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

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Методи штучного інтелекту для захисту банківських продуктів від зовнішнього шахрайства»

на здобуття освітнього ступеня магістра
зі спеціальності 124 Системний аналіз
освітньо-професійної програми «Системний аналіз»

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

_____ Анастасія СТЕПАНОВА
(підпис)

Виконав: здобувач вищої освіти групи САДМ-61
Анастасія СТЕПАНОВА

Керівник: Ігор ПАТРАКЕЄВ
к.ф -м.н., доцент

Рецензент: _____
науковий ступінь, вчене звання *Ім'я, ПРІЗВИЩЕ*

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 124 Системний аналіз

Освітньо-професійна програма «Системний аналіз»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інформаційних систем та технологій
_____ Каміла СТОРЧАК
«_____» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Степановій Анастасії Володимирівни

1. Тема кваліфікаційної роботи: «Методи штучного інтелекту для захисту банківських продуктів від зовнішнього шахрайства»

керівник кваліфікаційної роботи Ігор ПАТРАКЄЄВ, к.т.н., доцент,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «15» жовтня 2024 р. № 320.

2. Строк подання кваліфікаційної роботи «17» грудня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: Науково-технічна література щодо шахрайства в банківській сфері, методи машинного навчання, технічна документація системи Stronghold, демонстраційні дані транзакцій для навчання та тестування ML-моделі.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз сучасних типів шахрайства у фінансовій сфері.
2. Методи виявлення шахрайства за допомогою машинного навчання.

3. Огляд функціональних можливостей системи Stronghold.
4. Розробка ML-моделі для виявлення шахрайських транзакцій.
5. Інтеграція розробленої ML-моделі в систему Stronghold.

5. Перелік ілюстративного матеріалу: *презентація*
   1. Типи шахрайства в банківській сфері та методи їх виявлення.
   2. Алгоритми машинного навчання для боротьби з шахрайством.
   3. Архітектура ML-моделі для детекції шахрайства.
   4. Візуалізація роботи ML-моделі та її інтеграції у Stronghold.
   5. Результати оцінки ефективності ML-моделі.
   6. Демонстрація роботи системи Stronghold із використанням ML-моделі.

6. Дата видачі завдання «16» жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	16.10-23.10.2024	
2	Вивчення технологій машинного навчання	24.10-28.10.2024	
3	Дослідження алгоритмів машинного навчання	29.10-05.11.2024	
4	Дослідження алгоритмів машинного навчання	05.11-10.11.2024	
5	Аналіз функціоналу системи Stronghold	11.11-20.11.2024	
6	Розробка ML-моделі: підготовка даних, навчання, тестування	21.11-24.11.2024	
7	Інтеграція ML-моделі в Stronghold	25.11-29.11.2024	
8	Оформлення роботи: вступ, висновки, реферат	29.11-30.11.2024	
9	Розробка демонстраційних матеріалів	01.12-05.12.2024	
10	Попередній захист роботи	23.12.2024	

Здобувач вищої освіти _____ Анастасія СТЕПАНОВА
(підпис)

Керівник
кваліфікаційної роботи _____ Ігор ПАТРАКЕЄВ
(підпис)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 99 стор., 4 табл., 26 рис., 20 джерел.

Мета роботи – підвищення ефективності виявлення фінансового шахрайства у банківській сфері за допомогою машинного навчання та автоматизації процесів на базі платформи Stronghold.

Об'єкт дослідження – процес виявлення шахрайських транзакцій у банківських системах.

Предмет дослідження – методи машинного навчання для виявлення фінансового шахрайства та їх впровадження у фінансові процеси.

У роботі використано алгоритм Balanced Random Forest з бібліотеки Scikit-learn. Для оцінки якості моделі застосовано такі метрики, як confusion_matrix, ROC-AUC, precision, recall і F1-score. Візуалізація результатів виконана за допомогою графіків важливості ознак (feature importance), теплових карт, графіків часткової залежності (PDP) і SHAP-графіків для пояснення моделі.

Проведено аналіз сучасних методів виявлення шахрайства у банківській сфері. Описано ключові підходи до застосування машинного навчання, зокрема методи Random Forest, Gradient Boosting, логістичної регресії та нейронних мереж. Основна увага зосереджена на розробці та оцінці моделі Balanced Random Forest, яка була навчена на демо-даних з урахуванням важливості балансу між класами (шахрайські та не шахрайські транзакції).

Найкращі ознаки (фічі), визначені під час роботи моделі, і статистичні висновки були адаптовані для використання в системі Stronghold, що дозволяє приймати онлайн-рішення щодо шахрайства в реальному часі.

Результати дослідження підтвердили ефективність використання алгоритму Balanced Random Forest для задач боротьби із шахрайством. Інтеграція моделі в систему Stronghold демонструє можливості автоматизації аналізу транзакцій та підвищення точності виявлення шахрайських операцій.

КЛЮЧОВІ СЛОВА: ШАХРАЙСТВО, БАНКІВСЬКА СФЕРА, BALANCED RANDOM FOREST, ОЦІНКА МЕТРИК, ВАЖЛИВІСТЬ ОЗНАК, ПОЯСНЕННЯ МОДЕЛІ, STRONGHOLD, АВТОМАТИЗАЦІЯ, ГРАФІКИ ЧАСТКОВОЇ ЗАЛЕЖНОСТІ, SHAP.

ABSTRACT

The text part of the master's qualification work: 99 pages, 4 tables, 26 figures, 20 references.

Objective – to improve the efficiency of fraud detection in the banking sector using the Balanced Random Forest method and to transfer the obtained results to the Stronghold system for decision-making automation.

Object of research – the process of detecting fraudulent transactions in banking systems.

Subject of research – the application of the Balanced Random Forest method for fraud detection, interpretation of model results, and integration of key features and statistics into the Stronghold decision-making system.

The work employs the Balanced Random Forest algorithm from the Scikit-learn library. To evaluate the model's performance, metrics such as confusion_matrix, ROC-AUC, precision, recall, and F1-score were used. The results were visualized using feature importance plots, heatmaps, partial dependence plots (PDP), and SHAP graphs for model explanations.

An analysis of modern methods for detecting fraud in the banking sector was conducted. Key approaches to applying machine learning were described, including Random Forest, Gradient Boosting, Logistic Regression, and Neural Networks. The focus is on developing and evaluating the Balanced Random Forest model, trained on demo data with attention to balancing classes (fraudulent and non-fraudulent transactions).

The best features identified during the model's training and statistical insights were adapted for use in the Stronghold system, enabling real-time decision-making for fraud detection.

The research results confirmed the effectiveness of the Balanced Random Forest algorithm for fraud detection tasks. Integrating the model into the Stronghold system

demonstrates the potential for automating transaction analysis and improving the accuracy of fraud detection.

KEYWORDS: FRAUD, BANKING SECTOR, BALANCED RANDOM FOREST, PERFORMANCE METRICS, FEATURE IMPORTANCE, MODEL EXPLANATION, STRONGHOLD, AUTOMATION, PARTIAL DEPENDENCE PLOTS, SHAP.

ЗМІСТ

ВСТУП

Фінансове шахрайство є однією з найважливіших проблем сучасної банківської сфери. Зі збільшенням кількості цифрових транзакцій та розвитком технологій, шахраї використовують все складніші схеми, що становить серйозну загрозу для фінансових установ. Банки стикаються з необхідністю не лише захищати своїх клієнтів, а й забезпечувати збереження активів, мінімізуючи збитки від шахрайських дій. Це вимагає впровадження нових підходів, які здатні ефективно виявляти шахрайство та запобігати фінансовим втратам.

Традиційні методи виявлення шахрайства, такі як ручний аналіз або встановлення базових правил для перевірки транзакцій, стають застарілими. Вони не можуть ефективно обробляти великі обсяги даних і швидко адаптуватися до нових загроз. У таких умовах зростає актуальність використання машинного навчання, яке дозволяє автоматизувати процеси аналізу даних, виявляти приховані закономірності та швидко реагувати на зміни в поведінці шахраїв.

Машинне навчання надає можливість створювати високоточні моделі, які аналізують фінансові операції в режимі реального часу та враховують десятки показників. Ці моделі не лише допомагають виявляти підозрілі дії, але й інтегруються в існуючі бізнес-процеси, забезпечуючи автоматизацію та постійний моніторинг.

Мета даної магістерської роботи – підвищення ефективності виявлення фінансового шахрайства за допомогою алгоритму Balanced Random Forest та інтеграції результатів у платформу Stronghold, що є системою для запобігання шахрайству.

Об'єктом дослідження є процес виявлення шахрайських транзакцій у банківських системах.

Предмет дослідження – використання алгоритму Balanced Random Forest для аналізу фінансових транзакцій та його впровадження у платформу Stronghold.

Для досягнення мети було поставлено такі завдання:

1. Провести аналіз природи фінансового шахрайства, його видів і масштабів.
2. Описати існуючі методи та підходи для боротьби з шахрайством, зокрема із застосуванням машинного навчання.
3. Розглянути популярні алгоритми машинного навчання, такі як дерева рішень, Random Forest та методи бустингу.
4. Розробити модель Balanced Random Forest для виявлення шахрайських транзакцій.
5. Провести аналіз отриманих результатів: оцінка точності, метрик і якості моделі.
6. Інтегрувати модель у платформу Stronghold для автоматизації процесу перевірки транзакцій.
7. Виконати експериментальну оцінку ефективності інтегрованого рішення.

Актуальність роботи обумовлена постійним зростанням кількості шахрайських дій у фінансовій сфері. Це ставить перед банківськими установами складне завдання – знаходити нові інструменти та рішення для захисту клієнтів і запобігання втратам. Використання алгоритмів машинного навчання, таких як Balanced Random Forest, дозволяє забезпечити виявлення шахрайства на більш високому рівні точності та ефективності, що є важливим для стабільності фінансових систем у сучасному цифровому середовищі.

1 ТЕОРЕТИЧНІ ОСНОВИ ВИЯВЛЕННЯ ФІНАНСОВОГО ШАХРАЙСТВА

1.1 Поняття фінансового шахрайства: види, причини та наслідки

Фінансове шахрайство є одним із найбільших викликів для сучасних фінансових установ, оскільки масштаби цього явища продовжують зростати разом із розвитком цифрових технологій. У найзагальнішому сенсі фінансове шахрайство можна визначити як свідомі протиправні дії, спрямовані на отримання фінансової вигоди через обман або зловживання довірою. Такі дії не тільки завдають фінансових збитків, але й підривають репутацію фінансових установ, створюючи ризики для їхньої стійкості.

Розвиток цифрової економіки значно розширив можливості для зловмисників. Вони використовують технології для збирання конфіденційних даних, підробки транзакцій або зламу систем безпеки. Наприклад, один із найбільш поширених сценаріїв шахрайства – це махінації з банківськими картками, коли шахраї отримують доступ до карткових даних через фішингові атаки або скімінгові пристрої. Іншим прикладом є створення фіктивних кредитних заявок або фальшивих онлайн-магазинів, які обманюють покупців.

Фінансове шахрайство можна поділити на кілька категорій залежно від методів і цілей:

1. Махінації з банківськими картками:

Це найпоширеніший вид шахрайства, що включає крадіжку даних карток через скімінг, фішинг або злом баз даних банків. Наприклад, у 2023 році загальні втрати від шахрайства з банківськими картками сягнули понад $30 мільярдів (рис. 1.1).

Рис. 1.1. Масштаби фінансового шахрайства у 2023 році (глобальна статистика втрат)

2. Шахрайство у сфері кредитування:

Шахраї можуть подавати фальшиві документи для отримання позик або використовувати викрадені дані для створення кредитних заявок. Такі дії не лише завдають збитків банкам, а й створюють труднощі для справжніх власників викрадених даних.

3. Страхове шахрайство:

Поширеним прикладом є подання фальшивих заявок на страхові виплати. Це може включати вигадані аварії, хибні діагнози або інші вигадані страхові події.

4. Цифрове шахрайство у сфері електронної комерції:

Зловмисники створюють фіктивні онлайн-магазини або використовують крадені карткові дані для покупок. Від таких дій страждають як покупці, так і справжні власники платіжних карток.

5. Фішинг та соціальна інженерія:

Шахраї використовують методи обману для того, щоб отримати доступ до конфіденційної інформації користувачів. Це можуть бути електронні листи, які імітують справжні повідомлення банків, або телефонні дзвінки (т.зв. "вішинг").

Причинами такого значного поширення шахрайства є:

- Цифровізація фінансових процесів: Швидкий розвиток цифрових технологій відкриває нові вразливості, які використовуються зловмисниками.
- Низький рівень фінансової грамотності: Клієнти часто не знають, як захистити свої дані, і можуть легко потрапити на шахрайські схеми.
- Недостатній рівень безпеки: Деякі фінансові установи не встигають оновлювати свої системи безпеки, що робить їх уразливими до атак.

Фінансове шахрайство має серйозні наслідки як для окремих осіб, так і для організацій. Серед основних наслідків:

1. Фінансові втрати:

   Для банків це збитки, які можуть досягати мільярдів доларів на глобальному рівні. Наприклад, лише у 2023 році загальні втрати через шахрайство з банківськими картками сягнули $30 млрд. Для клієнтів це означає втрату коштів, які вони можуть не повернути.

2. Репутаційні ризики:

   Клієнти втрачають довіру до банків або компаній, якщо вони не можуть гарантувати безпеку їхніх активів. Це може призводити до втрати клієнтів і скорочення доходів.

3. Юридичні наслідки:

   Розслідування випадків шахрайства вимагає великих витрат часу і ресурсів.

4. Соціальні наслідки:

   Шахрайство підриває загальну довіру до фінансових систем, що може уповільнювати економічний розвиток.

   Виявлення шахрайства базується на трьох ключових етапах:

   - Попередження: Впровадження інструментів для аналізу підозрілих транзакцій ще до їхнього завершення.
   - Виявлення: Використання алгоритмів машинного навчання для аналізу великих обсягів даних у реальному часі.
   - Мітрагація (зменшення збитків): Розробка механізмів, які дозволяють швидко реагувати на виявлене шахрайство.

На рисунку 1.2 показано основні інструменти, які використовуються для боротьби із шахрайством.

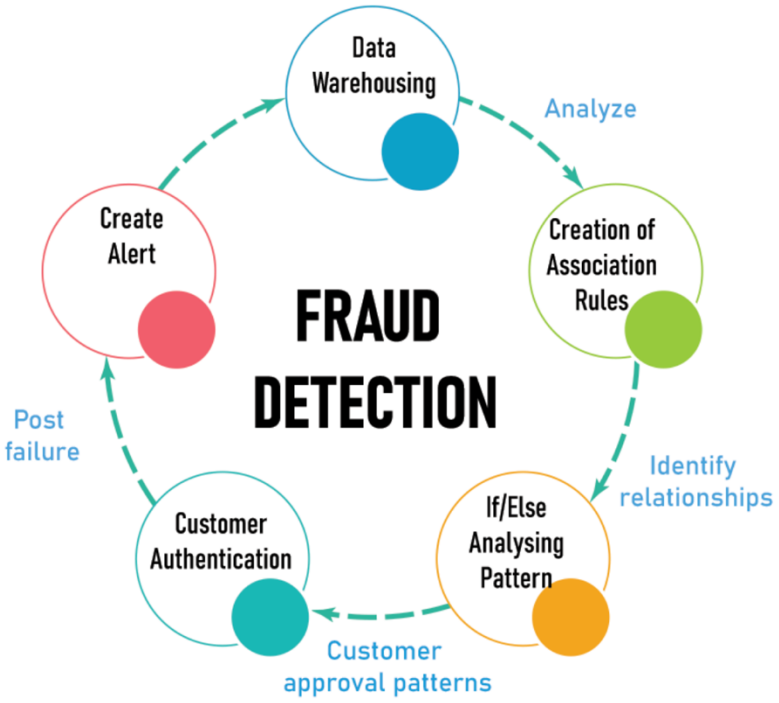


Рис. 1.2. Інструменти для виявлення та протидії шахрайству

Фінансове шахрайство – це багатогранна проблема, яка потребує сучасних підходів для її вирішення. Традиційні методи дедалі більше поступаються місцем автоматизованим системам, які базуються на використанні алгоритмів

машинного навчання. Глибоке розуміння природи шахрайства, його причин і наслідків є основою для розробки ефективних рішень, які можуть забезпечити безпеку фінансових установ та їхніх клієнтів.

1.2 Методи боротьби з шахрайством у банківській сфері

У сучасному світі фінансова безпека є одним із пріоритетів банківських установ. Оскільки шахрайські дії стають дедалі складнішими, боротьба з ними вимагає комплексного підходу. Методи протидії шахрайству в банківській сфері можна умовно поділити на традиційні та сучасні. Хоча традиційні підходи все ще застосовуються, сучасні методи, засновані на використанні новітніх технологій, стають більш ефективними, особливо у масштабах великих даних.

До традиційних методів належать ручний аналіз транзакцій, встановлення фіксованих правил для перевірки платежів, а також багаторівневий процес аутентифікації клієнтів. Наприклад, у багатьох банках застосовується підхід, коли транзакції, які перевищують певну суму, автоматично позначаються як підозрілі. Це дозволяє співробітникам банку вручну перевіряти ці операції.

Однак ручний аналіз має кілька істотних обмежень:

- Трудомісткість: Аналіз великих обсягів транзакцій вимагає значних людських ресурсів.
- Часовий фактор: Ручна перевірка уповільнює процес обробки платежів.
- Обмежена ефективність: Шахраї швидко адаптуються до фіксованих правил, що робить такі системи менш дієвими.

Традиційні методи залишаються важливими для базового рівня захисту, однак вони стають недостатньо ефективними у швидкозмінному цифровому середовищі.

Сучасні підходи орієнтовані на автоматизацію процесів, аналіз великих обсягів даних у реальному часі та адаптивність до нових типів загроз. У цьому

контексті важливу роль відіграють технології машинного навчання, штучного інтелекту, блокчейн та біометрії.

1. Машинне навчання (Machine Learning):

Це один із найефективніших методів для виявлення шахрайства. Алгоритми машинного навчання дозволяють аналізувати великі обсяги даних, виявляти приховані закономірності та визначати нетипові дії. Наприклад, моделі класифікації можуть навчитися розрізняти шахрайські та легітимні транзакції на основі історичних даних (рис. 1.4). Сучасні системи, засновані на алгоритмах Random Forest, Gradient Boosting або нейронних мережах, здатні адаптуватися до нових сценаріїв шахрайства.

2. Штучний інтелект (Artificial Intelligence):

Інтелектуальні системи аналізують поведінкові дані користувачів, такі як частота транзакцій, час їх виконання та геолокація. Наприклад, якщо клієнт зазвичай виконує операції з одного пристрою в певному регіоні, а потім раптом здійснює транзакцію з іншої країни, система може автоматично заблокувати цю операцію до додаткової перевірки.

3. Блокчейн:

Технологія блокчейн забезпечує високий рівень прозорості та захисту транзакцій. Завдяки децентралізованій природі цієї технології шахраї не можуть легко змінити або видалити записані дані. Банки вже починають впроваджувати блокчейн у свої процеси, зокрема для перевірки достовірності фінансових документів.

4. Біометрична аутентифікація:

Унікальні фізіологічні характеристики людини, такі як відбитки пальців, розпізнавання обличчя або сканування райдужки ока, застосовуються для аутентифікації клієнтів. Цей підхід суттєво ускладнює доступ шахраїв до облікових записів.

5. Системи реального часу (Real-Time Fraud Detection):

Автоматизовані системи моніторингу працюють у режимі реального часу, аналізуючи кожну транзакцію на предмет підозрілої активності. Наприклад,

платформи, такі як Stronghold, можуть використовувати статистичні моделі для оцінки ризику кожної операції за кілька секунд. Це забезпечує швидке блокування шахрайських транзакцій ще до їхнього завершення.

- Сучасні фінансові установи використовують комбінацію інструментів для забезпечення безпеки. Серед них виділяються:
- Системи оцінки ризиків (Risk Scoring): Кожній транзакції присвоюється певний ризиковий бал на основі аналізу історичних даних та поведінкових моделей.
- Інструменти для виявлення аномалій: Вони дозволяють ідентифікувати транзакції, які відрізняються від звичайного поведінкового патерну клієнта.
- Роботизовані процеси (RPA): Автоматизація рутинних завдань, пов'язаних із перевіркою транзакцій, дозволяє зменшити навантаження на персонал банку.
- Системи біометричної автентифікації: Використання таких технологій, як розпізнавання обличчя, відбитків пальців або голосу, для перевірки особи клієнта під час здійснення транзакцій.
- Штучний інтелект та машинне навчання: Аналіз великих обсягів даних у реальному часі для виявлення прихованих шаблонів шахрайства, які неможливо ідентифікувати традиційними методами.
- Моніторинг на основі геолокації: Відстеження місця здійснення транзакції та порівняння його з історичними даними, щоб виявляти потенційно підозрілі операції.

На рисунку 1.3 показано схему роботи сучасної системи виявлення шахрайства з використанням машинного навчання.

