення моделі. Ці накопичені дані використовуються модулем перенавчання моделі, який періодично здійснює оновлення параметрів машинного навчання з метою покращення точності прогнозування та адаптації до нових шахрайських шаблонів.

Завершальним елементом архітектури системи є застосунок адміністратора або інтерфейс персоналу служби безпеки, який забезпечує аналітичний контроль над ефективністю моделі, дозволяє переглядати історію виявлених транзакцій, здійснювати аудит процесів перевірки та керувати параметрами моделі в реальному часі. Таким чином, взаємодія між акторами системи забезпечує цілісний цикл від виявлення транзакції до її остаточної перевірки та навчання моделі, що гарантує безперервне вдосконалення рівня інформаційної безпеки банківських операцій.

Функціональні вимоги до системи представлено у таблиці 2.2

Таблиця 2.2 – Взаємозв’язок акторів, сценаріїв використання та функціональних вимог

Актор	Сценарій використання	Функціональні вимоги
1	2	3
Користувач / клієнтська програма	Надсилання даних транзакції до системи	Забезпечити інтерфейс для передачі даних транзакції до системи в реальному часі
Система виявлення шахрайства	Аналіз отриманої транзакції та визначення рівня ризику	Реалізувати алгоритм оцінки ризику (0–100); виконати класифікацію транзакції як низькоризикової або підозрілої
Система перевірки транзакцій	Перевірка підозрілих транзакцій	Забезпечити ручну або автоматизовану перевірку транзакцій з високим ризиком
Система зберігання даних	Збереження результатів аналізу до бази даних	Забезпечити надійне зберігання результатів перевірки, у тому числі для повторного використання
Навчальна система (детектор)	Перенавчання моделі на основі нових транзакцій	Забезпечити регулярне оновлення моделі машинного навчання на основі перевірених транзакцій

Продовження таблиці 2.2

1	2	3
Застосунок адміністратора	Візуалізація результатів та управління системою	Надати адміністративний інтерфейс для перегляду підозрілих транзакцій, їх статусів, історії рішень та моделі навчання

У процесі проєктування системи з особливу увагу слід приділити нефункціональним вимогам, які визначають якість, надійність, сумісність та ефективність роботи системи в реальних умовах. Ці вимоги не описують конкретні функції системи, однак безпосередньо впливають на її здатність до стабільного та безпечного функціонування в умовах динамічного навантаження та високих вимог до захисту даних.

В таблиці 2.3 представлено нефункціональні вимоги, критерії вимірювання та цільові значення.

Таблиця 2.3 – Нефункціональні вимоги, критерії вимірювання та цільові значення

№	Нефункціональна вимога	Критерій вимірювання	Цільове значення
1	2	3	4
1	Час обробки транзакції	Середній час від моменту надсилання до відповіді	Не більше 1 секунди

Продовження таблиці 2.3

1	2	3	4
2	Доступність системи	Частка часу, коли система доступна	Не менше 99,9%
3	Надійність	Частота відмов у роботі системи	Не більше 1 відмова на 10 000 транзакцій
4	Масштабованість	Можливість обробки зростаючого навантаження	Лінійне масштабування до 10 млн транзакцій/добу
5	Конфіденційність даних	Наявність механізмів шифрування	Використання SSL/TLS та AES-256
6	Безперервність обслуговування	Максимальний час відновлення після збою (RTO)	До 5 хвилин
7	Зручність інтерфейсу для аналітика	Суб'єктивна оцінка зручності (за шкалою SUS)	Не менше 80 балів
8	Узгодженість інтерфейсу	Відсутність критичних помилок у навігації	0 критичних помилок при тестуванні
9	Логування подій та аудит	Повнота журналу аудиту	100% транзакцій логуються
10	Можливість перенавчання моделі	Інтервал між циклами оновлення моделі	Не більше 24 годин

3 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА ПРОГРАМНИХ КОМПОНЕНТІВ

3.1 Проєктування архітектури системи

Система виявлення шахрайських банківських транзакцій будується на багаторівневій архітектурі, що складається з кількох основних компонентів: клієнтської частини, сервісного ядра, модуля машинного навчання, бази даних та адміністративного інтерфейсу. Архітектурна модель описується за допомогою UML-діаграми компонентів (Component Diagram) (рисунок 3.1), яка дозволяє візуалізувати логічні зв’язки між підсистемами, модулями та зовнішніми взаємодіями.

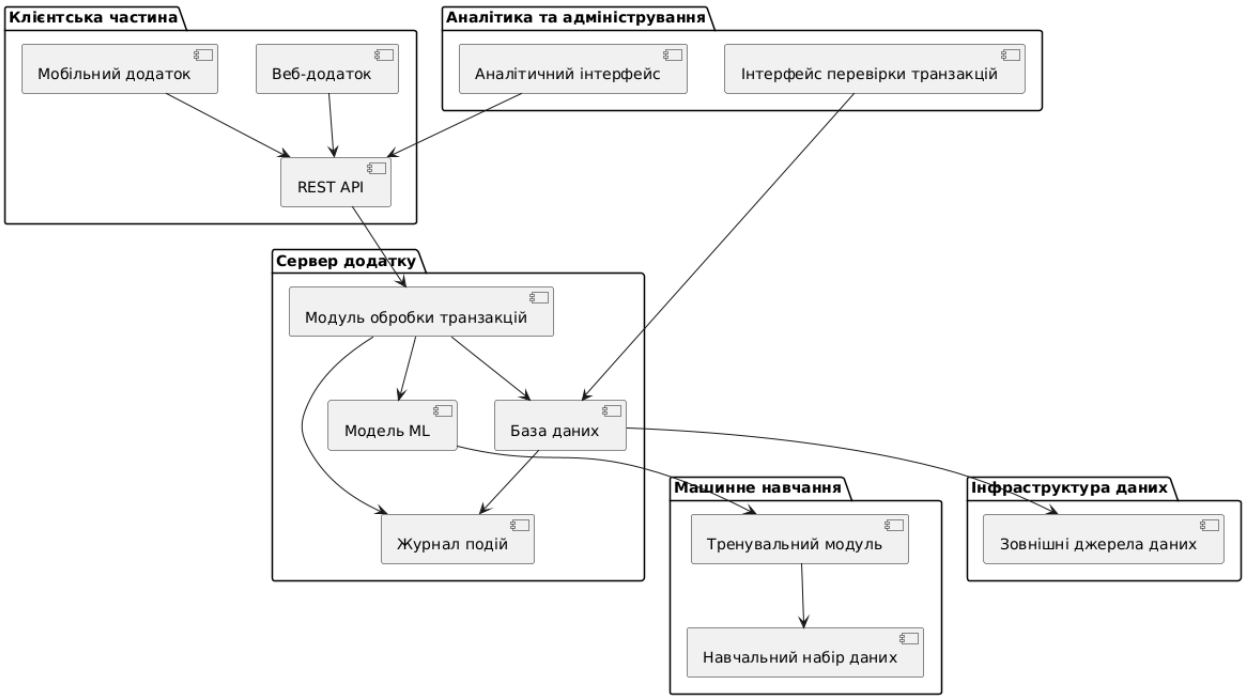

Рисунок 3.1 – Архітектура системи

Клієнтська частина – представлена мобільним і веб-додатком, які дозволяють користувачам виконувати транзакції та отримувати повідомлення про статуси перевірок. Всі запити до основної логіки проходять через REST API.

