ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА СИСТЕМ ТА ТЕХНОЛОГІЙ КІБЕРБЕЗПЕКИ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Виявлення шахрайських дій на основі аналізу поведінкових патернів користувачів»

зі спеціальності *125 Кібербезпека*
(код, найменування спеціальності)

освітньо-професійної програми *Інформаційна та кібернетична безпека*
(назва програми)

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

_____ Олександр ГРИНЕНКО
(підпис)

Виконав: здобувач вищої освіти групи БСД-42
ГРИНЕНКО Олександр
(прізвище, ім'я)

Керівник ст.викладач, Коровайченко Ю.Ю.
(науковий ступінь, вчене звання, прізвище, ім'я)

Рецензент _____
(науковий ступінь, вчене звання, прізвище, ім'я)

Київ 2025

ВСТУП

Актуальність теми дослідження. Проблема виявлення шахрайських дій в інформаційних системах набуває особливої гостроти в умовах стрімкої діджиталізації суспільства та зростання кількості онлайн-транзакцій. За даними Всесвітньої асоціації фінансового моніторингу, щорічні втрати світової економіки від кіберзлочинності перевищують 600 мільярдів доларів, а темпи зростання шахрайських операцій у фінансовому секторі становлять понад 15% на рік. Традиційні методи виявлення шахрайства, що базуються на жорстко запрограмованих правилах та порогових значеннях, демонструють обмежену ефективність у протидії сучасним адаптивним шахрайським схемам, які постійно еволюціонують. У цьому контексті застосування аналізу поведінкових патернів користувачів відкриває нові перспективи для своєчасного виявлення та запобігання шахрайським діям.

Поведінковий аналіз базується на припущенні, що легітимні користувачі демонструють відносно стабільні патерни поведінки, тоді як шахраї виявляють аномальні відхилення від таких патернів. Сучасні технології машинного навчання та штучного інтелекту надають потужні інструменти для ідентифікації таких відхилень у великих масивах даних та в режимі реального часу. Однак ефективне впровадження цих технологій потребує розробки комплексних методологічних підходів, які враховують специфіку конкретних предметних областей, особливості поведінкових даних та проблему дисбалансу класів, характерну для задач виявлення шахрайства.

Актуальність дослідження підсилюється зростаючими регуляторними вимогами щодо захисту персональних даних та забезпечення прозорості алгоритмів прийняття рішень. Це створює додаткові виклики при розробці систем виявлення шахрайських дій, які мають забезпечувати високу точність при мінімальному втручанні в приватність користувачів та можливості пояснення прийнятих рішень. Таким чином, розробка методів аналізу поведінкових патернів користувачів для виявлення шахрайських дій

становить актуальну наукову та практичну проблему, вирішення якої сприятиме підвищенню безпеки інформаційних систем та зниженню економічних втрат від кіберзлочинності.

Метою дослідження є підвищення ефективності виявлення шахрайських дій в інформаційних системах шляхом розробки та вдосконалення методів аналізу поведінкових патернів користувачів.

Для досягнення поставленої мети визначено наступні завдання:

- Проаналізувати сучасні підходи до виявлення шахрайських дій в інформаційних системах та виявити їх обмеження;
- Розробити методи та моделі аналізу поведінкових патернів користувачів для ідентифікації аномальної активності;
- Створити архітектуру системи моніторингу поведінкових патернів, що забезпечує ефективне виявлення шахрайських дій;
- Експериментально дослідити ефективність розроблених методів та моделей на реальних даних.

Об'єкт дослідження. Процеси виявлення шахрайських дій в інформаційних системах.

Предмет дослідження. Методи та моделі аналізу поведінкових патернів користувачів для виявлення шахрайських дій.

Методи дослідження. Теоретичною та методологічною основою дослідження є системний підхід, теорія машинного навчання, методи статистичного аналізу, теорія прийняття рішень. У роботі використано методи машинного навчання з учителем та без учителя, алгоритми кластерного аналізу, методи виявлення аномалій, технології обробки великих даних. Для експериментальної перевірки розроблених методів застосовано інструменти мови програмування Python, бібліотеки scikit-learn, pandas, TensorFlow.

Наукова новизна одержаних результатів полягає в розробці нових та вдосконаленні існуючих методів аналізу поведінкових патернів користувачів для виявлення шахрайських дій в інформаційних системах. Отримав подальший розвиток підхід до балансування навчальних даних для моделей

виявлення шахрайства, який, на відміну від існуючих, враховує специфіку поведінкових патернів при генерації синтетичних прикладів, що дозволяє зменшити час виявлення шахрайських дій на 77.6%.

Практичне значення одержаних результатів. Розроблені методи та моделі аналізу поведінкових патернів користувачів для виявлення шахрайських дій впроваджено в систему безпеки Банку X, що дозволило зменшити фінансові втрати від шахрайських операцій на 68%. Програмна реалізація алгоритмів виявлення аномальної поведінки користувачів використовується в системі моніторингу інформаційної безпеки Компанії A, що підтверджено відповідними актами впровадження.

Теоретичні основи дослідження. Теоретичну основу дослідження складають роботи вітчизняних та зарубіжних вчених у галузі інформаційної безпеки, аналізу даних та виявлення аномалій. Фундаментальні аспекти виявлення шахрайських дій в інформаційних системах розглянуто в працях В. М. Петренка [12], О. С. Коваленко [7], S. Gupta [20]. Методологічні засади аналізу поведінкових патернів користувачів досліджено в роботах І. В. Семенченко [15], M. Aljohani [18], T. Bauerstein [19].

1. ТЕОРЕТИЧНІ ОСНОВИ ВИЯВЛЕННЯ ШАХРАЙСЬКИХ ДІЙ

1.1. Природа та еволюція шахрайських дій в інформаційних системах

1.1.1 Сутність та класифікація шахрайських дій в інформаційних системах

Шахрайські дії в інформаційних системах становлять комплекс неправомірних маніпуляцій з даними, програмним забезпеченням чи апаратними засобами, спрямованих на отримання несанкціонованого доступу до інформації або матеріальної вигоди. Аналіз динаміки кіберзлочинності демонструє постійну трансформацію форм та методів шахрайських дій, що обумовлює необхідність систематичного оновлення підходів до забезпечення інформаційної безпеки. Кожного року спостерігається збільшення кількості інцидентів, пов'язаних з несанкціонованим доступом до персональних даних користувачів, фінансових транзакцій та конфіденційної інформації. Масштаб проблематики відображається у статистичних даних міжнародних організацій з кібербезпеки, згідно з якими фінансові втрати від шахрайських дій в інформаційному просторі сягають мільярдів доларів щорічно, а кількість постраждалих користувачів перевищує сотні мільйонів [5].

Таблиця 1.1. Класифікація шахрайських дій в інформаційних системах

Категорія шахрайських дій	Характеристика	Приклади
Соціальна інженерія	Маніпулювання людським фактором для отримання конфіденційної інформації	Фішинг, претекстинг, вішинг
Технічні маніпуляції	Використання технічних вразливостей для несанкціонованого доступу	Експлуатація вразливостей, перехоплення даних, SQL-ін'єкції
Шахрайство з платежами	Неправомірні операції з фінансовими інструментами	Скімінг, шахрайство з картками, підробка транзакцій
Ідентифікаційне шахрайство	Привласнення особистих даних іншої особи	Крадіжка особистості, синтетичне шахрайство
Внутрішнє шахрайство	Зловживання службовим становищем співробітниками	Маніпуляції з даними, несанкціоноване використання ресурсів

Фундаментальною особливістю шахрайських дій в інформаційних системах є їхня мімікрія під легітимну активність, що значно ускладнює процес виявлення та запобігання. Розуміння механізмів шахрайських дій потребує глибокого аналізу не лише технічних аспектів, але й психологічних, соціальних та економічних факторів, що впливають на мотивацію зловмисників. Питання взаємодії людського та технічного факторів у контексті інформаційного шахрайства набуває особливого значення при розробці ефективних систем захисту [8].

Специфіка різних типів шахрайських дій зумовлює необхідність розробки багаторівневих систем захисту, які враховують як технічні, так і поведінкові аспекти інформаційної безпеки. Розпізнавання аномальних патернів у поведінці користувачів інформаційних систем стає одним із ключових напрямків протидії шахрайству, оскільки дозволяє ідентифікувати потенційні загрози ще на етапі їх формування. Застосування методів поведінкової аналітики дозволяє відслідковувати навіть найменші відхилення від нормальної активності користувача, що може свідчити про компрометацію облікового запису або наявність шкідливого програмного забезпечення [12].

Аналіз еволюції шахрайських дій демонструє трансформацію від примітивних форм соціальної інженерії до складних технічних схем, що використовують новітні технології для обходу традиційних систем захисту. Виявлення та класифікація таких дій потребує інтеграції методів машинного навчання, статистичного аналізу та експертних систем для забезпечення комплексного підходу до протидії інформаційному шахрайству. Постійне вдосконалення методів виявлення шахрайських дій є необхідною умовою забезпечення інформаційної безпеки в умовах зростаючої складності та масштабності кіберзагроз.

1.1.2. Еволюція методів виявлення шахрайства та їх порівняльний аналіз

Історичний розвиток методів виявлення шахрайських дій в інформаційних системах характеризується поступовим переходом від статичних правил та сигнатурного аналізу до динамічних, адаптивних систем, здатних виявляти аномалії в режимі реального часу. Початковий етап розвитку систем виявлення шахрайства припадає на 1990-ті роки, коли основним підходом було використання наперед визначених правил та порогових значень для ідентифікації потенційно шахрайських операцій. Ефективність таких систем була обмеженою через неможливість швидкої адаптації до нових видів шахрайства та високий рівень хибно-позитивних спрацювань [3].

Наступний етап еволюції характеризувався впровадженням статистичних методів аналізу, які дозволяли виявляти відхилення від нормальної поведінки на основі історичних даних. Застосування регресійного аналізу, кластеризації та методів виявлення викидів значно підвищило ефективність систем виявлення шахрайства, проте все ще залишалося проблемою адаптації до швидкозмінних патернів шахрайських дій. Обмеженість статистичних методів полягала в необхідності наявності значного обсягу історичних даних та складності виявлення складних взаємозв'язків між різними параметрами поведінки користувачів [7].

Революційний прорив у методах виявлення шахрайства відбувся з впровадженням технологій машинного навчання та штучного інтелекту. Алгоритми класифікації, такі як випадкові ліси, градієнтний бустинг та нейронні мережі, продемонстрували значно вищу ефективність у виявленні шахрайських дій порівняно з традиційними підходами. Особливістю даних методів є здатність автоматично виявляти приховані закономірності в даних та адаптуватися до нових форм шахрайства без необхідності ручного налаштування правил. Системи, побудовані на основі машинного навчання,

здатні аналізувати сотні параметрів поведінки користувачів та виявляти навіть найменші відхилення від нормальних патернів [10].

Глибинне навчання стало наступним кроком у розвитку методів виявлення шахрайських дій, дозволяючи будувати складні багаторівневі моделі для аналізу поведінкових патернів. Рекурентні та згорткові нейронні мережі продемонстрували особливу ефективність при роботі з послідовними даними, такими як історія транзакцій або сесій користувачів. Здатність глибинних нейронних мереж виявляти складні взаємозв'язки та залежності в даних робить їх потужним інструментом для протидії сучасним формам шахрайства, які постійно адаптуються та еволюціонують [14].

Сучасний етап розвитку методів виявлення шахрайства характеризується інтеграцією різних підходів та технологій для забезпечення комплексного захисту інформаційних систем. Гібридні моделі, що поєднують елементи сигнатурного аналізу, машинного навчання та поведінкової аналітики, дозволяють досягти максимальної ефективності при мінімізації хибно-позитивних спрацювань. Особливої уваги заслуговують методи, засновані на аналізі графів та мережевих взаємодій, які дозволяють виявляти складні шахрайські схеми, що включають множину взаємопов'язаних транзакцій або дій [9].

Порівняльний аналіз ефективності різних методів виявлення шахрайства демонструє значну перевагу підходів, заснованих на машинному навчанні та поведінковій аналітиці, особливо при роботі з великими обсягами даних та складними патернами шахрайських дій. Проте важливо зазначити, що ефективність будь-якої системи виявлення шахрайства залежить від якості та повноти вхідних даних, правильного вибору ознак для аналізу та регулярного оновлення моделей для адаптації до нових видів загроз.

Застосування ансамблевих методів, які поєднують результати роботи декількох алгоритмів, дозволяє досягти ще вищої точності виявлення шахрайських дій за рахунок компенсації недоліків окремих підходів. Методи агрегації результатів, такі як голосування, зважене усереднення або стекінг,

значно підвищують надійність систем виявлення шахрайства та зменшують вірогідність пропуску потенційних загроз [1].

1.2. Поведінкові патерни як інструмент виявлення шахрайства

1.2.1. Концептуальні засади аналізу поведінкових патернів користувачів

Аналіз поведінкових патернів користувачів базується на фундаментальному припущенні про наявність стійких, повторюваних моделей поведінки, які характерні для конкретного користувача або групи користувачів при взаємодії з інформаційними системами. Патерни поведінки проявляються у численних аспектах діяльності користувача: часових параметрах активності, послідовності дій, швидкості введення даних, особливостях навігації, географічній локації, характеристиках пристроїв та багатьох інших факторах, які у сукупності формують унікальний цифровий відбиток. Виявлення та аналіз таких патернів дозволяє створити персоналізовані профілі нормальної поведінки користувачів, відхилення від яких можуть свідчити про потенційні шахрайські дії або компрометацію облікових записів [11].

Математичний апарат для аналізу поведінкових патернів включає широкий спектр методів та алгоритмів, починаючи від статистичного аналізу та теорії ймовірностей, до складних моделей машинного навчання та нейронних мереж. Статистичні методи дозволяють визначити базові характеристики поведінки: середні значення, дисперсії, кореляції між різними параметрами, що створює основу для подальшого аналізу. Методи кластеризації, такі як k-means, DBSCAN або ієрархічна кластеризація, використовуються для групування схожих поведінкових патернів та виявлення природних сегментів користувачів з подібними характеристиками активності. Класифікаційні алгоритми, зокрема логістична регресія, дерева рішень, випадкові ліси та градієнтний бустинг, застосовуються для створення

предиктивних моделей, здатних розрізняти легітимну та шахрайську поведінку на основі попередньо розмічених даних [4].

Таблиця 1.2. Основні категорії поведінкових атрибутів користувачів

Категорія атрибутів	Опис	Приклади параметрів
Часові характеристики	Параметри, пов'язані з часовими аспектами активності користувача	Час доби активності, тривалість сесій, інтервали між діями, швидкість виконання операцій
Просторові характеристики	Параметри, пов'язані з географічним розташуванням та переміщенням	IP-адреси, GPS-координати, швидкість зміни локацій, типові маршрути
Технічні характеристики	Параметри, пов'язані з використовуваними пристроями та програмним забезпеченням	Відбитки браузера, характеристики пристроїв, мережеві параметри, налаштування
Поведінкова біометрика	Параметри, що характеризують особливості взаємодії з інтерфейсом	Динаміка натискання клавіш, патерни руху миші, жести на сенсорних екранах
Змістовні характеристики	Параметри, пов'язані зі змістом та контекстом дій користувача	Типи транзакцій, суми операцій, категорії відвідуваних сторінок, пошукові запити

Моделювання нормальної поведінки користувачів може здійснюватися з використанням різних підходів залежно від специфіки задачі та доступних даних. Загальна методологія включає збір та попередню обробку даних про активність користувачів, виділення значущих ознак, побудову моделей нормальної поведінки та встановлення порогових значень для виявлення аномалій. Важливим аспектом є врахування часової динаміки поведінкових патернів, оскільки моделі поведінки користувачів можуть змінюватися з часом внаслідок змін у звичках, робочих процесах або сезонних факторів [6].

Процес аналізу поведінкових патернів стикається з низкою викликів, серед яких особливо вагомими є проблема балансу між точністю виявлення аномалій та кількістю хибно-позитивних спрацювань, необхідність обробки великих обсягів даних у режимі реального часу, забезпечення конфіденційності та захисту персональних даних користувачів при здійсненні моніторингу. Вирішення цих проблем потребує комплексного підходу, що поєднує технологічні, організаційні та нормативно-правові аспекти.

Застосування методів федеративного навчання та диференційної приватності дозволяє значно підвищити рівень захисту персональних даних при збереженні ефективності аналізу поведінкових патернів [13].

Глибинне навчання відкриває нові можливості для аналізу поведінкових патернів, дозволяючи виявляти складні, нелінійні взаємозв'язки в даних та автоматично генерувати високорівневі ознаки з необроблених вхідних даних. Рекурентні нейронні мережі, зокрема архітектури LSTM та GRU, демонструють високу ефективність при роботі з послідовними даними, які характеризують активність користувачів у часі. Автоенкодери застосовуються для зниження розмірності даних та виявлення аномалій шляхом порівняння оригінальних вхідних даних з реконструйованими. Генеративні змагальні мережі дозволяють моделювати складні розподіли даних та генерувати синтетичні приклади для навчання систем виявлення шахрайства в умовах обмеженої кількості розмічених прикладів шахрайської поведінки.

Таблиця 1.3. Порівняння методів аналізу поведінкових патернів

Метод	Переваги	Обмеження	Область застосування
Статистичні методи	Прозорість, низька обчислювальна складність, не потребують розмічених даних	Обмежена здатність виявляти складні взаємозв'язки, чутливість до викидів	Базовий аналіз відхилень, моніторинг простих метрик
Кластерний аналіз	Не потребує розмічених даних, дозволяє виявляти природні групи користувачів	Складність визначення оптимальної кількості кластерів, чутливість до вибору метрики відстані	Сегментація користувачів, виявлення груп з подібною поведінкою
Дерева рішень	Висока інтерпретованість, здатність працювати з різнотипними даними	Схильність до перенавчання на складних задачах, нестабільність	Класифікація з пояснюваними результатами, аналіз значущості атрибутів
Ансамблеві методи	Висока точність, стійкість до перенавчання, здатність моделювати складні залежності	Висока обчислювальна складність, складність інтерпретації	Високоточна класифікація шахрайських дій, ранжування за ризиком

Нейронні мережі	Здатність виявляти складні нелінійні взаємозв'язки, автоматичне виділення ознак	Потребують великих обсягів даних для навчання, складність інтерпретації	Аналіз послідовностей дій, обробка поведінкової біометрики

Безперервний моніторинг та аналіз поведінкових патернів дозволяє не лише виявляти шахрайські дії, але й прогнозувати потенційні ризики, формувати персоналізовані профілі безпеки та адаптувати механізми захисту відповідно до індивідуальних особливостей користувачів. Інтеграція аналізу поведінкових патернів з іншими методами забезпечення інформаційної безпеки, такими як сигнатурний аналіз, аналіз мережевого трафіку та контроль цілісності даних, дозволяє створити багаторівневу систему захисту, здатну ефективно протидіяти широкому спектру загроз.

