МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет: **ННІ Каразінський банківський інститут**
Кафедра: **Інформаційних технологій та математичного моделювання**
Спеціальність: **125 Кібербезпека**
Освітня програма: **Кібербезпека у фінансових технологіях**

Група: АБ-41Б **денна форма навчання**

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему:
«РОЗРОБКА МЕТОДУ ВИЯВЛЕННЯ ШАХРАЙСТВА В ОНЛАЙН-ТОРГІВЛІ»
ЗА НАКАЗОМ № 4601-5/335 ВІД 07 ЛЮТОГО 2025 РОКУ

здобувача вищої освіти **Андренко Катерини Віталіївни**

Робота допущена до захисту в ЕК

протокол кафедри ІТММ

Завідувач кафедри ІТММ
к. п. н., доцент
_____**Н. І. Стяглик**

Науковий керівник
к. ф.- м. н., доцент
_____**Л. Д. Філатова**

м. Харків 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В. Н. Каразіна

Факультет навчально-науковий інститут "Каразінський банківський інститут"
Кафедра інформаційних технологій та математичного моделювання
Рівень вищої освіти перший (бакалаврський)
Спеціальність 125 Кібербезпека
Освітня програма Кібербезпека у фінансових технологіях

ЗАТВЕРДЖУЮ
Завідувач кафедри

Н. І. Стяглик
“08” лютого 2025 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ (ПРОЄКТ)

Андренко Катерини Віталіївни
(прізвище, ім’я, по батькові студента)

1. Тема роботи «Розробка методу виявлення шахрайства в онлайн-торгівлі»

Керівник роботи к. ф.-м. н., доцент Л. Д. Філатова

затверджено наказом по університету від “08”лютого 2025 року № 4601-5/335

2. Строк подання студентом роботи 15 травня 2025 року

3. Перелік питань, які потрібно розробити:

У розділі 1: Проаналізувати статистику шахрайства в онлайн-торгівлі.

У розділі 2: Проаналізувати можливі ризики шахрайства в онлайн-торгівлі.

У розділі 3: Розробити метод виявлення шахрайства в онлайн-торгівлі.

4. План роботи

№ з/п	Назви етапів роботи
1	Вибір здобувачем теми кваліфікаційної магістерської роботи
2	Затвердження плану і завдання кваліфікаційної магістерської роботи
3	Здача кваліфікаційної магістерської роботи керівнику
4	Підпис кваліфікаційної магістерської роботи керівника
5	Підпис кваліфікаційної магістерської роботи у нормоконтролера
6	Допуск завідувачем кафедри до захисту кваліфікаційної магістерської роботи
7	Захист кваліфікаційної магістерської роботи

5. Дата видачі завдання 08 лютого 2025 року

Студент _____ **К. В. Андренко**
підпис ініціали, прізвище

Керівник роботи _____ **Л. Д. Філатова**
підпис ініціали, прізвище

РЕФЕРАТ
НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ
«РОЗРОБКА МЕТОДУ ВИЯВЛЕННЯ ШАХРАЙСТВА В ОНЛАЙН-ТОРГІВЛІ»
Андренко Катерини Віталіївни

Кваліфікаційна бакалаврська робота містить: 57 сторінок, 35 рисунків, список літератури з 13 найменувань.

Об’єктом дослідження є знешкодження ризиків шахрайства у процесі онлайн-торгівлі.

Предмет дослідження – Telegram-бот для виявлення шахрайства в онлайн-торгівлі.

Мета кваліфікаційної магістерської роботи полягає у розробці та дослідженні Telegram-бота, який виявляє та повідомляє про можливі шахрайські дії онлайн-магазинів, підтримуючи покупців та забезпечуючи зручний інтерфейс.

Завданнями кваліфікаційної магістерської роботи є:

У першому розділі проведено аналіз статистичних даних, пов'язаних із шахрайством в онлайн-торгівлі. Зокрема, основні типи шахрайських дій, такі як фейкові онлайн-магазини, підроблені відгуки, маніпуляції з цінами та непрозорі умови оплати. Описані джерела статистичних даних, включаючи звіти організацій із захисту прав споживачів, аналітику правоохоронних органів, дослідження ринку електронної комерції. Особливу увагу приділено аналізу ситуації на українському ринку онлайн-торгівлі в контексті глобальних тенденцій.