REST API – забезпечує з’єднання між клієнтом та серверною частиною, приймаючи запити і передаючи їх у модуль обробки.

Модуль обробки транзакцій – головний функціональний блок, який здійснює попередню перевірку, викликає модель машинного навчання, обробляє відповіді та оновлює базу даних.

Модель ML – реалізована за допомогою одного або кількох класифікаційних алгоритмів (наприклад, Random Forest, Logistic Regression тощо), які оцінюють ймовірність шахрайства.

Тренувальний модуль – відповідає за регулярне перенавчання моделі на основі нових даних.

База даних – зберігає інформацію про транзакції, ризикові оцінки, журнали дій, метадані користувачів тощо.

Журнал подій – використовується для аудиту та ретроспективного аналізу.

Аналітичний інтерфейс – дозволяє аналітикам переглядати підозрілі транзакції, налаштовувати моделі та формувати звіти.

Дана архітектурна діаграма відображає чіткий розподіл відповідальностей між компонентами та дозволяє забезпечити масштабованість, безпеку, надійність та можливість повторного використання елементів системи. Особлива увага приділена ізоляції моделі машинного навчання та можливості її регулярного перенавчання на оновлених даних, що забезпечує адаптацію системи до нових шаблонів шахрайства.

3.2 Проєктування програмних компонентів системи

3.2.1 Діаграма варіантів використання

На рисунку 3.2 представлено UML-діаграму варіантів використання (use case diagram), яка описує основні дії, які виконує користувач системи, а саме співробітник банку (Bank Staff). Діаграма демонструє функціональну

взаємодію між актором (користувачем) і системою через низку варіантів використання, що визначають функціональні можливості програмного забезпечення банку.

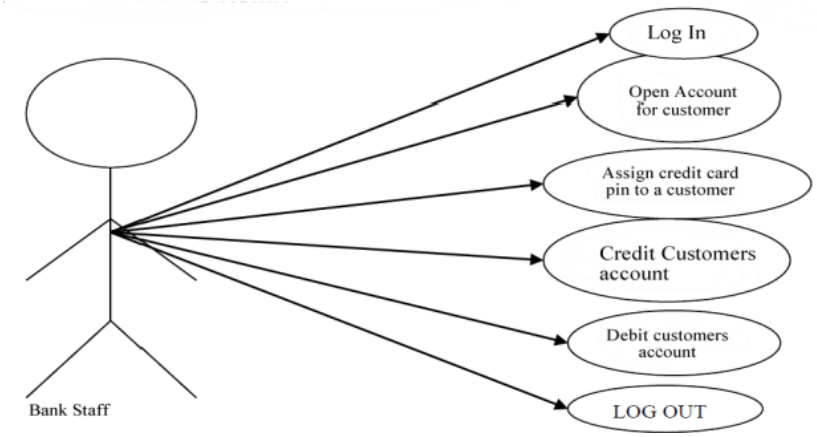

Рисунок 3.2 – Діаграмма варіантів використання для співробітника банку

Актор (Bank Staff), співробітник банку. Він має доступ до кількох основних сценаріїв взаємодії із системою, які представлені еліпсами праворуч:

- Log In – вхід до системи, що є початковим кроком для доступу до її функціоналу;
- Open Account for customer – відкриття нового банківського рахунку для клієнта;
- Assign credit card pin to a customer – призначення PIN-коду до кредитної картки клієнта;
- Credit Customers account – зарахування коштів на рахунок клієнта;
- Debit customers account – зняття коштів з рахунку клієнта;
- LOG OUT – вихід із системи після завершення роботи.

Кожен варіант використання (use case) з'єднаний стрілкою з актором, що означає його ініціацію співробітником банку. Діаграма не лише демонструє набір можливих функцій для даного типу користувача, але й формує основу для формулювання функціональних вимог у процесі розробки програмного забезпечення банківської системи.

Загалом, ця діаграма є корисним інструментом для аналізу сценаріїв

взаємодії персоналу з інформаційною системою банку, забезпечуючи візуалізацію ролей, обов’язків і прав доступу до основного функціоналу

На рисунку 3.3 представлено UML-діаграму варіантів використання (use case diagram), яка ілюструє взаємодію власника кредитної картки (Credit card Holder) із банківською інформаційною системою. Дана діаграма є частиною функціонального моделювання системи і призначена для візуалізації ключових дій, які може виконувати користувач у ролі власника картки.

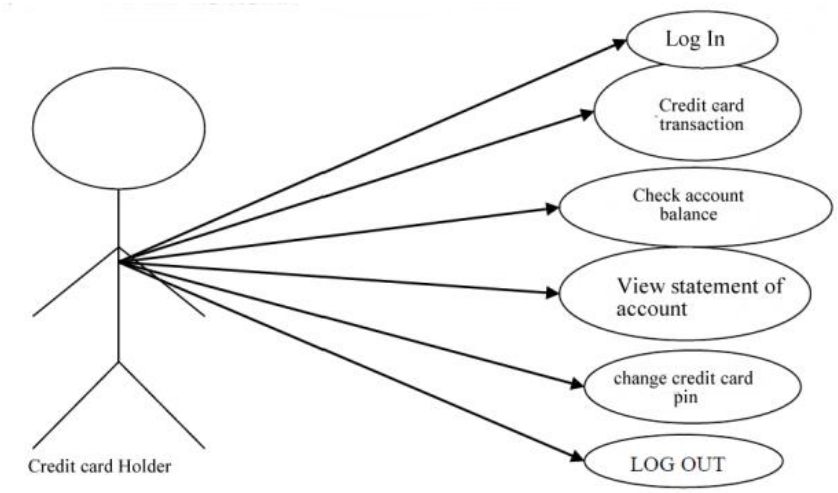

Рисунок 3.3 – Діаграмма варіантів використання для клієнта банку

Актор Credit card Holder (власник кредитної картки) має доступ до таких функцій системи:

- Log In – вхід до системи з метою авторизації;
- Credit card transaction – здійснення транзакції з використанням кредитної картки, наприклад, покупка або зняття коштів;
- Check account balance – перевірка залишку на рахунку, пов’язаному з кредитною карткою;
- View statement of account – перегляд виписки по рахунку, яка містить перелік операцій;
- Change credit card PIN – зміна PIN-коду до картки для забезпечення безпеки;
- LOG OUT – вихід із системи після завершення сесії.

Кожен варіант використання (use case), представлений у вигляді еліпса, поєднаний стрілкою з актором. Це демонструє, що власник картки ініціює ці дії у системі.

Діаграма використовується для визначення функціональних вимог до системи з точки зору кінцевого користувача і є важливим інструментом на етапі проектування програмного забезпечення для банківської сфери. Вона дозволяє ідентифікувати основні сценарії використання, забезпечує наочність та підтримує розробників і аналітиків у створенні системи, орієнтованої на потреби користувачів.

3.2.2 Діаграма послідовності

UML-діаграма послідовності (Sequence Diagram) моделює взаємодію між об'єктами у часовій послідовності (рисунок 3.4).