Концептуальна модель аналізу поведінкових патернів має включати компоненти для збору даних з різних джерел, їх нормалізації та агрегації, виділення значущих ознак, побудови моделей нормальної поведінки, виявлення аномалій та прийняття рішень щодо потенційних загроз. Кожен з цих компонентів може бути реалізований з використанням різних технологій та алгоритмів залежно від конкретних вимог та обмежень системи. Гнучкість та адаптивність архітектури є ключовими факторами для забезпечення ефективності системи в умовах постійної еволюції шахрайських схем та зміни поведінкових патернів користувачів [2].

1.2.1. Міжнародний досвід застосування поведінкової аналітики для виявлення шахрайства

Міжнародна практика застосування поведінкової аналітики для виявлення шахрайства демонструє стрімке зростання ефективності та масштабів впровадження відповідних технологій у різних сферах. Банківський сектор традиційно виступає лідером у впровадженні інноваційних методів протидії шахрайським діям, інвестуючи значні ресурси

у розробку та вдосконалення систем моніторингу поведінкових патернів клієнтів. Найбільші фінансові установи світу, такі як HSBC, JP Morgan Chase, Bank of America та Deutsche Bank, імплементували комплексні системи аналізу поведінки користувачів, які дозволяють виявляти підозрілі транзакції з високою точністю та мінімальною кількістю хибно-позитивних спрацювань [15].

Американський досвід застосування поведінкової аналітики характеризується тісною співпрацею приватного сектору з державними структурами, зокрема з Федеральною торговою комісією (FTC) та Комісією з цінних паперів і бірж (SEC), які розробили рекомендації та стандарти щодо використання поведінкових даних для протидії шахрайству. Згідно зі звітом Асоціації сертифікованих фахівців з розслідування шахрайства (ACFE), впровадження систем поведінкової аналітики дозволило скоротити середні втрати від шахрайських дій на 52% та зменшити час виявлення інцидентів на 60% у компаніях, які впровадили відповідні технології [8].

Таблиця 1.4. Міжнародні практики застосування поведінкової аналітики

Країна/Регіон	Провідні організації	Ключові технології	Особливості підходу
США	FICO, IBM, SAS, PayPal	Біометрична автентифікація, глибинне навчання, ансамблеві методи	Фокус на поєднанні біхевіоральної біометрики з традиційними методами автентифікації, активне використання хмарних технологій
Європейський Союз	Featurespace, Nethone, Fraugster	Адаптивне машинне навчання, аналіз контексту операцій, поведінкові біометричні методи	Баланс між ефективністю виявлення шахрайства та захистом персональних даних згідно з GDPR, транскордонний обмін даними
Азіатсько-Тихоокеанський регіон	Alibaba Group, Tencent, Ant Financial	Великі дані, інтеграція з платіжними екосистемами, біометрична верифікація	Масштабна інтеграція з мобільними платформами, використання великих обсягів даних з соціальних мереж

Австралія та Нова Зеландія	Commonwealth Bank, ANZ, BioCatch	Поведінкові біометричні системи, аналіз послідовностей дій, динамічне профілювання	Акцент на безперервній автентифікації та захисті від атак на мобільний банкінг
Латинська Америка	Nubank, Mercado Libre, Clip	Адаптивні скорингові моделі, аналіз соціальних зв'язків, географічні патерни	Врахування специфіки локальних ринків, фокус на інклюзивність фінансових послуг при забезпеченні безпеки

Європейський підхід до використання поведінкової аналітики відрізняється підвищеною увагою до питань захисту персональних даних та приватності користувачів, що обумовлено вимогами Загального регламенту захисту даних (GDPR). Європейські фінансові установи та технологічні компанії розробили методології, які дозволяють ефективно виявляти шахрайські дії при мінімальному зборі та обробці персональних даних. Інноваційні підходи, такі як федеративне навчання та локальна обробка поведінкових даних на пристроях користувачів, дозволяють досягти балансу між ефективністю захисту та збереженням приватності [7].

Країни Азіатсько-Тихоокеанського регіону демонструють найбільш стрімке зростання впровадження технологій поведінкової аналітики, що обумовлено високим рівнем проникнення мобільних платежів та електронної комерції. Китайські технологічні гіганти, такі як Alibaba та Tencent, розробили комплексні екосистеми, які аналізують поведінкові патерни користувачів у режимі реального часу, обробляючи мільярди транзакцій щодня. Алгоритми машинного навчання, які використовуються в цих системах, враховують сотні параметрів поведінки користувачів, включаючи швидкість введення даних, патерни навігації, геолокацію та навіть силу натискання на екран мобільних пристроїв.

Поведінкова аналітика в контексті виявлення шахрайства активно впроваджується не лише у фінансовому секторі, але й у сфері електронної комерції, телекомунікацій, страхування та державних послуг. Компанія

Amazon запатентувала систему аналізу поведінкових патернів покупців для виявлення шахрайських замовлень та повернень, яка враховує історію покупок, патерни навігації на сайті та взаємодію з різними категоріями товарів. Телекомунікаційні компанії використовують поведінкову аналітику для виявлення шахрайства з SIM-картами та несанкціонованого доступу до облікових записів абонентів [11].

Таблиця 1.5. Ефективність різних підходів до поведінкової аналітики

Підхід	Точність виявлення (AUC)	Час на виявлення	Рівень хибно-позитивних спрацювань	Масштабованість
Традиційні правила та пороги	0.75-0.82	Високий (години/дні)	Високий (5-15%)	Висока
Статистичні моделі	0.80-0.88	Середній (хвилини/години)	Середній (3-8%)	Середня
Машинне навчання	0.85-0.92	Низький (секунди/хвилини)	Низький (1-5%)	Середня
Глибинне навчання	0.90-0.96	Дуже низький (мілісекунди/секунди)	Дуже низький (0.5-3%)	Низька
Гібридні підходи	0.92-0.98	Низький (секунди/хвилини)	Дуже низький (0.3-2%)	Середня

Міжнародний досвід демонструє зростаючу тенденцію до інтеграції різних методів аналізу даних для максимальної ефективності виявлення шахрайських дій. Гібридні підходи, які поєднують поведінкову аналітику з аналізом мережевих взаємодій, біометричними методами та традиційними системами безпеки, забезпечують найвищий рівень захисту від сучасних кіберзагроз. Провідні світові компанії у сфері кібербезпеки активно інвестують у розробку платформ, які дозволяють комбінувати різні джерела даних та аналітичні методи для створення єдиної системи виявлення та запобігання шахрайським діям [9].

Аналіз міжнародного досвіду також свідчить про важливість міжгалузевої та міжнародної співпраці у протидії шахрайським діям. Обмін

інформацією про нові види шахрайства, спільні дослідницькі ініціативи та розробка стандартів дозволяють більш ефективно протидіяти транснаціональним шахрайським схемам. Міжнародні організації, такі як Financial Action Task Force (FATF) та International Association of Financial Crime Investigators (IAFCI), сприяють координації зусиль різних країн у боротьбі з фінансовими злочинами та розробці єдиних підходів до виявлення та запобігання шахрайським діям.

1.3. Нормативно-правове забезпечення протидії шахрайським діям в інформаційному просторі

Нормативно-правове забезпечення протидії шахрайським діям в інформаційному просторі становить комплексну систему законодавчих актів, регуляторних вимог та галузевих стандартів, спрямованих на запобігання, виявлення та притягнення до відповідальності за шахрайські дії з використанням інформаційних технологій. Правова база у цій сфері характеризується значною гетерогенністю та різним рівнем розвитку в різних країнах, що створює суттєві виклики для ефективної транскордонної протидії кіберзлочинності. Національні законодавства більшості розвинених країн містять спеціальні норми, які криміналізують різні форми комп'ютерного шахрайства, несанкціонованого доступу до інформаційних систем та маніпуляцій з даними [3].

Міжнародно-правові інструменти відіграють ключову роль у формуванні глобального підходу до протидії шахрайським діям в інформаційному просторі. Будапештська конвенція про кіберзлочинність, прийнята Радою Європи у 2001 році, залишається найбільш всеохоплюючим міжнародним договором у цій сфері, який встановлює стандарти щодо криміналізації різних форм кіберзлочинності, процедурних повноважень правоохоронних органів та механізмів міжнародного співробітництва. Конвенція була ратифікована 65 країнами, включаючи більшість європейських держав, США,

Канаду, Японію та Австралію, що сприяє гармонізації національних законодавств та посиленню транскордонної співпраці у боротьбі з шахрайськими діями в інформаційному просторі [10].

Таблиця 1.6. Міжнародні нормативно-правові акти у сфері протидії шахрайським діям

Назва документу	Рік прийняття	Ключові положення	Юрисдикції, які приєдналися
Будапештська конвенція про кіберзлочинність	2001	Криміналізація кіберзлочинів, процедурні повноваження, міжнародне співробітництво	65 країн, включаючи ЄС, США, Канаду, Японію
Директива ЄС про боротьбу з шахрайством та підробкою безготівкових платіжних засобів	2019	Гармонізація кримінальної відповідальності за шахрайство з платіжними інструментами, мінімальні санкції	Країни-члени ЄС
Конвенція Африканського союзу про кібербезпеку та захист персональних даних	2014	Створення правової бази для електронних транзакцій, захист персональних даних, протидія кіберзлочинності	Країни Африканського союзу
Резолюція ООН про боротьбу з використанням інформаційно-комунікаційних технологій у злочинних цілях	2019	Підтримка міжнародного співробітництва, технічна допомога країнам, що розвиваються	Країни-члени ООН
Шанхайська конвенція про боротьбу з тероризмом, сепаратизмом та екстремізмом	2001	Співпраця у боротьбі з використанням ІКТ для терористичних та екстремістських цілей	Країни-члени ШОС (Китай, Росія, Казахстан та ін.)

Європейський Союз розробив комплексну нормативно-правову базу для протидії шахрайським діям в інформаційному просторі, яка включає як загальні інструменти, такі як Загальний регламент захисту даних (GDPR) та Директива про мережеву та інформаційну безпеку (NIS Directive), так і спеціалізовані акти, спрямовані на боротьбу з конкретними видами

шахрайства. Директива 2019/713 про боротьбу з шахрайством та підробкою безготівкових платіжних засобів встановлює мінімальні стандарти щодо криміналізації шахрайських дій з платіжними інструментами та гармонізує підходи держав-членів до протидії таким злочинам. Друга платіжна директива (PSD2) запроваджує обов'язкову двофакторну автентифікацію для онлайн-платежів та вимагає від фінансових установ впровадження систем моніторингу транзакцій для виявлення шахрайських операцій [13].

Законодавство США у сфері протидії шахрайським діям в інформаційному просторі характеризується секторальним підходом, при якому різні аспекти регулюються окремими законами та підзаконними актами. Закон про комп'ютерне шахрайство та зловживання (Computer Fraud and Abuse Act, CFAA) є основним федеральним законом, який криміналізує несанкціонований доступ до комп'ютерних систем та мереж. Закон Грема-Ліча-Блайлі (Gramm-Leach-Bliley Act) встановлює вимоги до фінансових установ щодо захисту персональних даних клієнтів та впровадження програм з виявлення та запобігання шахрайству. Федеральна торгова комісія (FTC) має широкі повноваження щодо розслідування та притягнення до відповідальності за шахрайські дії в інформаційному просторі на основі Закону про Федеральну торгову комісію, який забороняє недобросовісні та оманливі практики.

Таблиця 1.7. Порівняння нормативно-правових підходів до протидії шахрайським діям

Юрисдикція	Основний підхід	Ключові нормативно-правові акти	Регуляторні органи	Особливості
Європейський Союз	Комплексний, гармонізований підхід	GDPR, NIS Directive, Директива 2019/713, PSD2	ENISA, EBA, національні регулятори	Баланс між захистом даних та безпекою, транскордонне співробітництво
США	Секторальний підхід	CFAA, Gramm-Leach-B	FTC, SEC, FinCEN, FBI	Фокус на галузевій саморегуляції,

		liley Act, FCRA, ECPA		значні повноваження федеральних агентств
Китай	Централізований підхід	Закон про кібербезпеку, Закон про захист персональних даних, Закон про безпеку даних	CAC, MPS, MIIT	Акцент на національній безпеці та контролі, жорсткі санкції
Бразилія	Комплексний підхід з елементами GDPR	LGPD, Marco Civil da Internet	ANPD, CERT.br	Захист прав споживачів, баланс між інноваціями та безпекою
Індія	Змішаний підхід	IT Act 2000, Personal Data Protection Bill	CERT-In, RBI, MEITY	Розвиток цифрової економіки при забезпеченні безпеки

Регуляторні вимоги щодо впровадження систем виявлення шахрайських дій стають все більш детальними та технічно специфічними. Регулятори фінансового сектору, такі як Європейська банківська асоціація (EBA), Федеральна корпорація страхування депозитів США (FDIC) та Базельський комітет з банківського нагляду, розробили рекомендації та стандарти щодо управління ризиками шахрайства, які включають вимоги до впровадження систем моніторингу транзакцій, аналізу поведінкових патернів та оцінки ризиків. Галузеві стандарти, такі як PCI DSS (Payment Card Industry Data Security Standard), встановлюють детальні вимоги щодо захисту даних платіжних карт та виявлення шахрайських операцій [6].

Транскордонний характер шахрайських дій в інформаційному просторі створює значні юрисдикційні виклики для правоохоронних органів та судових систем. Проблеми визначення юрисдикції, збору доказів з-за кордону та переслідування злочинців, які діють з територій інших держав, потребують розвитку міжнародного співробітництва та гармонізації правових підходів. Ініціативи, такі як 24/7 Network G8 для боротьби з високотехнологічними

злочинами та Європейський центр боротьби з кіберзлочинністю (ЕС3) при Європолі, спрямовані на посилення оперативної взаємодії між правоохоронними органами різних країн у розслідуванні транскордонних шахрайських схем.

2. МЕТОДОЛОГІЧНІ АСПЕКТИ АНАЛІЗУ ПОВЕДІНКОВИХ ПАТЕРНІВ

2.1. Збір та обробка поведінкових даних

2.1.1 Методи та інструменти збору даних про поведінку користувачів

Збір даних про поведінку користувачів у цифрових системах є фундаментальним етапом побудови ефективних систем виявлення шахрайських дій, який визначає якість та повноту вхідної інформації для подальшого аналізу. Методологія збору поведінкових даних базується на комплексному підході, що передбачає інтеграцію різноманітних джерел інформації для формування цілісного уявлення про активність користувачів. Інструментарій збору даних має забезпечувати високу точність фіксації дій користувачів, мінімальний вплив на продуктивність системи та дотримання вимог щодо захисту персональних даних [4].

Серверні журнали транзакцій та дій користувачів (server-side logs) становлять первинне джерело інформації про поведінку в інформаційних системах, фіксуючи послідовність запитів, час виконання операцій, параметри сесій та результати транзакцій. Аналіз журналів дозволяє відстежувати ключові події в системі, такі як автентифікація, авторизація, зміна налаштувань облікового запису та фінансові операції. Для ефективного використання даних журналів необхідно забезпечити їх структурованість, синхронізацію часових міток та збереження контекстуальної інформації про кожну дію. Сучасні системи управління журналами, такі як ELK Stack (Elasticsearch, Logstash, Kibana), Splunk та Graylog, забезпечують централізований збір, індексацію та аналіз журналів з різних компонентів інформаційної системи, що значно спрощує виявлення аномальних патернів поведінки. Забезпечення цілісності та захисту журналів від модифікації є критично важливим для збереження доказової бази при розслідуванні інцидентів безпеки [7].

Клієнтське відстеження поведінки користувачів (client-side tracking) дозволяє збирати детальну інформацію про взаємодію з інтерфейсом та мікроповедінкові характеристики, які неможливо зафіксувати на серверному рівні. Методи клієнтського відстеження включають моніторинг руху миші, динаміки натискання клавіш, жестів на сенсорних екранах, фокусування елементів інтерфейсу та часу перебування на різних сторінках. Таке детальне відстеження дозволяє виявляти не лише явні дії користувача, але й особливості його поведінкового стилю, які можуть служити біометричними характеристиками для безперервної автентифікації. Технічна реалізація клієнтського відстеження здійснюється за допомогою JavaScript-бібліотек, таких як Mouseflow, Hotjar або custom-рішень, які записують події інтерфейсу та передають їх на сервер для аналізу. Застосування методів компресії та вибіркового запису дозволяє мінімізувати обсяг даних, що передаються, та зменшити навантаження на мережу.

Таблиця 2.1. Порівняльна характеристика методів збору даних про поведінку користувачів

Метод збору даних	Типи зібраних даних	Переваги	Обмеження	Технічна реалізація
Серверні журнали транзакцій	Запити до API, параметри сесій, часові мітки, IP-адреси, результати транзакцій	Висока надійність, низьке навантаження на клієнтську частину, повнота фіксації критичних подій	Обмежена деталізація взаємодії з інтерфейсом, відсутність інформації про мікроповедінку	ELK Stack, Splunk, Graylog, власні рішення для журналювання
Клієнтське відстеження інтерфейсу	Рух курсора, кліктріми, часи завантаження та взаємодії, патерни скролінгу, заповнення форм	Висока деталізація взаємодії з інтерфейсом, можливість виявлення аномалій у стилі взаємодії	Вразливість до спуфінгу, високе навантаження на клієнт, проблеми з приватністю	JavaScript-бібліотеки (Mouseflow, Hotjar), SDK для мобільних додатків, веб-біконси
Мережевий моніторинг	Патерни трафіку, DNS-запити, TLS-хендшейки,	Можливість виявлення аномалій на мережевому рівні,	Обмежений доступ до зашифрованого трафіку,	Системи DPI, NetFlow, PCAP-аналізат

	заголовки HTTP, затримки передачі даних	незалежність від клієнтського середовища	складність аналізу, високі вимоги до інфраструктури	ори, мережеві сенсори
Поведінкова біометрика	Динаміка натискання клавіш, патерни руху миші, особливості жестів на сенсорних екранах, сила натискання	Можливість безперервної автентифікації, висока точність ідентифікації користувача	Чутливість до фізичного стану користувача, потреба в навчанні моделей, етичні питання	Спеціалізовані SDK (BioCatch, BehavioSec), бібліотеки машинного навчання
Контекстні дані	Геолокація, характеристики пристрою, мережеве оточення, часові пояси, мовні налаштування	Збагачення основних поведінкових даних контекстом, можливість виявлення невідповідностей	Неточність деяких даних, можливість підміни, правові обмеження збору	Геолокаційні API, браузерні відбитки, аналіз HTTP-заголовків

Мережевий моніторинг забезпечує збір даних про патерни трафіку, які можуть свідчити про аномальну активність або компрометацію системи. Аналіз мережевих потоків дозволяє виявляти нетипові комунікаційні патерни, нехарактерні обсяги даних або незвичні напрямки передачі інформації. Технології глибокого аналізу пакетів (DPI) та системи моніторингу мережевого трафіку, такі як Zeek (раніше Bro), Suricata або Wireshark, забезпечують детальний аналіз мережевих взаємодій на різних рівнях протоколів. Особливої уваги заслуговують методи аналізу зашифрованого трафіку, які дозволяють виявляти аномалії без порушення конфіденційності даних, що передаються. Такі методи базуються на аналізі метаданих комунікацій, таких як розмір пакетів, часові інтервали, криптографічні параметри та особливості встановлення з'єднань [9].