У другому розділі розглянуто ризики шахрайства, з якими можуть зіткнутися покупці в процесі онлайн-шопінгу. Проаналізовано основні фактори ризику, включаючи нечесні методи оплати, недостатню прозорість продавців, відсутність гарантій доставки. Особливий акцент зроблено на виявленні закономірностей у поведінці шахрайських магазинів та платформ. Також розглянуто підходи до мінімізації ризиків для покупців, через технологічні рішення, такі як чат-боти або автоматизовані системи моніторингу.

У третьому розділі детально описано розроблений метод виявлення шахрайства в онлайн-торгівлі, основою якого є Telegram-бот. Особливий акцент був зроблений на алгоритмах і механізмах перевірки магазинів, таких як реєстрація користувачів, транзакція та повідомлення про порушення.

Актуальність дослідження обумовлена зростаючою кількістю шахрайських дій в онлайн-торгівлі, що створює потребу в інноваційних інструментах для захисту прав споживачів. Розроблений Telegram-бот «БезпечнаТорговля» відповідає сучасним викликам цифрової економіки, пропонуючи зручний та ефективний спосіб виявлення потенційних ризиків та попередження користувачів.

За результатами дослідження було розроблено Telegram-бота, який забезпечує регулярне залучення користувачів, сприяє виявленню

шахрайських дій у сфері онлайн-торгівлі та допомагає покупцям уникати потенційних ризиків. Особливий акцент був зроблений на алгоритмах аналізу транзакцій, перевірці магазинів через реєстрацію користувачів та автоматизованих повідомленнях про порушення.

Практична новизна роботи у створенні інноваційного Telegram-бота «БезпечнаТорговля», який інтегрує алгоритми перевірки онлайн-магазинів, аналіз транзакцій і сповіщення користувачів про можливі ризики шахрайства. Бот забезпечує орієнтований на користувача підхід, пропонуючи персоналізацію досвіду, автоматизацію процесів перевірки та зручність у використанні.

Одержані результати включають розробку Telegram-бота, який забезпечує ефективне виявлення шахрайства в онлайн-торгівлі.

КЛЮЧОВІ СЛОВА: ШАХРАЙСТВО, БЕЗПЕКА, TELEGRAM-БОТ, ОНЛАЙН-ТОРГІВЛЯ, АЛГОРИТМИ ПЕРЕВІРКИ, РИЗИКИ, АВТОМАТИЗАЦІЯ, СПОВІЩЕННЯ, ПЕРСОНАЛІЗАЦІЯ.

ABSTRACT
FOR QUALIFICATION BACHELOR'S THESIS "DEVELOPMENT OF A METHOD FOR DETECTION OF FRAUD IN ONLINE TRADE"

Andrenko Kateryna Vitaliivna

Qualification bachelor's thesis contains: 56 pages, 35 drawings, list of references.

The object of the study is the damage of fraud risks in the process of online trading.

The subject of the study is Telegram-bot for fraud in online trading.

The meta-qualification master's thesis arises from the development and research of Telegram-bot, which shows and reports on the possibility of fraudulent actions of online stores, supporting buyers and providing a convenient interface.

The tasks of the qualification master's thesis are:

The first section analyzes statistical data related to fraud in online trading. The following are the main types of fraudulent actions, such as fake online stores, fake reviews, price manipulation and opaque payment terms. Sources of statistical data are described, including consumer reporting organizations, law enforcement analytics, and e-commerce market research. Particular attention is paid to market analysis in Ukrainian online commerce in the context of global trends.

The second section addresses the risk of fraud that buyers may encounter in the online shopping process. The main risk factors are analyzed, including dishonest payment methods, lack of transparency of sellers, and the availability of delivery guarantees. Special emphasis is placed on established patterns in the behavior of fraudulent stores and platforms. It also discusses approaches to minimizing risks for buyers through technological solutions, such as chatbots or automated monitoring systems.

The third section describes in detail the developed method for detecting fraud in online commerce, the basis of which is the Telegram bot. Special emphasis was placed on algorithms and mechanisms for checking stores, such as user registration, transactions, and violation notifications.

The relevance of the study is due to the growing number of fraudulent actions in online trading, which creates a need for innovative tools to protect consumer rights. The developed Telegram bot "BezpechnaTorgovlya" meets the modern challenges of the digital economy, offering a convenient and effective way to acquire existing risks and advance users.

According to the results of the study, a Telegram bot was developed that ensures regular user engagement, helps detect fraudulent actions in the field of online trading, and helps buyers avoid negative risks. Special emphasis was placed on transaction analysis algorithms, store verification through user registration, and automated violation notifications.