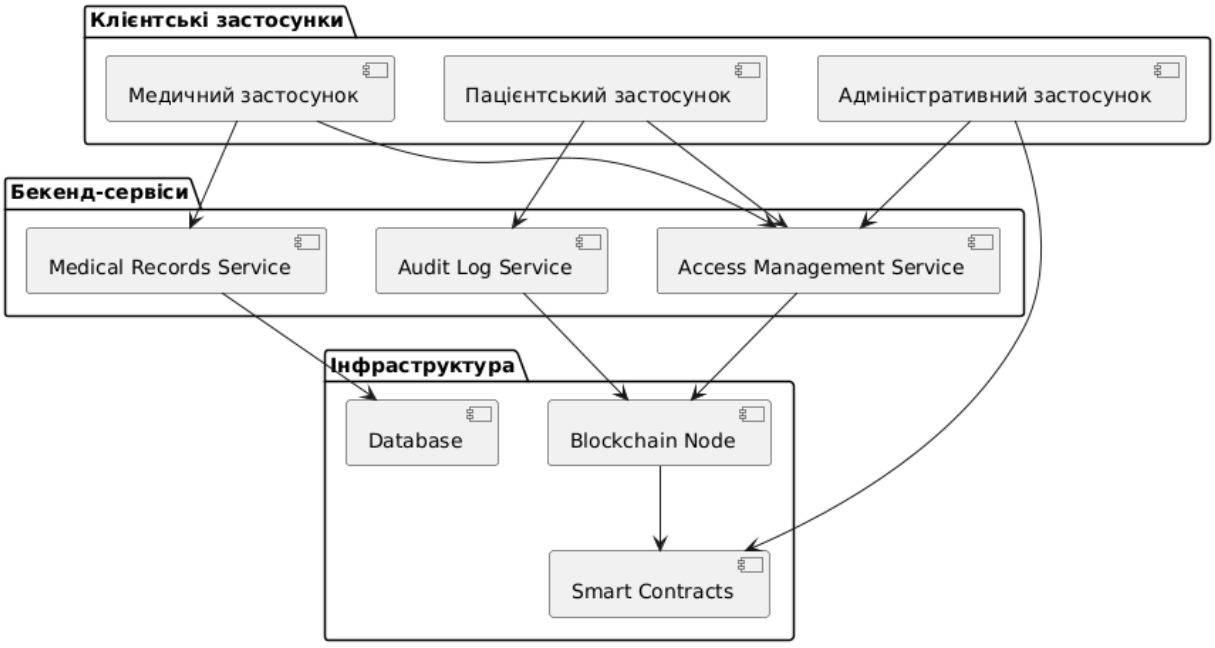

Рисунок 3.4 – Діаграма послідовності

На діаграмі послідовності моделюється логіка взаємодії учасників процесу виявлення шахрайських банківських транзакцій у часовій послідовності. У межах цього сценарію задіяні такі ключові компоненти:

користувач (клієнт), інтерфейс прикладного програмного забезпечення, система виявлення шахрайства, база даних транзакцій, аналітик з інформаційної безпеки та служба оповіщення.

Процес починається з ініціації транзакції користувачем через клієнтський застосунок. Введені дані передаються інтерфейсом до системи виявлення шахрайства, яка виконує аналіз за допомогою попередньо навченої моделі машинного навчання. Для цього система може звернутися до бази даних для отримання історичних транзакцій користувача, що є необхідним для коректної оцінки ризику операції.

Після обробки даних модель класифікує транзакцію за рівнем ризику. Якщо йдеться про операцію з низьким рівнем ризику, вона схвалюється автоматично, а результат повертається до інтерфейсу для інформування користувача про успішне виконання транзакції. У випадку, якщо рівень ризику перевищує допустимий поріг, транзакція маркується як підозріла й автоматично перенаправляється на ручну перевірку аналітику з безпеки. Одночасно клієнт інформується про затримку обробки за допомогою зовнішньої служби оповіщення.

Аналітик, отримавши відповідне повідомлення, здійснює перевірку операції та приймає рішення про її підтвердження або скасування. Після цього відповідь передається назад у систему, яка оновлює статус транзакції та передає остаточне рішення клієнту. Така архітектура забезпечує ефективне виявлення шахрайських дій, поєднуючи автоматизовану обробку з можливістю експертної перевірки, що дозволяє досягти високого рівня точності та безпеки в управлінні фінансовими операціями.

3.2.3 Діаграма класів

Діаграма класів (рисунок 3.5) системи виявлення шахрайських транзакцій відображає основну структуру програмного забезпечення, а також взаємозв’язки між його ключовими об’єктами. Вона моделює класи, їх

атрибути, методи та асоціації, які реалізують логіку функціонування системи.

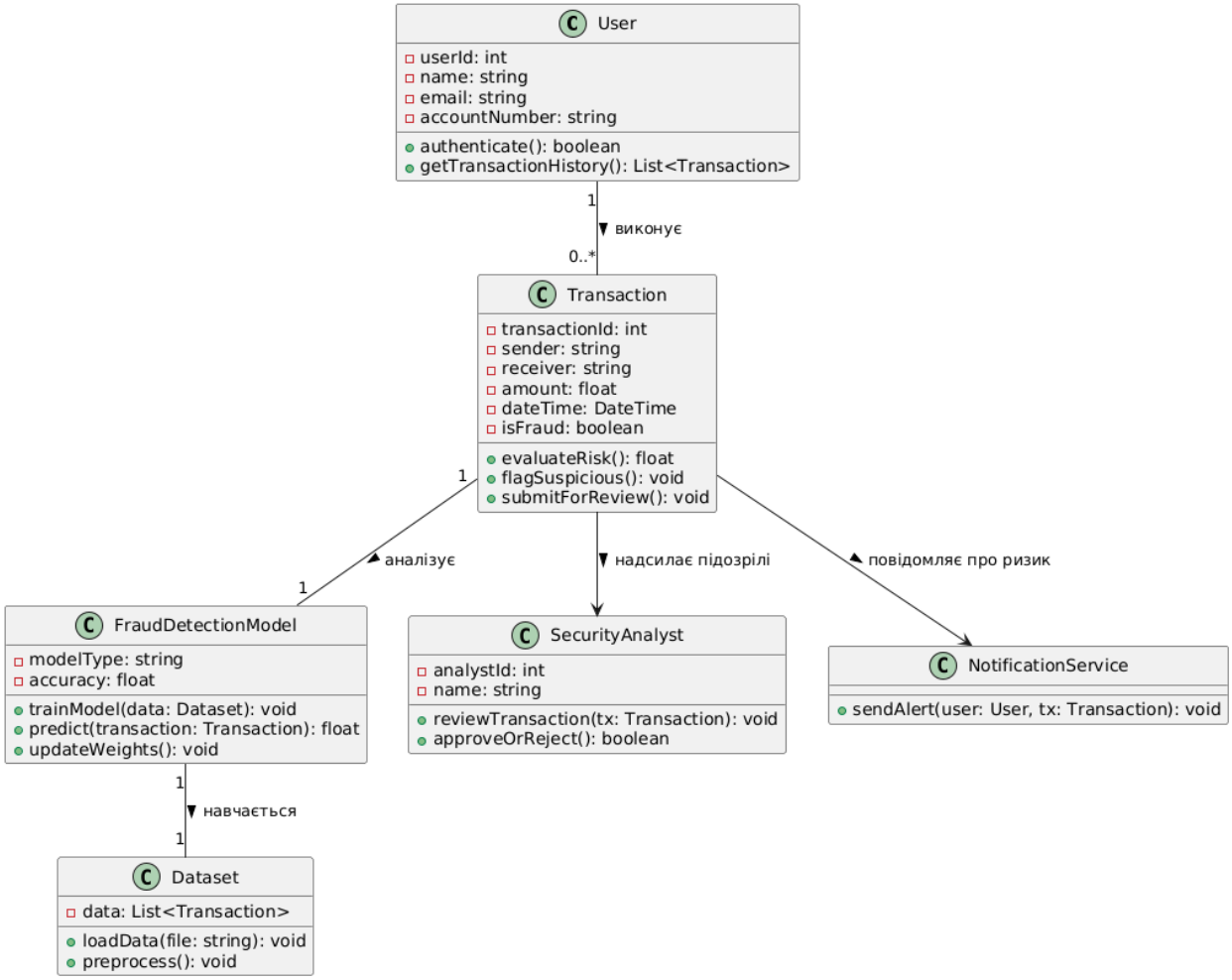

Рисунок 3.5 – Діаграма класів

Центральним елементом є клас Transaction, який містить такі атрибути, як ідентифікатор транзакції, сума, дата й час, відправник і одержувач, а також ознака «шахрайська/не шахрайська». До методів цього класу належать: evaluateRisk() – для розрахунку рівня ризику, flagSuspicious() – для маркування транзакції як підозрілої, та submitForReview() – для передачі операції на ручну перевірку.