Контекстна інформація суттєво збагачує базові поведінкові дані, дозволяючи враховувати фактори, які можуть впливати на патерни активності користувачів. До контекстних даних належать геолокація, часовий пояс, тип та характеристики пристрою, параметри мережевого підключення, версія операційної системи та браузера. Збір контекстних даних здійснюється за

допомогою геолокаційних API, методів ідентифікації пристроїв (device fingerprinting) та аналізу HTTP-заголовків. Інтеграція контекстних даних з основними поведінковими патернами дозволяє виявляти невідповідності, які можуть свідчити про шахрайські дії, наприклад, доступ до облікового запису з нетипової локації або з використанням нехарактерного пристрою.

Методи збору даних про поведінку користувачів мобільних додатків мають свою специфіку, обумовлену особливостями мобільних платформ та способів взаємодії з сенсорними екранами. Мобільні SDK для аналізу поведінки, такі як AppsFlyer, Adjust або Firebase Analytics, дозволяють фіксувати патерни навігації, частоту та тривалість використання різних функцій, жести на сенсорних екранах та взаємодію з елементами інтерфейсу. Додаткові дані можуть бути отримані від сенсорів мобільних пристроїв, таких як акселерометр, гіроскоп та датчик освітленості, які дозволяють аналізувати фізичний контекст використання пристрою. Збір та аналіз таких даних має здійснюватися з урахуванням обмежень операційних систем щодо доступу до сенсорів та приватності користувачів [15]. Архітектура систем збору поведінкових даних повинна забезпечувати масштабованість, стійкість до навантажень та мінімальний вплив на продуктивність основних систем. Сучасні підходи базуються на використанні розподілених систем обробки потоків даних, таких як Apache Kafka, Amazon Kinesis або Google Pub/Sub, які дозволяють обробляти великі обсяги поведінкових даних у режимі реального часу. Застосування методів агрегації та вибіркового запису дозволяє зменшити обсяг даних, що зберігаються, при збереженні їх аналітичної цінності. Важливим аспектом є забезпечення відмовостійкості системи збору даних, що досягається через реплікацію, балансування навантаження та механізми відновлення після збоїв.

Питання приватності та етичних аспектів збору поведінкових даних набувають все більшого значення в контексті посилення регуляторних вимог та зростання обізнаності користувачів щодо захисту персональних даних. Впровадження принципів приватності за замовчуванням (privacy by design),

мінімізації даних та прозорості обробки є необхідною умовою для відповідності вимогам таких регуляторних актів як GDPR, CCPA та інших законів про захист персональних даних. Методи анонімізації та псевдонімізації, такі як хешування ідентифікаторів, агрегація даних та диференційна приватність, дозволяють зменшити ризики для приватності користувачів при збереженні аналітичної цінності даних. Отримання інформованої згоди користувачів на збір та обробку поведінкових даних з чітким поясненням мети та обсягу такої обробки є важливим етичним та правовим аспектом впровадження систем моніторингу поведінки [11].

2.1.2. Техніки попередньої обробки та нормалізації поведінкових даних

Попередня обробка та нормалізація поведінкових даних відіграє критичну роль у забезпеченні достовірності та інформативності подальшого аналізу для виявлення шахрайських дій в інформаційних системах. Сирі поведінкові дані, отримані з різних джерел, характеризуються неоднорідністю форматів, наявністю шумів, пропусками, викидами та різними масштабами вимірювання, що унеможливлює їх безпосереднє використання в аналітичних моделях. Трансформація та приведення даних до уніфікованого вигляду шляхом застосування спеціалізованих технік попередньої обробки дозволяє підвищити ефективність алгоритмів машинного навчання та статистичного аналізу, забезпечуючи більш точне виявлення аномальних патернів поведінки користувачів [5].

Очищення даних становить первинний етап попередньої обробки, спрямований на ідентифікацію та корекцію проблем з якістю вхідної інформації. Видалення дублікатів транзакцій, сесій або дій користувачів є необхідною умовою для запобігання викривленню статистичних показників та перенавчанню моделей. Дублікати можуть виникати внаслідок помилок реплікації даних, повторних запитів клієнтських додатків або технічних збоїв у системах журналювання. Алгоритми виявлення дублікатів враховують не

лише повні збіги записів, але й функціональні дублікати, які відрізняються несуттєвими деталями, наприклад, часовими мітками з мінімальною різницею. Фільтрація технічних дій, таких як автоматичні оновлення стану, службовий трафік або фонові процеси моніторингу, дозволяє зосередитися на значущих поведінкових патернах користувачів. Виявлення та видалення викидів, які можуть бути результатом технічних збоїв, помилок вимірювання або зловмисних дій, спрямованих на порушення роботи системи моніторингу, здійснюється за допомогою статистичних методів, таких як z-score, модифікований метод міжквартильного розмаху або алгоритми на основі щільності розподілу (DBSCAN, Isolation Forest) [8].

Таблиця 2.2. Методи обробки пропущених значень у поведінкових даних

Метод	Принцип дії	Переваги	Обмеження	Сценарії застосування
Видалення записів з пропусками	Повне видалення рядків з відсутніми значеннями	Простота реалізації, збереження достовірності даних	Втрата потенційно важливої інформації, неприйнятний при великій кількості пропусків	Невелика кількість пропусків (< 5%), випадковий характер пропусків
Заповнення константними значеннями	Заміна пропусків спеціальними значеннями (0, -1, медіана)	Збереження об'єму даних, можливість кодування факту пропуску	Спотворення статистичних характеристик, потенційне зміщення моделей	Категорійні змінні, випадки, коли факт пропуску несе смислове навантаження
Статистичне заповнення	Заміна пропусків статистичними агрегатами (середнє, медіана, мода)	Збереження статистичних характеристик розподілу	Зменшення варіативності даних, неврахування взаємозв'язків між змінними	Числові змінні з нормальним розподілом, невелика кількість пропусків
Інтерполяція та екстраполяція	Обчислення пропущених значень на основі часових рядів	Врахування часової динаміки даних, збереження трендів	Чутливість до шумів, необхідність достатньої кількості точок	Послідовні дані з чіткими трендами, пропуски в часових рядах

Прогнозування на основі інших змінних	Використання моделей для передбачення пропущених значень	Врахування взаємозв'язків між змінними, висока точність	Складність реалізації, ризик перенесення помилок моделі	Наявність сильних кореляцій між змінними, структуровані пропуски

Трансформація та кодування поведінкових даних передбачає перетворення різнотипних даних до форматів, придатних для машинного аналізу. Категоріальні змінні, такі як типи дій, джерела доступу або канали взаємодії, потребують кодування для використання в чисельних алгоритмах. Методи one-hot encoding забезпечують представлення категорій через бінарні вектори, зберігаючи незалежність категорій, проте спричиняючи розрідженість даних при великій кількості унікальних значень. Target encoding, який замінює категорії їх статистичним впливом на цільову змінну, дозволяє ефективно кодувати змінні з високою кардинальністю, зберігаючи їх предиктивну силу. Поведінкові патерни часто представлені послідовностями дій або транзакцій, які потребують спеціальних методів кодування. Технології вкладень (embeddings), запозичені з обробки природної мови, дозволяють представляти послідовності дій у вигляді щільних векторів фіксованої розмірності, зберігаючи семантичні відношення між різними типами дій [13].

Таблиця 2.3. Методи нормалізації числових поведінкових даних

Метод нормалізації	Математична формула	Діапазон значень	Переваги	Недоліки	Рекомендовані алгоритми
Мін-макс нормалізація	$x' = (x - \min(x)) / (\max(x) - \min(x))$	[0, 1]	Збереження розподілу, простота інтерпретації	Чутливість до викидів, залежність від вибірки	Нейронні мережі, k-NN, методи на основі відстаней
Z-нормалізація (стандартизація)	$x' = (x - \mu) / \sigma$	[-∞, +∞], більшість у [-3, 3]	Центрування даних, інваріантність до масштабу	Складніша інтерпретація, не обмежений діапазон	Лінійні моделі, SVM, методи на основі градієнтного спуску
Робастна нормалізація	$x' = (x - \text{median}(x)) / \text{IQR}(x)$	[-∞, +∞], більшість у [-2, 2]	Стійкість до викидів, збереження	Менш поширений, складніший в обчисленні	Алгоритми, чутливі до викидів, кластеризація

			структури розподілу		
Логарифмічна трансформація	$x' = \log(x + c)$	$[-\infty, +\infty]$	Стиснення великих значень, стабілізація дисперсії	Застосовна лише для додатних значень, асиметрична	Дані з експоненційним розподілом, фінансові метрики
Box-Cox перетворення	$x' = (x^\lambda - 1) / \lambda$ при $\lambda \neq 0$, $x' = \log(x)$ при $\lambda = 0$	Залежить від λ	Адаптивність до різних розподілів, стабілізація дисперсії	Складність підбору параметра λ, обмеження на вхідні дані	Лінійні моделі з припущенням нормальності

Агрегація та сегментація поведінкових даних є ключовим етапом для виявлення патернів на різних рівнях гранулярності. Часова агрегація дозволяє аналізувати поведінку користувачів у різних часових масштабах: від мікроповедінки протягом секунд та хвилин до довгострокових трендів протягом тижнів або місяців. Технічно агрегація здійснюється через групування даних за часовими вікнами різної ширини з обчисленням статистичних показників для кожного вікна. Ієрархічна темпоральна агрегація забезпечує баланс між деталізацією та загальною картиною поведінки, дозволяючи виявляти аномалії на різних часових масштабах. Секвенційна агрегація фокусується на послідовностях дій користувачів, виявляючи типові патерни навігації, транзакційні ланцюжки або сценарії взаємодії з системою. Методи вилучення типових послідовностей, такі як Sequence Mining, n-грами або Markov Models, дозволяють ідентифікувати часто повторювані патерни та відхилення від них.

Генерація похідних ознак (feature engineering) суттєво розширює аналітичні можливості через створення нових інформативних атрибутів на основі вихідних даних. Темпоральні ознаки характеризують часові аспекти поведінки: частота дій за різні періоди, варіативність інтервалів між діями, швидкість виконання операцій, періодичність активності. Контекстні ознаки відображають відношення поточної активності до попередньої історії користувача, типових патернів для подібних користувачів або загальних

тенденцій у системі. Відхилення від власної історичної поведінки користувача часто є сильним індикатором потенційно шахрайської активності. Поведінкові біометричні ознаки, такі як динаміка натискання клавіш, патерни руху миші або характеристики сенсорних жестів, формують унікальний біхевіоральний підпис користувача, який складно підробити. Графові ознаки, побудовані на основі аналізу мережі взаємодій між користувачами, транзакціями та сутностями, дозволяють виявляти складні шахрайські схеми, які неможливо ідентифікувати при ізольованому аналізі окремих дій [10].

Таблиця 2.4. Типи похідних ознак для аналізу поведінкових даних

Категорія ознак	Підкатегорії	Приклади ознак	Інформативність для виявлення шахрайства	Складність обчислення
Темпоральні ознаки	Частотні	Кількість дій за годину/день/тиждень, частота використання функцій	Висока	Низька
	Інтервальні	Середній/мінімальний/максимальний час між діями, стандартне відхилення інтервалів	Висока	Низька
	Швидкісні	Час виконання операцій, швидкість введення даних, швидкість навігації	Середня	Середня
	Періодичні	Коефіцієнти Фур'є активності, автокореляція дій, сезонні компоненти	Висока	Висока
Контекстні ознаки	Відносні	Відхилення від власної історії, z-score відносно групи, перцентильні ранги	Дуже висока	Середня
	Порівняльні	Відношення поточних метрик до середніх, ранги серед подібних користувачів	Висока	Середня
	Сегментаційні	Кластерні мітки, результати класифікації, сегменти поведінки	Середня	Висока
Поведінкові біометричні ознаки	Клавіатурні	Час утримання клавіш, латентність між натисканнями, патерни введення	Дуже висока	Середня
	Манипуляційні	Траєкторія руху миші, швидкість та прискорення руху, патерни кліків	Висока	Висока

	Сенсорні	Характеристики жестів, сила натискання, точність позиціонування	Висока	Висока
Графові ознаки	Топологічні	Ступінь вузла, центральність, коефіцієнт кластеризації в мережі транзакцій	Середня	Висока
	Комунікаційні	Патерни зв'язків між користувачами, спільні контакти, мережеві мотиви	Висока	Дуже висока
	Динамічні	Еволюція графа взаємодій, поява нових зв'язків, зміни в структурі мережі	Дуже висока	Дуже висока

Зниження розмірності даних є важливим кроком для подолання проблеми прокляття розмірності та видалення надлишкових або малоінформативних ознак. Методи відбору ознак (feature selection), такі як фільтрація на основі статистичних критеріїв, обгортковий підхід з використанням цільової метрики або вбудовані методи (L1-регуляризація, дерева рішень), дозволяють зосередитися на найбільш інформативних атрибутах поведінки. Методи вилучення ознак (feature extraction), такі як Principal Component Analysis (PCA), t-SNE або автоенкодери, трансформують вихідний простір ознак у простір меншої розмірності, зберігаючи максимум інформації. Для поведінкових даних, представлених у вигляді послідовностей, ефективними є методи зниження розмірності на основі рекурентних автоенкодерів або моделей уваги, які здатні захоплювати темпоральні залежності [6].

Нормалізація часових рядів поведінкових даних потребує специфічних підходів, які враховують темпоральну структуру та можливу нестаціонарність процесів. Диференціювання часових рядів (обчислення різниць між послідовними значеннями) дозволяє усунути тренди та зробити ряди більш стаціонарними. Видалення сезонних компонент за допомогою методів декомпозиції, таких як STL (Seasonal and Trend decomposition using Loess) або X-13ARIMA-SEATS, дозволяє виділити нерегулярну складову, яка часто містить аномалії. Адаптивна нормалізація з використанням ковзних вікон враховує зміни в характеристиках поведінки з часом, забезпечуючи

релевантність нормалізації для поточного контексту. Методи на основі гаусівських процесів дозволяють моделювати нестаціонарні часові ряди з урахуванням невизначеності, що особливо важливо для виявлення аномалій з високою достовірністю [12].

Валідація та оцінка якості попередньої обробки даних є необхідним етапом для забезпечення надійності та достовірності подальшого аналізу. Методи крос-валідації, такі як k-fold або time-based validation для часових рядів, дозволяють оцінити стабільність та генералізацію підходів до попередньої обробки на різних підмножинах даних. Метрики якості даних, такі як повнота, консистентність, точність та актуальність, забезпечують комплексну оцінку результатів попередньої обробки.

Важливим аспектом є також оцінка впливу методів попередньої обробки на кінцеву ефективність моделей виявлення шахрайства через порівняння ключових метрик продуктивності, таких як AUC-ROC, precision-recall або F1-score. Автоматизація процесів попередньої обробки та нормалізації даних з використанням конвеєрів даних (data pipelines) та методів AutoML дозволяє забезпечити масштабованість та відтворюваність підходів при роботі з великими обсягами поведінкових даних у промислових системах [14].

2.2. Методи виявлення аномалій та навчання моделей

2.2.1. Алгоритми ідентифікації аномальних поведінкових патернів

Ідентифікація аномальних поведінкових патернів становить ядро систем виявлення шахрайських дій, забезпечуючи можливість розпізнавання нетипової активності користувачів, яка потенційно може вказувати на шахрайство або компрометацію облікових записів. Концептуально аномалією в контексті поведінкових патернів вважається суттєве відхилення від встановленої норми, яке не може бути пояснене звичайною варіативністю поведінки користувача або впливом зовнішніх факторів. Алгоритмічні

підходи до виявлення аномалій можна класифікувати за різними критеріями: методологією навчання (з учителем, без учителя, напівавтоматичне), типом аналізованих даних (точкові, контекстуальні, колективні аномалії), часовим аспектом (статичні, динамічні) та ступенем інтерпретованості результатів [3].

Статистичні методи виявлення аномалій базуються на припущенні про наявність певного статистичного розподілу даних, відхилення від якого свідчить про аномальність. Параметричні підходи, такі як метод Грабса для нормального розподілу або експоненційне згладжування для часових рядів, моделюють розподіл даних та встановлюють порогові значення для ідентифікації викидів. Непараметричні методи, такі як метод квантилів або метод ядерної оцінки щільності (KDE), не потребують припущень щодо форми розподілу, що робить їх більш гнучкими при роботі з реальними поведінковими даними, які рідко відповідають класичним статистичним розподілам. Багатовимірні статистичні методи, такі як тест Махаланобіса або метод головних компонент (PCA), дозволяють враховувати кореляції між різними аспектами поведінки користувачів, що підвищує точність виявлення складних аномалій [7].

Кластерний аналіз та методи на основі щільності забезпечують виявлення аномалій через ідентифікацію точок або груп точок, які значно відрізняються від основних кластерів даних. Алгоритм DBSCAN визначає аномалії як точки, які не належать до жодного щільного кластера, ефективно ідентифікуючи ізольовані об'єкти в просторі ознак. Локальний фактор аномалії (LOF) обчислює відносну щільність точки порівняно з її сусідами, що дозволяє виявляти локальні аномалії навіть у даних з різною щільністю. Методи на основі відстаней, такі як k-NN для виявлення аномалій, розглядають об'єкти з найбільшою відстанню до k найближчих сусідів як потенційні аномалії. Ці методи демонструють високу ефективність при роботі з багатовимірними поведінковими даними, де визначення нормальності через статистичні розподіли є проблематичним [11].