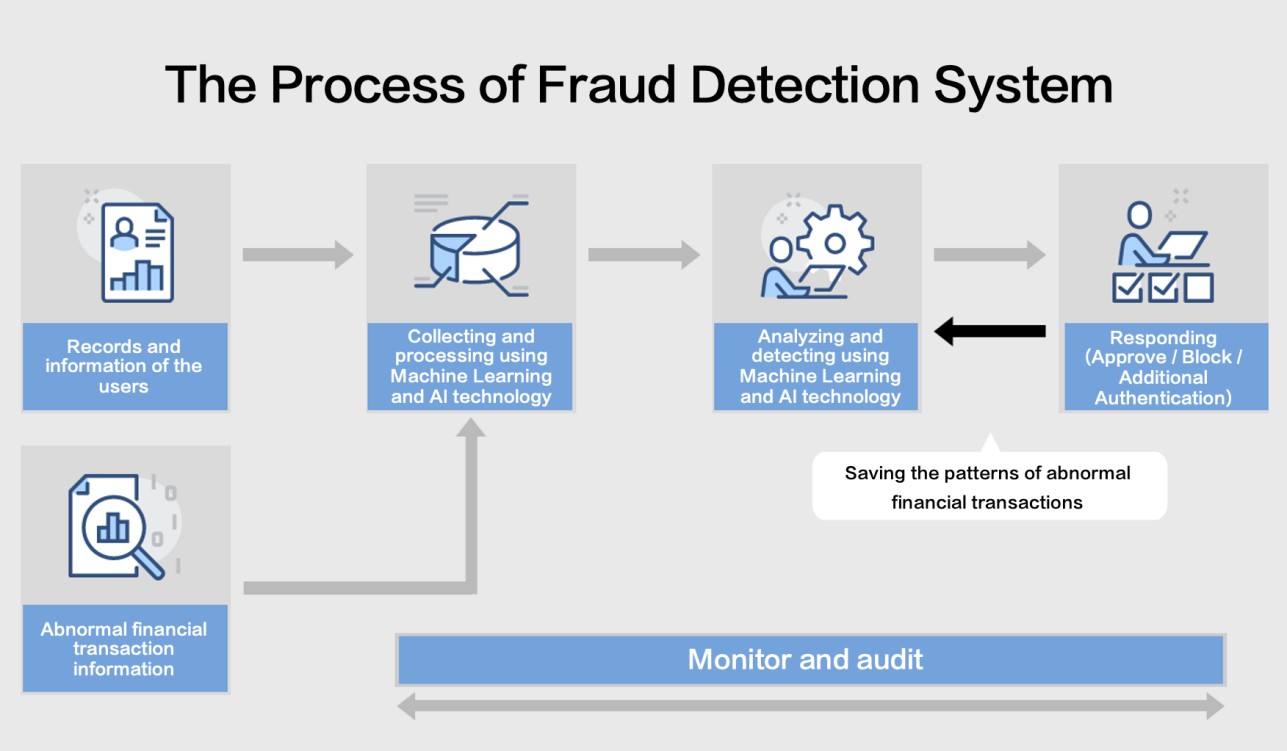


Рис. 1.3. Схема роботи системи виявлення шахрайства на основі машинного навчання.

Попри численні переваги сучасних методів, вони також мають свої виклики. По-перше, для навчання алгоритмів машинного навчання необхідна велика кількість якісних даних, що не завжди доступно. По-друге, шахраї постійно вдосконалюють свої техніки, створюючи нові виклики для систем безпеки. Тому важливим є не тільки впровадження новітніх технологій, але й постійна адаптація до нових загроз.

1.3 Використання алгоритмів машинного навчання для виявлення шахрайства

Машинне навчання (ML) є одним із найефективніших підходів для виявлення фінансового шахрайства в сучасному цифровому середовищі. Завдяки здатності аналізувати великі обсяги даних, виявляти приховані закономірності та швидко адаптуватися до нових типів загроз, алгоритми машинного навчання

займають провідне місце серед інструментів для боротьби з шахрайськими діями. Використання ML дозволяє автоматизувати процеси аналізу транзакцій та скоротити час реакції на потенційно небезпечні операції.

Алгоритми машинного навчання здатні будувати прогнози на основі аналізу даних минулих транзакцій, розпізнаючи патерни шахрайської та легітимної поведінки. Ключовими етапами процесу застосування ML для виявлення шахрайства є:

1. Збір та підготовка даних

Машинне навчання працює лише з якісними та релевантними даними. У випадку з фінансовим шахрайством це можуть бути історичні транзакції, дані про клієнтів, час виконання операцій, геолокація та навіть поведінкові фактори. Для забезпечення високої точності моделі важливо, щоб дані були очищені, нормалізовані та структуровані.

2. Навчання моделі

Алгоритм навчається на історичних даних, де кожна транзакція маркована як шахрайська або легітимна. Це дозволяє моделі ідентифікувати закономірності, які відрізняють нормальні дії від шахрайських.

3. Тестування та валідація

Перед впровадженням модель тестується на нових даних для оцінки її точності, стабільності та здатності до узагальнення. Основними метриками для оцінки є precision, recall, ROC-AUC та F1-score.

4. Впровадження в реальну систему

Модель інтегрується у фінансову платформу (наприклад, Stronghold), де в режимі реального часу аналізує транзакції та визначає ймовірність шахрайства.

На рисунку 1.4 зображено процес впровадження моделі машинного навчання для виявлення шахрайства.

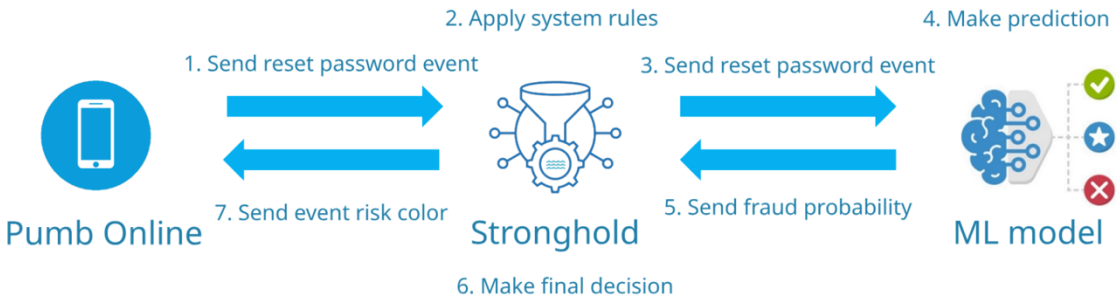


Рис. 1.4. Етапи впровадження машинного навчання для виявлення шахрайства.

У практиці боротьби з шахрайством використовуються як прості, так і складніші моделі. Нижче наведено найпопулярніші алгоритми:

1. Дерева рішень

Дерева рішень є базовим методом машинного навчання. Вони працюють, будуючи ієрархічні правила для класифікації даних. Наприклад, транзакція може бути класифікована як шахрайська, якщо вона виконується з незвичної геолокації у незвичний час.

Переваги дерев рішень:

- Простота у розумінні та реалізації.
- Візуалізація процесу прийняття рішень.

Недоліки:

- Низька стійкість до шуму в даних.
- Схильність до перенавчання, якщо модель є занадто складною.

2. Random Forest

Random Forest — це ансамблевий метод, який використовує кілька дерев рішень для створення більш стабільної та точної моделі. Кожне дерево голосує за класифікацію, і кінцеве рішення приймається на основі більшості голосів.

Random Forest є ефективним у задачах виявлення шахрайства завдяки таким особливостям:

- Висока точність: Алгоритм добре працює з великими обсягами даних, забезпечуючи надійні результати навіть за наявності складних залежностей.
- Можливість обробляти незбалансовані набори даних: Використання таких варіантів, як Balanced Random Forest, дозволяє вирівнювати кількість шахрайських і легітимних транзакцій у навчальному наборі, що зменшує ризик упередженості моделі.
- Стійкість до шуму: Завдяки своїй структурі Random Forest зменшує вплив нерелевантних або шумових даних, що підвищує точність моделі.
- Інтерпретація результатів: Алгоритм дозволяє оцінювати важливість кожної змінної, що спрощує аналіз ключових факторів, пов’язаних із шахрайством.
- Масштабованість: Random Forest легко адаптується до великих обсягів даних і складних структур, що робить його придатним для використання в реальних банківських системах.

На рисунку 1.5 показано, як Random Forest аналізує дані для виявлення шахрайства.

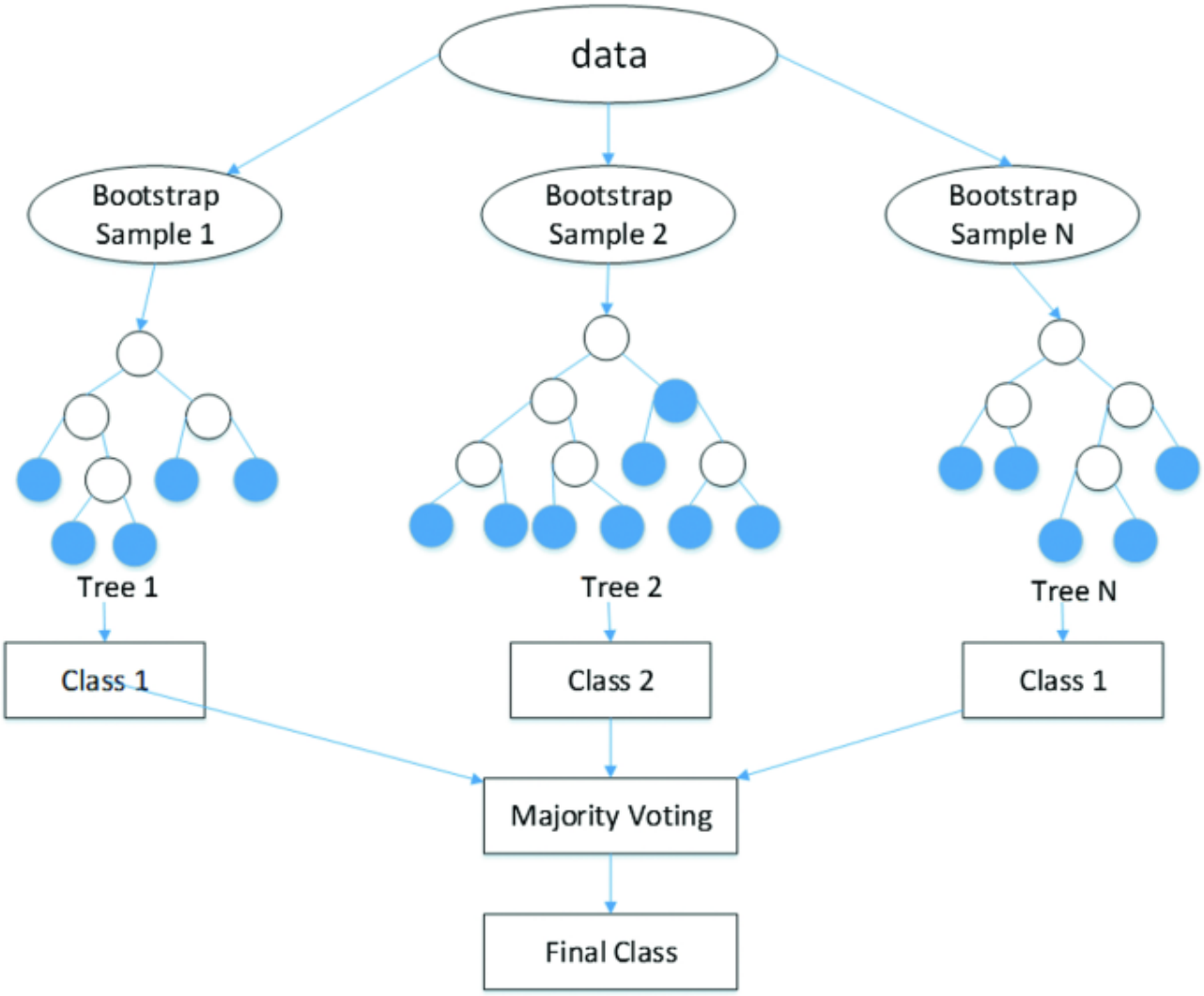


Рис. 1.5. Робота алгоритму Random Forest у задачі класифікації транзакцій.

3. Gradient Boosting (XGBoost, LightGBM, CatBoost)

Boosting — це метод, що створює послідовність моделей, де кожна наступна модель покращує попередню, зосереджуючись на помилках. Ці алгоритми, як-от XGBoost, є одними з найефективніших для виявлення шахрайства завдяки їхній високій точності та гнучкості.

Переваги:

- Висока точність і адаптивність.

- Можливість обробляти великі набори даних.

Недоліки:

- Висока обчислювальна складність.
- Необхідність ретельного налаштування гіперпараметрів.

3. Gradient Boosting (XGBoost, LightGBM, CatBoost)

Boosting — це метод, що створює послідовність моделей, де кожна наступна модель покращує попередню, зосереджуючись на помилках. Ці алгоритми, як-от XGBoost, є одними з найефективніших для виявлення шахрайства завдяки їхній високій точності та гнучкості.

Переваги:

- Висока точність і адаптивність.
- Можливість обробляти великі набори даних.

Недоліки:

- Висока обчислювальна складність.
- Необхідність ретельного налаштування гіперпараметрів.

4. Нейронні мережі

Нейронні мережі здатні аналізувати складні взаємозв’язки між даними, що робить їх перспективними для виявлення шахрайства. Наприклад, рекурентні нейронні мережі (RNN) можуть враховувати послідовність транзакцій, а згорткові нейронні мережі (CNN) використовуються для аналізу графів зв’язків між користувачами.

Головною перевагою ML-алгоритмів є їхня здатність аналізувати аномалії у великих обсягах даних. Наприклад, якщо користувач виконує транзакції з різних країн протягом короткого часу, модель може позначити ці дії як підозрілі.

Інша сильна сторона — це адаптивність. Алгоритми машинного навчання можуть автоматично "навчатися" нових сценаріїв шахрайства, що робить їх більш гнучкими порівняно з традиційними методами.

Попри численні переваги, використання машинного навчання для виявлення шахрайства стикається з певними викликами:

1. Незбалансованість даних: У реальних умовах кількість шахрайських транзакцій набагато менша, ніж легітимних. Це може вплинути на точність моделі.
2. Якість даних: Помилки, відсутні значення та шум у даних можуть значно погіршити результати моделі.
3. Обчислювальні ресурси: Складні алгоритми, такі як Gradient Boosting або нейронні мережі, вимагають значних обчислювальних ресурсів.

Машинне навчання є ключовим інструментом у боротьбі з шахрайством. Алгоритми, такі як Random Forest, Gradient Boosting та нейронні мережі, забезпечують високу точність, швидкість і масштабованість у аналізі транзакцій. Вони дозволяють фінансовим установам не тільки виявляти шахрайство, але й адаптуватися до нових загроз у реальному часі.

2 ПЛАТФОРМА STRONGHOLD: ФУНКЦІЇ ТА ЗАСТОСУВАННЯ У БАНКІВСЬКИХ ПРОЦЕСАХ

2.1 Основні функції та можливості платформи Stronghold

Stronghold — це сучасна система виявлення та запобігання шахрайським операціям, розроблена для забезпечення безпеки банківських транзакцій у режимі реального часу. Платформа використовує гнучкий підхід до аналізу транзакцій, включаючи правила ризику, статистичні моделі та алгоритми машинного навчання.

Мета Stronghold — зменшення фінансових ризиків, пов'язаних із шахрайством, завдяки автоматизації процесів моніторингу та прийняття рішень, що допомагає банківським установам забезпечувати довіру клієнтів та мінімізувати збитки.

Основні функціональні можливості Stronghold:

1. Реальний час аналізу транзакцій

Платформа дозволяє обробляти вхідні транзакції у режимі реального часу, використовуючи складні алгоритми перевірки. Система миттєво аналізує операції, визначаючи їхній ризик на основі:

- історії транзакцій клієнта,
- геолокаційних даних,
- шаблонів поведінки користувачів.

Цей підхід дозволяє блокувати потенційно шахрайські транзакції до їх завершення.

2. Гнучка система ризикових правил

Stronghold забезпечує можливість створення складних правил ризику з використанням:

- логічних операторів (AND, OR),
- математичних функцій,
- статистичних даних, таких як сума операцій за певний період або кількість транзакцій за день.

Ці правила дозволяють банкам адаптувати систему до власних потреб і типів загроз.

3. Скорингові моделі

Система автоматично присвоює транзакціям ризиковий бал (score) на основі параметрів операції. Наприклад:

- операції з незвичних місць,
- великі суми переказів,
- незвична частота транзакцій.

Транзакції з високим ризиковим балом або блокуються, або позначаються для подальшого аналізу.

4. Статистичний аналіз

Stronghold включає модуль збору статистичних даних, який дозволяє:

- накопичувати інформацію про транзакції клієнтів,
- використовувати статистику для модифікації правил ризику,
- створювати профілі поведінки клієнтів.

Stronghold підтримує інтеграцію з банківськими системами, такими як:

- інтернет-банкінг,
- мобільний банкінг,
- CRM-системи.

Завдяки цьому забезпечується повна автоматизація процесу обробки транзакцій.

Система має розвинений модуль управління подіями, який дозволяє:

- відслідковувати підозрілі дії клієнтів,
- створювати події для автоматичного виконання певних дій, таких як відправлення повідомлень або блокування акаунтів.

Stronghold надає API для інтеграції з зовнішніми сервісами та розширення функціональності. Це включає:

- Web API (REST/JSON),
- підтримку мікросервісної архітектури.

Додаткові функції:

1. Моніторинг у реальному часі

   Stronghold дозволяє відслідковувати активність системи в реальному часі, зокрема:

   - кількість транзакцій за секунду,
   - статус транзакцій (схвалено, відхилено, підозріле),
   - технічні характеристики сервера (використання пам’яті та завантаження CPU).

2. Інструменти звітності

   Система генерує звіти про виявлені шахрайські дії, включаючи:

   - кількість шахрайських транзакцій,
   - ефективність правил ризику,
   - порівняння статистичних даних.

3. Безпека та аудит

   Stronghold відповідає вимогам PCI DSS і підтримує:

   - контроль доступу користувачів,
   - аудит дій користувачів,

- шифрування даних.

Stronghold є потужним інструментом для виявлення шахрайства, який поєднує сучасні алгоритми, гнучкі налаштування та високу продуктивність.

2.2 Інтеграція аналітики даних у бізнес-логіку Stronghold

Інтеграція аналітики даних у бізнес-процеси є одним із ключових етапів роботи системи Stronghold. Ця платформа забезпечує безперервний зв’язок між великими обсягами даних, їх аналізом та прийняттям рішень у реальному часі. Завдяки використанню гнучких алгоритмів і аналітичних інструментів, Stronghold перетворює "сирі" дані в конкретні дії, такі як блокування шахрайських транзакцій, створення звітів для аналітиків або зміна параметрів ризикових правил.

Основні принципи інтеграції аналітики в Stronghold:

1. Статистичний аналіз для прогнозування ризиків

Stronghold використовує вбудовані модулі статистичного аналізу для накопичення, зберігання та обробки історичних даних. Це дозволяє системі:

- визначати шаблони поведінки клієнтів,
- оцінювати аномалії у транзакціях,
- прогнозувати можливі ризики на основі минулих подій.

2. Система правил ризику (Rule Engine)

Аналітичні дані інтегруються у Rule Engine для перевірки кожної транзакції на відповідність попередньо налаштованим правилам. Наприклад:

- Якщо клієнт виконує транзакцію з IP-адреси, що не відповідає його типовій геолокації, це правило може відразу позначити операцію як підозрілу.
- Дані про попередні шахрайські дії додаються до "чорних списків", що забезпечує автоматичне блокування повторних порушень.

3. Моделі машинного навчання для адаптації

Інтеграція алгоритмів машинного навчання дозволяє Stronghold використовувати сучасні методи прогнозування ризиків. Алгоритми навчаються на історичних даних, забезпечуючи динамічне оновлення моделей без необхідності ручного втручання. Це дає змогу:

- оцінювати ризик у реальному часі,
- адаптувати систему до нових типів загроз,
- підвищувати точність і швидкість прийняття рішень.

4. Інтеграція з бізнес-процесами банків

Stronghold легко інтегрується з різними банківськими системами, такими як:

- CRM (управління відносинами з клієнтами),
- процесингові системи карткових транзакцій,
- мобільний та інтернет-банкінг.

Це дозволяє збирати дані з усіх каналів і забезпечує наскрізний контроль за шахрайськими операціями.

Ключові інструменти інтеграції аналітики

1. Робота з великими обсягами даних

Stronghold підтримує роботу з мільйонами транзакцій щомісяця, використовуючи такі механізми:

- Сегментація даних: розподіл транзакцій за клієнтами, геолокацією або типами операцій.
- Групування аномалій: створення кластерів для спрощення аналізу підозрілих дій.

2. Моніторинг у реальному часі

Система надає можливість:

- переглядати поточний стан транзакцій,
- відстежувати ключові показники, такі як кількість підозрілих операцій або частота шахрайських дій у певному регіоні.

3. Автоматизація реагування

На основі аналітичних даних система може:

- автоматично блокувати підозрілі транзакції,
- генерувати повідомлення для аналітиків,
- запускати розслідування інцидентів.