Клас User представляє зареєстрованого користувача банківської системи. Він містить атрибути userId, name, accountDetails, а також методи автентифікації та доступу до історії транзакцій. Користувач пов’язаний з множиною об’єктів класу Transaction, що відображає можливість виконання

кількох транзакцій одним користувачем.

Клас FraudDetectionModel відповідає за машинне навчання і має методи trainModel() для навчання на історичних даних, predict() – для оцінки ризику нової транзакції, updateWeights() – для адаптації моделі до нових даних. Цей клас взаємодіє з класом Dataset, у якому зберігається навчальний набір транзакцій, а також методи завантаження й попередньої обробки даних (loadData(), preprocess()).

Клас SecurityAnalyst реалізує взаємодію людини з системою у разі, коли необхідна ручна перевірка. Він має методи reviewTransaction() та approveOrReject(), що дають змогу фахівцю приймати остаточне рішення щодо транзакції.

У системі також присутній клас NotificationService, який відповідає за інформування користувача. Він має метод sendAlert() для надсилання повідомлень у разі підозрілих дій.

Взаємозв’язки між класами представлені асоціаціями: транзакції належать користувачам, модель аналізує транзакції з використанням набору даних, а підозрілі транзакції надсилаються аналітику безпеки. Така структура забезпечує масштабованість, повторне використання коду та ефективну інтеграцію алгоритмів машинного навчання в середовище банківської безпеки.

3.2.4 Діаграма розгортання

Діаграма розгортання (рисунок 3.6) моделює фізичну структуру системи, зображуючи вузли (сервери, клієнтські пристрої) та компоненти програмного забезпечення, розгорнуті на цих вузлах.

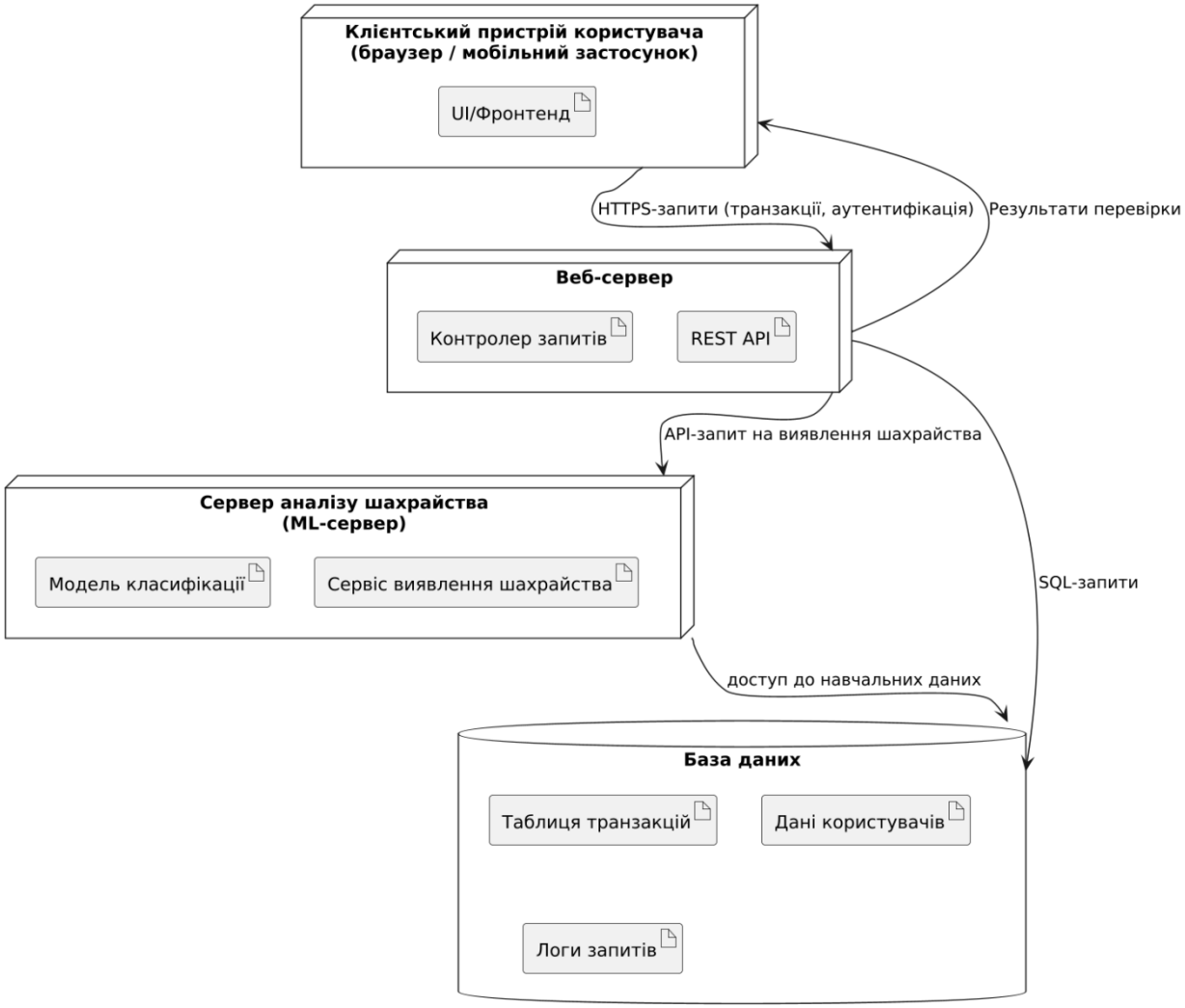

Рисунок 3.6 – Діаграма розгортання

Клієнтський пристрій користувача, через який ініціюється транзакція та відображається її статус.

Веб-сервер, що забезпечує обробку запитів, зв’язок із базою даних та передає інформацію до модуля аналізу.

Сервер моделей машинного навчання, що реалізує алгоритми виявлення шахрайства (наприклад, Random Forest або SVM).

Сервер бази даних, де зберігаються історичні транзакції, дані користувачів, налаштування та результати перевірки.

Система передбачає обмін даними між модулями через захищені інтерфейси API, із використанням протоколів HTTPS.

4 ДОСЛІДЖЕННЯ ПРОТОТИПУ СИСТЕМИ

4.1 Експериментальна платформа

Технічна реалізація завдання, викладеного в даній роботі, здійснювалася з використанням мови програмування високого рівня Python. Завдяки простоті синтаксису, гнучкості, підтримці різноманітних бібліотек і фреймворків для штучного інтелекту та машинного навчання, платформній незалежності, а також великій спільноті розробників, Python вважається ідеальним інструментом для реалізації проєктів, пов’язаних з інтелектуальним аналізом даних.