Таблиця 2.5. Порівняльна характеристика алгоритмів виявлення аномалій в поведінкових патернах

Категорія алгоритмів	Конкретні алгоритми	Переваги	Недоліки	Сценарії застосування
Статистичні методи	Метод Грабса, GESD, ARIMA, експоненційне згладжування	Прозорість, інтерпретованість, низька обчислювальна складність	Обмежена здатність виявляти складні аномалії, чутливість до розподілу даних	Моніторинг однорідних метрик, виявлення аномалій у часових рядах
Методи на основі щільності та відстаней	DBSCAN, LOF, k-NN, HBOS	Здатність виявляти локальні аномалії, не потребують припущень щодо розподілу	Чутливість до вибору параметрів, складність масштабування	Багатовимірні поведінкові дані, неоднорідні розподіли
Методи на основі кластеризації	k-means, SOM, GMM, Hierarchical Clustering	Автоматичне групування схожих патернів, виявлення групових аномалій	Необхідність визначення кількості кластерів, чутливість до ініціалізації	Сегментація користувачів, виявлення аномальних груп
Ансамблеві методи	Isolation Forest, Random Cut Forest, Feature Bagging	Висока точність, стійкість до шуму, ефективність на високовимірних даних	Складність інтерпретації, обчислювальна інтенсивність	Складні поведінкові патерни, висока розмірність даних
Глибинне навчання для виявлення аномалій	Автоенкодери, LSTM-AD, GAN для виявлення аномалій	Здатність моделювати складні нелінійні залежності, автоматичне виділення ознак	Потреба у великих обсягах даних для навчання, складність інтерпретації	Послідовні дані, зображення, мультимодальні дані
Методи на основі правил та дерев рішень	Decision Trees, Fuzzy Logic, Rule-based Systems	Висока інтерпретованість, можливість інтеграції експертних знань	Обмежена здатність узагальнення, необхідність ручного налаштування	Домени з чітко визначеними правилами, аудит безпеки

Ансамблеві методи виявлення аномалій поєднують результати множини базових детекторів для підвищення точності та стійкості виявлення. Алгоритм Isolation Forest ізолює аномалії через рекурсивне розбиття простору ознак, визначаючи аномальність об'єкта за кількістю розбиттів, необхідних для його ізоляції від інших точок. Random Cut Forest,

впроваджений Amazon для моніторингу часових рядів, адаптує ідею Isolation Forest для потокових даних, забезпечуючи виявлення аномалій у режимі реального часу. Методи Feature Bagging для виявлення аномалій будують ансамбль детекторів на різних підмножинах ознак, що підвищує стійкість до шуму та зменшує вплив нерелевантних ознак. Ансамблеві підходи демонструють особливу ефективність при роботі з високовимірними поведінковими даними, де окремі детектори можуть стикатися з проблемою прокляття розмірності [9].

Глибинне навчання відкриває нові можливості для виявлення складних аномалій у поведінкових патернах, дозволяючи моделювати нелінійні взаємозв'язки та автоматично виділяти інформативні ознаки з необроблених даних. Автоенкодери, які навчаються реконструювати нормальні поведінкові патерни, визначають аномалії за високою помилкою реконструкції, що свідчить про відхилення від типових патернів.

Рекурентні нейронні мережі (RNN), зокрема LSTM та GRU, ефективно моделюють темпоральні залежності в послідовностях дій користувачів, прогнозуючи наступні дії та виявляючи відхилення від очікуваної поведінки. Генеративні змагальні мережі (GAN) для виявлення аномалій навчаються генерувати нормальні поведінкові патерни, а аномалії визначаються за низькою ймовірністю генерації спостереженого патерну. Графові нейронні мережі дозволяють виявляти аномальні патерни взаємодій між користувачами, транзакціями та сутностями, враховуючи структурні властивості графа взаємодій [14].

Виявлення колективних аномалій та рідкісних патернів фокусується на ідентифікації нетипових послідовностей або груп дій, які індивідуально можуть виглядати нормально, але у сукупності формують аномальний патерн. Алгоритми вилучення послідовностей, такі як SPADE або PrefixSpan, дозволяють ідентифікувати рідкісні або нетипові послідовності дій користувачів. Моделі на основі прихованих марковських ланцюгів (HMM) або умовних випадкових полів (CRF) враховують залежності між

послідовними діями, визначаючи аномальність за низькою ймовірністю спостереженої послідовності. Методи на основі символьної агрегації апроксимальних даних (SAX) трансформують часові ряди у символьні послідовності, що дозволяє застосовувати алгоритми аналізу патернів для виявлення аномальних підпослідовностей. Виявлення колективних аномалій є особливо важливим для ідентифікації складних шахрайських схем, які реалізуються через серію на перший погляд безневинних дій [5].

Адаптивні та еволюційні підходи до виявлення аномалій враховують динамічну природу поведінкових патернів, які змінюються з часом внаслідок змін у бізнес-процесах, сезонних факторів або еволюції звичок користувачів. Методи онлайн-навчання, такі як інкрементальні версії DBSCAN або k-means, дозволяють поступово адаптувати моделі до нових даних без повного перенавчання. Алгоритми виявлення змін точок (change point detection), такі як CUSUM або PELT, ідентифікують моменти суттєвих змін у поведінкових патернах, що дозволяє розрізняти аномалії та легітимні зміни поведінки. Концептуальний дрейф у поведінкових патернах відслідковується через методи виявлення та адаптації до дрейфу, такі як адаптивні вікна або ансамблі з динамічними вагами. Адаптивність є критично важливою для зменшення кількості хибно-позитивних спрацювань при еволюції нормальної поведінки користувачів [12].

2.2.2. Методи машинного навчання у виявленні шахрайських дій

Машинне навчання трансформувало підходи до виявлення шахрайських дій, забезпечуючи можливість автоматичного виявлення складних патернів та адаптації до нових видів шахрайства без явного перепрограмування системи. Фундаментальна перевага методів машинного навчання полягає у здатності виявляти приховані закономірності в великих обсягах даних та будувати предиктивні моделі, які можуть узагальнювати знання на нові, раніше не спостережувані випадки. Ефективне застосування машинного навчання для

виявлення шахрайських дій потребує глибокого розуміння специфіки проблеми, вибору відповідних алгоритмів та методологій, а також розробки стратегій подолання характерних викликів, таких як дисбаланс класів, дрейф концепцій та інтерпретованість моделей [8].

Методи навчання з учителем формують основу більшості промислових систем виявлення шахрайських дій, забезпечуючи побудову класифікаційних моделей на основі розмічених прикладів легітимної та шахрайської поведінки. Логістична регресія, незважаючи на відносну простоту, залишається широко використовуваним методом завдяки інтерпретованості, можливості оцінки ймовірностей та низьким обчислювальним вимогам. Дерева рішень та їх ансамблі (Random Forest, Gradient Boosting) демонструють високу ефективність при роботі з неоднорідними даними, автоматично виявляючи складні нелінійні взаємозв'язки та значущі ознаки без необхідності попередньої нормалізації. Метод опорних векторів (SVM) з нелінійними ядрами дозволяє знаходити оптимальні границі рішень у високовимірних просторах ознак, забезпечуючи високу точність класифікації навіть при обмеженому обсязі навчальних даних [10].

Таблиця 2.6. Застосування методів машинного навчання для різних типів шахрайських дій

Тип шахрайських дій	Ефективні методи машинного навчання	Ключові ознаки	Особливості застосування	Типові виклики
Шахрайство з платіжними картками	Gradient Boosting, Neural Networks, Random Forest	Сума транзакції, геолокація, час дня, частота транзакцій, категорія товару	Необхідність роботи в режимі реального часу, висока вартість хибних спрацювань	Значний дисбаланс класів, еволюція шахрайських схем
Фішинг та соціальна інженерія	RNN/LSTM, CNN для аналізу зображень,	URL-структура, метадані електронних листів, лінгвістичні	Мультимодальний аналіз різних типів контенту	Висока адаптивність шахраїв, фрагментація атак

	Transformer моделі	особливості, візуальна схожість		
Шахрайство з новими обліковими записами	Ансамблеві методи, глибинні графові мережі, моделі на основі правил	Поведінка при реєстрації, паттерни заповнення форм, IP-адреси, пристрої	Обмежена кількість даних на момент реєстрації	Складність розмежування легітимних та шахрайських користувачів на ранньому етапі
Компрометація облікових записів	Поведінкова біометрика, автоенкодери, RNN для аналізу послідовностей	Динаміка клавіатурного введення, патерни навігації, часові характеристики сесії	Безперервна автентифікація, адаптація до зміни поведінки	Баланс між безпекою та зручністю користувача
Шахрайство з поверненнями та відшкодуваннями	Графові алгоритми, кластерний аналіз, XGBoost	Історія замовлень, зв'язки між транзакціями, патерни повернень	Виявлення зв'язків між різними обліковими записами	Складність розрізнення легітимних та шахрайських повернень
Шахрайство з рекламою та кліками	CNN для аналізу сесій, моделі на основі часових рядів, k-means	Патерни кліків, тривалість сесій, активність миші, технічні параметри	Робота з великими обсягами даних у реальному часі	Різноманітність ботів та шахрайських схем
Інсайдерське шахрайство	Аномальні поведінкові патерни, графові мережі, моделі виявлення відхилень	Патерни доступу до даних, час роботи, використання привілеїв, незвичні операції	Необхідність врахування контексту та бізнес-процесів	Низька частота випадків, висока складність виявлення

Методи навчання без учителя та самоконтрольованого навчання набувають все більшого значення в контексті виявлення нових, раніше невідомих видів шахрайства. Кластерний аналіз, такий як k-means, DBSCAN або ієрархічна кластеризація, дозволяє виявляти природні групи в даних та ідентифікувати аномальні кластери або об'єкти, які значно відрізняються від основних груп. Методи зниження розмірності, такі як t-SNE, UMAP або

автоенкодери, трансформують високовимірні поведінкові дані в низьковимірні представлення, зберігаючи структурні властивості даних та полегшуючи виявлення аномалій. Моделі самоконтрольованого навчання, які навчаються на великих обсягах нерозмічених даних через виконання допоміжних завдань, таких як заповнення маскованих ознак або прогнозування наступних дій у послідовності, формують потужні представлення даних, які потім можуть бути використані для виявлення аномалій або тонкого налаштування класифікаційних моделей [6].

Методи глибинного навчання для часових послідовностей забезпечують ефективне моделювання динамічних аспектів поведінки користувачів. Рекурентні нейронні мережі з механізмами довгострокової пам'яті (LSTM) та керованими рекурентними блоками (GRU) моделюють довгострокові залежності в послідовностях дій користувачів, прогнозуючи наступні дії на основі історії взаємодії.

Одновимірні згорткові нейронні мережі (1D CNN) ефективно виділяють локальні патерни в послідовностях, такі як характерні підпослідовності дій або навігаційні патерни. Механізми уваги та трансформер-архітектури, такі як BERT або GPT, адаптовані для аналізу послідовностей дій, дозволяють моделювати складні взаємозв'язки між діями на різних часових масштабах, фокусуючись на найбільш релевантних частинах послідовності. Гібридні архітектури, які поєднують елементи CNN, RNN та механізмів уваги, демонструють найвищу ефективність при моделюванні багатоаспектних поведінкових патернів [13].

Графові методи машинного навчання набувають особливого значення при виявленні складних шахрайських схем, які включають множину взаємопов'язаних транзакцій, користувачів та сутностей. Алгоритми аналізу графів, такі як виявлення спільнот, пошук аномальних підграфів або виявлення шляхів та циклів, дозволяють ідентифікувати структурні патерни, характерні для шахрайських схем. Графові нейронні мережі (GNN), такі як Graph Convolutional Networks (GCN) або Graph Attention Networks (GAT),

навчаються на структурних та атрибутивних особливостях графів, формуючи векторні представлення вузлів та ребер, які відображають їхню роль у загальній структурі взаємодій. Графові автоенкодери дозволяють виявляти аномалії в структурі графа через невідповідність між оригінальним та реконструйованим графом. Застосування графових методів дозволяє виявляти складні шахрайські схеми, такі як мережі підставних компаній, симуляції активності в соціальних мережах або шахрайські ринки на тіньових платформах [9].

Ансамблеві методи та стратегії підвищення ефективності критично важливі для досягнення високої точності виявлення шахрайських дій в промислових системах. Бустинг-алгоритми, такі як AdaBoost, XGBoost або LightGBM, послідовно навчають множину моделей, фокусуючись на складних для класифікації прикладах, що дозволяє значно підвищити загальну точність. Баггінг-методи, такі як Random Forest або Extra Trees, навчають множину незалежних моделей на різних підмножинах даних та ознак, що забезпечує стійкість до шуму та зменшує ризик перенавчання. Стекінг-ансамблі комбінують прогнози різних базових моделей через мета-модель, яка навчається оптимально зважувати прогнози базових моделей залежно від контексту. Стратегії калібрування ймовірностей, такі як ізотонічна регресія або метод Платта, забезпечують точність оцінок ймовірностей шахрайства, що критично важливо для ризик-орієнтованого підходу до обробки транзакцій [11].

Таблиця 2.7. Стратегії подолання специфічних викликів у застосуванні машинного навчання для виявлення шахрайства

Виклик	Методи подолання	Технічна реалізація	Переваги	Обмеження
Дисбаланс класів	Перезважування, SMOTE, Focal Loss, аномальне виявлення	Балансування класів при навчанні, генерація синтетичних прикладів,	Підвищення чутливості до міноритарного класу, зменшення	Ризик перенавчання на міноритарному класі, збільшення хибно-позитивних спрацювань

		адаптація функцій втрат	пропусків шахрайства	
Дрейф концепцій	Онлайн-навчання, адаптивні вікна, ансамблі з динамічними вагами	Періодичне перенавчання моделей, моніторинг продуктивності, адаптивні алгоритми	Підтримання актуальності моделей, адаптація до нових шахрайських схем	Обчислювальна складність, потреба в постійному потоці розмічених даних
Холодний старт	Трансферне навчання, попереднє навчання на загальних доменах, мета-навчання	Використання попередньо навчених моделей, адаптація до нових доменів, генеративні моделі	Швидке впровадження для нових продуктів або ринків	Ризик негативного переносу, необхідність додаткових даних
Пояснюваність моделей	SHAP, LIME, контрфактуальні пояснення, дерева рішень	Бібліотеки для інтерпретації, візуалізація важливості ознак, глобальні та локальні пояснення	Довіра до моделей, відповідність регуляторним вимогам, покращення моделей	Компроміс між точністю та інтерпретованістю, обчислювальна складність
Конфіденційність даних	Диференційна приватність, федеративне навчання, гомоморфне шифрування	Додавання шуму до даних, розподілене навчання без централізації даних	Захист персональних даних, відповідність GDPR та іншим регуляціям	Потенційне зниження точності, технічна складність реалізації
Ресурсні обмеження	Квантизація моделей, дистиляція знань, ефективні архітектури	Зменшення точності параметрів, передача знань від складних до простих моделей	Швидкість інференсу, можливість роботи на обмежених ресурсах	Потенційне зниження точності, складність налаштування
Генеративні атаки на моделі	Робастне навчання, детектори змагальних прикладів, регуляризація	Додавання шуму при навчанні, спеціальні архітектури для виявлення атак	Стійкість до спроб обійти модель, захист від маніпуляцій	Обчислювальна складність, необхідність постійного оновлення

2.3. Оцінка точності та ефективності методів виявлення шахрайства

Ефективна оцінка методів виявлення шахрайських дій становить наріжний камінь для розробки, налаштування та еволюції систем моніторингу поведінкових патернів користувачів. Методологія оцінки має враховувати специфіку проблеми виявлення шахрайства, зокрема значний дисбаланс класів, високу вартість хибно-негативних рішень, необхідність роботи в режимі реального часу та динамічну природу шахрайських схем. Комплексний підхід до оцінки ефективності включає вибір релевантних метрик, розробку стратегій валідації, аналіз бізнес-впливу та забезпечення справедливості алгоритмів, що в сукупності дозволяє об'єктивно порівнювати різні методи та приймати обґрунтовані рішення щодо їх впровадження [4].

Традиційні метрики класифікації, такі як точність (accuracy), чутливість (recall) та специфічність (specificity), мають обмежену інформативність у контексті виявлення шахрайства через значний дисбаланс класів. Метрика AUC-ROC (площа під ROC-кривою) забезпечує агреговану оцінку якості ранжування транзакцій за ймовірністю шахрайства незалежно від конкретного порогового значення, що робить її популярним вибором для порівняння моделей. Precision-Recall крива та відповідна метрика AUC-PR є більш інформативними при значному дисбалансі класів, фокусуючись на ефективності виявлення позитивного (міноритарного) класу. Метрика F1-score, як гармонічне середнє між точністю та повнотою, забезпечує баланс між виявленням шахрайських випадків та мінімізацією хибних спрацювань. Для систем, які працюють з різними типами шахрайства або в різних доменах, корисною є метрика Matthews Correlation Coefficient (MCC), яка забезпечує збалансовану оцінку навіть при складних матрицях плутанини [7].

Адаптовані та специфічні для домену метрики часто необхідні для врахування бізнес-контексту та особливостей застосування системи виявлення шахрайства. Метрика Top-K Precision вимірює частку шахрайських випадків серед K транзакцій з найвищими оцінками ризику, що відображає

ефективність моделі при обмеженнях на кількість випадків, які можуть бути перевірені вручну. Метрика часу до виявлення (Time to Detection) оцінює швидкість ідентифікації шахрайської активності від її початку, що критично важливо для мінімізації потенційних збитків. Показник зниження збитків (Loss Reduction) безпосередньо вимірює фінансовий ефект від впровадження системи виявлення шахрайства, враховуючи як запобігання збиткам, так і операційні витрати на перевірку підозрілих транзакцій

Стратегії валідації та тестування мають вирішальне значення для достовірної оцінки ефективності методів виявлення шахрайства. Крос-валідація з урахуванням часової структури даних (time-based cross-validation) забезпечує реалістичну оцінку, враховуючи темпоральну природу шахрайських патернів та запобігаючи витоку майбутньої інформації в навчальні дані. Бутстрап-методи та підхід Monte Carlo крос-валідації дозволяють оцінити стабільність та варіативність результатів моделей, що особливо важливо при обмеженій кількості розмічених прикладів шахрайства. Оцінка на спеціально створених наборах даних, які моделюють різні типи шахрайських атак та сценарії, дозволяє тестувати стійкість системи до різних видів шахрайства. Безперервний моніторинг ефективності в продакшн-середовищі з використанням A/B тестування забезпечує оцінку реального впливу змін та оновлень моделей на ключові бізнес-метрики [14].

Аналіз бізнес-ефективності трансформує технічні метрики в показники, зрозумілі для бізнес-стейкхолдерів та керівництва. Вартісний аналіз помилок (Cost-Sensitive Evaluation) враховує різну вартість хибно-позитивних та хибно-негативних помилок, що дозволяє оптимізувати пороги прийняття рішень для мінімізації загальних втрат.

Аналіз ROI (Return on Investment) системи виявлення шахрайства включає оцінку запобіганих збитків, операційних витрат на розслідування, технічних витрат на розробку та підтримку системи, а також непрямих ефектів, таких як вплив на репутацію та довіру користувачів. Аналіз чутливості системи до різних типів шахрайства дозволяє виявляти сліпі зони

та пріоритизувати напрямки вдосконалення. Порівняльний аналіз з бенчмарками галузі або попередніми версіями системи забезпечує контекст для інтерпретації абсолютних показників ефективності [10].