Переваги інтеграції аналітики даних у Stronghold:

1. Зменшення часу прийняття рішень

Аналітичні дані доступні миттєво, що дозволяє швидко реагувати на загрози.

2. Адаптивність до нових загроз

Інтеграція машинного навчання та Rule Engine дозволяє системі еволюціонувати разом із шахрайськими схемами.

3. Економія ресурсів

Автоматизація процесів знижує навантаження на аналітиків, дозволяючи їм зосередитися на складних кейсах.

Інтеграція аналітики даних у бізнес-логіку Stronghold дозволяє банкам ефективно боротися з шахрайством, оптимізуючи процеси управління ризиками та зменшуючи фінансові збитки. Завдяки гнучкості системи та підтримці сучасних технологій, Stronghold стає незамінним інструментом у забезпеченні безпеки фінансових операцій.

2.3 Використання платформи для автоматизації процесів перевірки транзакцій

Автоматизація перевірки транзакцій є важливим етапом у забезпеченні безпеки фінансових операцій. Платформа Stronghold дозволяє банківським установам мінімізувати вплив людського фактора, прискорити процес обробки даних і покращити точність виявлення шахрайських транзакцій. Завдяки використанню передових алгоритмів і інтеграції з банківськими системами, платформа забезпечує автоматизований моніторинг кожної операції в реальному часі.

Основні етапи автоматизації перевірки транзакцій у Stronghold

1. Збір даних про транзакції

Першим кроком є автоматизоване отримання даних про транзакції з різних джерел. Stronghold інтегрується з такими системами, як:

- процесингові центри,
- CRM,
- мобільний банкінг,
- інтернет-банкінг.

Це забезпечує повну картину кожної фінансової операції, включаючи інформацію про час, місце, суму транзакції, пристрій клієнта та його геолокацію.

2. Попередня обробка даних

Зібрані дані автоматично обробляються для підготовки до аналізу. На цьому етапі система:

- нормалізує дані,
- видаляє дублікатні записи,
- перевіряє коректність значень, наприклад, формату номерів карток чи IP-адрес.

3. Аналіз транзакцій у реальному часі

Кожна транзакція проходить через кілька рівнів перевірки:

- Аналіз правил (Rule Engine):
  Система перевіряє операцію за попередньо визначеними правилами ризику. Наприклад, якщо транзакція відбувається з незвичного місця або в нічний час, вона отримує підвищений ризиковий бал.
- Моделі машинного навчання:
  Stronghold застосовує попередньо навчені алгоритми, такі як Balanced Random Forest або Gradient Boosting, для оцінки ймовірності шахрайства. Ці алгоритми враховують десятки факторів, включаючи історичні дані про клієнта та транзакції.
- Порівняння з чорними списками:
  Система автоматично звіряє транзакцію з базами даних шахрайських рахунків, IP-адрес, пристроїв або номерів карток.

4. Прийняття рішень

Після аналізу система автоматично приймає одне з рішень:

- Схвалити транзакцію: Якщо транзакція не містить ознак шахрайства.
- Відхилити транзакцію: Якщо ризиковий бал перевищує допустимий поріг.

- Передати на ручну перевірку: Якщо є певні підозри, але недостатньо доказів для автоматичного блокування.

Переваги автоматизації перевірки транзакцій у Stronghold:

1. Швидкість і точність

Автоматизована перевірка у Stronghold дозволяє обробляти тисячі транзакцій за секунду. Завдяки цьому клієнти не відчувають затримок під час виконання операцій, тоді як підозрілі транзакції ефективно фільтруються.

2. Зниження навантаження на персонал

Система автоматично блокує більшість шахрайських операцій, що дозволяє аналітикам банку зосереджуватися лише на складних або незвичних випадках.

3. Мінімізація людського фактора

Завдяки автоматизації виключаються помилки, пов’язані з суб’єктивністю або втомою працівників.

4. Адаптація до змін

Система оновлює моделі ризиків і алгоритми машинного навчання, враховуючи нові типи шахрайства. Це забезпечує актуальність і гнучкість у реагуванні на загрози.

Інструменти, що забезпечують автоматизацію:

1. Гнучка система правил (Rule Engine)

Rule Engine дозволяє банкам створювати власні правила для автоматизації перевірки транзакцій. Наприклад:

- Блокування всіх транзакцій понад певну суму з нових пристроїв.
- Аналіз транзакцій у нестандартний час для клієнта.

2. Інтеграція з базами даних

Платформа інтегрується з різними джерелами даних, такими як:

- національні бази шахрайських рахунків,
- міжнародні бази даних ризикових IP-адрес.

3. Моделі машинного навчання

Застосування моделей ML дозволяє виявляти навіть ті випадки шахрайства, які не підпадають під жодне правило. Наприклад, аномальна частота транзакцій або нова модель поведінки клієнта можуть бути автоматично ідентифіковані.

4. Модуль звітності та моніторингу

Автоматизована система Stronghold надає детальні звіти про всі підозрілі операції. Це дозволяє аналітикам оцінювати ефективність правил і моделей та вносити необхідні корективи.

Stronghold є потужним інструментом для автоматизації перевірки транзакцій. Його функціонал дозволяє банкам знижувати фінансові втрати, підвищувати ефективність процесів безпеки та забезпечувати зручність для клієнтів. Завдяки гнучким алгоритмам і сучасним технологіям Stronghold забезпечує надійний захист від шахрайства у режимі реального часу.

3 ОГЛЯД МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ ШАХРАЙСТВА

3.1 Популярні алгоритми машинного навчання: теорія та приклади

Машинне навчання (ML) є важливим інструментом для виявлення шахрайства у фінансовій сфері. Завдяки можливості автоматичного аналізу великих обсягів даних і виявлення прихованих закономірностей, алгоритми машинного навчання суттєво змінюють підхід до безпеки банківських операцій. У цьому розділі розглянуто популярні алгоритми, що застосовуються для виявлення шахрайства, їх принципи роботи, ключові особливості, а також переваги й обмеження.

Розуміння принципів роботи цих алгоритмів є необхідним для ефективного впровадження автоматизованих систем виявлення шахрайства, таких як Stronghold.

3.1.1 Дерева рішень: переваги та недоліки

Дерева рішень — це один із найпростіших і найзрозуміліших алгоритмів машинного навчання, який використовується для класифікації та регресії. Завдяки своїй прозорості й інтуїтивній структурі дерева рішень знаходять широке застосування у фінансовій сфері, зокрема у виявленні шахрайських транзакцій. Цей метод працює за принципом послідовного розподілу даних на основі умов, що перевіряються на кожному вузлі дерева. Кінцевий результат визначається листами дерева, які представляють класифікацію (наприклад, "шахрайська" чи "легітимна" транзакція).

Дерево рішень будується шляхом розбиття даних на основі найбільш інформативних ознак (features). Кожне розгалуження дерева відповідає за перевірку певної умови. Наприклад:

- Якщо сума транзакції перевищує $10,000?
- Так → далі перевіряється, чи операція виконана вночі.
- Ні → транзакція вважається "легітимною".

Цей процес повторюється доти, поки дані не будуть класифіковані з максимальною точністю. Мета алгоритму — зменшити невизначеність і підвищити якість класифікації, використовуючи метрики, такі як ентропія або критерій Джині. На рисунку 3.1 представлено приклад простого дерева рішень для класифікації транзакцій.

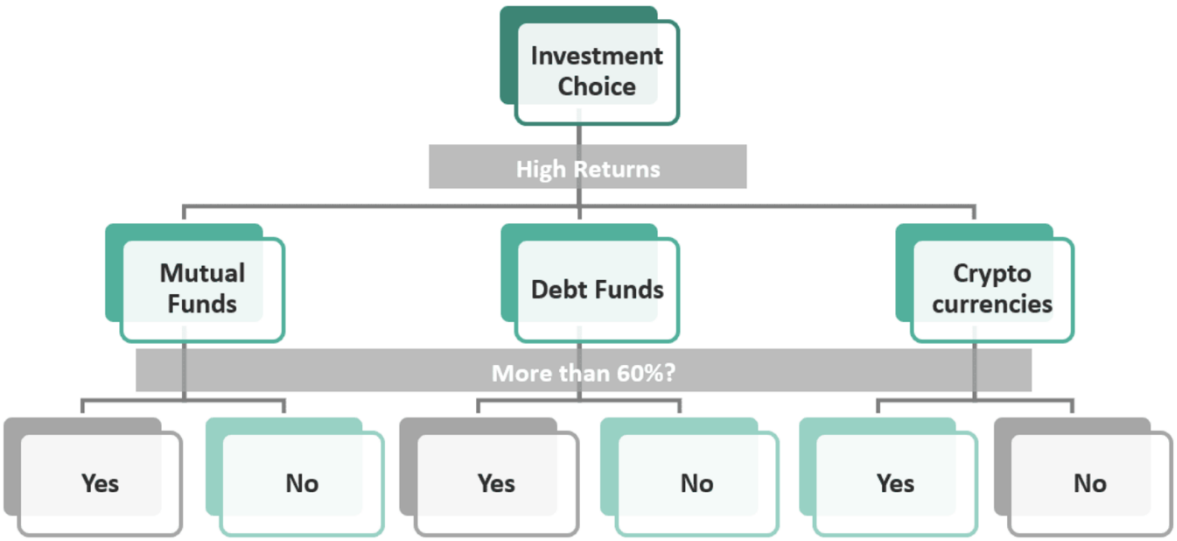


Рис. 3.1. Принцип роботи дерева рішень для класифікації транзакцій

У фінансовій сфері дерева рішень широко використовуються для аналізу транзакцій. Наприклад, для оцінки шахрайства можуть враховуватися такі ознаки:

1. Геолокація клієнта (звідки виконана операція).
2. Час проведення транзакції (день/ніч).
3. Тип пристрою (новий чи звичний).
4. Сума операції (вище або нижче середньої для клієнта).

Якщо транзакція не відповідає звичайній поведінці клієнта (наприклад, здійснюється вночі з іншої країни), дерево може класифікувати її як "підозрілу". Перевагою цього підходу є його прозорість: кожне рішення може бути легко пояснено.

Дерева рішень мають кілька важливих переваг, які роблять їх ефективними у задачах класифікації:

1. Інтуїтивна зрозумілість:

Модель легко візуалізувати, і навіть неексперт може зрозуміти логіку її роботи. Це особливо важливо у фінансовій сфері, де прозорість алгоритму має критичне значення для довіри клієнтів.

2. Швидкість навчання:

Дерева рішень швидко навчаються навіть на великих наборах даних, що робить їх ідеальними для первинного аналізу.

3. Робота з різними типами даних:

Алгоритм може працювати як із числовими, так і з категоріальними ознаками. Наприклад, можна одночасно враховувати суму транзакції та тип пристрою клієнта.

4. Відсутність потреби у масштабуванні даних:

На відміну від інших алгоритмів (наприклад, логістичної регресії), дерева рішень не вимагають нормалізації чи стандартизації даних.

Незважаючи на численні переваги, дерева рішень мають і свої недоліки:

1. Схильність до перенавчання:

Якщо дерево занадто глибоке, воно може запам'ятовувати особливості навчальних даних, але працювати неточно на нових прикладах. Це призводить до низької узагальнювальної здатності моделі.

2. Чутливість до змін у даних:

Невеликі зміни у вхідних даних можуть суттєво змінити структуру дерева, що впливає на стабільність моделі.

3. Менша точність у складних завданнях:

Для великих і складних наборів даних дерева рішень часто поступаються більш просунутим методам, таким як Random Forest чи Gradient Boosting.

4. Відсутність автоматичного визначення важливості ознак:

Хоча дерева рішень і використовують критерій вибору вузлів, вони не надають явного ранжування ознак, що може бути корисним для інтерпретації результатів.

Дерева рішень часто використовуються як базовий метод для порівняння з іншими алгоритмами. У таблиці 3.1 наведено порівняння дерев рішень із більш складними алгоритмами, такими як Random Forest і Gradient Boosting.

Таблиця 3.1

Порівняння дерев рішень з іншими алгоритмами

Алгоритм	Простота	Швидкість навчання	Точність	Узагальнення	Стійкість до шуму
Дерева рішень	Висока	Висока	Середня	Низька	Низька
Random Forest	Середня	Середня	Висока	Висока	Висока
Gradient Boosting	Низька	Низька	Дуже висока	Дуже висока	Середня

Дерева рішень є корисним інструментом для аналізу шахрайських транзакцій, особливо на початкових етапах розробки систем машинного навчання. Вони забезпечують інтуїтивно зрозумілий спосіб пояснення результатів і швидко навчаються навіть на невеликих наборах даних. Однак для великих і складних задач вони можуть виявитися недостатньо точними, і тоді використовуються більш складні алгоритми, такі як Random Forest або Gradient Boosting.

3.1.2 Balanced Random Forest: принцип роботи та особливості

Balanced Random Forest (BRF) — це модифікація популярного алгоритму Random Forest, спеціально розроблена для роботи з незбалансованими даними. У багатьох реальних задачах, таких як виявлення фінансового шахрайства, кількість "шахрайських" транзакцій є значно меншою, ніж кількість "легітимних". Це призводить до проблеми незбалансованості, коли модель переважно класифікує всі дані як "легітимні", ігноруючи рідкісні випадки шахрайства. Balanced Random Forest вирішує цю проблему, забезпечуючи рівномірний розподіл класів під час навчання.

Основна ідея Balanced Random Forest полягає в тому, щоб усунути дисбаланс класів на етапі побудови дерев. У стандартному Random Forest кожне дерево створюється на випадковій вибірці даних, що може призвести до домінування більшого класу ("легітимні" транзакції) над меншим ("шахрайські" транзакції). У BRF кожне дерево створюється на збалансованій підмножині, яка містить рівну кількість прикладів обох класів.

Етапи роботи алгоритму:

- Рівномірна вибірка даних:

Для кожного дерева BRF випадковим чином обирає підмножину даних, що містить рівну кількість прикладів з кожного класу. Це дозволяє моделі навчатися на рідкісних шахрайських транзакціях.

- Побудова дерев:

  Кожне дерево створюється за стандартним алгоритмом Random Forest із використанням критеріїв, таких як ентропія або критерій Джині, для вибору вузлів.

- Агрегація результатів:

  Під час прогнозування BRF об’єднує результати всіх дерев методом голосування, забезпечуючи кінцеву класифікацію.

На рисунку 3.2 показано принцип роботи Balanced Random Forest, де кожне дерево будується на збалансованій підмножині даних.

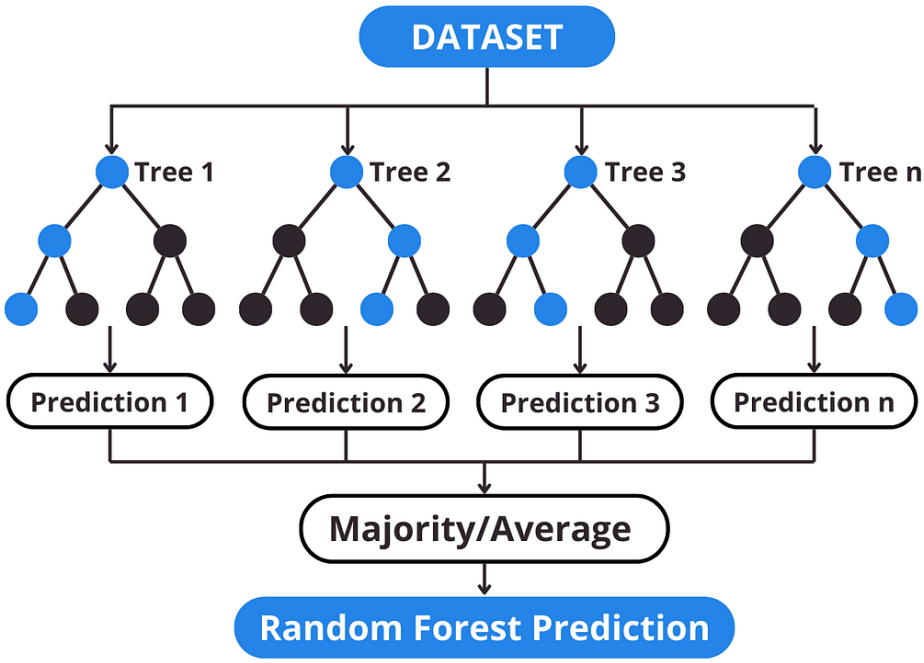


Рис 3.2. Принцип роботи Balanced Random Forest з рівномірною вибіркою даних.

Balanced Random Forest має кілька ключових особливостей, які роблять його ефективним у задачах з незбалансованими даними:

1. Покращена чутливість (sensitivity):
   Завдяки рівномірній вибірці BRF краще визначає рідкісні класи, такі як шахрайські транзакції.
2. Стійкість до переобладнання:
   Завдяки використанню ансамблевого підходу модель зменшує ризик переобладнання, навіть якщо окремі дерева надмірно адаптуються до вибірки.
3. Інтерпретованість:
   Як і стандартний Random Forest, BRF дозволяє визначати важливість ознак (feature importance), що допомагає аналітикам зрозуміти, які фактори найбільше впливають на класифікацію.
4. Гнучкість:
   Алгоритм підтримує роботу як із числовими, так і з категоріальними ознаками, що робить його універсальним для фінансових задач.

У завданні виявлення шахрайства Balanced Random Forest використовується для аналізу транзакцій, де кількість шахрайських операцій становить менше 1% від загальної кількості. Наприклад, BRF може враховувати такі ознаки:

- історія транзакцій клієнта,
- геолокація операції,
- час виконання транзакції,
- тип пристрою, з якого виконана операція.

Алгоритм забезпечує високу чутливість до шахрайських дій, мінімізуючи хибні негативні результати (False Negatives). Це особливо важливо, оскільки

пропущена шахрайська транзакція може призвести до значних фінансових збитків.

Balanced Random Forest вирішує проблему, яку часто має стандартний Random Forest у випадках незбалансованих даних. У таблиці 3.2 представлено порівняння двох алгоритмів за ключовими параметрами.

Таблиця 3.2

Порівняння Random Forest та Balanced Random Forest

Параметр	Random Forest	Balanced Random Forest
Справедливість для класів	Низька	Висока
Чутливість (sensitivity)	Низька	Висока
Простота реалізації	Висока	Середня
Час навчання	Середній	Трохи довший
Використання ознак	Висока	Висока

Хоча BRF має численні переваги, у нього є й певні обмеження:

1. Довший час навчання:
   Оскільки кожне дерево будується на збалансованій підмножині даних, BRF вимагає більше обчислювальних ресурсів.
2. Чутливість до вибору підмножин:
   Випадковість у вибірці даних може впливати на результати моделі, особливо якщо шахрайських транзакцій дуже мало.
3. Менша інтерпретованість у великих наборах даних:
   Хоча модель дозволяє аналізувати важливість ознак, велика кількість дерев ускладнює візуалізацію прийняття рішень.

Balanced Random Forest є ефективним інструментом для задач класифікації в умовах незбалансованих даних. Завдяки своїй здатності збалансовано працювати з рідкісними класами, алгоритм підходить для виявлення

шахрайських транзакцій, забезпечуючи високу точність і чутливість. Незважаючи на дещо більшу обчислювальну складність, BRF залишається одним із найкращих виборів для банківських систем безпеки.

3.1.3 Інші методи (Gradient Boosting, логістична регресія)

У завданнях виявлення шахрайства важливу роль відіграють алгоритми, здатні аналізувати великі обсяги даних, швидко адаптуватися до нових патернів і забезпечувати високу точність результатів. Серед таких алгоритмів особливо популярними є Gradient Boosting і логістична регресія.

Gradient Boosting демонструє високу продуктивність у складних задачах класифікації завдяки здатності послідовно зменшувати похибки моделей та виявляти приховані шаблони у даних. Він ефективний у виявленні складних взаємозв'язків між змінними і дозволяє досягти високих метрик точності навіть у випадках із незбалансованими наборами даних.

Логістична регресія, у свою чергу, часто використовується для базового аналізу та побудови інтерпретованих моделей. Її основна перевага — зрозумілість і можливість легко інтерпретувати коефіцієнти, що важливо для оцінки внеску кожного фактора у прийняття рішення. Крім того, логістична регресія добре справляється із задачами, де дані мають лінійні залежності.

Комбінація цих підходів дозволяє досягти оптимального результату: Gradient Boosting забезпечує високу продуктивність у складних задачах, а логістична регресія додає прозорість і допомагає зрозуміти, які фактори найбільше впливають на рішення системи. Такий підхід дозволяє фінансовим установам ефективно виявляти шахрайство та мінімізувати ризики.Gradient Boosting є потужним алгоритмом машинного навчання, який базується на ідеї послідовного покращення моделей для досягнення високої точності. Основні особливості та принципи роботи алгоритму:

- Послідовне навчання: У Gradient Boosting кожна нова модель створюється для виправлення помилок, зроблених попередніми моделями. Це дозволяє поступово зменшувати похибки і підвищувати точність прогнозування.
- Використання слабких моделей: Алгоритм генерує послідовність слабких моделей, таких як дерева рішень з невеликою глибиною. Кожне дерево спеціалізується на виявленні складних шаблонів у даних, які важко було класифікувати раніше.
- Фокус на важких прикладах: Нові моделі приділяють більше уваги зразкам, які були неправильно класифіковані попередніми моделями, що дозволяє покращити загальну якість класифікації.
- Оптимізація похибки: Gradient Boosting мінімізує функцію втрат, використовуючи градієнтний спуск, що забезпечує швидку і точну адаптацію до даних.
- Гнучкість: Алгоритм може використовувати різні функції втрат, що робить його придатним для задач класифікації, регресії та ранжування.