Як середовище розробки було обрано інтерактивні блокноти (notebooks) на платформі Kaggle, звідки також був отриманий набір даних. Kaggle-ноутбуки являють собою послідовність осередків, які можуть містити текст, оформлений у форматі Markdown, або програмний код мовою Python. Це дозволяє ефективно організувати логіку обробки, візуалізації та аналізу даних безпосередньо у веббраузері, без необхідності встановлення програмного забезпечення на локальний комп’ютер користувача.

У процесі реалізації було використано такі основні бібліотеки:

- Scikit-learn – бібліотека з відкритим кодом для машинного навчання, яка підтримує як контрольоване (supervised), так і неконтрольоване (unsupervised) навчання. Вона надає широкий спектр інструментів для побудови моделей, попередньої обробки даних, вибору моделей, їх оцінювання, а також оптимізації параметрів;
- Pandas – потужна бібліотека для обробки табличних та часових даних. Вона забезпечує високу продуктивність завдяки інтеграції з бібліотекою NumPy та є однією з найзручніших у використанні для маніпуляції великими обсягами структурованих даних;
- Matplotlib – бібліотека для побудови графіків і візуалізації даних. Вона

є аналогом MATLAB для мови Python і забезпечує сумісність із NumPy. API бібліотеки дозволяє легко вбудовувати графіки у графічні інтерфейси та забезпечує кросплатформність розробки.

Оскільки реалізація проєкту здійснювалася безпосередньо в середовищі Kaggle, виконання коду можливе у браузері будь-якого користувача, що усуває необхідність у високопродуктивному локальному обладнанні. Достатньо лише стабільного з'єднання з Інтернетом.

Після завантаження набору даних у структуру типу DataFrame (що надається бібліотекою Pandas), відбувається його послідовна обробка відповідно до описаних раніше методів: стандартизація ознак та випадкове зменшення вибірки (random undersampling) для балансування класів. Стандартизація здійснюється за допомогою стандартного скейлера, а undersampling дозволяє уникнути перекосу моделі на користь домінуючого класу.

Для оцінювання здатності моделей до узагальнення, вхідний набір даних було розділено на три підмножини: навчальну (70%), валідаційну (15%) та тестову (15%). Такий підхід забезпечує об'єктивну оцінку ефективності обраних моделей та дає змогу налаштувати їхні гіперпараметри з урахуванням продуктивності на валідаційному наборі.

Усі обрані моделі навчаються з використанням відповідних гіперпараметрів, що дозволяє підвищити точність виявлення шахрайських банківських транзакцій та покращити загальну надійність запропонованого програмного рішення.

4.2 Експериментальні данні

У процесі реалізації дослідження було використано три загальнодоступні набори даних, що присвячені виявленню шахрайських транзакцій з банківськими картками. Всі вони були отримані з платформи Kaggle, яка є одним із провідних ресурсів у сфері аналізу даних та машинного

навчання. Кожен із цих наборів має свої особливості щодо обсягу, ступеня збалансованості класів та джерела даних.

Перший набір даних — Credit Card Fraud Detection обсягом 150.83 МБ (джерело: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud, дата доступу: 15 листопада 2024 року). Він містить 284 807 транзакцій, що відбулися протягом двох днів. З них лише 492 транзакції класифіковані як шахрайські, що свідчить про сильну дисбалансованість класів — близько 0.17% даних становлять позитивні приклади шахрайства. Через таку нерівномірність даних особливого значення набувають методи обробки дисбалансованих вибірок, зокрема методи undersampling та oversampling, що були застосовані у межах даного дослідження.

Другий набір — Credit Card Fraud обсягом 76.28 МБ (джерело: https://www.kaggle.com/datasets/dhanushnarayananr/credit-card-fraud, дата доступу: 15 листопада 2024 року). Цей набір є змодельованим (симульованим), що дозволяє проводити тестування алгоритмів без ризику роботи з чутливою персональною інформацією. В обговоренні на сторінці датасету (URL: https://www.kaggle.com/datasets/dhanushnarayananr/credit-card-fraud/discussion/335338) зазначено, що один із розв’язків на основі цього датасету демонструє точність, що дорівнює одиниці. Такий результат є непридатним для реального використання, оскільки свідчить про імовірну надмірну узгодженість даних, яка спотворює об’єктивність моделей машинного навчання.

Третій набір — Fraud Detection—Credit Card обсягом 102.92 МБ (джерело: https://www.kaggle.com/datasets/yashpaloswal/fraud-detection-credit-card, дата доступу: 15 листопада 2024 року). Він є похідним від першого набору: було вилучено записи з пропущеними значеннями. Незважаючи на менший

розмір, структура даних та розподіл класів повністю відповідають першому набору, що зумовлює їх подібне використання.

З трьох описаних наборів для проведення основного дослідження було обрано Credit Card Fraud Detection (перший набір), оскільки він містить повні, автентичні транзакційні дані, які найкраще ілюструють реальні виклики у виявленні шахрайства. До того ж, цей набір активно використовується в науковій та практичній спільноті: на його основі створено понад 4050 інтерактивних ноутбуків у середовищі Kaggle. Це забезпечує можливість прямого порівняння отриманих результатів із вже існуючими методами, що є критично важливим для оцінки ефективності запропонованої моделі.

4.3 Метрики оцінювання

Для оцінювання ефективності моделей, запропонованих у даному дослідженні, було використано криву робочих характеристик приймача (ROC-криву) та площу під цією кривою (AUC). ROC-крива є графічним представленням точності класифікаційної моделі за всіма можливими порогами класифікації. Вона відображає співвідношення між показником істинно позитивних результатів (True Positive Rate, TPR) та хибнопозитивних результатів (False Positive Rate, FPR).

Показник TPR (або повнота, recall) визначається як відношення кількості істинно позитивних передбачень до суми істинно позитивних і хибнонегативних результатів. Математично це виражається формулою:

$$TPR = \frac{TP}{TP + FN},$$

де TP — кількість об'єктів, правильно класифікованих як позитивні,
FN — кількість позитивних об'єктів, які були класифіковані як негативні.

Показник FPR демонструє частку негативних об’єктів, помилково класифікованих як позитивні. Це співвідношення хибнопозитивних результатів до загальної кількості дійсно негативних зразків (включаючи як правильно, так і неправильно класифіковані). Визначається як:

$$FPR = \frac{FP}{FP + TN},$$

де FP — кількість об'єктів, помилково класифікованих як позитивні,
TN — кількість об'єктів, правильно класифікованих як негативні.

Побудова ROC-кривої здійснюється шляхом нанесення значень TPR на осі Y та значень FPR на осі X для кожного порогу класифікації. Зменшення порогу класифікації призводить до того, що більша кількість прикладів класифікується як позитивні, що, своєю чергою, збільшує як кількість істинно позитивних, так і хибнопозитивних передбачень. Такий підхід дає змогу оцінити здатність моделі розрізняти класи незалежно від обраного порогу.

Площа під ROC-кривою — AUC (Area Under Curve) — є загальноприйнятим інтегральним показником якості моделі. Значення AUC відображає ймовірність того, що модель присвоїть вищий прогнозний бал випадковому позитивному прикладу, ніж випадковому негативному. Ідеальна модель має AUC = 1.0, тоді як модель з випадковими прогнозами — AUC = 0.5.