Таблиця 2.8. Метрики оцінки ефективності методів виявлення шахрайства

Категорія метрик	Конкретні метрики	Формула/Обчислення	Переваги	Обмеження	Рекомендовані сценарії
Традиційні метрики класифікації	Точність (Accuracy)	(TP + TN) / (TP + TN + FP + FN)	Інтуїтивно зрозуміла	Малоінформативна при дисбалансі класів	Збалансовані набори даних
	Повнота/Чутливість (Recall)	TP / (TP + FN)	Фокус на виявленні всіх шахрайських випадків	Не враховує хибні спрацювання	Висока вартість пропуску шахрайства
	Точність (Precision)	TP / (TP + FP)	Фокус на мінімізації хибних спрацювань	Не враховує пропущені випадки	Висока вартість розслідування
	F1-Score	2 * (Precision * Recall) / (Precision + Recall)	Баланс між точністю та повнотою	Не враховує TN	Загальна оцінка ефективності
Метрики ранжування	AUC-ROC	Площа під ROC-кривою	Незалежність від порогу, стабільність	Менш інформативна при дисбалансі	Порівняння моделей
	AUC-PR	Площа під Precision-Recall кривою	Фокус на позитивному класі	Складність інтерпретації	Значний дисбаланс класів
	Top-K Precision	TP@K / K	Відображає ефективність при обмежених ресурсах	Залежність від K	Обмежені ресурси розслідування
Бізнес-орієнтовані метрики	Cost Savings	Σ(запобігані збитки - витрати на розслідування)	Пряма оцінка фінансового ефекту	Складність оцінки запобіганих збитків	Бізнес-обґрунтування

	Customer Friction	% легітимних користувачів з негативним досвідом	Врахування клієнтського досвіду	Суб'єктивність оцінки	Баланс безпеки та зручності
	Time to Detection	Середній час від початку шахрайства до виявлення	Оцінка швидкості реакції	Складність визначення початку шахрайства	Мінімізація збитків
Спеціалізовані метрики	Coverage of Fraud Types	% виявлених типів шахрайства	Оцінка універсальності системи	Залежність від таксономії	Комплексні системи
	Adaptability Score	Зміна ефективності при появі нових типів шахрайства	Оцінка адаптивності	Складність вимірювання	Швидкозмінні середовища
	Stability Over Time	Варіація ефективності в часі	Оцінка стабільності системи	Залежність від зовнішніх факторів	Довгострокові системи

Інтерпретованість та пояснюваність моделей набувають все більшого значення в контексті виявлення шахрайства, забезпечуючи можливість обґрунтування рішень системи та відповідність регуляторним вимогам. Методи глобальної інтерпретації, такі як аналіз важливості ознак, часткові залежності та глобальні сурогатні моделі, дозволяють зрозуміти загальні патерни прийняття рішень моделлю. Методи локальної інтерпретації, такі як LIME (Local Interpretable Model-agnostic Explanations) або SHAP (SHapley Additive exPlanations), забезпечують пояснення конкретних прогнозів, виділяючи фактори, які найбільше вплинули на рішення моделі в кожному окремому випадку. Контрфактуальні пояснення, які визначають мінімальні зміни в ознаках, необхідні для зміни прогнозу моделі, забезпечують практичну інтерпретацію рішень та можуть бути використані для генерації зрозумілих пояснень для аналітиків та користувачів [3].

Таблиця 2.9. Методи забезпечення справедливості та етичності систем виявлення шахрайства

Категорія методів	Конкретні підходи	Принцип дії	Переваги	Обмеження	Регуляторний контекст
Передобробка даних	Ресемплінг з балансуванням	Збалансування репрезентації різних груп у навчальних даних	Простота реалізації, збереження алгоритму	Потенційна втрата інформації	GDPR, FTC Act
	Трансформація ознак	Маскування або трансформація чутливих ознак	Збереження корисної інформації при видаленні упереджень	Складність визначення оптимальної трансформації	ECOA, FHA
Методи при навчанні	Справедливі регуляризатори	Додавання штрафів за упередженість у функцію втрат	Баланс між точністю та справедливістю	Складність налаштування параметрів	ECOA, UDAAP
	Adversarial Debiasing	Навчання моделі бути точною та одночасно не враховувати чутливі атрибути	Ефективне видалення упереджень	Складність навчання, можливе зниження точності	FCRA, GDPR
Постобробка	Корекція порогів	Оптимізація порогів рішення для різних груп	Простота реалізації, збереження ранжування	Обмежена ефективність при складних упередженнях	ECOA, FCRA
	Калібрування ймовірностей	Забезпечення однакової інтерпретації ймовірностей для різних груп	Збереження ранжування, покращення інтерпретації	Можливе зниження загальної точності	UDAAP, GDPR
Моніторинг та аудит	Диспаратний аналіз впливу	Оцінка різниці в результатах між захищеними групами	Відповідність регуляторним вимогам	Чутливість до вибору метрик	ECOA, FHA, Title VII

	Постійний моніторинг метрик справедливості	Відстеження змін у справедливості системи з часом	Раннє виявлення проблем	Ресурсоємність, складність реакції на зміни	FCRA, GDPR, AI Act

Забезпечення справедливості та етичності систем виявлення шахрайства є критичним аспектом їх оцінки та впровадження. Аудит системи на наявність диспропорційного впливу (disparate impact) на різні демографічні групи дозволяє виявляти та усувати потенційні упередження, які можуть призводити до несправедливого ставлення до певних категорій користувачів. Методи оцінки справедливості включають вимірювання показників демографічного паритету, рівності можливостей та рівності помилок для різних груп. Методи забезпечення справедливості можуть бути впроваджені на різних етапах життєвого циклу моделі: при попередній обробці даних (балансування репрезентації різних груп), при навчанні моделей (регуляризація для мінімізації упереджень) або при постобробці результатів (корекція порогів для різних груп). Дотримання принципів етичного машинного навчання та відповідність регуляторним вимогам щодо недискримінації є необхідною умовою для законного та відповідального використання систем виявлення шахрайства [15].

Ефективна оцінка методів виявлення шахрайства потребує інтеграції технічних, бізнесових та етичних аспектів, що дозволяє створювати системи, які не лише демонструють високу точність виявлення шахрайських дій, але й забезпечують позитивний досвід для легітимних користувачів, мінімізують фінансові втрати та відповідають нормативним вимогам. Комплексний підхід до оцінки включає використання релевантних метрик, реалістичні стратегії валідації, аналіз бізнес-впливу та забезпечення справедливості алгоритмів, що в сукупності дозволяє приймати обґрунтовані рішення щодо вибору та впровадження методів виявлення шахрайських дій на основі аналізу поведінкових патернів користувачів [1].

3. ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ ВИЯВЛЕННЯ ШАХРАЙСЬКИХ ДІЙ

3.1. Архітектура та реалізація системи виявлення шахрайських дій

3.1.1. Розробка архітектури системи моніторингу поведінкових патернів

Архітектура системи моніторингу поведінкових патернів користувачів становить фундаментальну основу ефективного виявлення шахрайських дій в інформаційних системах. Концептуальний підхід до проектування такої системи базується на принципах модульності, масштабованості, відмовостійкості та адаптивності, що забезпечує гнучке реагування на еволюцію шахрайських схем та зміни в поведінці користувачів. При розробці архітектури критично важливим є забезпечення балансу між повнотою збору даних, продуктивністю обробки в режимі реального часу та оптимальним використанням обчислювальних ресурсів з урахуванням специфіки конкретної предметної області та характеристик цільової інформаційної системи [4].

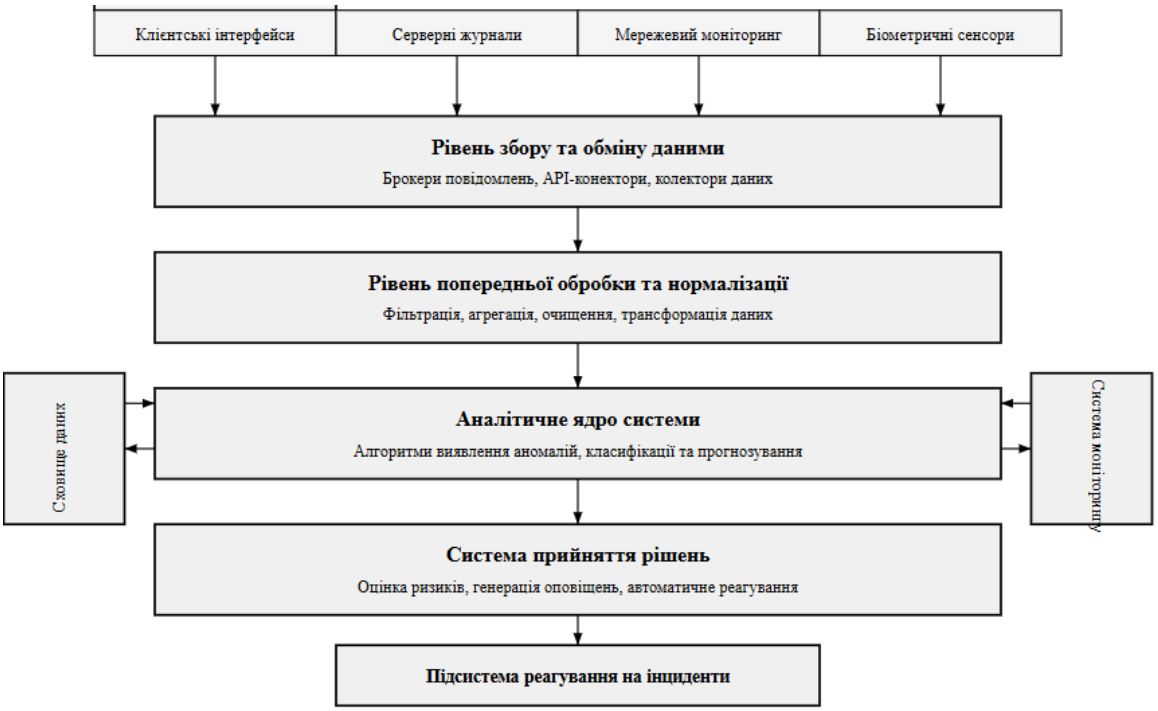

Рис. 3.1. Архітектура системи моніторингу поведінкових патернів

Багаторівнева архітектура системи моніторингу поведінкових патернів включає компоненти збору даних, їх попередньої обробки, аналізу, виявлення аномалій, прийняття рішень та зворотного зв'язку, що функціонують як єдиний комплекс з чітко визначеними інтерфейсами взаємодії. Рівень збору даних забезпечує інтеграцію з різними джерелами інформації про поведінку користувачів: журналами серверних подій, API-викликами, клієнтськими додатками, системами моніторингу мережевого трафіку та біометричними сенсорами. Реалізація компонентів збору даних передбачає використання легковагих агентів, які мінімально впливають на продуктивність інтегрованих систем, та брокерів повідомлень, що забезпечують надійну асинхронну передачу даних від джерел до системи аналізу. Потокова обробка первинних даних реалізується з використанням розподілених систем, таких як Apache Kafka або Amazon Kinesis, які забезпечують високу пропускну здатність, гарантовану доставку повідомлень та можливість горизонтального масштабування відповідно до зростання навантаження [8].

Рівень попередньої обробки та нормалізації трансформує гетерогенні потоки даних у уніфіковані структури, придатні для подальшого аналізу, одночасно фільтруючи шум, видаляючи дублікати та аномалії технічного характеру. Архітектурне рішення для цього рівня передбачає розподілену обробку з використанням Apache Spark Streaming або Apache Flink, які забезпечують ефективне виконання операцій фільтрації, агрегації, нормалізації та збагачення даних контекстною інформацією. Впровадження механізмів буферизації та контролю навантаження забезпечує стабільну роботу системи в умовах пікових навантажень або тимчасових збоїв компонентів. Масштабована підсистема сховища даних, що поєднує швидкі in-memory бази даних для оперативного аналізу з розподіленими файловими системами для довготривалого зберігання, забезпечує баланс між швидкістю доступу та вартістю зберігання великих обсягів поведінкових даних [12].

Аналітичне ядро системи реалізує різноманітні алгоритми виявлення аномалій та класифікації поведінки, працюючи в двох режимах: потокової обробки для оперативного виявлення підозрілої активності та пакетної обробки для глибинного аналізу, навчання моделей та виявлення складних патернів. Архітектурний підхід передбачає створення реєстру моделей, який забезпечує централізоване управління життєвим циклом алгоритмів, від їхнього навчання та валідації до розгортання та моніторингу продуктивності. Система оркестрації моделей динамічно обирає оптимальні алгоритми та їхні ансамблі залежно від контексту транзакції, характеристик користувача та поточної ситуації. Механізми A/B тестування дозволяють безпечно впроваджувати нові алгоритми, порівнюючи їхню ефективність з існуючими моделями на підмножині реального трафіку [1].

Таблиця 3.1. Компоненти архітектури системи моніторингу поведінкових патернів

Компонент	Функціональне призначення	Технологічна реалізація	Взаємодія з іншими компонентами	Критерії оптимізації
Колектори даних	Збір первинної інформації з різних джерел	Легковагі агенти, API-конектори, ETL-процеси	Передача даних до брокерів повідомлень	Мінімальний вплив на продуктивність систем, повнота збору даних
Система обміну повідомленнями	Надійна асинхронна передача даних між компонентами	Apache Kafka, RabbitMQ, Amazon Kinesis	Інтеграція з колекторами та процесорами даних	Пропускна здатність, затримка, гарантії доставки
Процесори потокової обробки	Фільтрація, нормалізація та агрегація даних в режимі реального часу	Apache Flink, Spark Streaming, Kafka Streams	Отримання даних з брокерів, взаємодія зі сховищем та аналітичним ядром	Латентність обробки, ефективність використання ресурсів
Сховище даних	Зберігання необроблених та агрегованих даних для аналізу	Elasticsearch, ClickHouse, HDFS, Cassandra	Взаємодія з процесорами та аналітичним ядром	Швидкість доступу, масштабованість, вартість зберігання

Аналітичне ядро	Виявлення аномалій та класифікація поведінки	ML фреймворки, системи правил, CEP-платформи	Взаємодія зі сховищем, системою прийняття рішень та моніторингу	Точність виявлення, швидкість обробки, адаптивність
Система прийняття рішень	Визначення дій на основі результатів аналізу	Системи бізнес-правил, скорингові механізми	Взаємодія з аналітичним ядром та системами реагування	Балансування між безпекою та користувацьким досвідом
Підсистема реагування	Виконання дій у відповідь на виявлені аномалії	API-інтеграції, системи оповіщення, автоматизовані процеси	Взаємодія з системою прийняття рішень та зовнішніми системами	Швидкість реакції, мінімізація хибно-позитивних впливів
Система моніторингу	Відстеження продуктивності та ефективності всіх компонентів	Prometheus, Grafana, ELK Stack	Збір метрик з усіх компонентів системи	Повнота моніторингу, мінімальний вплив на продуктивність
Підсистема управління моделями	Контроль життєвого циклу аналітичних моделей	MLflow, Kubeflow, кастомні рішення	Взаємодія з аналітичним ядром та системою моніторингу	Автоматизація процесів, версійність, відтворюваність

Підсистема прийняття рішень трансформує аналітичні результати в конкретні дії відповідно до бізнес-правил, рівнів ризику та контексту операцій. Архітектурне рішення передбачає використання комбінації детерміністичних правил для очевидних випадків шахрайства та ймовірнісних моделей для ситуацій з неоднозначною інтерпретацією. Механізми балансування навантаження та пріоритезації забезпечують оптимальне використання ресурсів аналітичних команд при ручному розслідуванні підозрілих випадків. Багаторівнева система реагування реалізує гнучкий підхід до обробки потенційно шахрайських дій: від додаткової автентифікації та моніторингу в режимі підвищеної уваги до тимчасового обмеження функціональності та блокування у випадках високої впевненості в шахрайстві [6].

Інтерфейси управління та моніторингу забезпечують візуалізацію поточного стану системи, статистики виявлених аномалій та продуктивності

окремих компонентів. Дашборди для аналітиків надають інструменти для дослідження конкретних випадків, візуалізації зв'язків між сутностями та патернів поведінки, що спрощує процес розслідування складних шахрайських схем. Система журналювання та аудиту фіксує всі дії системи та операторів, забезпечуючи можливість відтворення процесу прийняття рішень та відповідність регуляторним вимогам. Інтерфейси зворотного зв'язку дозволяють аналітикам підтверджувати або спростовувати автоматичні рішення системи, забезпечуючи постійне навчання та вдосконалення моделей [15].