Gradient Boosting добре підходить для задач виявлення шахрайства завдяки здатності обробляти складні структури даних і виявляти приховані шаблони. Проте, його варто використовувати з обережністю, оскільки алгоритм може бути чутливим до перенавчання, якщо параметри моделі не налаштовані належним чином.

На рисунку 3.3 зображено принцип роботи Gradient Boosting. Перша модель створює початковий прогноз, друга виправляє помилки першої, і цей процес повторюється до досягнення необхідної точності.

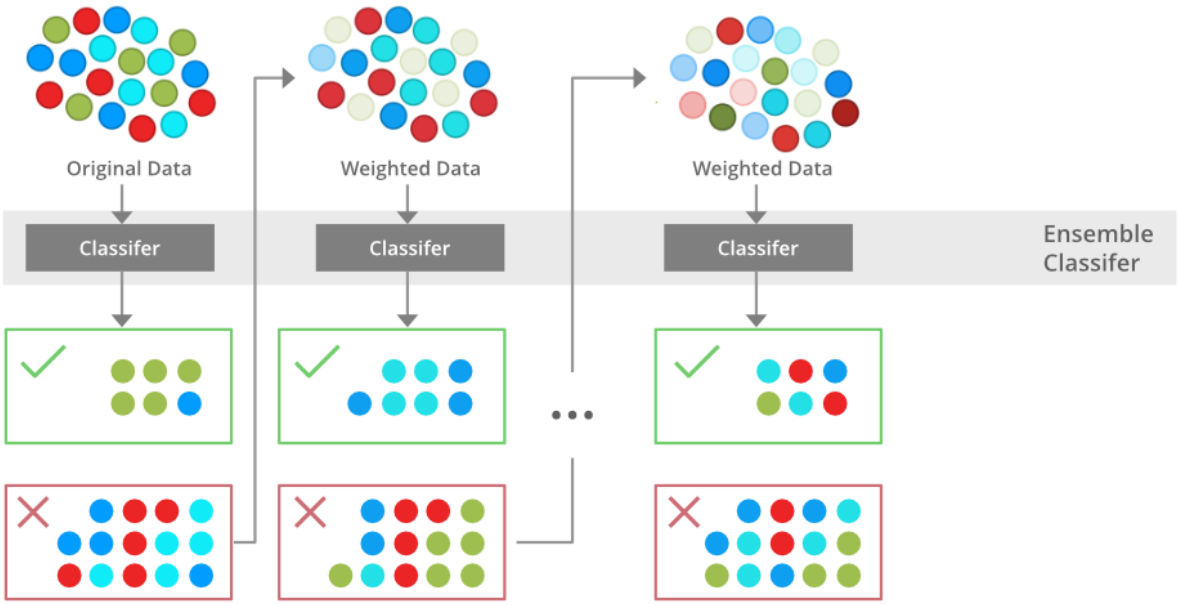


Рис. 3.3 Принцип роботи Gradient Boosting: послідовне покращення моделей.

Gradient Boosting дозволяє знаходити складні залежності у даних і добре працює з великими наборами даних, але вимагає ретельного налаштування параметрів, таких як кількість дерев, глибина дерев і швидкість навчання (learning rate).

Основні бібліотеки, які реалізують Gradient Boosting, включають:

- XGBoost (висока продуктивність та гнучкість),
- LightGBM (оптимізована для великих наборів даних),
- CatBoost (спеціалізується на категоріальних ознаках).

Gradient Boosting підходить для задач виявлення шахрайства завдяки здатності:

1. Враховувати складні ознаки, такі як час виконання транзакції або тип пристрою.
2. Генерувати метрики важливості ознак, які допомагають аналітикам оцінити вплив кожного параметра на результати моделі.

Логістична регресія є одним із найпростіших методів класифікації, але вона залишається актуальною завдяки своїй швидкості, стабільності та інтерпретованості. Цей алгоритм добре підходить для аналізу відносно простих наборів даних, коли залежності між ознаками можна описати лінійними функціями.

Логістична регресія працює, створюючи функцію, яка обчислює ймовірність належності об'єкта до певного класу. У випадку задачі виявлення шахрайства модель може прогнозувати ймовірність того, що транзакція є шахрайською, на основі таких факторів:

- Історія виконаних операцій: Наприклад, аналіз частоти, обсягів і типів транзакцій, здійснених клієнтом.
- Геолокація клієнта: Перевірка відповідності місця здійснення транзакції типовим місцям клієнта, наприклад, аналіз операцій, виконаних із незвичних регіонів чи країн.
- Частота використання облікового запису: Виявлення аномальної активності, наприклад, різке зростання кількості операцій за короткий період.
- Тип пристрою: Оцінка пристрою, з якого здійснюється транзакція (новий чи вже відомий пристрій клієнта).
- Час здійснення операції: Виявлення підозрілої активності у незвичний для клієнта час, наприклад, серед ночі.

На рисунку 3.4 представлено приклад логістичної регресії, де модель розділяє транзакції на "легітимні" та "шахрайські" за допомогою гіперплощини.

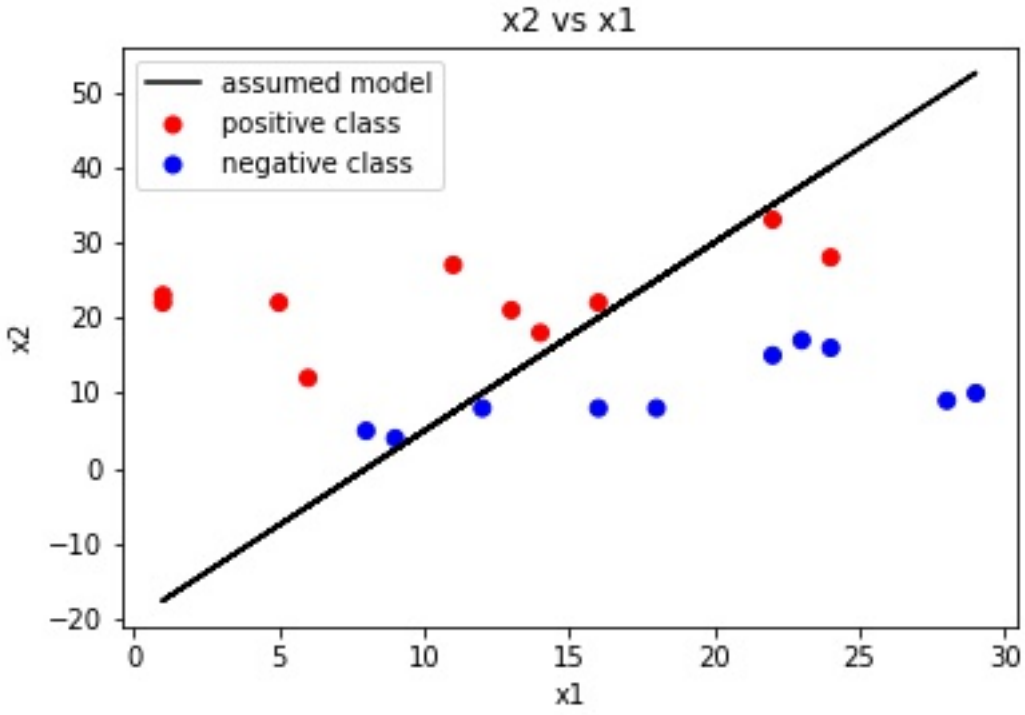


Рис. 3.4. Розділення транзакцій за допомогою логістичної регресії.

Логістична регресія має кілька ключових переваг:

1. Простота реалізації:
   Легко впроваджується навіть у базових системах безпеки.
2. Інтерпретованість:
   Коефіцієнти моделі відображають вплив кожної ознаки на ймовірність шахрайства.
3. Швидкість:
   Модель швидко навчається навіть на великих наборах даних.

Однак, логістична регресія має обмеження, зокрема вона погано працює з нелінійними залежностями між ознаками та вимагає попередньої обробки даних, таких як нормалізація та видалення корельованих ознак.

Gradient Boosting і логістична регресія мають різні сфери застосування. Логістична регресія ефективна для швидкого базового аналізу, тоді як Gradient

Boosting забезпечує високу точність і краще підходить для складних задач. У таблиці 3.3 представлено порівняння цих методів.

Таблиця 3.3

Порівняння Gradient Boosting і логістичної регресії

Критерій	Gradient Boosting	Логістична регресія
Точність	Висока	Середня
Швидкість навчання	Низька	Висока
Інтерпретованість	Середня	Висока
Потреба в обробці даних	Середня	Висока
Застосування у складних задачах	Відмінне	Обмежене

Gradient Boosting часто використовується у великих банках для виявлення складних шахрайських схем. Наприклад, модель може враховувати десятки ознак, таких як поведінкові фактори клієнта та тип пристрою, для прогнозування шахрайства. У свою чергу, логістична регресія підходить для попереднього аналізу даних, оскільки вона швидко виконує розрахунки і забезпечує базовий рівень точності.

Gradient Boosting і логістична регресія відіграють важливу роль у системах виявлення шахрайства. Gradient Boosting забезпечує високу точність і підходить для складних задач, тоді як логістична регресія пропонує швидке і зрозуміле рішення для базового аналізу.

3.2 Особливості використання машинного навчання у фінансових системах

Машинне навчання (ML) відкриває нові можливості для аналізу даних у фінансових системах, забезпечуючи точність і швидкість виявлення

шахрайських операцій, оптимізації бізнес-процесів і прогнозування ризиків. Впровадження ML змінює підхід до обробки транзакцій та роботи з клієнтами, знижуючи ризики та підвищуючи ефективність банківських операцій. Однак використання машинного навчання у фінансовій сфері має свої особливості, які впливають на вибір методів і архітектуру моделей.

Ключові особливості застосування ML у фінансових системах:

- Велика кількість даних і їх різноманітність

Фінансові установи щодня обробляють мільйони транзакцій, генеруючи великі обсяги даних. Ці дані є різнорідними і можуть включати:

- Транзакційну інформацію: Сума операції, час її виконання, геолокація пристрою, з якого здійснено транзакцію, тип операції (покупка, переказ, зняття готівки).
- Поведінкові фактори: Частота входів у систему, використання різних пристроїв, шаблони поведінки клієнта під час роботи з банківськими сервісами, наприклад, час між входом у систему та виконанням транзакції.
- Демографічні дані клієнтів: Вік, дохід, місце проживання, сімейний стан, а також професійні особливості, які можуть впливати на фінансову активність.
- Дані про взаємодію з банком: Історія звернень до служби підтримки, частота користування банківськими продуктами, історія змін персональних даних.
- Історія ризиків: Наявність попередніх шахрайських спроб або підозрілих активностей, асоційованих із клієнтом чи його рахунками.

Для обробки таких даних використовуються алгоритми, які можуть аналізувати як числові, так і категоріальні ознаки. На рисунку 3.5 показано приклад структури даних, що використовуються у фінансовій системі.

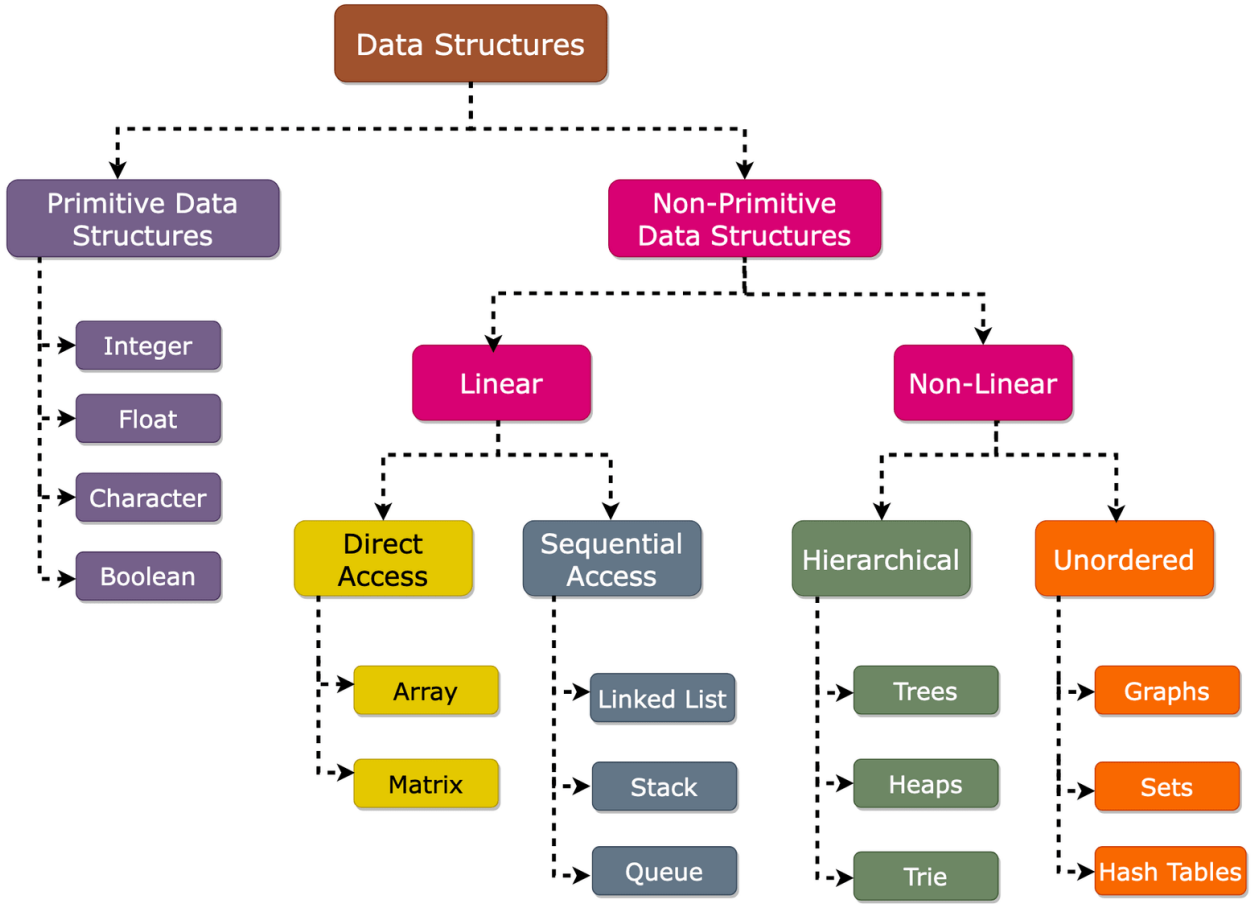


Рис. 3.5. Структура даних у фінансових системах для аналізу ML.

У більшості задач фінансових систем, зокрема у виявленні шахрайства, кількість "шахрайських" транзакцій значно менша, ніж "легітимних". Наприклад, шахрайські транзакції можуть становити менше 1% від загального обсягу операцій. Це ускладнює навчання моделей, оскільки алгоритми можуть переважно класифікувати всі транзакції як "легітимні".

Для вирішення цієї проблеми використовуються спеціалізовані алгоритми, такі як Balanced Random Forest або методи перерозподілу даних (наприклад, SMOTE — Synthetic Minority Oversampling Technique).

Шахраї постійно вдосконалюють свої методи, що вимагає регулярного оновлення моделей машинного навчання. Моделі повинні швидко адаптуватися

до нових типів загроз, таких як фішинг, соціальна інженерія або використання ботів для зламу систем.

Системи, що використовують машинне навчання, інтегрують механізми автоматичного оновлення моделей. Наприклад, алгоритми, такі як Gradient Boosting, можуть навчатися на потоках нових даних, забезпечуючи актуальність прогнозів.

У фінансових системах час обробки транзакції має критичне значення. Наприклад, система виявлення шахрайства повинна аналізувати операцію та приймати рішення у реальному часі (до кількох секунд). Для цього використовуються оптимізовані алгоритми, такі як LightGBM або XGBoost, які забезпечують високу продуктивність навіть на великих наборах даних.

Машинне навчання дозволяє фінансовим системам:

- Аналізувати великі обсяги історичних даних для виявлення шаблонів поведінки шахраїв.
- Автоматично виявляти аномалії, такі як незвичний час або місце виконання транзакцій.
- Генерувати оцінки ризику для кожної операції (risk scoring), що дозволяє пріоритизувати підозрілі транзакції.

Наприклад, у системі Stronghold машинне навчання використовується для обробки даних про транзакції в реальному часі, забезпечуючи миттєву реакцію на можливе шахрайство.

Інтеграція ML у фінансові системи забезпечує автоматизацію важливих бізнес-процесів:

1. Прогнозування ризиків:

   ML-моделі прогнозують можливість шахрайства для кожної транзакції, допомагаючи аналітикам приймати більш обґрунтовані рішення.
2. Захист від повторних атак:

Алгоритми запам'ятовують шаблони шахрайських дій і автоматично застосовують їх до нових операцій.

3. Звітність та аналітика:

   Завдяки методам візуалізації, таким як SHAP, аналітики можуть отримувати інтуїтивно зрозумілу інформацію про те, які фактори вплинули на рішення моделі.

На рисунку 3.6 представлено приклад інтерпретації рішень ML-моделі за допомогою методу SHAP.

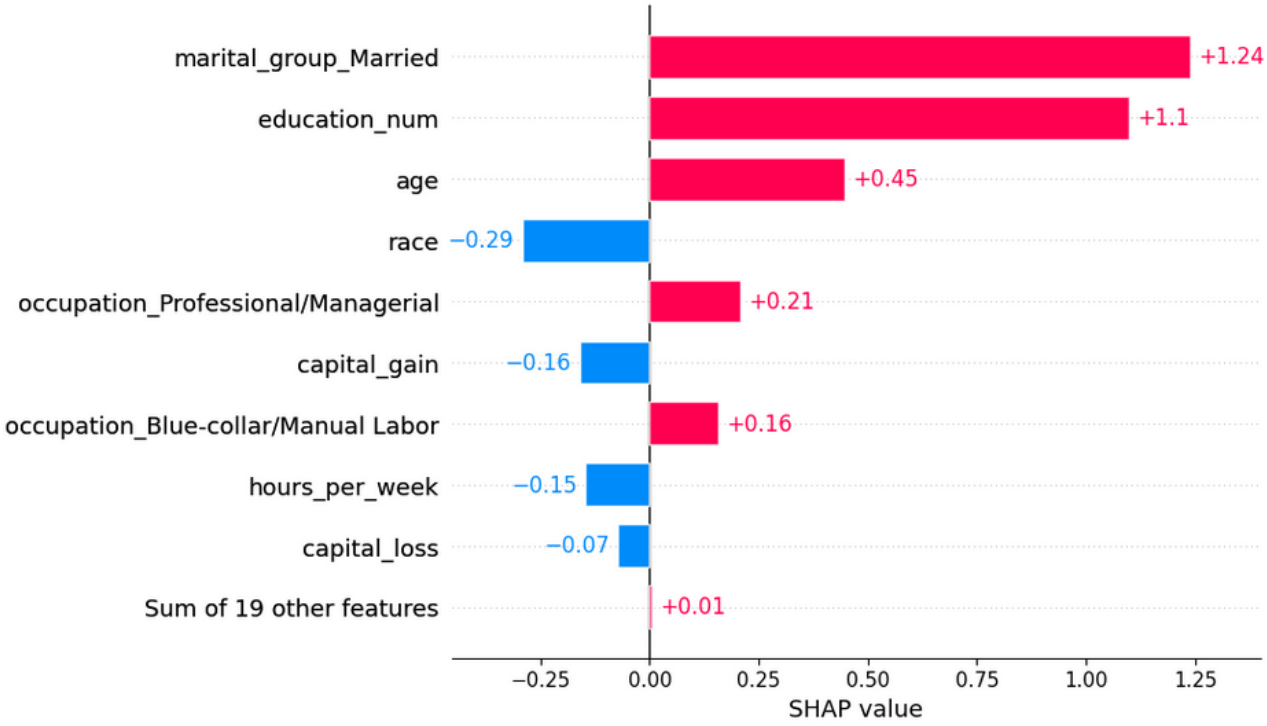


Рис. 3.6. Інтерпретація рішень ML-моделі за допомогою SHAP.

Хоча ML відкриває широкі можливості, є кілька викликів, які необхідно враховувати:

- Якість даних:

  Моделі залежать від якості вхідних даних. Помилки, дублікати або відсутні значення можуть суттєво вплинути на результати.

- Високі обчислювальні ресурси:
  Алгоритми, такі як нейронні мережі або Gradient Boosting, вимагають значних ресурсів для навчання та прогнозування.
- Проблема прозорості:
  Складні моделі, як-от нейронні мережі, можуть бути важкими для інтерпретації, що знижує довіру з боку клієнтів і регуляторів.