Перевага AUC полягає в тому, що вона не залежить від масштабу (scale-independent) і не змінюється при зміні порогу (threshold-invariant), що дозволяє коректно порівнювати ефективність різних моделей або моделей на різних наборах даних. Однак у деяких сценаріях ці переваги можуть виявитися недоліками. Зокрема, в задачах, де критично важливо мати добре калібровані ймовірнісні передбачення, AUC може не бути адекватним показником. Також, якщо вартість хибнопозитивних результатів значно перевищує вартість хибнонегативних (або навпаки), AUC не враховує ці пріоритети. Наприклад, у

системах виявлення спаму важливішим може бути мінімізація кількості хибнопозитивних результатів, тобто ситуацій, коли легітимне повідомлення помилково класифіковано як спам.

Таким чином, використання ROC-кривої та AUC як метрик у цьому дослідженні є доцільним і забезпечує об'єктивну оцінку здатності моделей розрізняти шахрайські та легітимні банківські транзакції, однак для практичних впроваджень варто також враховувати контекст використання моделей.

4.4 Аналіз результатів

У цьому дослідженні для вирішення задачі класифікації було використано сім основних алгоритмів машинного навчання: дерево рішень, логістична регресія, метод опорних векторів (SVC), метод k найближчих сусідів (KNN), стохастичний градієнтний спуск (SGD), наївний баєсівський класифікатор та випадковий ліс (Random Forest). Після ініціалізації моделей та попередньої обробки даних було проведено налаштування гіперпараметрів за допомогою методу Grid Search, що забезпечило оптимізацію кожного з моделей у межах обраного простору параметрів.

На наступному етапі виконувалося поетапне тренування моделей та оцінювання їх ефективності за допомогою AUC-метрики (Area Under the Curve). Для візуального представлення якості класифікації було побудовано ROC-криві для кожного алгоритму. На рисунку 4.1 наведено графіки ROC-кривих, які демонструють роботу кожної з моделей.

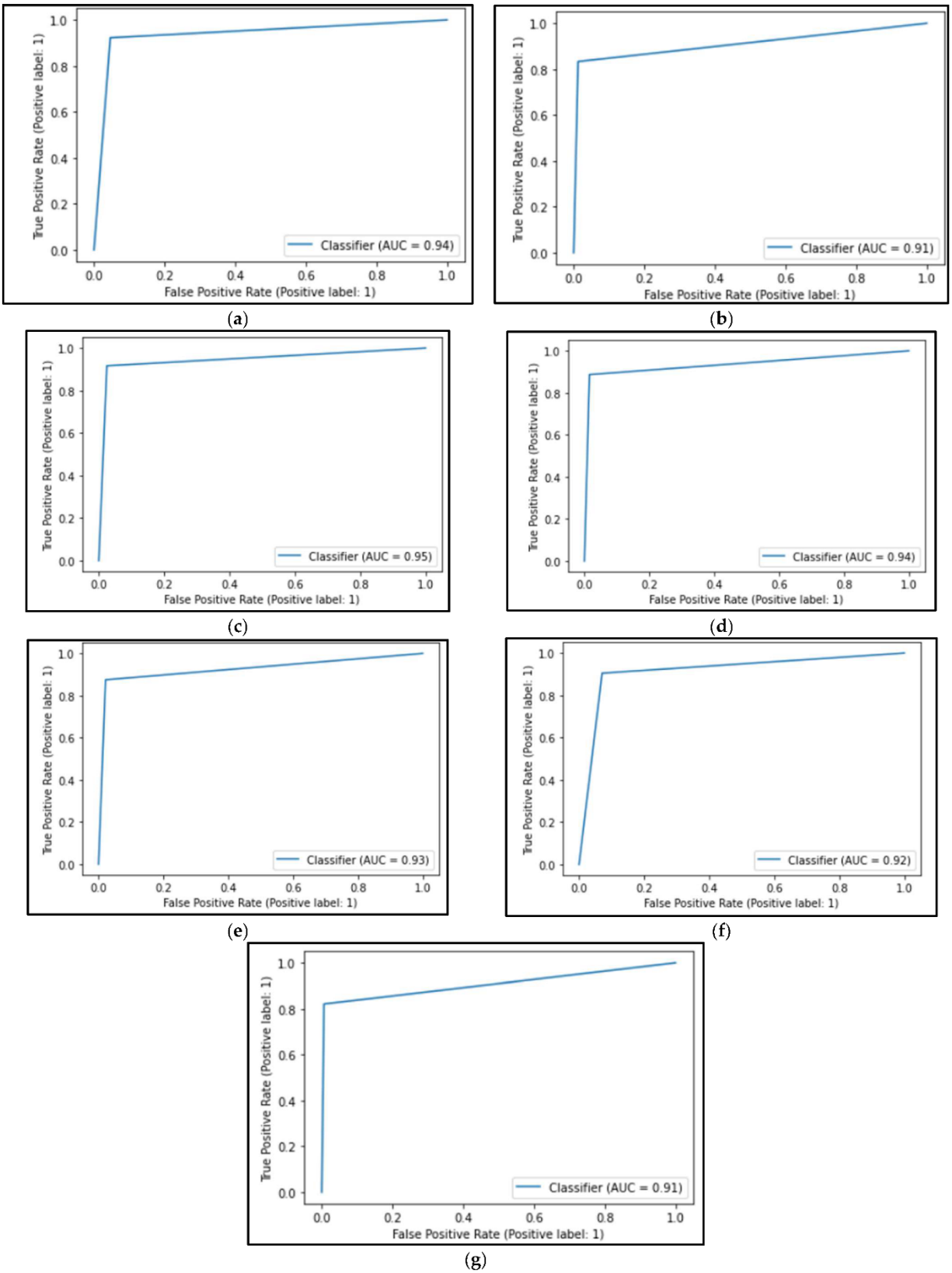

Рисунок 4.1 - Графіки ROC-кривих наступних алгоритмів: (a) алгоритму дерева рішень; (b) алгоритму випадкового лісу; (c) алгоритму логістичної регресії; (d) алгоритму SVC; (e) алгоритму k-найближчих сусідів; (f) алгоритму SGD; (g) наївного алгоритму Байєса.

Результати метрики AUC для відповідних алгоритмів були такими:

- дерево рішень — 0.938;
- логістична регресія — 0.946;
- метод опорних векторів (SVC) — 0.936;
- метод k найближчих сусідів — 0.927;
- SGD-класифікатор — 0.917;
- наївний баєсівський класифікатор — 0.908;
- випадковий ліс — 0.911.

Загальний висновок після завершення класифікаційного процесу полягав у тому, що найкращим результатом за метрикою AUC серед базових моделей відзначилася логістична регресія, яка досягла значення AUC ≈ 0.946, що також підтверджено програмним виводом (рисунок 4.2).