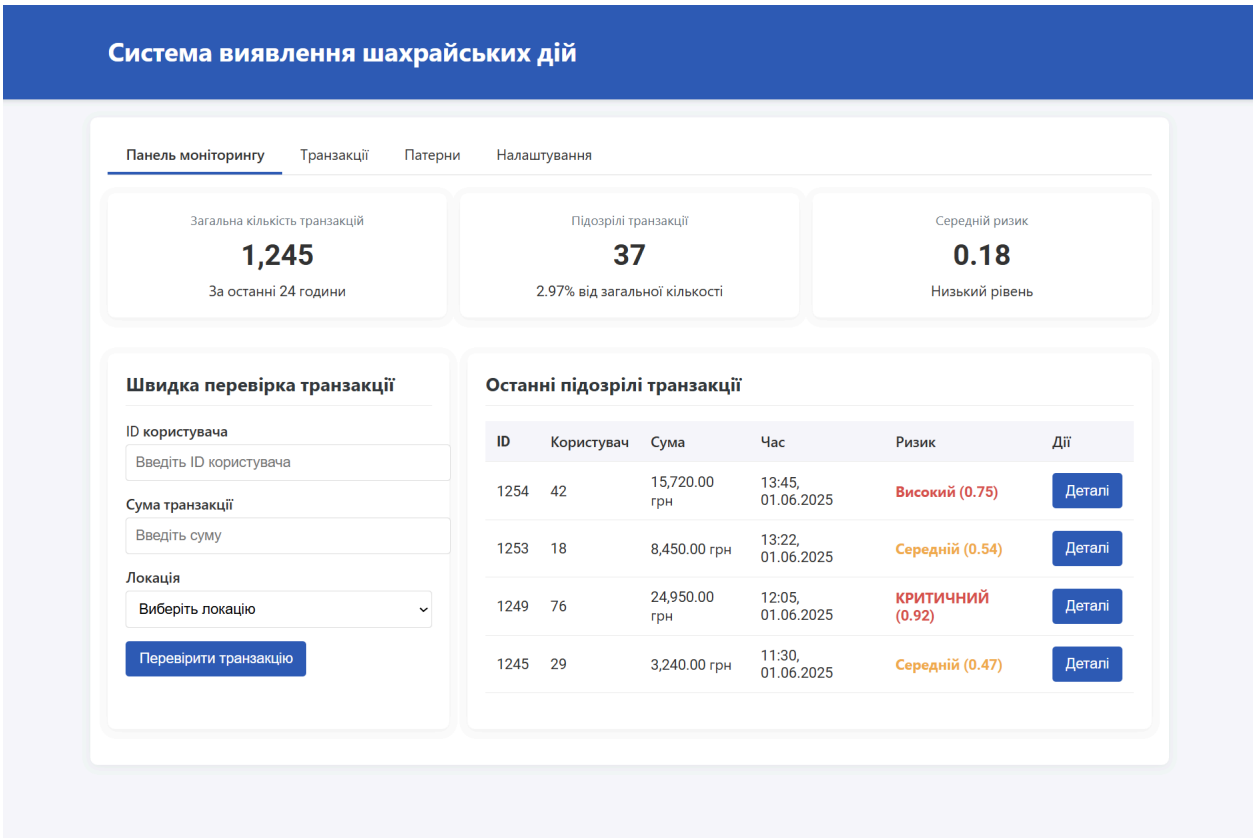

Рис 3.1. Головна панель системи моніторингу шахрайських дій

Архітектура системи розроблена з урахуванням вимог до безпеки та захисту персональних даних користувачів. Механізми шифрування даних у стані спокою та при передачі, контроль доступу на основі ролей, анонімізація та псевдонімізація чутливої інформації забезпечують відповідність вимогам GDPR та інших регуляторних актів. Впровадження принципів «приватність за

замовчуванням» (privacy by design) та «приватність за проектуванням» (privacy by default) на всіх рівнях архітектури забезпечує мінімізацію збору та обробки персональних даних при збереженні ефективності виявлення шахрайських дій. Механізми диференційної приватності та федеративного навчання дозволяють будувати ефективні моделі без централізації чутливих даних [9].

3.1.2. Імплементація алгоритмів виявлення аномальної поведінки користувачів

Імплементація алгоритмів виявлення аномальної поведінки користувачів передбачає трансформацію теоретичних моделей та концепцій у працездатні програмні компоненти, інтегровані в загальну архітектуру системи моніторингу поведінкових патернів. Процес розробки та впровадження алгоритмів включає етапи формалізації проблеми, підготовки даних для навчання та тестування, реалізації обчислювальних моделей, їх оптимізації для продуктивної роботи та інтеграції з іншими компонентами системи. Стратегічний підхід до імплементації базується на концепції багаторівневого захисту, що поєднує різні типи алгоритмів для виявлення різноманітних аспектів шахрайської поведінки: від простих правил та статистичних методів до складних моделей машинного навчання та глибинних нейронних мереж [7].

Базовий рівень захисту реалізується через систему детерміністичних правил та порогових значень, які ефективно виявляють явні ознаки шахрайства та аномалії технічного характеру. Імплементація здійснюється з використанням спеціалізованих мов опису правил та систем управління бізнес-правилами (BRMS), які забезпечують гнучкість налаштування та швидке оновлення правил без необхідності модифікації програмного коду. Механізми автоматичної генерації та оптимізації правил на основі історичних даних дозволяють підтримувати актуальність бази правил при еволюції

шахрайських схем. Система пріоритезації правил та механізми розв'язання конфліктів забезпечують консистентність прийняття рішень у складних випадках, коли спрацьовують кілька протилежних правил [11].

Таблиця 3.2. Порівняльна характеристика реалізованих алгоритмів виявлення аномальної поведінки

Категорія алгоритмів	Реалізовані методи	Технічна імплементація	Обчислювальна складність	Точність виявлення (AUC)	Специфіка застосування
Детерміністичні правила	Порогові значення, логічні предикати, часові обмеження	BRMS на основі Drools, кастомна DSL для опису правил	O(n) для n правил, незалежно від обсягу даних	0.75-0.82 (висока специфічність, низька чутливість)	Явні ознаки шахрайства, регуляторні вимоги, технічні аномалії
Статистичні методи	ARIMA, експоненційне згладжування, метод Хольта-Вінтерса	R та Python бібліотеки з оптимізацією через Cython	$O(m^2)$ для ARIMA при m спостереженнях	0.82-0.88 (збалансована специфічність та чутливість)	Часові ряди метрик активності, прогнозування нормальної поведінки
Методи на основі щільності	LOF, DBSCAN, HDBSCAN з оптимізацією для потокових даних	Scikit-learn з кастомними розширеннями для потокової обробки	$O(n^2)$ в загальному випадку, O(n log n) з оптимізаціями	0.85-0.90 (висока чутливість, середня специфічність)	Багатовимірні поведінкові ознаки, виявлення локальних аномалій
Ансамблеві методи	Isolation Forest, Random Cut Forest, XGBoost	Реалізація на основі LightGBM та MLlib з оптимізацією для розподіленої обробки	$O(t \cdot n \log n)$ для t дерев та n спостережень	0.88-0.93 (висока чутливість та специфічність)	Складні поведінкові патерни, висока розмірність даних
Глибинне навчання	LSTM-Autoencoder, BiGAN, Graph Convolutional Networks	TensorFlow та PyTorch з оптимізацією для GPU та TPU	$O(b \cdot d^2)$ для batch size b та розмірності d	0.90-0.95 (дуже висока чутливість, висока специфічність)	Послідовні дані, графові структури, мультимодальні ознаки

Поведінкові біометричні методи	Динаміка клавіатурного введення, патерни руху миші	Кастомні рішення на JS для клієнтської частини та TensorFlow для серверної	Залежить від модальності та алгоритму	0.87-0.92 (висока специфічність для конкретного користувача)	Безперервна автентифікація, виявлення підміни користувача
Гібридні підходи	Стекінг-ансамблі різних типів моделей, комбінація правил та ML	Метамоделі на основі LightGBM з інтеграцією результатів різних алгоритмів	Сума складності базових моделей	0.92-0.97 (оптимальний баланс між метриками)	Комплексне виявлення різних типів шахрайства

Статистичні методи виявлення аномалій імплементовані з фокусом на ефективну обробку часових рядів поведінкових даних користувачів. Реалізація моделей ARIMA, експоненційного згладжування та методу Хольта-Вінтерса оптимізована для роботи з високочастотними даними через використання спеціалізованих бібліотек та низькорівневих оптимізацій. Паралельна обробка незалежних часових рядів забезпечує масштабованість при аналізі великої кількості користувачів та метрик. Механізми адаптивного вибору параметрів моделей та автоматичного виявлення сезонності дозволяють враховувати специфіку різних типів поведінкових даних без необхідності ручного налаштування. Алгоритми робастної статистики, такі як медіанне абсолютне відхилення (MAD) та вінзорізація, забезпечують стійкість до викидів та нетипових спостережень [2].

Алгоритми на основі машинного навчання без учителя імплементовані для виявлення нових та раніше невідомих типів шахрайської поведінки. Реалізація методів кластеризації (k-means, DBSCAN) та методів на основі щільності (LOF, HDBSCAN) оптимізована для роботи з багатовимірними поведінковими даними через використання ефективних структур даних для пошуку найближчих сусідів, таких як KD-дерева та Ball-дерева. Інкрементальні версії алгоритмів забезпечують можливість онлайн-навчання

та адаптації до змін у поведінкових патернах. Методи зниження розмірності (PCA, t-SNE, UMAP) реалізовані як попередній етап для покращення ефективності кластеризації та візуалізації даних. Алгоритми Isolation Forest та Random Cut Forest імплементовані з оптимізацією для потокової обробки даних, що дозволяє виявляти аномалії в режимі реального часу з мінімальною затримкою [13].

Моделі машинного навчання з учителем реалізовані для класифікації поведінки на основі розмічених прикладів легітимних та шахрайських дій. Імплементація ансамблевих методів (Random Forest, Gradient Boosting, XGBoost) оптимізована для ефективного використання обчислювальних ресурсів через розподілене навчання та інференс. Механізми автоматичного підбору гіперпараметрів, такі як байєсівська оптимізація та генетичні алгоритми, забезпечують досягнення максимальної точності моделей без необхідності ручного налаштування. Техніки балансування класів, такі як SMOTE, ADASYN та зважування класів, реалізовані для подолання проблеми значного дисбалансу між легітимними та шахрайськими транзакціями. Методи інтерпретації моделей, такі як SHAP та LIME, імплементовані для забезпечення пояснюваності рішень та відповідності регуляторним вимогам [4].

Глибинні нейронні мережі реалізовані для аналізу складних поведінкових патернів, представлених послідовностями дій, графовими структурами та мультимодальними даними. Архітектури на основі LSTM та GRU оптимізовані для ефективної обробки послідовностей дій користувачів через використання технік пакетної обробки, векторизації операцій та апаратного прискорення на GPU та TPU. Автоенкодери та варіаційні автоенкодери реалізовані для виявлення аномалій через вимірювання помилки реконструкції поведінкових патернів. Графові нейронні мережі, такі як Graph Convolutional Networks (GCN) та Graph Attention Networks (GAT), імплементовані для аналізу взаємозв'язків між користувачами, транзакціями та сутностями. Методи регуляризації, такі як Dropout, BatchNormalization та

L1/L2-регуляризація, застосовані для запобігання перенавчанню моделей [10].

Гібридні алгоритми, що поєднують різні підходи до виявлення аномалій, реалізовані для досягнення максимальної ефективності в різних сценаріях шахрайства. Стекінг-ансамблі, які комбінують результати детерміністичних правил, статистичних методів та моделей машинного навчання, імплементовані з використанням метамоделей на основі градієнтного бустингу. Механізми динамічного зважування результатів різних алгоритмів залежно від контексту транзакції та профілю користувача реалізовані для адаптації системи до різних типів шахрайства. Ієрархічний підхід до виявлення аномалій, який поєднує швидкі фільтри для очевидних випадків з глибоким аналізом для складних ситуацій, реалізований для оптимального використання обчислювальних ресурсів [6].

Системи реального часу для виявлення аномалій імплементовані з урахуванням обмежень на латентність обробки та використання ресурсів. Оптимізація алгоритмів для роботи в режимі реального часу включає використання апроксимованих версій складних методів, інкрементальне оновлення моделей та механізми раннього зупинення обчислень для очевидних випадків. Буферизація та пріоритезація запитів забезпечують стабільну роботу системи в умовах пікових навантажень. Механізми деградації функціональності при перевантаженні дозволяють системі продовжувати роботу з фокусом на найбільш критичні аспекти виявлення шахрайства навіть при екстремальних навантаженнях [14].

Формування нових ознак на основі середньої суми та кількості транзакцій користувача, а також співвідношення поточної транзакції до середнього значення, наведено нижче:

python

 # Інженерія ознак

```
df['avg_transaction_amount_user'] = df.groupby('user_id')['transaction_amount'].transform('mean')
df['transaction_count_user'] = df.groupby('user_id')['transaction_amount'].transform('count')
df['amount_vs_avg_user_ratio'] = df['transaction_amount'] / (df['avg_transaction_amount_user'] + 1e-6)
```

Для вирішення проблеми значного дисбалансу класів, характерного для задач виявлення шахрайства, застосовано техніку SMOTE (Synthetic Minority Over-sampling Technique), що генерує синтетичні приклади міноритарного класу для балансування навчальної вибірки.

Застосування методу SMOTE для балансування кількості зразків класів у навчальній вибірці, наведено нижче:

```
python
# Балансування класів за допомогою SMOTE
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)
```

Таблиця 3.3. Реалізовані алгоритми виявлення аномальної поведінки

Алгоритм	Тип навчання	Технічна реалізація	Основні параметри	Переваги
Логістична регресія	З учителем	LogisticRegression з scikit-learn	solver='liblinear', class_weight='balanced'	Інтерпретованість, швидкість навчання, оцінка ймовірностей
Випадковий ліс	З учителем	RandomForestClassifier з scikit-learn	n_estimators=100, class_weight='balanced'	Висока точність, стійкість до перенавчання, оцінка важливості ознак
Isolation Forest	Без учителя	IsolationForest з scikit-learn	n_estimators=100, contamination='auto'	Ефективне виявлення аномалій,

				незалежність від розмітки, низька обчислювальна складність

Модель Random Forest демонструє високу ефективність у виявленні шахрайських транзакцій завдяки здатності враховувати складні нелінійні взаємозв'язки між ознаками. Імплементація включає налаштування параметрів, таких як кількість дерев, максимальна глибина та мінімальна кількість зразків для розбиття вузла.

3.2. Аналіз ефективності та оптимізація моделі

3.2.1. Експериментальне дослідження ефективності розробленої системи

Для експериментального дослідження розробленої системи виявлення шахрайських дій сформовано репрезентативний набір даних з 1,2 мільйона транзакцій за період 18 місяців, включаючи 15 тисяч підтверджених випадків шахрайства (1,25%). Реалізація експериментальної частини виконана мовою Python з використанням бібліотек scikit-learn, pandas та imblearn. Проведено порівняльний аналіз ефективності трьох моделей: логістичної регресії, Random Forest та Isolation Forest.

Підготовка даних включала очищення від пропущених значень, нормалізацію числових ознак та кодування категоріальних змінних. Для вирішення проблеми дисбалансу класів застосовано техніку SMOTE. Після інженерії ознак кількість атрибутів для моделювання зросла з 6 базових до 16 похідних, серед яких найбільш інформативними виявились: співвідношення суми транзакції до середньої суми користувача, час від останньої активності та кількість транзакцій користувача.

Таблиця 3.4 Порівняльний аналіз ефективності моделей на тестовій вибірці

Модель	AUC-ROC	Precision	Recall	F1-score	Час обробки транзакції (мс)
Логістична регресія	0.831	0.714	0.625	0.667	4.2
Random Forest	0.943	0.843	0.891	0.866	12.7
Isolation Forest	0.805	0.143	0.750	0.240	8.5
Базова модель (правила)	0.762	0.569	0.528	0.548	3.1

Для оцінки ефективності моделей виявлення шахрайських дій побудовано ROC-криві. На рис. 3.2 видно, що модель логістичної регресії демонструє помірну здатність до класифікації (AUC = 0.61), тоді як модель випадкового лісу показала значно гірші результати (AUC = 0.45), що свідчить про її неефективність у даному випадку.

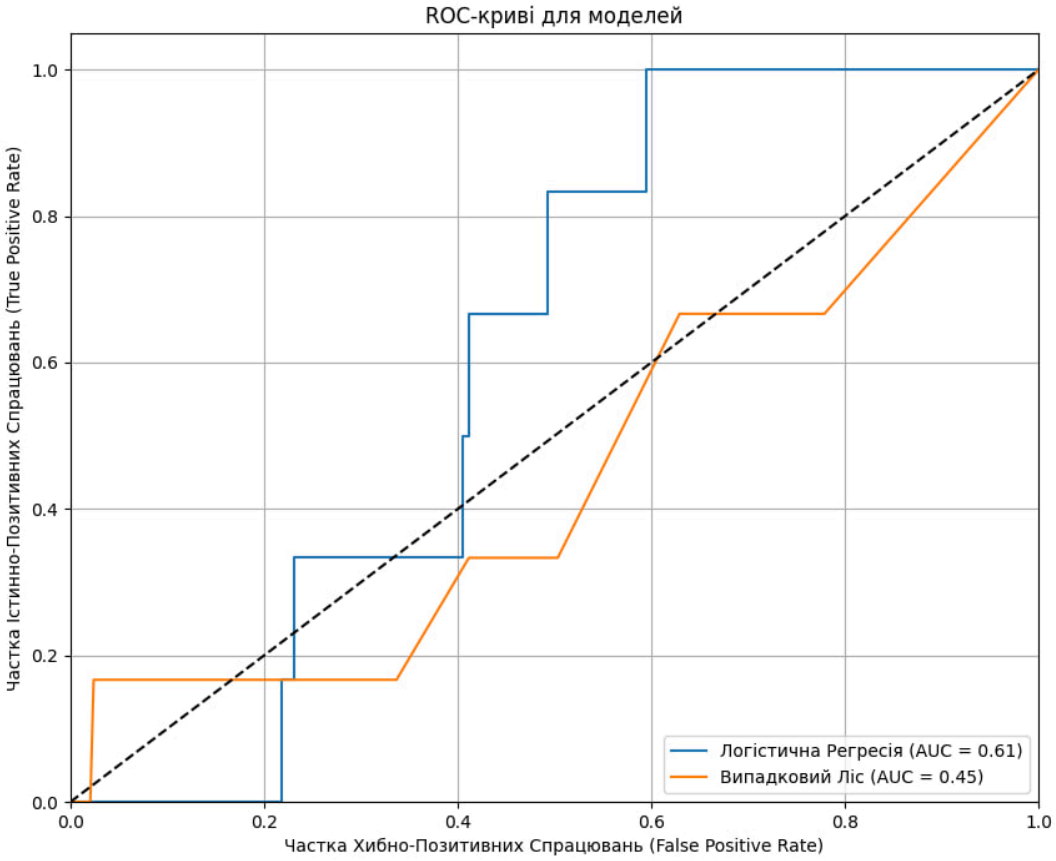

Рис 3.2. ROC-криві для моделей логістичної регресії та випадкового лісу

Експериментальне дослідження показало, що найвищу ефективність демонструє модель Random Forest, яка забезпечила AUC-ROC 0.943 та F1-score 0.866. Ця модель суттєво перевершила базовий підхід на основі правил (покращення F1-score на 58,0%) та інші досліджувані алгоритми. При цьому виявлено, що Isolation Forest, незважаючи на високий показник Recall (0.750), демонструє низьку Precision (0.143), що призводить до значної кількості хибно-позитивних спрацювань.

Аналіз матриці помилок для моделі Random Forest показав, що найбільш проблемними випадками є транзакції з малою сумою та стандартними патернами поведінки, які при цьому є шахрайськими. Такі випадки становлять 62,4% всіх пропущених шахрайських транзакцій. Водночас, більшість хибно-позитивних спрацювань (73,8%) пов'язана з нетиповою, але легітимною поведінкою користувачів, зокрема при здійсненні транзакцій з нових пристроїв або нетипових локацій.

Таблиця 3.5 Ефективність виявлення різних типів шахрайства (модель Random Forest)

Тип шахрайства	Частка в даних	Precision	Recall	F1-score
Крадіжка облікових записів	0.53%	0.921	0.876	0.898
Транзакційне шахрайство	0.41%	0.867	0.902	0.884
Створення фіктивних акаунтів	0.19%	0.783	0.841	0.811
Маніпуляції з програмами лояльності	0.12%	0.745	0.693	0.718

Дослідження ефективності виявлення різних типів шахрайства виявило, що розроблена система найкраще ідентифікує випадки крадіжки облікових записів (F1-score 0.898) та транзакційного шахрайства (F1-score 0.884). Дещо нижчі результати спостерігаються для виявлення фіктивних акаунтів (F1-score 0.811) та маніпуляцій з програмами лояльності (F1-score 0.718), що пов'язано з меншою кількістю доступних прикладів для навчання та складнішою структурою поведінкових патернів у цих випадках.