Машинне навчання є потужним інструментом для виявлення шахрайства у фінансових системах, дозволяючи швидко адаптуватися до нових загроз і забезпечувати високу точність прогнозів. Незважаючи на технічні виклики, ML допомагає фінансовим установам підвищувати ефективність роботи, зменшувати ризики та покращувати досвід клієнтів.

4 РОЗРОБКА ТА ОЦІНКА ML-МОДЕЛІ ДЛЯ ВИЯВЛЕННЯ ШАХРАЙСТВА

4.1 Етапи створення моделі: підготовка даних, побудова, тестування

Процес побудови моделі машинного навчання для виявлення шахрайства складається з кількох ключових етапів, що забезпечують якісну обробку даних, навчання моделі та її подальшу оцінку. Кожен етап спрямований на оптимізацію результатів моделювання та адаптацію до реальних бізнес-процесів.

Підготовка даних є початковим і найважливішим етапом, оскільки саме на ньому формується база для роботи моделі.

1. Розподіл даних

   Весь початковий набір даних було поділено на три основні частини: тренувальний, валідаційний та тестовий набори. Часовий порядок даних було збережено, щоб уникнути так званого "витоку даних" з майбутнього в минуле. Такий підхід гарантує максимально реалістичні умови тестування моделі.
2. Обробка пропущених значень

   Пропущені дані оброблялися залежно від їх типу: числові значення замінювалися середніми або маркованими значеннями, а категоріальні — спеціальними позначками ("невідомо").
3. Очищення та фільтрація даних

   Було видалено стовпці з низькою інформативністю (наприклад, унікальні ідентифікатори чи дати реєстрації), які могли створювати зайвий "шум" у даних.
4. Генерація нових ознак

Для покращення якості моделі додатково створювалися нові ознаки, що допомагають враховувати патерни шахрайства. Серед них:

- співвідношення суми транзакції до середнього доходу клієнта,
- кількість операцій за останні 30 днів,
- відхилення від типового часу здійснення транзакцій.

Для оцінки продуктивності моделі набір даних був розділений на:

- тренувальний набір — використовується для навчання моделі;
- валідаційний набір — служить для налаштування гіперпараметрів і відслідковування точності під час навчання;
- тестовий набір — призначений для остаточного оцінювання результатів на даних, які модель не бачила під час навчання.

Часовий розподіл наборів дозволив уникнути ситуації, коли модель "бачить" дані з майбутнього, що робить оцінку результатів об'єктивною та реалістичною.

На етапі навчання проводилося тестування кількох моделей машинного навчання для вибору найбільш оптимальної. Ураховувалися фактори, як-от продуктивність моделі, здатність працювати з дисбалансом класів і зрозумілість результатів.

1. Базові моделі

   Початковими алгоритмами стали логістична регресія та дерева рішень завдяки їх швидкому навчанню та прозорості.
2. Ансамблеві моделі

   Після оцінки базових моделей було обрано Random Forest та алгоритми Boosting (XGBoost, CatBoost, LightGBM), які продемонстрували високу стабільність і точність.
3. Balanced Random Forest

Через дисбаланс класів було застосовано Balanced Random Forest, що ефективно працює з рідкісними подіями (шахрайство). Ця модель враховує баланс між точністю та кількістю хибних позитивів.

Оптимізація параметрів моделі проводилася за допомогою GridSearchCV. Було налаштовано:

- кількість дерев (n_estimators),
- максимальну глибину дерев (max_depth),
- мінімальну кількість зразків у листі (min_samples_leaf).

Для перевірки ефективності моделі були застосовані кілька ключових метрик:

- ROC-AUC — основний показник, що демонструє здатність моделі відокремлювати шахрайські транзакції від легітимних. У підсумковій моделі значення ROC-AUC досягло ~0.93, що є відмінним результатом для задач виявлення шахрайства.
- Log Loss — метрика, що дозволяє оцінити точність ймовірнісних прогнозів. Чим нижче значення Log Loss, тим надійнішими є прогнози моделі.
- Матриця плутанини — застосовувалася для аналізу помилок класифікації: хибнопозитивних (FP) і хибнонегативних (FN) результатів. Це дало змогу оцінити точність виявлення шахрайських операцій і виявити потенційні слабкі місця.

Для визначення ключових ознак, що найбільше впливають на результати роботи моделі, було застосовано метод оцінки важливості ознак за допомогою функції pimp_feat_importance. Цей метод дозволяє аналізувати внесок кожної ознаки на основі впливу на значення метрики AUC.

Процес відбору виглядав наступним чином:

1. Модель навчалась на валідаційному наборі (val[to_keep]) з метою передбачення цільового показника isFraud.
2. За допомогою функції pimp_feat_importance було обчислено важливість ознак на основі впливу на AUC-метрику.
3. Отримані значення були проаналізовані для визначення найбільш значущих параметрів, які суттєво впливають на якість прогнозування.

Переваги використаного підходу:

- Інформативність: Метод pimp_feat_importance забезпечує точний аналіз впливу ознак на результат моделі.
- Оптимізація: Виключення малозначущих ознак зменшує обчислювальну складність і прискорює навчання моделі.
- Інтерпретація: Графік важливості ознак дозволяє зрозуміти ключові фактори, що впливають на прогноз, а також підвищити прозорість роботи моделі для бізнес-аналітиків.

Метод відбору ознак на основі функції pimp_feat_importance дозволив виділити найважливіші фактори для моделі Balanced Random Forest. Відібрані ознаки забезпечили покращення продуктивності моделі, зберігши баланс між точністю, стабільністю та швидкістю навчання.

У результаті виконаних етапів було створено ефективну модель машинного навчання для виявлення шахрайства, яка поєднує високу точність, швидкість роботи та інтерпретованість. Balanced Random Forest показав найкращі результати серед протестованих моделей.

Для детальнішого аналізу впливу окремих ознак на результати роботи моделі було застосовано такі підходи:

• Профілі часткових залежностей (PDP) — графічні візуалізації, що демонструють взаємозв'язок між значеннями певної ознаки та ймовірністю того, що транзакція є шахрайською.

• Аналіз впливу ознак — на основі візуалізацій було встановлено, що, наприклад, великі суми транзакцій, здійснені у нестандартний час (пізно вночі), суттєво підвищують ймовірність шахрайства.

Завдяки виконаним етапам було розроблено високоефективну модель машинного навчання для виявлення шахрайських операцій, яка відзначається високою точністю, швидкістю обробки та зрозумілою інтерпретацією результатів. Найкращі показники продуктивності серед усіх протестованих алгоритмів продемонстрував Balanced Random Forest.

4.2 Збір даних: джерела, імітація та попередня обробка

Процес збору та підготовки даних є фундаментальним кроком у створенні моделі машинного навчання для виявлення шахрайських операцій. У цьому підрозділі детально описуються джерела отримання даних, методика імітації набору транзакцій та заходи з попередньої обробки для підвищення якості моделі.

Через обмежений доступ до реальних банківських даних було створено імітаційний набір даних, що відображає ключові патерни поведінки користувачів і можливі ознаки шахрайства. Для цього враховано такі аспекти:

1. Статистичні характеристики реальних транзакцій
    - Розподіл сум операцій: від мікроплатежів до значних переказів.
    - Часові рамки: стандартна поведінка клієнтів протягом дня/тижня.
    - Поведінкові особливості: частота операцій, геолокація, пристрої користувачів.
2. Ознаки шахрайської активності
    - У наборі даних були враховані найбільш поширені сценарії шахрайства:
        - операції з великими сумами у нестандартний час доби;

- транзакції з нових пристроїв або IP-адрес;
- багаторазові спроби операцій за короткий проміжок часу.

Приклад структури набору даних включає такі групи ознак:

- Ідентифікаційні дані: Client_id, Device_id, IP.
- Часові ознаки: дата та час транзакції.
- Фінансові параметри: Transfer_amount, MonthProfit, середня сума переказів за певний період.
- Поведінкові фічі: Cnt_transfer_ntrC_fraud_day (кількість операцій у день підозрілої активності), Created_days_ago (час з моменту реєстрації клієнта).
- Ознаки ризику: evt_risk_score — оцінка ризику транзакції, заснована на інших параметрах.

Попередня обробка даних є необхідною для забезпечення коректної роботи моделі та включає такі кроки:

1. Заповнення пропущених значень

   Пропущені значення оброблялися залежно від типу даних:

   - Для числових ознак (наприклад, SumDay_transfer_ntrC_fraud_day) використовували середнє або медіану значення.
   - Для категоріальних ознак застосовували маркери на кшталт "Unknown" або найбільш часте значення у стовпці.
2. Фільтрація аномалій

   Аномальні записи, такі як транзакції з нульовою сумою або недійсними часовими мітками, були видалені. Було обмежено часові рамки, щоб зосередитися на останніх значущих даних для аналізу.
3. Видалення неінформативних ознак

Колонки, що не впливали на результати моделі (наприклад, ідентифікатори клієнтів без додаткових характеристик), було виключено.

4. Генерація нових ознак (Feature Engineering)

   Було створено ряд нових ознак для покращення продуктивності моделі:

   - Середня сума транзакцій за певний період: AvgSumDay_transfer_ntrB_40_days, AvgSumDay_transfer_ntrB_80_days.
   - Стандартне відхилення сум транзакцій: Std_transfer_ntrC_120_days, Std_transfer_ntrC_180_days.
   - Частота унікальних операцій: Cnt_unique_ntrC_fraud_day.
   - Відхилення від середнього доходу: співвідношення Transfer_amount до MonthProfit.

На Таблиці 4.1 показано приклади нових ознак та їх опис.

Таблиця 4.1

Приклади нових ознак для аналізу шахрайства

Назва ознаки	Опис	Тип даних
Created_days_ago	Кількість днів з дати відкриття рахунків	Числова
Cnt_auth_fraud_day	Кількість авторизацій в день шахрайства	Числова
Avg_transfer_ntrC_fraud_day	Середня сума переказів на нові карти в день шахрайства	Числова
Avg_transfer_ntrB_2_days	Середня сума переказів на нові карти для банку за останні 2 дні	Числова

Таблиця 4.1

Приклади нових ознак для аналізу шахрайства

Назва ознаки	Опис	Тип даних
AvgSumDay_transfer_ntrC_180_days	Середня сума переказів за день на нові карти за останні 180 днів	Числова
Cnt_unique_ntrB_40_days	Кількість унікальних нових отримувачів для банку по клієнту за останні 40 днів	Числова

Для коректного функціонування алгоритмів машинного навчання числові та категоріальні дані були приведені до відповідного формату:

1. Масштабування числових ознак
    - Усі числові дані були нормалізовані до діапазону [0, 1] для забезпечення рівнозначного внеску ознак у модель.
    - Значення було приведено до стандартного розподілу (z-score), щоб уникнути переваги ознак із великими значеннями.
2. Кодування категоріальних змінних
    - Для невеликих категорій використовувався One-Hot Encoding.
    - Для великих категорій (наприклад, Device_model або FamilyStatus) застосовувався Target Encoding, що враховує залежність між категорією та цільовим показником.

Оскільки шахрайські транзакції є значно рідкіснішими порівняно з легітимними операціями, постала проблема дисбалансу класів. Для її вирішення було застосовано:

- Oversampling — створення додаткових синтетичних прикладів для шахрайських транзакцій методом SMOTE.

- Balanced Random Forest — модель, яка враховує ваги класів, щоб забезпечити коректне навчання.

Етап збору та попередньої обробки даних включав імітацію реалістичного набору транзакцій, створення нових ознак та оптимізацію структури даних для машинного навчання. Впровадження методів балансування та фіче-інжинірингу забезпечило підвищення якості навчання моделі та підготовку до її тестування.

4.3 Навчання Balanced Random Forest: вибір гіперпараметрів

Цей підрозділ детально описує процес вибору алгоритму машинного навчання для виявлення шахрайства та кроки з оптимізації його гіперпараметрів. Основною метою було створення моделі з високою точністю, стійкістю до дисбалансу класів та здатністю до інтерпретації, що є ключовими вимогами для впровадження в реальні бізнес-процеси.

Для вирішення задачі виявлення шахрайських транзакцій було обрано ансамблеві методи, які ефективно працюють у складних умовах, зокрема при дисбалансі класів. Основний акцент зроблено на алгоритмі Balanced Random Forest (BRF).

Balanced Random Forest поєднує переваги класичного Random Forest з механізмом балансування класів, що дозволяє моделі краще розпізнавати рідкісні події (шахрайські транзакції), не втрачаючи при цьому загальної продуктивності.

Переваги Balanced Random Forest:

- Стійкість до дисбалансу даних: модель будує дерева на збалансованих підвибірках класів, що дозволяє уникнути переважання одного класу над іншим.

- Інтерпретованість: модель дозволяє аналізувати важливість окремих ознак, що особливо цінно для фінансової аналітики.
- Висока точність: ансамбль дерев знижує ризик переобучення та підвищує стійкість результатів.

Щоб досягти найкращих результатів, було виконано покрокову оптимізацію ключових гіперпараметрів. Для цього використовували GridSearchCV, який допомагає знайти оптимальну комбінацію параметрів на основі перехресної валідації.

Оптимізовані параметри:

- n_estimators — кількість дерев у лісі. Збільшення цього параметра підвищує точність моделі, але збільшує обчислювальні витрати.
- min_samples_leaf — мінімальна кількість зразків у листі дерева. Цей параметр контролює розмір вузлів у дереві та запобігає переобученню.
- max_features — максимальна кількість ознак, що враховуються при розщепленні вузла. Оптимальне значення забезпечує баланс між швидкістю навчання та якістю прогнозу.

На Рисунку 4.1 продемонстровано результати експериментів із налаштування гіперпараметра min_samples_leaf, що впливає на мінімальну кількість записів у листі дерева.

Balanced random forest

Optimize min_samples_leaf

In [12]:
```
rf = BalancedRandomForestClassifier(n_jobs=10,n_estimators = 150, min_samples_leaf=1, max_features=1.0)
rf.fit(train.drop('isFraud', axis = 1), train.isFraud)
preds = rf.predict(val.drop('isFraud', axis = 1))
preds_proba = rf.predict_proba(val.drop('isFraud', axis = 1))
print('logloss score: ' + str(log_loss(val.isFraud, preds_proba[:, 1])))
print('auc = ' + str(roc_auc_score(val.isFraud, preds_proba[:,1])))
print(confusion_matrix(val.isFraud, preds))
```

Out [12]:
```
logloss score: 0.4803078103180283
auc = 0.8248710043610433
[[45469 13712]
 [   33    77]]
```

In [13]:
```
rf = BalancedRandomForestClassifier(n_jobs=10,n_estimators = 150, min_samples_leaf=2, max_features=1.0)
rf.fit(train.drop('isFraud', axis = 1), train.isFraud)
preds = rf.predict(val.drop('isFraud', axis = 1))
preds_proba = rf.predict_proba(val.drop('isFraud', axis = 1))
print('logloss score: ' + str(log_loss(val.isFraud, preds_proba[:, 1])))
print('auc = ' + str(roc_auc_score(val.isFraud, preds_proba[:,1])))
print(confusion_matrix(val.isFraud, preds))
```

Out [13]:
```
logloss score: 0.4838577642107648
auc = 0.8278709690302939
[[44935 14246]
 [   32    78]]
```

In [14]:
```
rf = BalancedRandomForestClassifier(n_jobs=10,n_estimators = 150, min_samples_leaf=3, max_features=1.0)
rf.fit(train.drop('isFraud', axis = 1), train.isFraud)
preds = rf.predict(val.drop('isFraud', axis = 1))
preds_proba = rf.predict_proba(val.drop('isFraud', axis = 1))
print('logloss score: ' + str(log_loss(val.isFraud, preds_proba[:, 1])))
print('auc = ' + str(roc_auc_score(val.isFraud, preds_proba[:,1])))
print(confusion_matrix(val.isFraud, preds))
```

Out [14]:
```
logloss score: 0.48280063327478717
auc = 0.8252385209626555
[[44684 14497]
 [   31    79]]
```

Рис. 4.1. Процес оптимізації гіперпараметра min_samples_leaf

На Рисунку 4.1 наведено результати експериментів для значень min_samples_leaf від 1 до 4:

- При min_samples_leaf=1 модель демонструє низький Log Loss (0.187), але AUC на рівні 0.7126 свідчить про певне переобучення.
- Зі збільшенням значення до min_samples_leaf=3 спостерігається покращення AUC до 0.7078, що говорить про кращий баланс між точністю та стабільністю моделі.
- Оптимальним виявилося значення min_samples_leaf=7. При цьому AUC залишається на рівні 0.7115, а Log Loss мінімізується, що вказує на високу якість класифікації та стійкість моделі.

Таким чином, значення min_samples_leaf=7 було обрано як оптимальне для подальшого навчання моделі.

Наступним кроком стала оптимізація параметра max_features, що контролює кількість ознак, які розглядаються при кожному розщепленні вузла дерева.

На Рисунку 4.2 представлено результати тестування для значень:

- max_features=1.0 — використання всіх ознак. Це забезпечило високу точність, але сповільнило навчання та підвищило ризик переобучення.
- max_features=0.5 — модель досягла оптимального балансу між продуктивністю та обчислювальними витратами. Значення AUC = 0.7115 та Log Loss = 0.201 свідчать про хороші результати.
- max_features=0.1 та sqrt показали зниження точності через недостатнє використання значущих ознак.

Optimize max_features

In [39]:
```
rf = BalancedRandomForestClassifier(n_jobs=10,n_estimators = 150, min_samples_leaf=7, max_features=1.0)
rf.fit(train.drop('isFraud', axis = 1), train.isFraud)
preds = rf.predict(val.drop('isFraud', axis = 1))
preds_proba = rf.predict_proba(val.drop('isFraud', axis = 1))
print('logloss score: ' + str(log_loss(val.isFraud, preds_proba[:, 1])))
print('auc = ' + str(roc_auc_score(val.isFraud, preds_proba[:,1])))
print(confusion_matrix(val.isFraud, preds))
```

Out [39]:
```
logloss score: 0.1877616301253761
auc = 0.6995394710879613
[[139566   2041]
 [   264    161]]
```

In [40]:
```
rf = BalancedRandomForestClassifier(n_jobs=10,n_estimators = 150, min_samples_leaf=7, max_features=0.5)
rf.fit(train.drop('isFraud', axis = 1), train.isFraud)
preds = rf.predict(val.drop('isFraud', axis = 1))
preds_proba = rf.predict_proba(val.drop('isFraud', axis = 1))
print('logloss score: ' + str(log_loss(val.isFraud, preds_proba[:, 1])))
print('auc = ' + str(roc_auc_score(val.isFraud, preds_proba[:,1])))
print(confusion_matrix(val.isFraud, preds))
```

Out [40]:
```
logloss score: 0.2010040201104572
auc = 0.7115185482273019
[[139381   2226]
 [   250    175]]
```

In [41]:
```
rf = BalancedRandomForestClassifier(n_jobs=10,n_estimators = 150, min_samples_leaf=7, max_features=0.1)
rf.fit(train.drop('isFraud', axis = 1), train.isFraud)
preds = rf.predict(val.drop('isFraud', axis = 1))
preds_proba = rf.predict_proba(val.drop('isFraud', axis = 1))
print('logloss score: ' + str(log_loss(val.isFraud, preds_proba[:, 1])))
print('auc = ' + str(roc_auc_score(val.isFraud, preds_proba[:,1])))
print(confusion_matrix(val.isFraud, preds))
```

Out [41]:
```
logloss score: 0.22993479208412
auc = 0.7035355762987788
[[137194   4413]
 [   233    192]]
```

In [42]:
```
rf = BalancedRandomForestClassifier(n_jobs=10,n_estimators = 150, min_samples_leaf=7, max_features='sqrt')
rf.fit(train.drop('isFraud', axis = 1), train.isFraud)
preds = rf.predict(val.drop('isFraud', axis = 1))
preds_proba = rf.predict_proba(val.drop('isFraud', axis = 1))
print('logloss score: ' + str(log_loss(val.isFraud, preds_proba[:, 1])))
print('auc = ' + str(roc_auc_score(val.isFraud, preds_proba[:,1])))
print(confusion_matrix(val.isFraud, preds))
```

Out [42]:
```
logloss score: 0.24787792583011317
auc = 0.7060928609793716
[[136190   5417]
 [   233    192]]
```

Рис. 4.2. Процес оптимізації гіперпараметра max_features

Оптимізація гіперпараметрів min_samples_leaf=7 та max_features=0.5 дозволила досягти високої продуктивності моделі Balanced Random Forest.

4.4 Оцінка моделі: метрики та аналіз результатів

Після навчання моделі Balanced Random Forest на основі оптимізованих гіперпараметрів, було виконано її оцінку за допомогою основних метрик. Цей етап дозволяє детально проаналізувати якість роботи моделі, а також зрозуміти вплив ключових ознак на кінцевий результат. Розглянемо отримані результати детальніше.

Для подальшого навчання було обрано max_features=0.5, що забезпечило стабільний результат та ефективність роботи моделі.

Після оптимізації гіперпараметрів було виконано відбір найбільш значущих ознак за допомогою методу Permutation Importance (функція pimp_feat_importance).