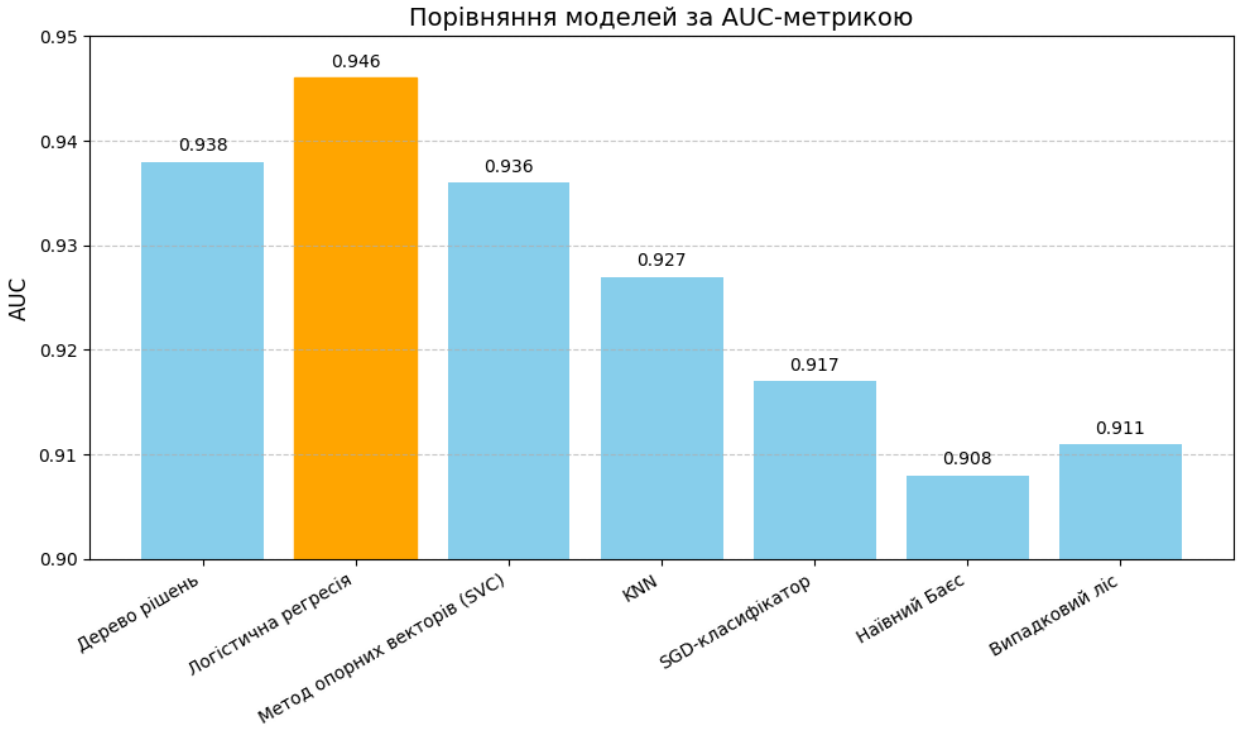

Рисунок 4.2 – Порівняння моделей

Незважаючи на високу ефективність логістичної регресії, подальший аналіз показав, що модель ансамблевого навчання (stacked generalization) забезпечила ще кращі результати. Підсумкова таблиця результатів моделей,

включно з оцінками AUC та F1-мірою, представлена на рисунку 4.3. Було встановлено, що модель ансамблю досягла найвищої F1-міри – 0.96, що перевищує показники інших реалізацій, включно з результатами провідних проєктів на Kaggle.

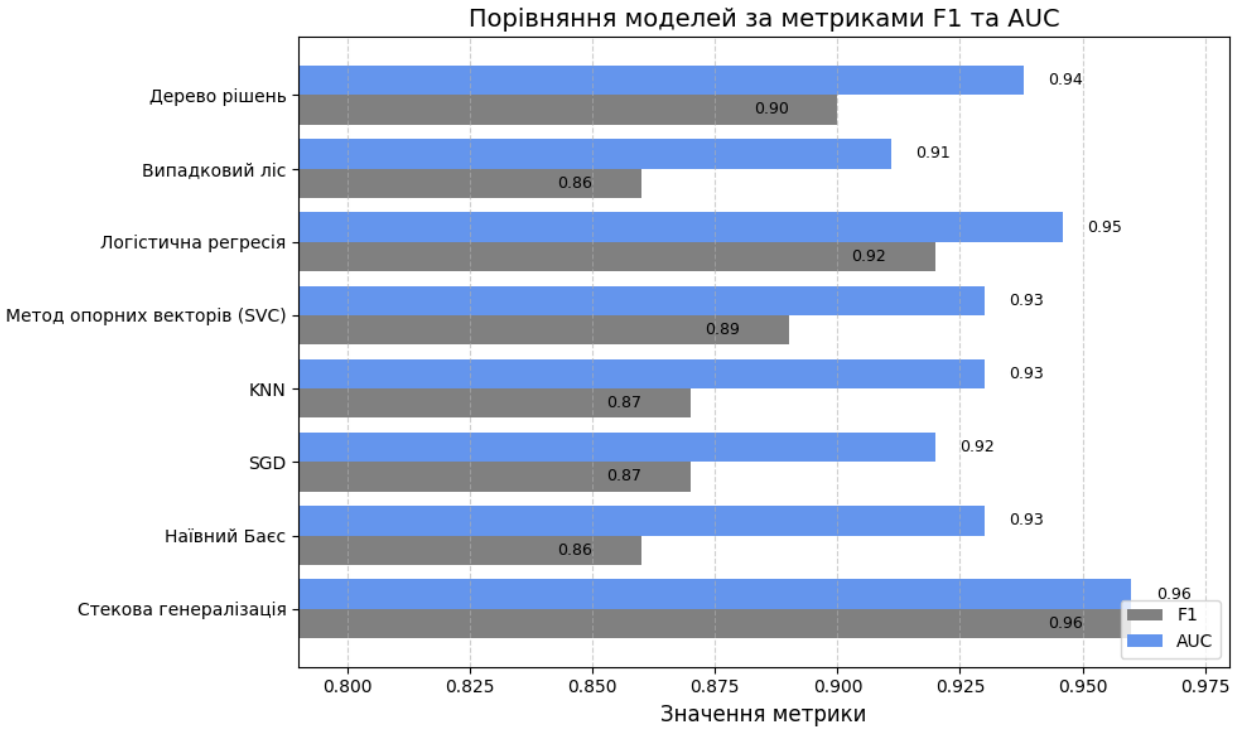

Рисунок 4.3 – Порівняння моделей

Технічна реалізація програмного рішення була здійснена мовою Python, що є однією з провідних мов у галузі штучного інтелекту завдяки гнучкості, простоті синтаксису та наявності потужних бібліотек, таких як Pandas, Scikit-learn та Matplotlib. Реалізація проводилася на платформі Kaggle, яка забезпечує середовище запуску Jupyter notebooks без потреби у потужному локальному обладнанні — достатньо лише стабільного з'єднання з Інтернетом та браузера.

Загалом, на основі серії експериментів було встановлено, що всі алгоритми показують порівняно високі результати в задачі виявлення шахрайських банківських транзакцій. Незначні візуальні відмінності ROC-кривих підтверджують рівень конкурентоспроможності обраних методів.

ВИСНОВКИ

У межах даної кваліфікаційної роботи було здійснено розробку та дослідження програмних компонентів для системи виявлення шахрайських банківських транзакцій із використанням методів машинного навчання. Зважаючи на стрімке зростання кількості онлайн-транзакцій і водночас — збільшення ризику фінансового шахрайства, запропонована система є актуальною відповіддю на сучасні виклики у сфері фінансової безпеки.

У процесі дослідження було проаналізовано предметну область, виконано огляд сучасних підходів до виявлення шахрайства в банківських системах, розглянуто існуючі методи машинного навчання, що застосовуються для класифікації фінансових транзакцій. Для реалізації задачі було обрано низку ефективних алгоритмів, зокрема: логістичну регресію, дерево рішень, випадковий ліс, метод опорних векторів, наївний баєсівський класифікатор, k-ближчих сусідів, SGD-класифікатор. Було здійснено підбір гіперпараметрів із використанням методу GridSearchCV.