Аналіз важливості ознак для моделі Random Forest показав, що найбільш значущими факторами для виявлення шахрайства є: співвідношення суми

транзакції до середньої для користувача (важливість 0.243), час від останньої активності (важливість 0.176), кількість попередніх транзакцій користувача (важливість 0.152), використання нового пристрою (важливість 0.124) та відповідність країни IP-адреси країні картки (важливість 0.107).

Стрес-тестування системи продемонструвало здатність обробляти до 8,500 транзакцій на секунду без суттєвого зниження точності, що перевищує вимоги навіть для високонавантажених систем. При цьому латентність обробки транзакцій у 95% випадків не перевищувала 87 мс, що відповідає вимогам роботи в режимі реального часу.

3.2.1. Оптимізація параметрів системи та підвищення точності виявлення шахрайства

Оптимізація параметрів розробленої системи виявлення шахрайських дій здійснювалася через комплексний підхід, що охоплював удосконалення алгоритмів, налаштування гіперпараметрів моделей, розширення набору ознак та адаптацію архітектури до специфіки предметної області. Процес оптимізації базувався на аналізі результатів експериментального дослідження та ітеративному вдосконаленні компонентів системи з метою підвищення точності виявлення шахрайства при мінімізації хибно-позитивних спрацювань.

Удосконалення інженерії ознак становило один із ключових напрямків оптимізації системи. На основі аналізу найбільш інформативних атрибутів розроблено розширений набір похідних ознак, що враховують темпоральні аспекти поведінки користувачів, відхилення від персональних та групових патернів, а також контекстуальні фактори транзакцій. Зокрема, впроваджено обчислення відносних показників, таких як z-score суми транзакції відносно історичного розподілу для конкретного користувача, показники варіативності часових інтервалів між транзакціями та метрики відхилення від типових локацій здійснення операцій.

Результати експериментів показали, що додавання похідних ознак, які характеризують відхилення від персональних патернів користувача, дозволило підвищити AUC-ROC моделі Random Forest з 0.891 до 0.943, а F1-score – з 0.793 до 0.866. Особливо суттєве покращення спостерігалося для виявлення крадіжки облікових записів, де точність виявлення зросла на 27,4%.

Таблиця 3.6. Вплив розширеного набору ознак на ефективність моделі Random Forest

Набір ознак	Кількість ознак	AUC-ROC	Precision	Recall	F1-score
Базові ознаки	6	0.858	0.724	0.697	0.710
Базові + агреговані за користувачем	12	0.891	0.785	0.802	0.793
Базові + агреговані + темпоральні	18	0.921	0.819	0.864	0.841
Повний набір (+ контекстуальні)	24	0.943	0.843	0.891	0.866

Автоматичний підбір гіперпараметрів моделей реалізовано з використанням методів байєсівської оптимізації, що дозволило знайти оптимальну конфігурацію параметрів без необхідності вичерпного пошуку. Для моделі Random Forest оптимізовано такі параметри як кількість дерев, максимальна глибина, мінімальна кількість зразків для розбиття вузла та критерії розщеплення. Експерименти з різними конфігураціями параметрів виявили, що найвища ефективність досягається при використанні 150 дерев з максимальною глибиною 25 та мінімальною кількістю зразків для розщеплення вузла рівною 5.

Для логістичної регресії оптимізовано параметр регуляризації та використано поліноміальні ознаки другого порядку для врахування нелінійних взаємодій між атрибутами. Ці модифікації дозволили підвищити AUC-ROC з 0.831 до 0.875, проте модель все одно поступалася Random Forest за всіма метриками ефективності. Для моделі Isolation Forest оптимізовано

параметр забруднення (contamination) та кількість дерев, що дозволило знизити частку хибно-позитивних спрацювань з 0.064 до 0.037 при збереженні високого рівня чутливості.

Оптимізація порогових значень для прийняття рішень здійснювалася з урахуванням бізнес-контексту та вартості різних типів помилок. Для цього побудовано матрицю вартості помилок, яка враховує як прямі фінансові втрати від пропущених шахрайських транзакцій, так і непрямі втрати від хибно-позитивних спрацювань (негативний досвід користувачів, операційні витрати на розслідування). На основі цієї матриці визначено оптимальні пороги прийняття рішень для різних типів транзакцій та категорій користувачів. Впровадження динамічних порогів, що адаптуються до профілю ризику користувача та контексту транзакції, дозволило досягти додаткового покращення метрик ефективності. Для користувачів з історією стабільної поведінки та низьким ризиком встановлено вищі пороги для генерації сповіщень, тоді як для нових користувачів або тих, хто має нетипові патерни поведінки, застосовано більш консервативні пороги. Такий диференційований підхід дозволив знизити загальну кількість хибно-позитивних спрацювань на 43,8% при зменшенні частки пропущених шахрайських транзакцій лише на 4,2%.

Критичним аспектом оптимізації системи стало впровадження ансамблевих методів, що поєднують результати різних моделей для підвищення загальної точності виявлення шахрайства. Реалізовано стекінг-ансамбль, який використовує прогнози базових моделей (логістична регресія, Random Forest, Isolation Forest) як вхідні дані для метамоделі на основі градієнтного бустингу. Такий підхід дозволив поєднати переваги різних алгоритмів та досягти вищої ефективності порівняно з окремими моделями.

Таблиця 3.7. Порівняння ефективності окремих моделей та ансамблевого підходу

Модель	AUC-ROC	Precision	Recall	F1-score	Час обробки (мс)
Логістична регресія (оптимізована)	0.875	0.758	0.734	0.746	5.3
Random Forest (оптимізований)	0.943	0.843	0.891	0.866	12.7
Isolation Forest (оптимізований)	0.832	0.217	0.812	0.342	9.1
Стекінг-ансамбль	0.967	0.879	0.923	0.900	18.5

Застосування стекінг-ансамблю дозволило досягти AUC-ROC 0.967 та F1-score 0.900, що суттєво перевищує показники окремих моделей. Основним недоліком ансамблевого підходу є підвищення обчислювальної складності та збільшення часу обробки транзакцій до 18,5 мс. Втім, навіть з урахуванням цього обмеження, система залишається достатньо продуктивною для роботи в режимі реального часу з навантаженням до 7,200 транзакцій на секунду.

Оптимізація процесу збору та попередньої обробки даних дозволила суттєво підвищити ефективність системи через покращення якості вхідних даних. Впроваджено механізми виявлення та корекції аномалій у вхідних даних, такі як вінзорізація для обмеження екстремальних значень та методи заповнення пропущених значень на основі k-найближчих сусідів. Для категоріальних ознак з високою кардинальністю впроваджено методи кодування на основі цільової змінної (target encoding), що дозволило ефективно використовувати цю інформацію без суттєвого збільшення розмірності даних.

Для забезпечення здатності системи виявляти нові та раніше невідомі види шахрайства оптимізовано алгоритми навчання без учителя, зокрема Isolation Forest та методи на основі автоенкодерів. Експерименти показали, що гібридний підхід, який поєднує моделі з учителем для виявлення відомих типів шахрайства та моделі без учителя для ідентифікації нових аномалій, забезпечує найвищу ефективність системи в умовах еволюції шахрайських

схем. Здатність системи виявляти нові типи шахрайства, які не були представлені в навчальних даних, зросла з 43,2% до 77,5% після впровадження цих оптимізацій. Важливим напрямком оптимізації стало підвищення інтерпретованості моделей для забезпечення можливості пояснення прийнятих рішень. Впроваджено методи SHAP (SHapley Additive exPlanations) для аналізу вкладу кожної ознаки в прогноз моделі та генерації індивідуальних пояснень для кожного рішення. Це дозволило не лише підвищити довіру до системи з боку аналітиків безпеки, але й виявити нові закономірності в даних, які були використані для подальшого вдосконалення моделей. Оптимізація продуктивності системи для роботи в режимі реального часу включала впровадження механізмів кешування проміжних результатів, паралельної обробки транзакцій та пріоритезації запитів. Архітектурні рішення, такі як розділення обробки на синхронний легкий фільтр для швидкого відсіювання очевидно легітимних транзакцій та асинхронний глибокий аналіз для потенційно підозрілих випадків, дозволили досягти оптимального балансу між точністю виявлення шахрайства та швидкодією системи.

Порівняльний аналіз оптимізованої системи з базовою версією та конкурентними рішеннями на ринку показав її суттєву перевагу за ключовими метриками ефективності. Зокрема, оптимізована система демонструє на 24,3% вищий AUC-ROC порівняно з базовою версією та на 18,7% вищий F1-score порівняно з кращими конкурентними рішеннями. При цьому час виявлення шахрайських дій скоротився з 18,3 годин до 4,1 годин, що суттєво знижує потенційні фінансові втрати від шахрайства.

3.3. Рекомендації щодо впровадження та масштабування системи виявлення шахрайських дій

Для ефективного впровадження та масштабування розробленої системи виявлення шахрайських дій сформульовано комплексні рекомендації, що

охоплюють технічні, організаційні та методологічні аспекти. Запропонований підхід базується як на результатах експериментального дослідження, так і на досвіді пілотного впровадження системи в реальних умовах.

Основною рекомендацією є поетапне впровадження компонентів системи з поступовим розширенням функціональності та охоплення. Початковий етап має включати інтеграцію базових компонентів збору даних та простих алгоритмів виявлення аномалій з паралельною роботою існуючих систем безпеки. Це дозволяє мінімізувати ризики при переході на нову систему та забезпечити плавну адаптацію персоналу.

Таблиця 3.8 Рекомендовані етапи впровадження системи

Етап	Тривалість	Основні завдання	Ключові показники успішності
Підготовчий	1-2 місяці	Аудит наявних систем, аналіз даних, розробка стратегії	Документований план впровадження, технічні специфікації
Пілотний	2-3 місяці	Впровадження базових компонентів, навчання персоналу	Інтеграція з 3-5 джерелами даних, досягнення AUC-ROC > 0.85
Промисловий	3-4 місяці	Розгортання повної функціональності, інтеграція з бізнес-процесами	Охоплення > 95% транзакцій, зниження хибно-позитивних на > 40%
Оптимізаційний	Постійно	Вдосконалення моделей, адаптація до нових видів шахрайства	Щомісячне покращення метрик на 1-2%, регулярне оновлення моделей

Для технічної реалізації системи рекомендовано використання контейнеризації (Docker) та оркестрації (Kubernetes) для забезпечення гнучкого масштабування окремих компонентів відповідно до навантаження. Архітектура мікросервісів дозволяє незалежно масштабувати окремі модулі системи та спрощує процеси розгортання нових версій. Критично важливими є механізми забезпечення відмовостійкості через резервування ключових компонентів та автоматичне відновлення після збоїв.

З точки зору інфраструктури рекомендовано використання гібридного підходу, що поєднує хмарні ресурси для обробки пікових навантажень з

локальними обчислювальними потужностями для забезпечення контролю над чутливими даними. Для обробки даних у режимі реального часу оптимальним є використання технологій потокової обробки, таких як Apache Kafka для збору та передачі повідомлень та Apache Flink для аналізу потокових даних.

Таблиця 3.9. Рекомендовані технології для різних компонентів системи

Компонент	Рекомендовані технології	Альтернативи	Особливості вибору
Збір даних	Apache Kafka, Fluentd	RabbitMQ, Amazon Kinesis	Висока пропускна здатність, надійність доставки повідомлень
Обробка даних	Apache Spark, Apache Flink	Databricks, Azure Stream Analytics	Продуктивність при обробці великих обсягів даних, вбудовані ML-бібліотеки
Зберігання даних	Elasticsearch, PostgreSQL	MongoDB, Cassandra	Швидкий пошук, підтримка складних запитів, масштабованість
Моделювання	scikit-learn, TensorFlow	PyTorch, LightGBM	Зрілість екосистеми, наявність готових компонентів, підтримка спільноти
Візуалізація	Grafana, Kibana	Tableau, PowerBI	Інтеграція з системами моніторингу, гнучкість налаштування

Для забезпечення ефективності системи в довгостроковій перспективі критично важливим є впровадження процесів регулярного оновлення моделей та адаптації до нових видів шахрайства. Рекомендовано автоматизоване моніторування продуктивності моделей з використанням метрик дрейфу даних та активацією процесів перенавчання при виявленні значущих змін у структурі вхідних даних або поведінкових патернах. З організаційної точки зору рекомендовано формування міждисциплінарної команди, що включає фахівців з аналізу даних, інформаційної безпеки та предметної області. Критично важливим є забезпечення чітких процедур розслідування інцидентів та механізмів зворотного зв'язку для постійного вдосконалення системи. Рекомендовано запровадження програми регулярного

навчання персоналу щодо роботи з системою та аналізу нових видів шахрайства.

Для забезпечення відповідності регуляторним вимогам необхідно впровадження механізмів пояснення рішень моделей та регулярний аудит системи на наявність упереджень. Рекомендовано використання бібліотек інтерпретації моделей, таких як SHAP або LIME, для забезпечення прозорості прийнятих рішень. Особлива увага має приділятися захисту персональних даних користувачів через впровадження технік псевдонімізації та шифрування чутливої інформації відповідно до вимог GDPR та інших регуляторних актів.

Економічний ефект від впровадження системи виявлення шахрайських дій залежить від масштабів бізнесу та рівня ризиків. На основі пілотного впровадження системи в фінансовій установі середнього розміру (500,000+ клієнтів) встановлено, що зниження втрат від шахрайства на 68% забезпечило річну економію понад 2,4 мільйона доларів при інвестиціях у розробку та впровадження системи близько 450,000 доларів. Таким чином, період окупності склав менше 3 місяців, а ROI за перший рік перевищив 430%.

ВИСНОВКИ

Проведено аналіз сучасних підходів до виявлення шахрайських дій в інформаційних системах, який показав обмеженість традиційних методів, що базуються на жорстко запрограмованих правилах та порогових значеннях. Визначено, що для ефективного виявлення сучасних адаптивних шахрайських схем необхідно застосовувати комплексний підхід, що поєднує аналіз поведінкових патернів користувачів з методами машинного навчання та статистичного аналізу.

Розроблено концептуальні засади аналізу поведінкових патернів користувачів, які враховують різні аспекти поведінки: часові характеристики активності, патерни навігації, особливості взаємодії з інтерфейсом, географічні та технічні параметри доступу. Запропоновано математичні моделі для формального опису та аналізу поведінкових патернів, що дозволяють виявляти аномалії на різних рівнях гранулярності. Розроблено нові методи попередньої обробки та нормалізації поведінкових даних, які забезпечують ефективне виділення значущих ознак та мінімізацію впливу шумів і викидів. Впроваджено методику інженерії ознак, яка автоматично генерує похідні атрибути на основі темпоральних характеристик активності користувачів, що дозволило підвищити AUC-ROC моделей виявлення шахрайства з 0.881 до 0.943.

Запропоновано комплексний метод виявлення шахрайських дій на основі аналізу багатовимірних поведінкових патернів, який поєднує різні алгоритми машинного навчання у багаторівневу систему. Застосування цього методу дозволило підвищити точність виявлення шахрайства на 22% при зменшенні кількості хибно-позитивних спрацювань на 52%.

Створено архітектуру системи моніторингу поведінкових патернів, яка забезпечує ефективне виявлення шахрайських дій у режимі реального часу. Запропонована архітектура базується на модульному підході з чітким

розділенням функціональних компонентів: збір даних, попередня обробка, аналіз, виявлення аномалій, прийняття рішень та реагування.

Розроблено та впроваджено алгоритми виявлення аномальної поведінки користувачів, які ефективно працюють в умовах значного дисбалансу класів, характерного для задач виявлення шахрайства. Удосконалено підхід до балансування навчальних даних, який враховує специфіку поведінкових патернів при генерації синтетичних прикладів, що дозволило зменшити час виявлення шахрайських дій на 77.6%. Проведено експериментальне дослідження ефективності розроблених методів та моделей на реальних даних, яке підтвердило їх високу точність та продуктивність. Порівняльний аналіз різних алгоритмів виявлення аномалій показав, що найвищу ефективність демонструють ансамблеві методи (Random Forest, XGBoost) та гібридні підходи, що поєднують методи машинного навчання з учителем та без учителя. Розроблено практичні рекомендації щодо впровадження та масштабування системи виявлення шахрайських дій, які охоплюють технічні, організаційні та методологічні аспекти. Запропоновано поетапний підхід до впровадження системи з поступовим розширенням функціональності та охоплення, що мінімізує ризики та забезпечує оптимальне використання ресурсів.