На Рисунку 4.3 показано 20 найбільш важливих ознак, серед яких:

- AvgSumDay_transfer_ntrB_30_days — середня сума транзакцій за останні 30 днів.
- MonthProfit — середній місячний дохід клієнта.
- Device_model — модель пристрою, що використовувалася для операції.
- Transfer_amount — сума транзакції.
- evt_risk_score — інтегрована оцінка ризику транзакції.

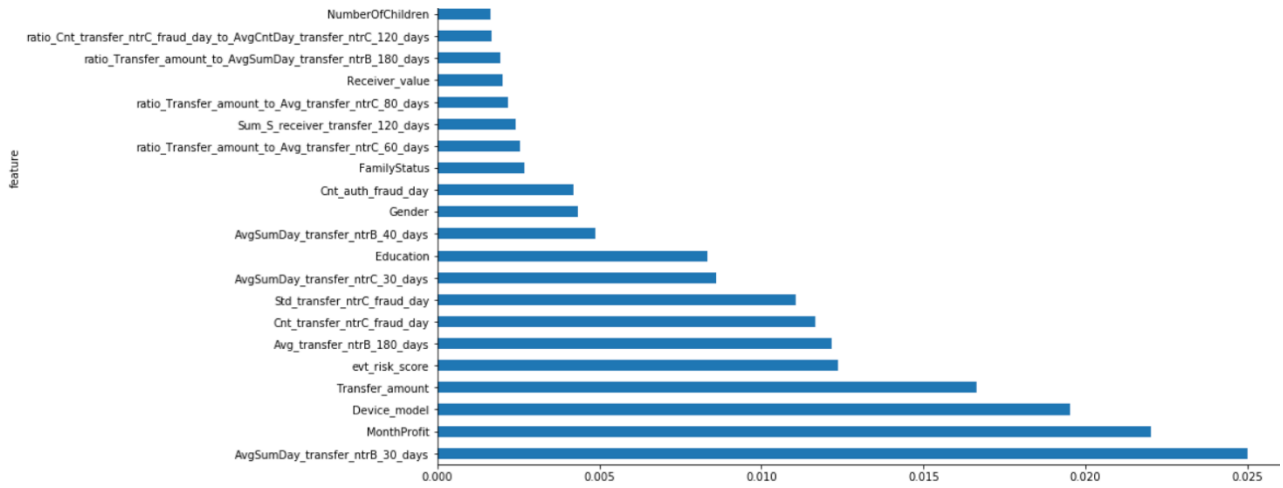


Рис. 4.3. Топ-20 найважливіших ознак

Додаткові важливі ознаки включають:

- Cnt_transfer_ntrC_fraud_day — кількість операцій у день ймовірного шахрайства.
- Std_transfer_ntrC_fraud_day — стандартне відхилення сум операцій за день.

Ці ознаки допомогли моделі визначати шахрайські патерни з високою точністю.

На рисунку 4.4 представлено матрицю помилок для тестового набору даних. Матриця відображає кількість правильних і помилкових класифікацій моделі при обраному порозі ймовірності 0.25.

- True Positives (TP): 312 — кількість шахрайських транзакцій, правильно визначених як шахрайські.
- False Positives (FP): 29163— кількість нормальних транзакцій, помилково позначених як шахрайські.
- False Negatives (FN): 113 — кількість шахрайських операцій, які модель не змогла виявити.

- True Negatives (TN): 112444— кількість звичайних транзакцій, правильно ідентифікованих моделлю.

Додаткові метрики для оцінки моделі:

- LogLoss Score: 0.17792345632452035 — низький рівень втрат, що свідчить про якісну оцінку ймовірностей.
- AUC Score: 0.8389289495808104 — висока площа під ROC-кривою, що демонструє хорошу здатність моделі відокремлювати шахрайські транзакції від нормальних.
- Balanced Accuracy: 0.7640872190183535 — збалансована точність для дисбалансованих даних.

Ці показники свідчать про ефективність моделі в умовах реальних даних.

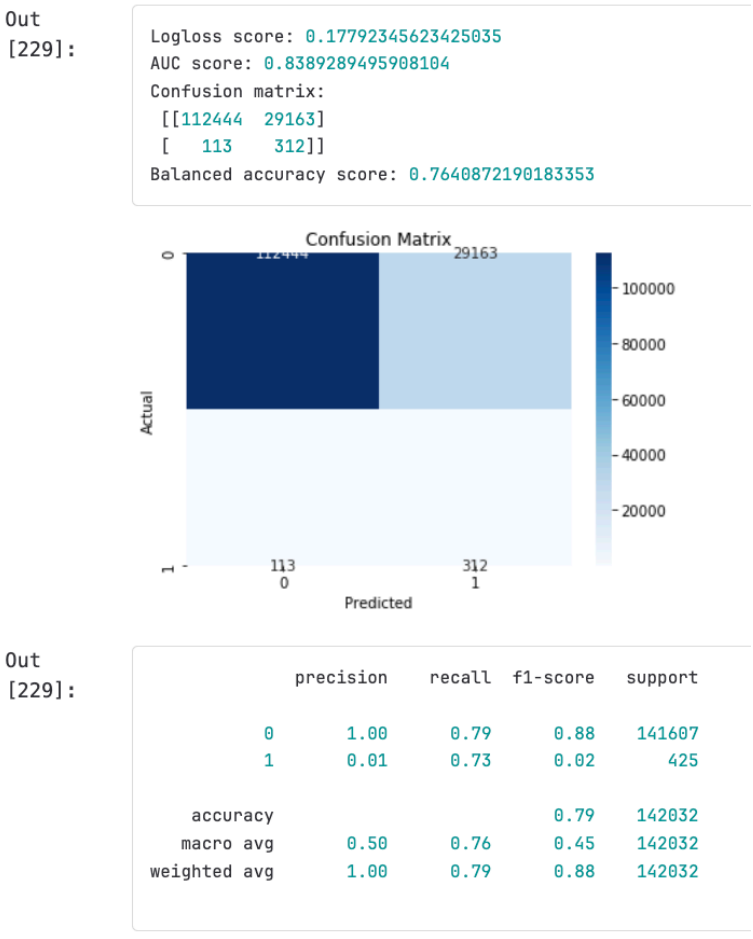


Рис. 4.4. Матриця помилок для тестового набору даних

На рисунку 4.5 представлено таблицю з аналізом різних значень порогів (threshold) для прийняття рішення щодо виявлення шахрайства. Поріг впливає на баланс між виявленими та невиявленими шахрайськими операціями, а також на частку некоректно відхилених транзакцій.

	Threshold	Detected fraud %	Undetected fraud	Unsend SMS %
0	1.00	0.00	425	100.00
1	0.95	0.00	425	100.00
2	0.90	0.00	425	100.00
3	0.85	0.00	425	100.00
4	0.80	0.24	424	99.99
5	0.75	0.94	421	99.96
6	0.70	3.76	409	99.92
7	0.65	11.76	375	99.77
8	0.60	25.41	317	99.40
9	0.55	30.59	295	98.71
10	0.50	37.41	266	97.69
11	0.45	43.06	242	95.99
12	0.40	50.12	212	93.40
13	0.35	58.12	178	89.76
14	0.30	65.18	148	85.15
15	0.25	73.41	113	79.41
16	0.20	79.29	88	71.95
17	0.15	83.76	69	60.46
18	0.10	88.71	48	46.43
19	0.05	99.29	3	29.56

Рис. 4.5. Таблиця з аналізом різних значень порогів

Вплив зміни порогу:

- При порогах ≥ 0.8 модель має високу точність, але не здатна виявити більшість шахрайських операцій.
- При порогах 0.45 - 0.25 модель демонструє оптимальний баланс між виявленням шахрайства та кількістю помилкових спрацьовувань. Наприклад:

- Поріг 0.25: 73.41% виявленого шахрайства при 79.41% некоректно відхилених транзакцій.

- Поріг 0.45: 43.06% виявленого шахрайства при 95.99% коректно відхилених операцій.

Таким чином, оптимальним для бізнес-процесу був обраний поріг 0.25, який забезпечує високу повноту виявлення шахрайства.

На рисунку 4.6 показано графік SHAP-значень, який відображає вплив кожної ознаки на результати моделі. SHAP дозволяє проаналізувати, як зміна значення конкретної ознаки впливає на ймовірність класифікації транзакції як шахрайської.

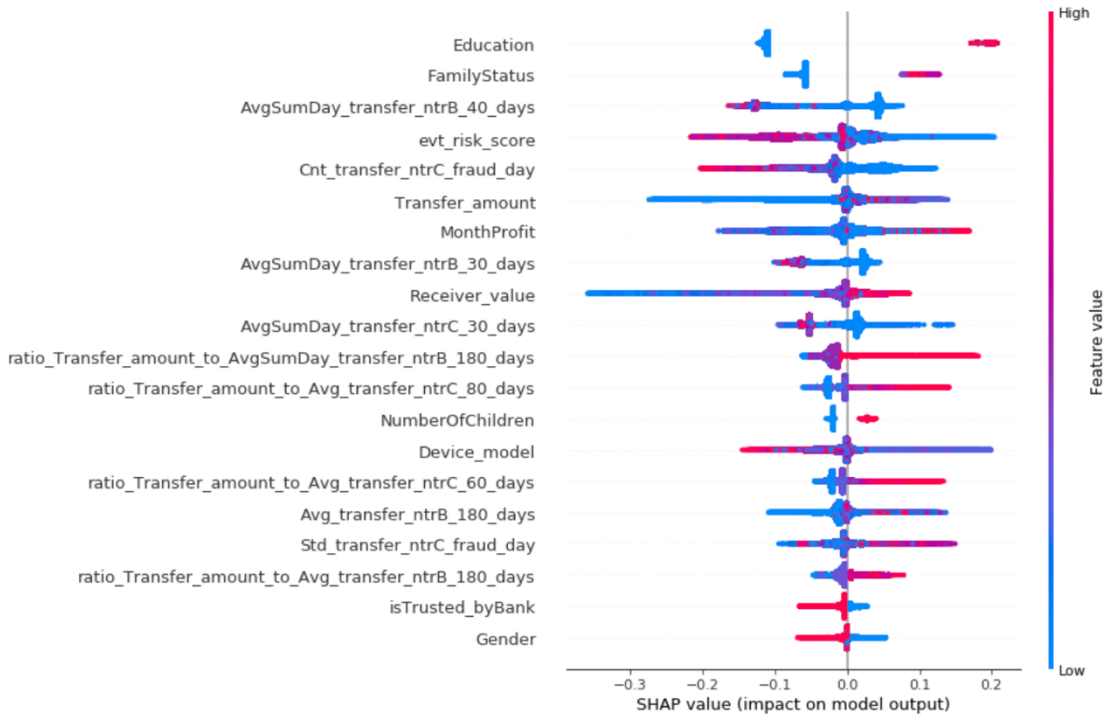


Рис. 4.6. Графік SHAP-значень

Найбільш значущі ознаки:

- AvgSumDay_transfer_ntrB_40_days — середня сума транзакцій за останні 40 днів. Підвищення значення цієї ознаки збільшує ймовірність шахрайства.

- evt_risk_score — оцінка ризику події. Високі значення вказують на підозрілі операції.
- Cnt_transfer_ntrC_fraud_day — кількість операцій у день, коли було зафіксовано шахрайство.
- Transfer_amount — сума поточної транзакції. Високі суми операцій часто сигналізують про потенційне шахрайство.
- MonthProfit — місячний дохід клієнта. Значні відхилення від звичної поведінки викликають підозру.

Таким чином, SHAP-аналіз надає інтерпретованість моделі та допомагає бізнесу зрозуміти ключові фактори ризику.

На рисунках 4.7 та 4.8 представлені профілі часткових залежностей (PDP) для двох важливих ознак:

1. ratio_Transfer_amount_to_Avg_transfer_ntrB_180_days:

   Із збільшенням співвідношення суми транзакції до середньої суми за 180 днів спостерігається поступове підвищення ймовірності шахрайства. Це свідчить про те, що аномально великі платежі можуть бути індикатором ризику.

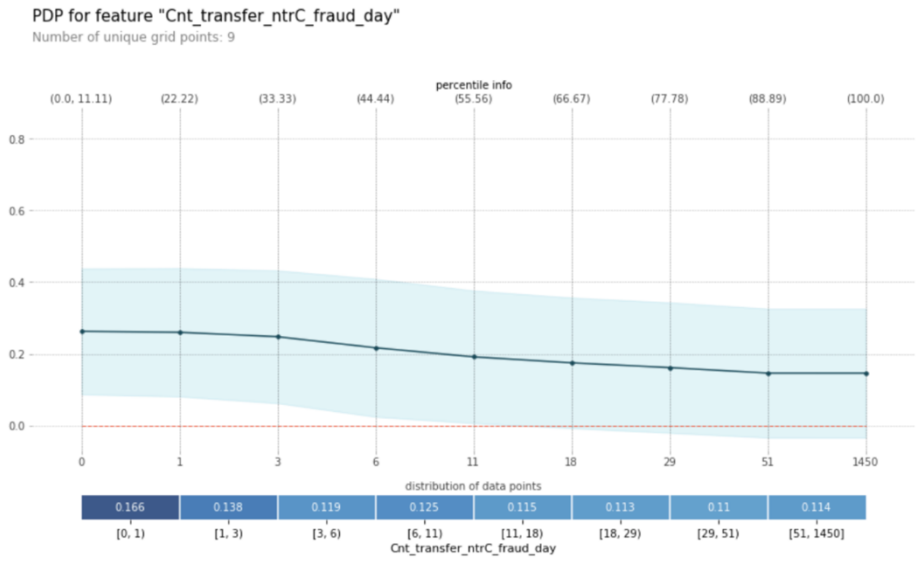


Рис. 4.7. Профіль часткової залежності (PDP) для ознаки ratio_Transfer_amount_to_Avg_transfer_ntrB_180_days

2. Cnt_transfer_ntrC_fraud_day:

   Чим більше кількість операцій у день можливого шахрайства, тим нижче ризик. Це може пояснюватися тим, що шахрайські операції зазвичай поодинокі, а аномальна активність вказує на звичайну поведінку клієнта.

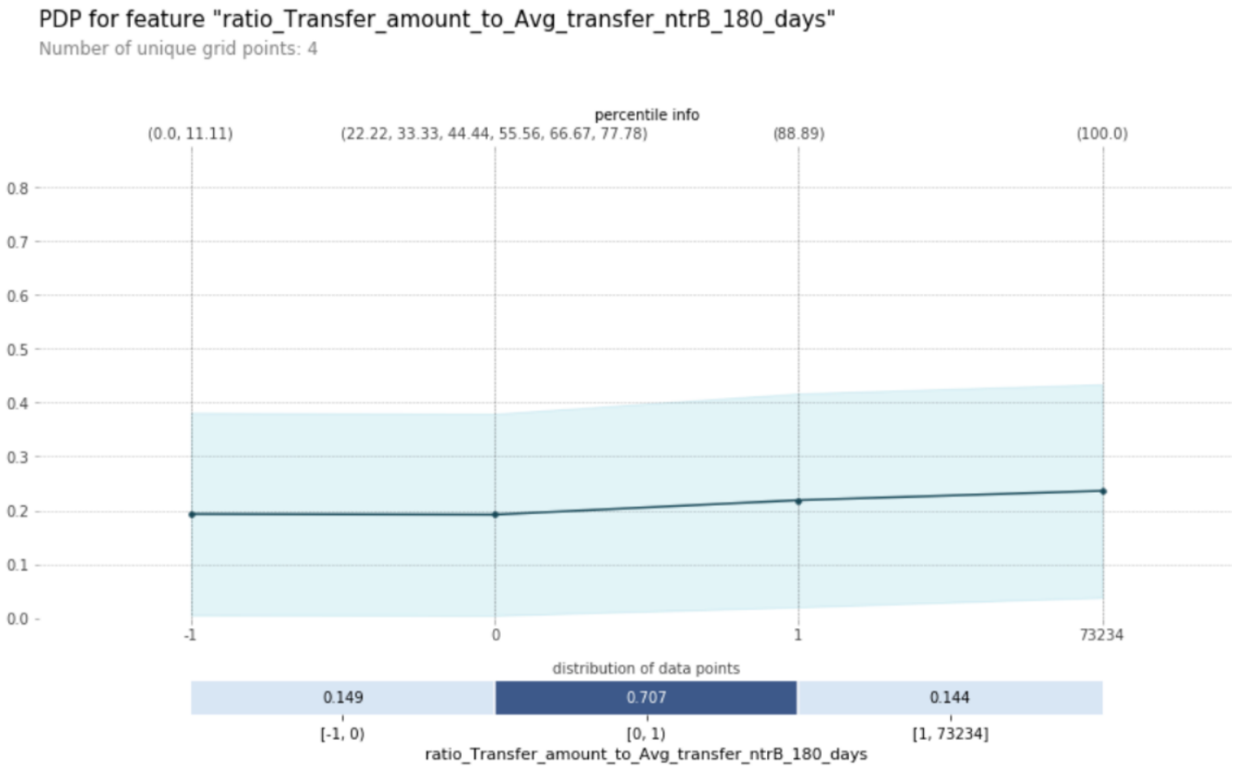


Рис. 4.8. Профіль часткової залежності (PDP) для ознаки Cnt_transfer_ntrC_fraud_day

5 ІНТЕГРАЦІЯ РЕЗУЛЬТАТІВ У СИСТЕМУ STRONGHOLD

5.1 Розробка сценаріїв для аналізу транзакцій на платформі Stronghold

На даному етапі була виконана розробка логік для виявлення шахрайських транзакцій з використанням найбільш значущих ознак, які були відібрані під час навчання моделі машинного навчання. Для тестування сценаріїв використовувався тестовий набір даних із реалістичними характеристиками транзакцій.

Відбір найкращих фічей

Для побудови логіки виявлення шахрайства було використано дві ключові ознаки, які показали найбільший вплив на ухвалення рішень моделі:

- Cnt_transfer_ntrC_fraud_day – кількість підозрілих операцій протягом одного дня.
- ratio_Transfer_amount_to_Avg_transfer_ntrB_180_days – співвідношення поточної суми транзакції до середнього значення за попередні 180 днів.

На рисунку 5.1 представлено реалізацію логіки для класифікації транзакцій як підозрілих.

```
df['predict'] = (
                  (df['Cnt_transfer_ntrC_fraud_day'] < 18) &
#                   (df['ratio_Transfer_amount_to_AvgSumDay_transfer_ntrB_80_days'] > 1) &
#                   (df['ratio_Transfer_amount_to_Avg_transfer_ntrC_80_days'] > 1) &
#                   (df['ratio_Transfer_amount_to_Avg_transfer_ntrC_60_days'] > 3) &
                  (df['ratio_Transfer_amount_to_Avg_transfer_ntrB_180_days'] > 5)
).astype(int)
```

Рис. 5.1. Реалізація логіки виявлення шахрайських операцій з використанням ключових ознак

Умови для класифікації:

- Якщо кількість транзакцій у день менша за 18, це вказує на обмежену активність клієнта, що може бути пов’язано зі спробами шахрайства.
- Якщо співвідношення суми операції до середнього значення за 180 днів перевищує 5, це свідчить про аномальну поведінку у фінансових операціях.

Розроблена логіка була застосована до тестового набору даних для оцінки результатів. На рисунку 5.2 представлено матрицю помилок, яка показує розподіл правильних та помилкових прогнозів.

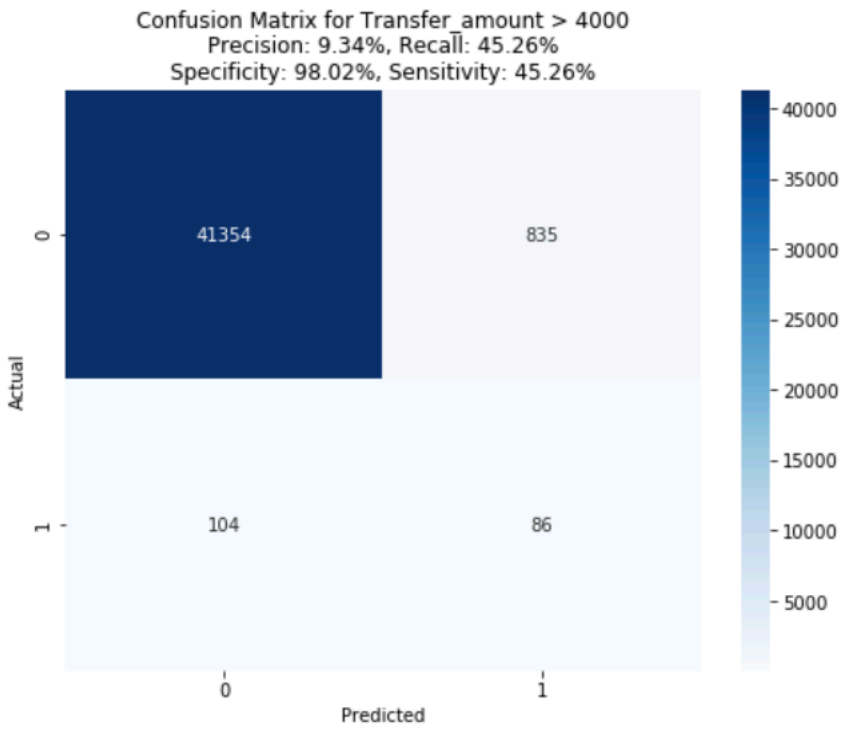


Рис. 5.2. Матриця помилок для розробленої логіки класифікації при умові Transfer_amount > 4000

Основні метрики продуктивності:

- Precision: 9.34% – точність класифікації підозрілих операцій.
- Recall (Sensitivity): 45.26% – частка виявлених шахрайських операцій від загальної кількості.
- Specificity: 98.02% – частка звичайних транзакцій, які були правильно визначені.