В якості експериментальної бази були використані три датасети, серед яких основним став набір Credit Card Fraud Detection з платформи Kaggle. Було проведено попередню обробку даних, включно зі стандартизацією ознак та балансуванням класів методом випадкового зменшення кількості записів більшості. Для оцінювання ефективності моделей використовувались метрики AUC (Area Under Curve) та F1-міра, а також побудовані ROC-криві.

Найвищу точність і ефективність продемонструвала модель логістичної регресії з AUC = 0.946, а також метод стекової генералізації, який забезпечив найкраще значення F1 = 0.96. Це свідчить про доцільність використання ансамблевих моделей для виявлення шахрайських транзакцій в умовах незбалансованих даних.

Запропоноване програмне рішення реалізовано мовою Python із використанням середовища виконання Kaggle Notebooks, що забезпечує зручність, масштабованість і незалежність від ресурсів локального

комп’ютера. Розроблена система може бути інтегрована у програмне забезпечення банківських установ для автоматизованого виявлення підозрілих транзакцій в реальному часі.

Таким чином, у роботі досягнуто поставленої мети: реалізовано ефективну систему класифікації фінансових транзакцій, підтверджено доцільність застосування алгоритмів машинного навчання у сфері банківської безпеки, обґрунтовано вибір моделей та здійснено їх оцінку. Отримані результати можуть бути використані як основа для подальших наукових розвідок і практичних впроваджень у сфері фінансових технологій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Jansen J., Leukfeldt R. How people help fraudsters steal their money: An analysis of 600 online banking fraud cases // Proceedings of the Workshop on Socio-Technical Aspects in Security and Trust. Verona, Italy, 13 July 2015. P. 24–31.

2. Top 5 Banking Fraud Prevention Methods, SailPoint [Електронний ресурс]. – Режим доступу: https://www.sailpoint.com/identity-library/top-5-banking-fraud-prevention-methods/ (дата звернення: 15.11.2024).

3. Law B. Bank Fraud—Definitions & Penalties, Berry Law, 24 October 2017 [Електронний ресурс]. – Режим доступу: https://jsberrylaw.com/blog/bankfraud-definition-penalties/.

4. Scopus. Search “Fraudulent Banking” [Електронний ресурс]. – Режим доступу: https://www.scopus.com .

5. Barker R. The use of proactive communication through knowledge management to create awareness and educate clients on e-banking fraud prevention // South African Journal of Business Management. 2020. Vol. 51. Article a1941.

6. Abidoye A.P., Kabaso B. Hybrid machine learning: A tool to detect phishing attacks in communication networks // International Journal of Advanced Computer Science and Applications. 2020. Vol. 11. P. 559–569.

7. Shah S.S.H. et al. Memory forensics-based malware detection using computer vision and machine learning // Electronics. 2022. Vol. 11. Article 2579.

8. Maulana L.R., Fajar A.N., Meyliana. Extending the design of smart mobile application to detect fraud theft of E-banking access using big data analytic and SOA // Proceedings of the 2021 IEEE 5th Int. Conf. on Information Technology, Information Systems and Electrical Engineering (ICITISEE). Purwokerto, Indonesia, 24–25 Nov. 2021. P. 360–364.

9. Khalaf Al Hattali S.S., Hussain S.M., Frank A. Design and development for detection and prevention of ATM skimming frauds // Indonesian Journal of Electrical Engineering and Computer Science. 2019. Vol. 17. P. 1224–1231.

10. Tsai C., Su P. The application of multi-server authentication scheme in internet banking transaction environments // Information Systems and e-Business Management. 2021. Vol. 19. P. 77–105.

11. Hammi B. et al. Blockchain-based solution for detecting and preventing fake check scams // IEEE Transactions on Engineering Management. 2022. Vol. 69. P. 3710–3725.

12. Abdul Rani M.I. et al. A systematic literature review of money mule: Its roles, recruitment and awareness // Journal of Financial Crime. 2023. (ahead-of-print).

13. Ileberi E., Sun Y., Wang Z. A machine learning based credit card fraud detection using the GA algorithm for feature selection // Journal of Big Data. 2022. Vol. 9. Article 24.

14. Chaquet-Ulldemolins J. et al. On the Black-Box Challenge for Fraud Detection Using Machine Learning (I): Linear Models and Informative Feature Selection // Applied Sciences. 2022. Vol. 12. Article 3328.

15. Kasasbeh B. et al. Multilayer perceptron artificial neural networks-based model for credit card fraud detection // Indonesian Journal of Electrical Engineering and Computer Science. 2022. Vol. 26. P. 362–373.

16. Nguyen N. et al. A Proposed Model for Card Fraud Detection Based on CatBoost and Deep Neural Network // IEEE Access. 2022. Vol. 10. P. 96852–96861.

17. Esenogho E. et al. A Neural Network Ensemble with Feature Engineering for Improved Credit Card Fraud Detection // IEEE Access. 2022. Vol. 10. P. 16400–16407.

18. Sharma P. et al. Machine learning model for credit card fraud detection – A comparative analysis // International Arab Journal of Information Technology. 2021. Vol. 18. P. 789–796.

19. Benchaji I. et al. Credit card fraud detection model based on LSTM recurrent neural networks // Journal of Advanced Information Technology. 2021. Vol. 12. P. 113–118.

20. Mehbodniya A. et al. Financial Fraud Detection in Healthcare Using

Machine Learning and Deep Learning Techniques // Security and Communication Networks. 2021. Article ID 9293877.

21. Cauteruccio F., Terracina G., Ursino D. Generalizing identity-based string comparison metrics: Framework and techniques // Knowledge-Based Systems. 2020. Vol. 187. Article 104820.

22. Ojagh S. et al. Enhanced air quality prediction by edge-based spatiotemporal data pre-processing // Computers and Electrical Engineering. 2021. Vol. 96. Article 107572.

23. Arora M., Bhardwaj I. Artificial Intelligence in Collaborative Information System // International Journal of Modern Education and Computer Science (IJMECS). 2022. Vol. 14. P. 44–55.

24. Junejo M. et al. Quality of Experience Assessment of Banking Service // International Journal of Information Engineering and Electronic Business (IJIEEB). 2020. Vol. 12. P. 39–50.

25. Gupta P. et al. Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques // Procedia Computer Science. 2023. Vol. 218. P. 2575–2584.

26. Navaneethakrishnan P., Viswanath R. Fraud Detection on Credit Cards Using Artificial Intelligence Methods // Ilkogretim Online – Elementary Education Online. 2020. Vol. 19. P. 2086–2096.

27. Khan M., Mahmood W. Technology Adoption in Pakistani Banking Industry using UTAUT // International Journal of Information Technology and Computer Science (IJITCS). 2022. Vol. 14. P. 32–42.

28. Zimba A. A Bayesian Attack-Network Modeling Approach to Mitigating Malware-Based Banking Cyberattacks // International Journal of Computer Network and Information Security (IJCNIS). 2022. Vol. 14. P. 25–39.

29. Elhassan R., Yousif A., Suliman T. Assessment of Knowledge Management Application in Banking Sector of Sudan: Case Study Farmer's Commercial Bank // International Journal of Information Engineering and Electronic Business (IJIEEB). 2021. Vol. 13. P. 1–19.