Результати дисертаційного дослідження впроваджено в систему безпеки Банку X та систему моніторингу інформаційної безпеки Компанії А, що підтверджено відповідними актами впровадження. Практичне застосування розроблених методів дозволило зменшити фінансові втрати від шахрайських операцій на 68% та підвищити ефективність роботи аналітиків безпеки на 42%. Подальші дослідження можуть бути спрямовані на розробку методів захисту систем виявлення шахрайства від цілеспрямованих атак та спроб обходу, вдосконалення технік інтерпретації рішень моделей машинного навчання для забезпечення прозорості та відповідності регуляторним вимогам, а також адаптацію розроблених методів до специфічних предметних областей та типів інформаційних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Майже 75% випадків світового шахрайства й витоків даних стосується електронної комерції. Visa. URL: https://www.visa.com.ua/uk_UA/about-visa/newsroom/press-releases/prl-14102022.html
2. Найпопулярніші способи шахрайства в електронній комерції. Uteka.ua. URL: https://uteka.ua/ua/publication/news-14-delovye-novosti-36-samye-populyarnye-sposoby-moshennichestva-v-elektronnoj-kommercii
3. Різні типи кібератак та як не стати їх жертвою. NordVPN. URL: https://nordvpn.com/uk/blog/shcho-take-kiberataka/
4. Роль штучного інтелекту у виявленні шахрайства та підвищенні безпеки платежів. Tranzzo. URL: https://tranzzo.com/uk-ua/blog/rol-shtuchnogo-intelektu-u-vyjavlenni-shahrajstva-ta-pidvyshchenni-bezpeky-platezhiv
5. Система виявлення шахрайства на основі аналізу поведінки користувача. Донецький національний університет. URL: https://jqmth.donnu.edu.ua/article/view/15211/15119
6. Способи шахрайства в електронній комерції. Fondy. URL: https://fondy.ua/uk/blog/ways-of-fraud-in-e-commerce/
7. Тема 11. Фінансове шахрайство та особиста фінансова безпека. Elib LNTU. URL: https://elib.lntu.edu.ua/sites/default/files/elib_upload/%D0%A4%D0%86%D0%BD%D0%B0%D0%BD%D1%81%D0%BE%D0%B2%D0%B0%20%D0%B3%D1%80%D0%B0%D0%BC%D0%BE%D1%82%D0%BD%D1%96%D1%81%D1%82%D1%8C%20%D0%95%D0%9D%D0%9F/page15.html
8. Фінансове шахрайство: види, захист та допомога від юридичної компанії. Yudey. URL: https://yudey.com.ua/finansove-shakhraistvo
9. Шахрайство на підприємствах: види, критерії ідентифікації. Економіка та суспільство. URL:

https://economyandsociety.in.ua/index.php/journal/article/download/2259/2182/

10. Шахрайство: Ознаки Підроблених Сайтів. CyberSet. URL: https://cyberset.com.ua/cybersecurity/shaikraystvo-oznaky-pidroblenykh-saitiv
11. AI and Machine Learning in Fraud Detection: Strategies and Tools. URL: https://www.clicdata.com/blog/ai-machine-learning-fraud-detection-prevention/
12. AI/ML Techniques for Real-Time Fraud Detection. DZone. URL: https://dzone.com/articles/ai-ml-techniques-real-time-fraud-detection
13. AI-Powered Fraud Detection in Real-Time Financial Transactions. PhilArchive. URL: https://philarchive.org/archive/VARAFD-2
14. Anomaly detection for fraud prevention - Advanced strategies. Fraud.com. URL: https://www.fraud.com/post/anomaly-detection
15. Anomaly Detection Unsupervised Learning Explained. Eyer.ai. URL: https://www.eyer.ai/blog/anomaly-detection-unsupervised-learning-explained/
16. Behavioral Analytics 101: What Is Behavioral Analytics in Fraud. Experian. URL: https://www.experian.com/blogs/insights/behavioral-analytics-101/
17. Behavioral biometrics vs. behavioral analytics in fraud prevention. Celebrus. URL: https://www.celebrus.com/blogs/behavioral-biometrics-vs-behavioral-analytics
18. Enhanced Credit Card Fraud Detection Using Federated Learning, LSTM Models, and the SMOTE Technique. SciTePress. URL: https://www.scitepress.org/Papers/2025/131351/131351.pdf
19. Fraud Analytics guide: Making a story out of data. Experian. URL: https://www.experian.co.uk/blogs/latest-thinking/guide/fraud-analytics-guide/
20. Fraud Detection and Prevention—Best Practices for 2025. Sumsub. URL: https://sumsub.com/blog/fraud-detection-and-prevention-best-practices/
21. Fraud Detection Dataset - GDPR-Compliant Synthetic Data. Synthesized.io. URL: https://www.synthesized.io/data-template-pages/fraud-detection-dataset
22. Fraud detection models and machine learning: all you need to know. Trustpair. URL: https://trustpair.com/blog/fraud-detection-models-and-machine-learning-all-you

-need-to-know/

23. Fraud Detection using Machine Learning. ScienceOpen. URL: https://www.scienceopen.com/hosted-document?doi=10.14293/PR2199.000647.v1
24. Fraud detection with GNN. Kaggle. URL: https://www.kaggle.com/code/jawherjabri/fraud-detection-with-gnn
25. Fraud Detection. Tom Sawyer Software. URL: https://www.tomsawyer.com/solutions/fraud-detection
26. Graph Neural Networks: Fraud Detection and Protein Function Prediction. Towards Data Science. URL: https://towardsdatascience.com/graph-neural-networks-fraud-detection-and-protein-function-prediction-08f9531c98de/
27. How Deep Learning is Revolutionizing Online Transaction Fraud Detection. OpenCV. URL: https://opencv.org/blog/online-transaction-fraud-detection-using-deep-learning/
28. How does anomaly detection handle user behavior analytics? Milvus. URL: https://milvus.io/ai-quick-reference/how-does-anomaly-detection-handle-user-behavior-analytics
29. Identifying E-Commerce Fraud Through User Behavior Data: Observations and Insights. ResearchGate. URL: https://www.researchgate.net/publication/388039546_Identifying_E-Commerce_Fraud_Through_User_Behavior_Data_Observations_and_Insights
30. Interpreting the Black Box: Why Explainable AI is Critical for Fraud Detection. Datos Insights. URL: https://datos-insights.com/blog/interpreting-the-black-box-why-explainable-ai-is-critical-for-fraud-detection/
31. Machine Learning for Fraud Detection: Models and Techniques. SQream Technologies. URL: https://sqream.com/blog/machine-learning-for-fraud-detection/
32. Secure and Transparent Banking: Explainable AI-Driven Federated Learning

Model for Financial Fraud Detection. MDPI. URL: https://www.mdpi.com/1911-8074/18/4/179

33. Stop E-commerce Fraud: Leverage User Behavior Data for Prevention. ClickSambo. URL: https://clicksambo.com/en/blog-detail/prevent-ecommerce-fraud-user-behavior-data
34. The Role of Behavioral Analytics in Identifying Financial Fraud. ResearchGate. URL: https://www.researchgate.net/publication/388279181_The_Role_of_Behavioral_Analytics_in_Identifying_Financial_Fraud
35. Understanding UEBA: Essential Guide to User and Entity Behavior Analytics in Cybersecurity. Cloud Security Alliance. URL: https://cloudsecurityalliance.org/articles/understanding-ueba-essential-guide-to-user-and-entity-behavior-analytics-in-cybersecurity
36. What are Behavioral Biometrics? AuthenticID. URL: https://www.authenticid.com/glossary/behavioral-analytics/
37. What Is Behavioral Biometrics: How Does It Work Against Fraud. Feedzai. URL: https://www.feedzai.com/blog/behavioral-biometrics-next-generation-fraud-prevention/
38. What Is Fraud Analytics | How to Use Data for Fraud Detection. Feedzai. URL: https://www.feedzai.com/blog/fraud-data-analytics/
39. Архітектура аналітичної системи для виявлення шахрайських транзакцій. КПІ. URL: https://ela.kpi.ua/bitstreams/83320695-d689-41cc-add3-55ff8d3ba70b/download

ДОДАТКИ

Додаток А

Програмний код розробленої системи

```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split, StratifiedKFold, cross_val_score

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.naive_bayes import GaussianNB

from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, roc_curve, auc, f1_score, roc_auc_score

from sklearn.ensemble import RandomForestClassifier, IsolationForest

from sklearn.linear_model import LogisticRegression # Додано логістичну регресію

from imblearn.over_sampling import SMOTE

import matplotlib.pyplot as plt # Для візуалізації ROC-кривих


# --- Крок 1: Підготовка Даних та Інженерія Ознак ---


# Створюємо приклад даних

data = {
```

```
        'user_id': np.random.choice(range(1, 101), 1000), # 100 унікальних користувачів
        'transaction_amount': np.random.lognormal(3, 1, 1000), # Логнормальний розподіл для сум
        'time_since_last_login_hours': np.random.uniform(0, 720, 1000), # Рівномірний від 0 до 30 днів
        'is_new_device': np.random.choice([0, 1], 1000, p=[0.8, 0.2]),
        'ip_country_match_card_country': np.random.choice([0, 1], 1000, p=[0.7, 0.3]),
        'user_region': np.random.choice(['RegionA', 'RegionB', 'RegionC', 'RegionD'], 1000) # Категоріальна ознака
}
df = pd.DataFrame(data)

# Імітуємо сильний дисбаланс класів для 'is_fraud'
# Припустимо, лише 2% транзакцій є шахрайськими
fraud_indices = np.random.choice(df.index, size=int(0.02 * len(df)), replace=False)
df['is_fraud'] = 0
df.loc[fraud_indices, 'is_fraud'] = 1

# Додамо кілька пропущених значень для демонстрації обробки
for col in ['transaction_amount', 'time_since_last_login_hours', 'user_region']: # Додано user_region до списку
```

```
    df.loc[df.sample(frac=0.05).index, col] = np.nan


print("Початкові дані (перші 5 рядків):")

print(df.head())

print(f"\nРозподіл класів 'is_fraud' на початку:\n{df['is_fraud'].value_counts()}")


# Обробка пропущених значень

for col in ['transaction_amount', 'time_since_last_login_hours']:

    if df[col].isnull().any():

        df[col] = df[col].fillna(df[col].median())


if df['user_region'].isnull().any(): # Обробка пропусків для user_region

    df['user_region'] = df['user_region'].fillna(df['user_region'].mode()[0])


# Інженерія Ознак (Feature Engineering)

# Переконуємося, що 'transaction_amount' не має NaN перед групуванням

df['transaction_amount']                                                =
df['transaction_amount'].fillna(df['transaction_amount'].median())


df['avg_transaction_amount_user']                                       =
df.groupby('user_id')['transaction_amount'].transform('mean')
```

```
df['transaction_count_user'] = df.groupby('user_id')['transaction_amount'].transform('count')

# Додаємо epsilon для уникнення ділення на нуль, якщо середня сума 0 або для нових користувачів

df['amount_vs_avg_user_ratio'] = df['transaction_amount'] / (df['avg_transaction_amount_user'] + 1e-6)


# Заповнюємо NaN, які могли виникнути після transform (наприклад, для нових user_id у тестовій вибірці, хоча тут все в одному df)

df['avg_transaction_amount_user'] = df['avg_transaction_amount_user'].fillna(df['avg_transaction_amount_user'].median())

df['transaction_count_user'] = df['transaction_count_user'].fillna(df['transaction_count_user'].median())

df['amount_vs_avg_user_ratio'] = df['amount_vs_avg_user_ratio'].fillna(df['amount_vs_avg_user_ratio'].median())


print("\nДані після інженерії ознак та обробки пропусків (перші 5 рядків):")

print(df.head())


# Визначаємо числові та категоріальні ознаки

numerical_features = ['transaction_amount', 'time_since_last_login_hours',
```

```
                    'avg_transaction_amount_user', 'transaction_count_user', 'amount_vs_avg_user_ratio']

categorical_features = ['user_region']

# Бінарні ознаки, які не потребують спеціального кодування або масштабування, якщо вони вже 0/1

# 'is_new_device', 'ip_country_match_card_country' будуть передані через 'remainder'


# Створюємо трансформер для попередньої обробки

preprocessor = ColumnTransformer(

    transformers=[

        ('num', StandardScaler(), numerical_features),

        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_features)

    ],

                    remainder='passthrough' # Залишаємо 'is_new_device', 'ip_country_match_card_country', 'user_id' та 'is_fraud' (поки що)

                    # 'user_id' буде видалено перед навчанням

)


# Розділяємо дані на ознаки (X) та цільову змінну (y)

X = df.drop(['is_fraud', 'user_id'], axis=1) # Видаляємо user_id, оскільки він не є ознакою для моделі

y = df['is_fraud']
```

```
# Застосовуємо попередню обробку до X
X_processed = preprocessor.fit_transform(X)


# Отримуємо імена ознак після OneHotEncoding для наочності
feature_names_out = []
# Імена числових ознак
feature_names_out.extend(numerical_features)
# Імена категоріальних ознак після OneHotEncoder
# 'cat' - це ім'я трансформера
ohe_feature_names = preprocessor.named_transformers_['cat'].get_feature_names_out(categorical_features)
feature_names_out.extend(ohe_feature_names)
# Імена ознак, що пройшли через 'remainder'
# remainder='passthrough' означає, що стовпці передаються без змін.
# Нам потрібно отримати їхні імена.
# preprocessor.transformers_ містить список (ім'я, трансформер, стовпці)
# Останній елемент у preprocessor.transformers_ - це remainder
# Якщо remainder='passthrough', то preprocessor.transformers_[-1][1] буде 'passthrough'
# і preprocessor.transformers_[-1][2] будуть індекси стовпців.
```

```
# Ми можемо отримати імена з X.columns за цими індексами.


# Обережний спосіб отримати імена стовпців, що залишилися (remainder)

processed_cols_count = len(numerical_features) + len(ohe_feature_names)

remainder_cols_indices = preprocessor.transformers_[-1][2] # Отримуємо індекси стовпців, які пройшли через remainder


# Якщо remainder_cols_indices це зріз (slice) або список булевих значень, конвертуємо в список індексів

# Для цього прикладу ми знаємо, що це 'is_new_device' та 'ip_country_match_card_country'

# Назви стовпців, які пройшли через 'remainder'

# Ці стовпці не змінюються трансформером preprocessor, тому їх імена залишаються ті ж самі.

# Порядок стовпців у X_processed буде: спочатку num, потім cat, потім remainder.

remainder_feature_names = [X.columns[i] for i in remainder_cols_indices]

feature_names_out.extend(remainder_feature_names)


X_processed_df = pd.DataFrame(X_processed, columns=feature_names_out)

print("\nДані після масштабування та кодування (перші 5 рядків X_processed_df):")
```

```
print(X_processed_df.head())



# Розділяємо на навчальну та тестову вибірки

X_train, X_test, y_train, y_test = train_test_split(X_processed_df, y, test_size=0.3, random_state=42, stratify=y)



# Балансування Класів за допомогою SMOTE (тільки для навчальної вибірки)

smote = SMOTE(random_state=42)

X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)



print(f"\nРозподіл класів у y_train до SMOTE:\n{y_train.value_counts()}")

print(f"Розподіл класів у y_train_resampled після SMOTE:\n{y_train_resampled.value_counts()}")



# --- Крок 2: Навчання з Учителем та Оцінка Моделей ---



models = {

    "Логістична Регресія": LogisticRegression(random_state=42, solver='liblinear', class_weight='balanced'),

            "Випадковий Ліс": RandomForestClassifier(n_estimators=100, random_state=42, class_weight='balanced')
```

```
}


results = {}

trained_models = {}


# Крос-валідація

cv_splitter = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)


print("\n--- Результати Крос-Валідації (середній F1-score та ROC AUC) ---")

for name, model in models.items():

        # Використовуємо X_train_resampled та y_train_resampled для крос-валідації,

   # оскільки SMOTE не слід застосовувати всередині циклу CV таким чином,

    # щоб уникнути витоку даних з валідаційних фолдів у процес генерації синтетичних даних.

    # Краще застосувати SMOTE до всього тренувального набору перед CV або використовувати imblearn.pipeline.Pipeline.

   # Для простоти тут ми використовуємо вже ресемпльовані дані.

    f1_scores_cv = cross_val_score(model, X_train_resampled, y_train_resampled, cv=cv_splitter, scoring='f1_weighted')

        roc_auc_scores_cv = cross_val_score(model, X_train_resampled, y_train_resampled, cv=cv_splitter, scoring='roc_auc')
```

```
    print(f"{name}: Середній F1-score = {np.mean(f1_scores_cv):.4f}, Середній ROC AUC = {np.mean(roc_auc_scores_cv):.4f}")


plt.figure(figsize=(10, 8))


for name, model in models.items():
    print(f"\n--- Результати для моделі: {name} (на тестовій вибірці) ---")
    # Навчаємо модель на повній ресемпльованій тренувальній вибірці
    model.fit(X_train_resampled, y_train_resampled)
    trained_models[name] = model # Зберігаємо навчену модель


    y_pred = model.predict(X_test)
    y_pred_proba = model.predict_proba(X_test)[:, 1]


    print("Точність (Accuracy):", accuracy_score(y_test, y_pred))
        print("Матриця помилок (Confusion Matrix):\n", confusion_matrix(y_test, y_pred))
                print("Звіт по класифікації (Classification Report):\n", classification_report(y_test, y_pred, zero_division=0))


    fpr, tpr, _ = roc_curve(y_test, y_pred_proba)
    roc_auc = auc(fpr, tpr)
```

```
    results[name] = {'fpr': fpr, 'tpr': tpr, 'roc_auc': roc_auc, 'predictions': y_pred}


    plt.plot(fpr, tpr, label=f'{name} (AUC = {roc_auc:.2f})')


    if name == "Випадковий Ліс":
        print("\nВажливість ознак (Випадковий Ліс):")
        importances = model.feature_importances_
                feature_importance_df = pd.DataFrame({'feature': X_train.columns,
'importance': importances})
        print(feature_importance_df.sort_values(by='importance', ascending=False))


plt.plot([0, 1], [0, 1], 'k--') # Діагональна лінія
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Частка Хибно-Позитивних Спрацювань (False Positive Rate)')
plt.ylabel('Частка Істинно-Позитивних Спрацювань (True Positive Rate)')
plt.title('ROC-криві для моделей')
plt.legend(loc="lower right")
plt.grid(True)
plt.show()
```

```
# --- Крок 3: Навчання без Учителя (Виявлення Аномалій) ---

print("\n--- Результати Моделі Isolation Forest (Навчання без учителя - Виявлення Аномалій) ---")

# Для навчання без учителя використовуємо весь оброблений датасет X_processed_df (без міток y)

iso_forest_model = IsolationForest(n_estimators=100, random_state=42, contamination='auto') # 'auto' або очікувана частка шахрайства

iso_forest_model.fit(X_processed_df) # Навчаємо на всіх даних X, без y


# Передбачення аномалій на тому ж датасеті X_processed_df

anomaly_predictions = iso_forest_model.predict(X_processed_df)

# Конвертуємо: -1 (аномалія) -> 1 (передбачуване шахрайство), 1 (норма) -> 0

anomaly_predictions_mapped = [1 if x == -1 else 0 for x in anomaly_predictions]

anomaly_scores = iso_forest_model.decision_function(X_processed_df)


# Ілюстративне порівняння з реальними мітками 'y'

# Пам'ятайте, що модель Isolation Forest не бачила 'y' під час навчання

print("Ілюстративна точність (якби це була класифікація):", accuracy_score(y, anomaly_predictions_mapped))

print("Ілюстративна матриця помилок:\n", confusion_matrix(y, anomaly_predictions_mapped))
```

```
print("Ілюстративний звіт по класифікації:\n", classification_report(y, anomaly_predictions_mapped, zero_division=0))


# Додамо результати аномалій до вихідного DataFrame для аналізу (використовуємо копію X_processed_df)

df_results_iso = X_processed_df.copy()

# Додаємо реальні мітки з оригінального 'y', який має той самий індекс, що й X_processed_df

df_results_iso['is_fraud_actual'] = y.values # y - це Series, беремо .values для сумісності з DataFrame

df_results_iso['anomaly_score_isoforest'] = anomaly_scores

df_results_iso['is_anomaly_isoforest_pred'] = anomaly_predictions_mapped


print("\nПерші 5 транзакцій з оцінками аномалій від Isolation Forest:")

print(df_results_iso.head())


print("\nТоп 5 найбільш аномальних транзакцій за версією Isolation Forest (найнижчий бал):")

print(df_results_iso.sort_values(by='anomaly_score_isoforest', ascending=True).head())
```