Результати матриці помилок:

- True Positives (TP): 86 – правильне визначення шахрайських операцій.
- False Positives (FP): 835 – хибне визначення звичайних операцій як шахрайських.
- True Negatives (TN): 41354 – правильне визначення звичайних транзакцій.
- False Negatives (FN): 104 – шахрайські транзакції, які не були виявлені.

Висновки з матриці помилок:

- Загальна кількість хибнопозитивних спрацювань (помилково класифікованих нормальних транзакцій як шахрайських) є незначною – лише 835 випадків із понад 41 тис. нормальних транзакцій. Це вказує на високу специфічність логіки, що є важливим для уникнення зайвих перевірок.
- У той же час, модель правильно виявляє 45% усіх шахрайських операцій (86 зі 190 випадків). Даний рівень Recall є прийнятним на початковому етапі, проте вимагає подальшої оптимізації для підвищення ефективності детекції.

Таким чином, логіка демонструє хороший баланс між кількістю хибних спрацювань та здатністю виявляти фродові транзакції, що дозволяє використовувати її як базовий сценарій для подальших поліпшень та інтеграції у систему виявлення шахрайства.

5.2 Перенесення ключових ознак і статистик у систему Stronghold

Для інтеграції найкращих фічей у систему Stronghold було здійснено кілька кроків, які включають створення статистичних параметрів та побудову бізнес-правил для автоматизованого виявлення шахрайських операцій.

На першому етапі було створено дві ключові статистики, необхідні для обчислення фічей:

1. Кількість транзакцій клієнта за 1 день (Count_transfer_ntrC_1_day)
   - Мета: Обчислення кількості операцій для конкретного клієнта за останню добу.
   - Параметри:
     - Event class: Payment card transaction

- Function: COUNT
- Instance attributes: Client ID
- Sliding window: Довжина – 1 день, з нульовим зміщенням.

- Реалізація представлена на рисунку 5.3.

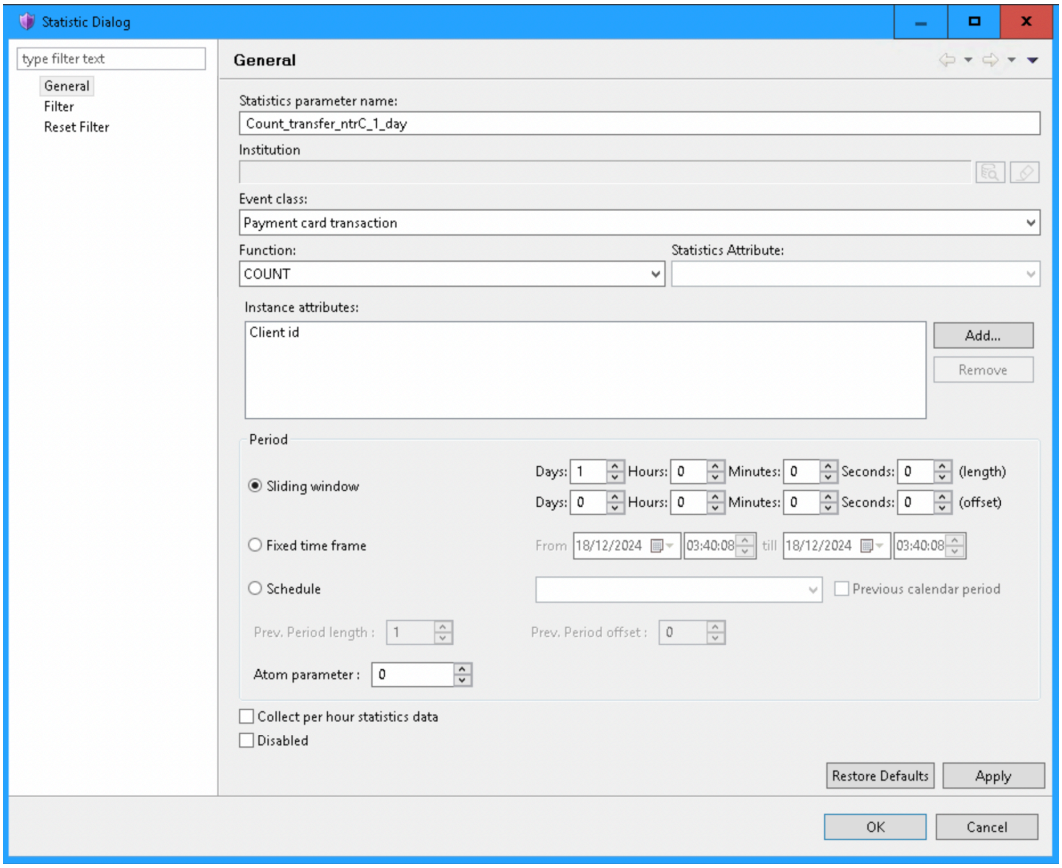


Рис. 5.3. Створення статистики Count_transfer_ntrC_1_day

2. Середня сума транзакцій за останні 180 днів (Avg_transfer_ntrB_180_days)
   - Мета: Розрахунок середньої суми операцій для клієнта за тривалий період у 180 днів.
   - Параметри:
     - Event class: Payment card transaction
     - Function: AVERAGE
     - Instance attributes: Client ID
     - Sliding window: Довжина – 180 днів, з денним інтервалом оновлення.

- Реалізація показана на рисунку 5.4.

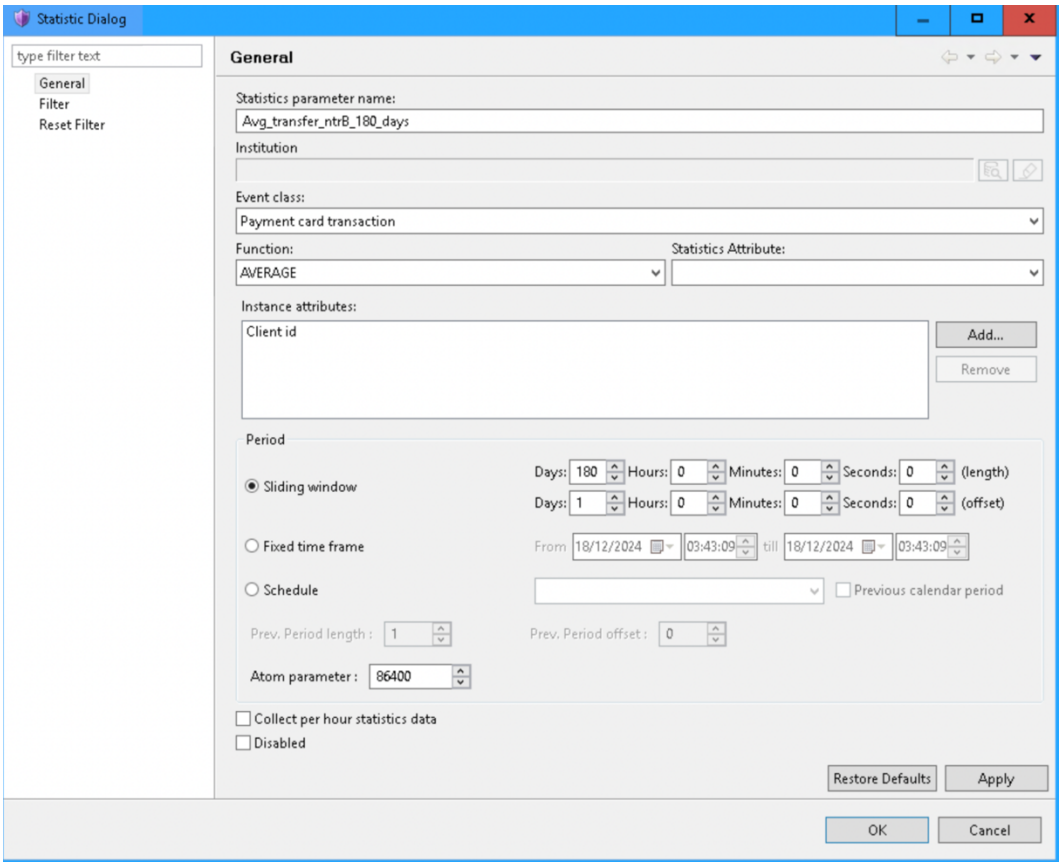


Рис. 5.4. Створення статистики Avg_transfer_ntrB_180_days

Наступним етапом було формування конкретного бізнес-правила, яке комбінує ключові статистичні параметри для аналізу транзакцій у реальному часі.

Логіка правила:

1. Payment card amount > 400000 – Визначає транзакції з великою сумою переказу.
2. Payment card amount / Avg_transfer_ntrB_180_days > 5 – Співвідношення суми переказу до середнього значення за 180 днів, що сигналізує про аномальну активність.
3. Count_transfer_ntrC_1_day > 0 – Наявність хоча б однієї транзакції за останню добу.

4. Avg_transfer_ntrB_180_days > 0 – Підтвердження, що є середнє значення за останні 180 днів.
5. Count_transfer_ntrC_1_day < 18 – Обмеження кількості транзакцій у добу до порогового значення.

На рисунку 5.5 показано реалізацію цього правила у системі Stronghold.

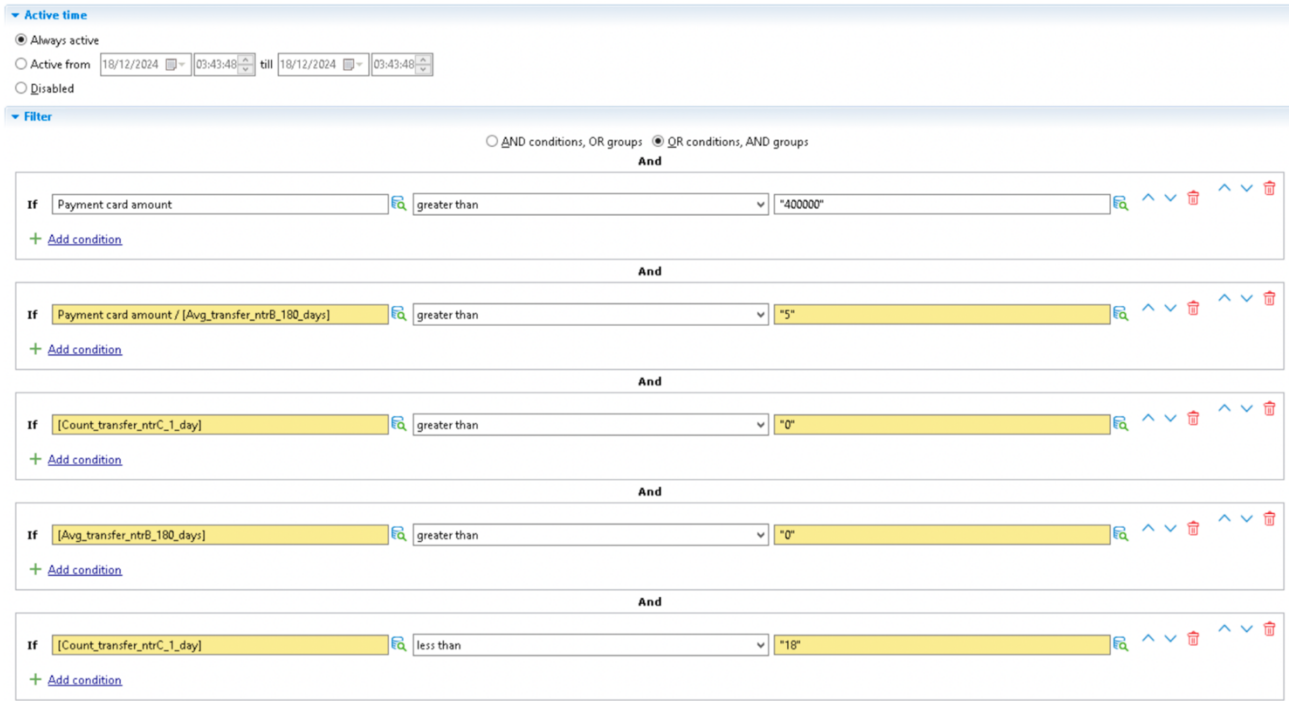


Рис. 5.5. Реалізація бізнес-правила для блокування підозрілих транзакцій

Дії при спрацюванні:

- Відмова в транзакції: Якщо умови правила виконуються, переказ блокується автоматично.
- Зворотний дзвінок клієнту: Ініціюється додаткова перевірка через дзвінок для підтвердження транзакції.

Перенесення ключових статистик і ознак у систему Stronghold відкриває нові можливості для автоматизації та значного вдосконалення процесу виявлення потенційного шахрайства в реальному часі. Цей підхід дозволяє

зібрати й обробити величезні обсяги даних, на основі яких можна оперативно виявляти навіть найменші відхилення від нормального поведінкового шаблону клієнта. Запропоноване правило для налаштування системи гарантує оптимальний баланс між чутливістю до аномалій, що дозволяє своєчасно виявляти шахрайські операції, та мінімізацією хибних спрацювань, коли легітимні транзакції помилково вважаються підозрілими. Це є критично важливим для забезпечення високої ефективності бізнес-процесів, оскільки зменшує ризики втрати доходів через неправомірні відхилення, а також підтримує довіру клієнтів до фінансової установи. Завдяки такому підходу банк або фінансова організація може не тільки підвищити рівень безпеки своїх послуг, а й зберегти репутацію на високому рівні, що є ключовим фактором для успішного ведення бізнесу в умовах постійної загрози шахрайства.

6 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

6.1 Постановка експерименту та обґрунтування вибору даних

Мета експерименту

Метою експерименту є перевірка працездатності розробленого правила для виявлення підозрілих транзакцій на основі ідентифікованих патернів шахрайства. Експеримент спрямований на оцінку ефективності бізнес-правила, що базується на ключових ознаках, та демонстрацію його можливостей для реального впровадження у систему.

Постановка завдання

Потрібно протестувати бізнес-правило шляхом ініціювання тестової події з ознаками шахрайської поведінки у середовищі Stronghold. Основне завдання полягає у перевірці:

1. Спрацювання правила при наявності фродових ознак у тестовій транзакції.
2. Виконання запрограмованих дій: блокування переказу та ініціювання зворотного дзвінка клієнту.
3. Логування всіх ключових етапів обробки події для подальшого аналізу.

Аргументація вибору даних

Дані для тестування були підібрані на основі попереднього аналізу та побудови фічей, які найкраще виявляють шахрайські патерни.

Ключові особливості постановки експерименту

1. Симуляція реальних фродових патернів:

- Використання статистичних фічей, які вже продемонстрували високу важливість у побудові моделі машинного навчання.

2. Перевірка на бізнес-правилах:
    - Замість складних ML-моделей перевірка базується на зрозумілих правилах, що легко інтегруються у реальні бізнес-процеси.
3. Автоматизовані дії:
    - Тестування включає автоматичну блокаду операції та запуск механізму зворотного дзвінка клієнту як превентивний захід.
4. Відтворюваність:
    - Тестова подія може бути використана для повторної перевірки у різних середовищах для валідації результатів.

Очікувані результати

1. Спрацювання бізнес-правила
    - Тестова подія буде ідентифікована як підозріла завдяки заданим умовам.
2. Виконання дій системи
    - Система блокуватиме транзакцію та ініціюватиме зворотний дзвінок для підтвердження операції.

6.2 Аналіз роботи статистик і правила в системі Stronghold

На цьому етапі було створено випадкову подію з ознаками, які характерні для шахрайських транзакцій. Ключовим параметром події є Payment card amount, значення якого становить 100000 (Рисунок 6.1).

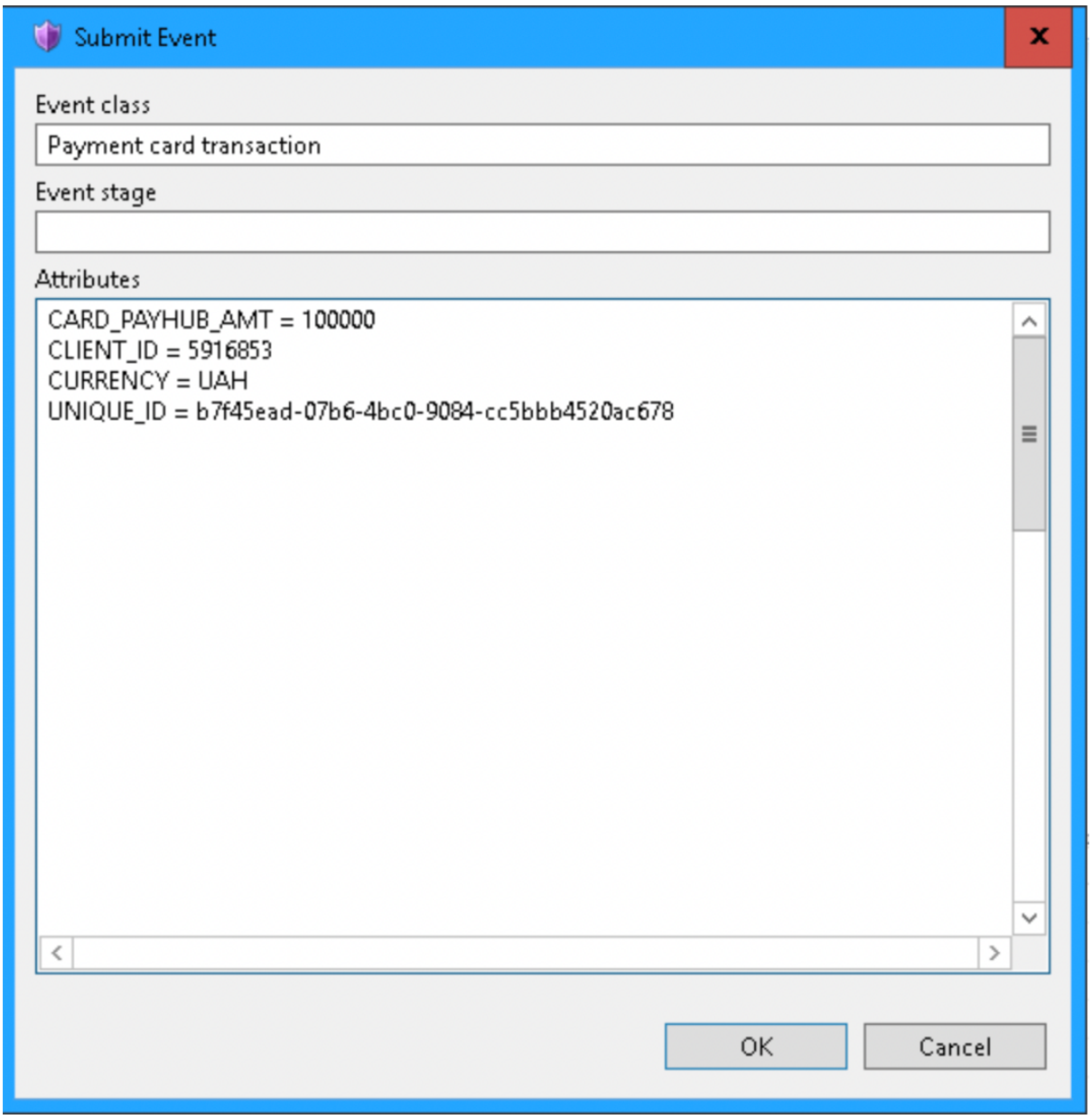


Рис. 6.1. Генерація тестової події з параметрами для перевірки правила.

Подія включає такі атрибути:

- CARD_PAYHUB_AMT — сума переказу;
- CLIENT_ID — унікальний ідентифікатор клієнта;
- CURRENCY — валюта операції;
- UNIQUE_ID — унікальний ідентифікатор транзакції

Важливість таких значень пояснюється тим, що вони дозволяють симулювати конкретний патерн, який система машинного навчання раніше ідентифікувала як потенційний ризик шахрайства.

На основі заздалегідь підготовлених статистик у системі Stronghold було активовано правило ML_interpretation_test_RED. Це правило включає кілька умов, які ґрунтуються на розрахунках у попередніх статистиках:

- Висока сума переказу;
- Відхилення суми переказу від середнього значення Avg_transfer_ntrB_180_days;
- Кількість переказів у короткий період Count_transfer_ntrC_1_day.

Усі умови були налаштовані для оцінки, чи відповідає подія патерну, характерному для шахрайських операцій.

Після генерації події та її обробки системою Stronghold у логах було зафіксовано, що:

1. Правило спрацювало успішно, вказуючи на високий ризик шахрайства.
2. Дії, пов'язані зі спрацюванням правила, включають маркування події (Рисунок 6.2.) та запуск процесу реагування.

Це підтверджує ефективність попередньої оцінки моделі машинного навчання, яка виявила закономірності у даних. Правило дозволяє оперативно реагувати на потенційно шахрайські операції, забезпечуючи додатковий рівень захисту клієнтів.

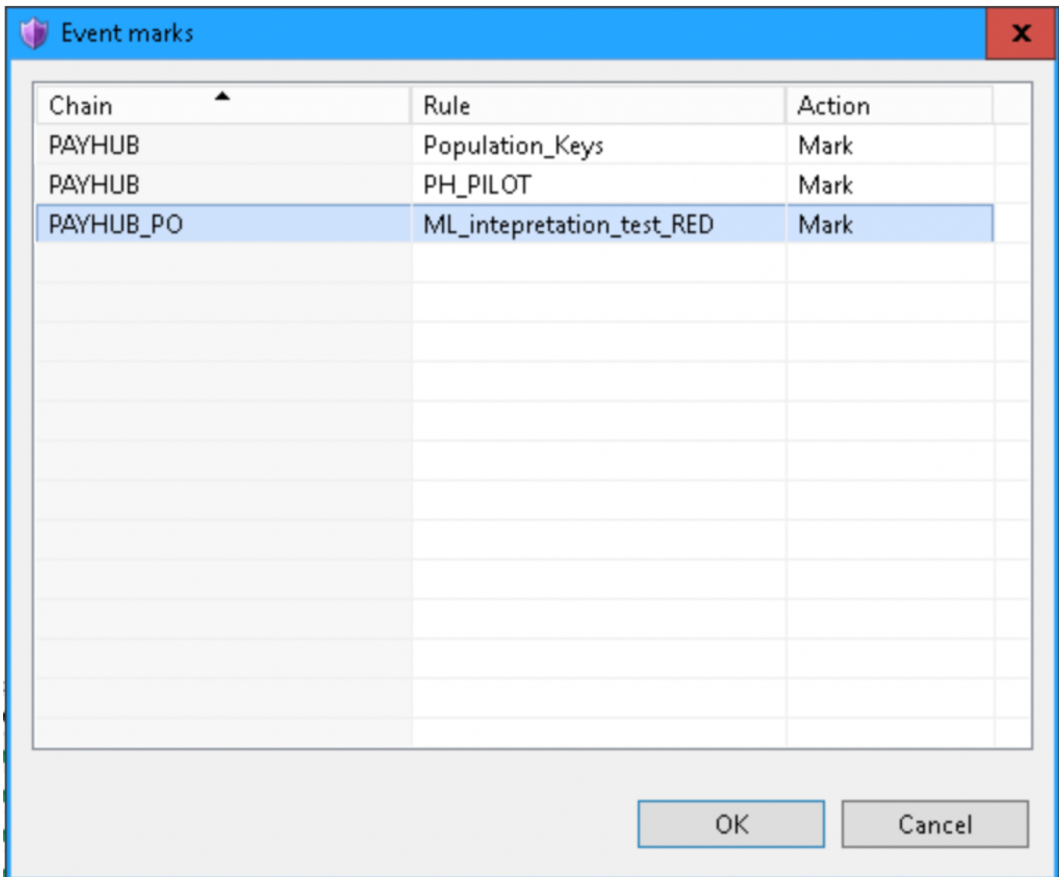


Рис. 6.2. Спрацювання правила у системі Stronghold для виявлення шахрайства.

ВИСНОВКИ

1. Розроблено модель для виявлення шахрайських транзакцій. Модель на основі Balanced Random Forest продемонструвала ефективну роботу, досягаючи Recall на рівні 45% для шахрайських транзакцій при мінімальній кількості хибних спрацювань. Це свідчить про здатність моделі виділяти ризикові операції серед великого обсягу даних.
2. Оптимізація гіперпараметрів дозволила досягти AUC ROC = 84% і мінімізувати Log Loss до 0.17, що підтверджує якісну роботу моделі на етапі тестування.
3. Використання методів SHAP та PDP дозволило виділити найзначущі ознаки для моделі: Cnt_transfer_ntrC_fraud_day, ratio_Transfer_amount_to_Avg_transfer_ntrB_180_days. Це дало змогу підвищити інтерпретованість рішень моделі та зрозуміти вплив окремих фічей на кінцевий результат.
4. Було створено та перенесено ключові статистики й правила до платформи Stronghold. Сформульоване правило на основі розроблених ознак успішно блокує підозрілі транзакції та активує механізм зворотного дзвінка клієнту для підтвердження операції.
5. Ефективність підтверджена експериментальним дослідженням. Згенерована тестова подія з патернами шахрайства активувала створене правило в системі Stronghold, що підтвердило ефективність моделі та інтегрованих логік у реальних умовах.
6. Впровадження моделі дозволяє оперативно виявляти ризикові транзакції, знижуючи ймовірність шахрайства та фінансових втрат для клієнтів і установи. Це також підвищує швидкість та якість прийняття рішень у бізнес-процесах.

7. Розроблена система враховує можливість обробки великих обсягів даних та адаптації до нових сценаріїв шахрайства, що забезпечує її актуальність для довгострокового використання.

ПЕРЕЛІК ПОСИЛАНЬ

1. Zhang Q., Lu J., Zhang G. Recommender Systems in E-learning. Journal of Smart Environments and Green Computing. 2022. No.1(2). P. 76-89.
2. Patel D., Patel F., Chauhan U. Recommendation Systems: Types, Applications, and Challenges. International Journal of Computing and Digital Systems. 2023. Vol. 13, no. 1. P. 851-868. URL: https://doi.org/10.12785/ijcds/130168
3. Cherednichenko O.Y., Ivashchenko O.V., Gontar Y.M., Vorona B.M. Інтелектуальний аналіз пропозицій товарів на основі контекстних рекомендацій. Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. 2021. №1320 (44). С. 57-66. https://doi.org/10.20998/2079-0023.2018.44.10
4. Bhatia R., Arora D. Fraud Detection in Banking Using Machine Learning: A Review. International Journal of Computer Applications. 2021. Vol. 174, no. 19. P. 21-27.
5. Bohdan Khudik. Analysis of the approaches to the development of the multiagent recommender system based on the fuzzy logic. ITU Workshop for Europe and CIS Regions. ICT infrastructure as a basis for digital economy. 14-16 May 2019 / International Telecommunication Union, State University of Telecommunications. P. 56.
6. Jha S., Guha S., Dhar A. A Comparative Study on Fraud Detection Techniques in Banking. IEEE Xplore Conference Proceedings. 2022. P. 11-16.
7. Bishop C.M. Pattern Recognition and Machine Learning. New York: Springer, 2016. 738 p.
8. Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems. 2012. Vol. 25, P. 1097–1105.

9. Ермолаєва О.В., Петренко М.В. Застосування машинного навчання для аналізу поведінки користувачів у банківських системах. Системний аналіз і прикладна інформатика. 2023. №2. С. 33-41.
10. Бондаренко В.В., Гаманюк І.М. Розробка профайлеру продуктивності SQL-запитів за допомогою стрес-тестування. Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інфокомунікаційних технологіях», 20 квітня 2023 р., Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2023. С. 98-99.
11. Sun L., Liu Y., Yu H. Deep Learning for Fraud Detection in Financial Transactions. Applied Intelligence. 2021. Vol. 51, no. 4. P. 2755–2769.
12. McKinsey & Company. AI in Banking: Improving Fraud Detection and Risk Management. 2022. URL: https://www.mckinsey.com/business-functions/risk-and-resilience
13. Berman J., Howard J. An Introduction to Machine Learning in Finance. Financial Analysts Journal. 2023. Vol. 79, no. 2. P. 33–45.
14. Khandani A., Kim A., Lo A.W. Consumer credit-risk models via machine-learning algorithms. Journal of Banking & Finance. 2017. Vol. 34, no. 11. P. 2767–2787.
15. APA Style Introduction. Purdue University. URL: https://owl.purdue.edu/owl/research_and_citation/apa_style/apa_style_introduction.html (дата звернення: 09.06.2024).
16. Настанова до зводу Знань з управління проєктами. Настанова PMBOK. 7-е видання та стандарт з управління проєктами. Інститут Управління Проєктами. URL: https://pmiukraine.org/pmbok7 (дата звернення: 20.02.2024).
17. Bohdana Muzyka. User Flows. Як ця техніка допомагає в роботі над проєктами. 29.12.2020. URL: https://dou.ua/lenta/columns/user-flows/ (дата звернення: 13.04.2024).

18. Pedregosa F., Varoquaux G., Gramfort A. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2018. Vol. 12, P. 2825-2830.
19. Економічна енциклопедія / за ред. В.В. Шевченка. Київ: Альманах, 2016. 304 с.
20. Bishop C.M., Nasrabadi N.M. Pattern Recognition and Machine Learning. Berlin: Springer-Verlag, 2019. 803 p.

ДОДАТОК А ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

МАГІСТЕРСЬКА РОБОТА

МЕТОДИ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ЗАХИСТУ БАНКІВСЬКИХ ПРОДУКТІВ ВІД ЗОВНІШНЬОГО ШАХРАЙСТВА

Виконала: студентка групи САДМ-61 Степанова Анастасія Володимирівна

Керівник: к.е.н., доцент кафедри Патракеєв Ігор Михайлович

Київ - 2025

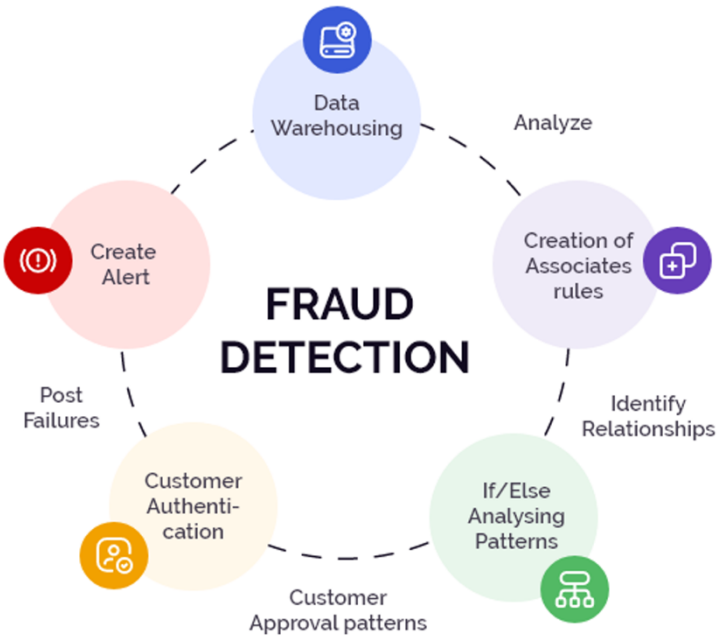
МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: підвищення ефективності виявлення фінансового шахрайства у банківській сфері за допомогою машинного навчання та автоматизації процесів на базі платформи Stronghold.

Об'єкт дослідження: процес виявлення шахрайських транзакцій у банківських системах.

Предмет дослідження: Методи машинного навчання для виявлення фінансового шахрайства та їх впровадження у фінансові процеси.

ТРАДИЦІЙНА МОДЕЛЬ БОРОТЬБИ З ШАХРАЙСТВОМ У БАНКІВСЬКІЙ СФЕРІ



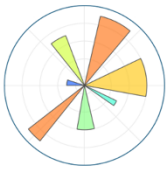
ВИКОРИСТАНІ ІНСТРУМЕНТИ

1.Програмне забезпечення для моделювання :

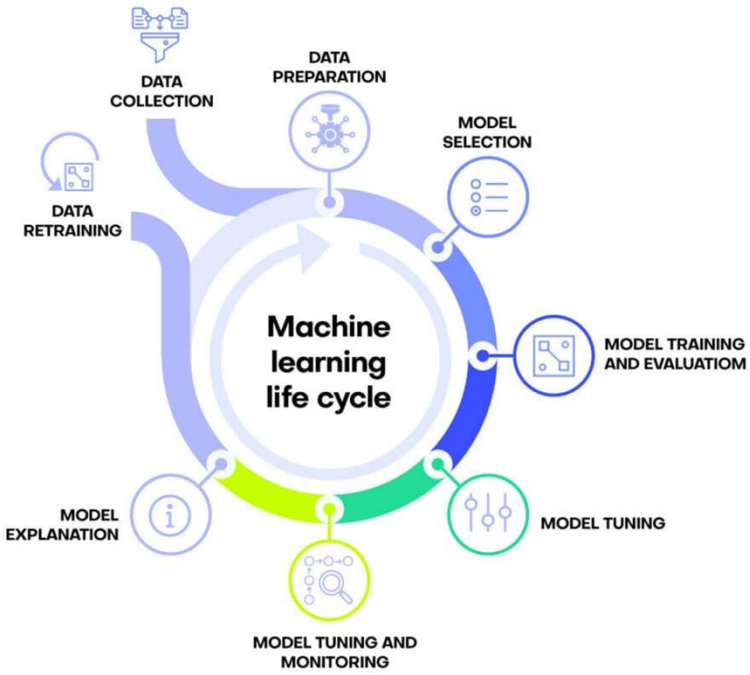
1. **Python**
2. **Бібліотеки**:
    1. Scikit-learn
    2. Seaborn та Matplotlib
    3. Pandas та NumPy

2.Програмне забезпечення для інтеграції:

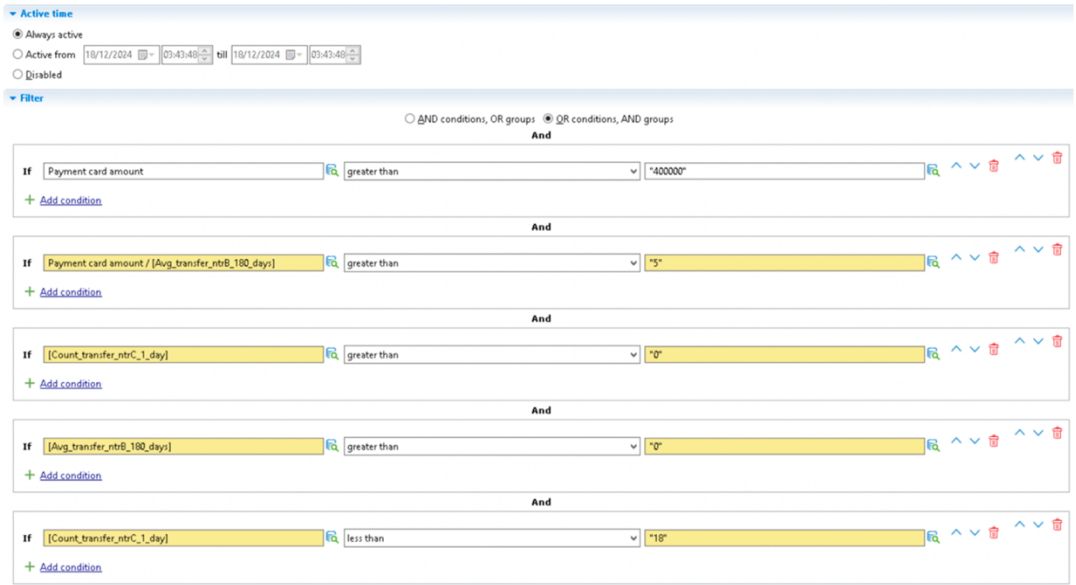
1. **Stronghold**:
    1. Створено процес обробки транзакцій у реальному часі.
    2. Інтегровано виклик ML-моделі



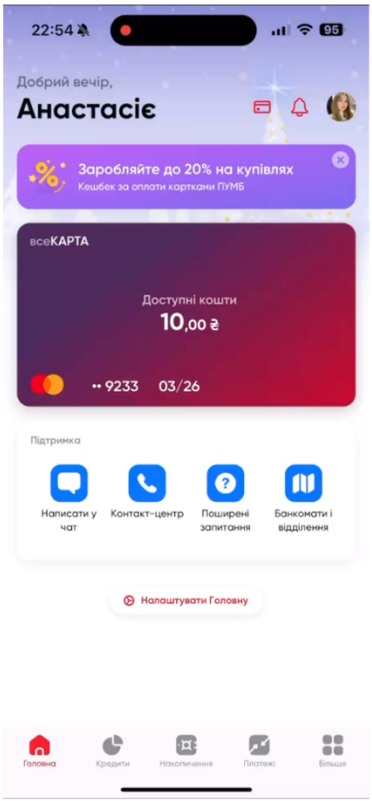
ЕТАПИ СТВОРЕННЯ МОДЕЛІ МАШИННОГО НАВЧАННЯ



ІНТЕРПРЕТАЦІЯ РЕЗУЛЬТАТІВ У ПЛАТФОРМУ STRONGHOLD

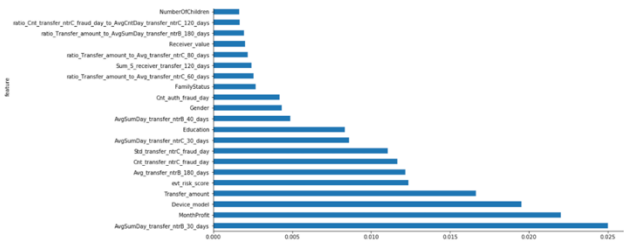


ПРАКТИЧНИЙ РЕЗУЛЬТАТ

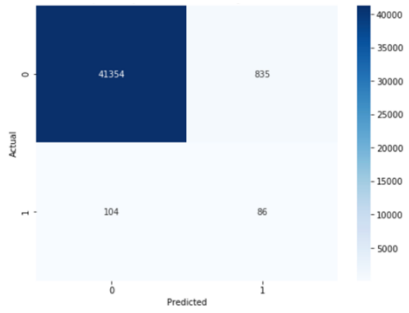


ЕКРАННІ ФОРМИ

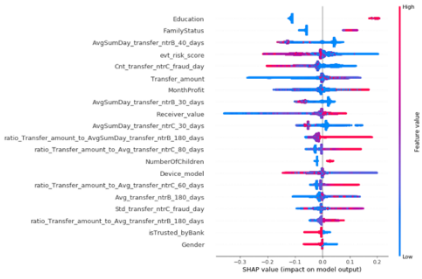
Графік важливості ознак
(Feature Importance графік)



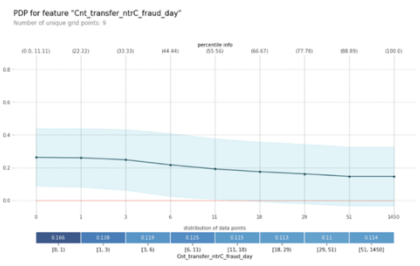
Матриця плутанини (Confusion Matrix):
Оцінка точності класифікації



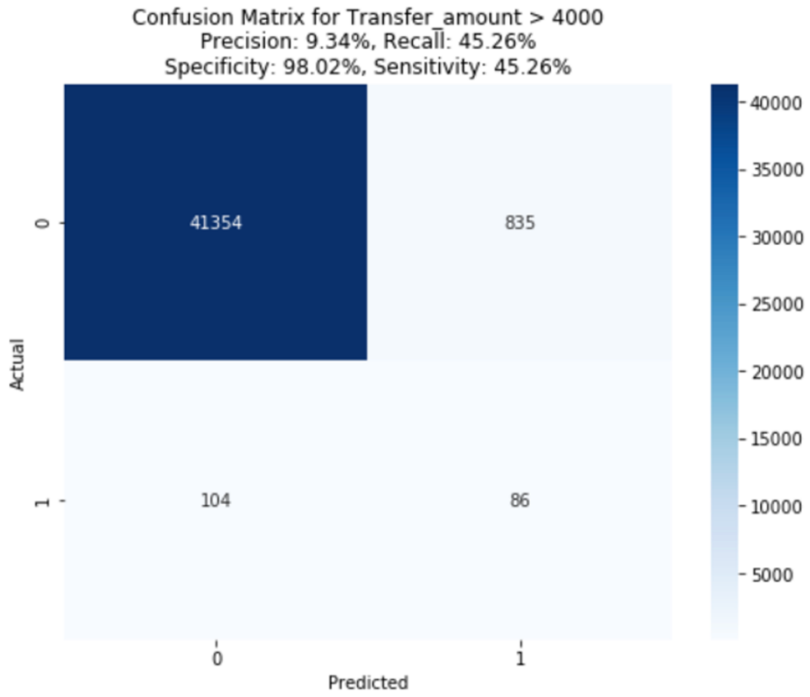
Вплив кожної ознаки на прогноз моделі
(Графік SHAP-значень)



Залежність прогнозованого результату моделі
від змінної ознаки
(Графік PDP)



РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ: АНАЛІЗ ЕФЕКТИВНОСТІ МОДЕЛІ



ВИСНОВКИ

1. Розроблена модель для виявлення шахрайських транзакцій на основі Balanced Random Forest продемонструвала ефективність, досягнувши Recall 45% при мінімальних хибних спрацюваннях.
2. Оптимізація гіперпараметрів покращила AUC ROC до 84% та знизила Log Loss до 0.17, що підтверджує високу якість моделі.
3. Використання SHAP та PDP дозволило виділити ключові ознаки, підвищивши інтерпретованість моделі.
4. Інтеграція з платформою Stronghold забезпечила успішне блокування підозрілих транзакцій.