The practical novelty of the work is the creation of an innovative Telegram bot "BezpechnaTorgovlya", which integrates algorithms for checking online stores,

analyzing transactions, and supporting users with the possibility of fraud risks. The bot provides a user-centric approach, offering personalization of the experience, automation of the verification process, and ease of use.

The results include the development of a Telegram bot that provides effective fraud detection in online trading.

KEYWORDS: FRAUD, SECURITY, TELEGRAM BOT, ONLINE TRADING, VERIFICATION ALGORITHMS, RISKS, AUTOMATION, NOTIFICATIONS, PERSONALIZATIO.

ЗМІСТ

ВСТУП

У сучасному світі інформаційних технологій, стрімко зростає популярність онлайн-торгівлі, тому питання безпеки даних набули актуального значення. Шахрайство в онлайн-торгівлі, стало серйозною загрозою в умовах глобалізації та цифровізації бізнесу, саме це впливає на репутацію підприємств та довіру клієнтів. Кожного дня у соціальних мережах, новинах, газетах показують, що фінансові втрати, через шахрайство в електронній комерції, значно зросли [3;11]. Зловмисники постійно вдосконалюють методи атак, що значно збільшує ризики приватності (Рис. 1).

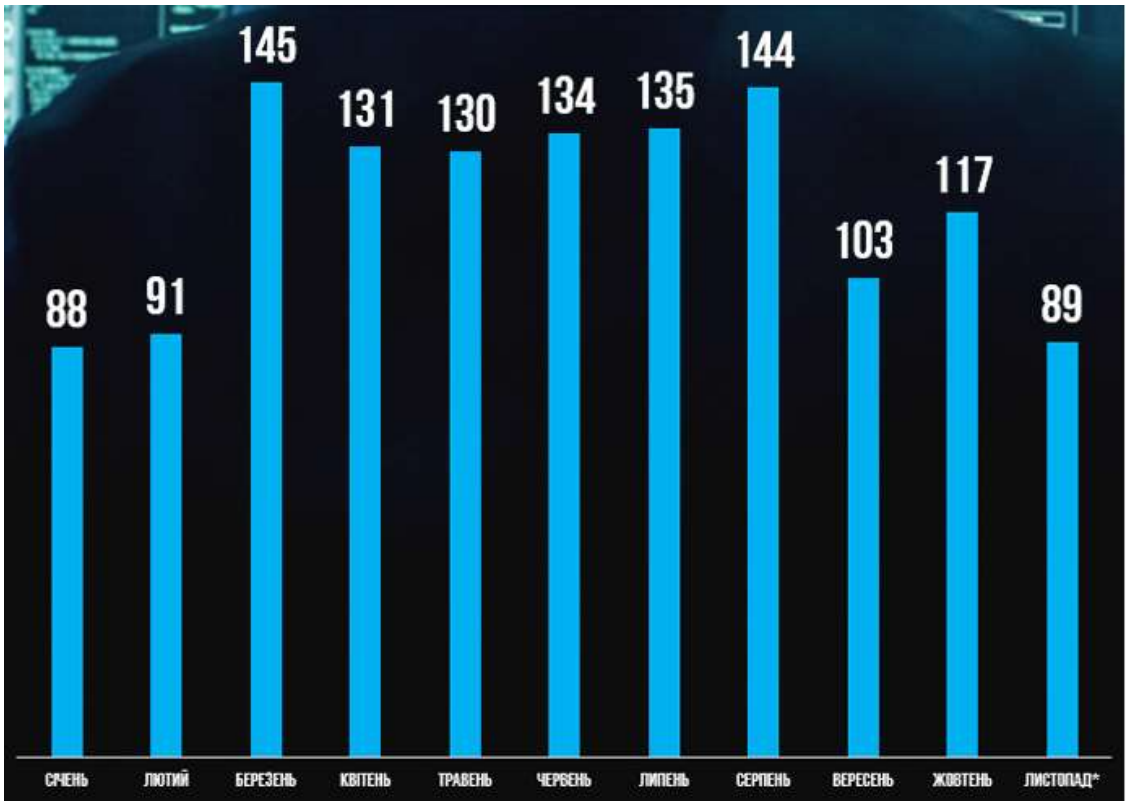

Рис. 1. «Кількість українців, які скаржилися на шахрайські схеми з початку 2024 року»

Основною причиною зростання шахрайства є недостатня обізнаність споживачів та підприємств про потенційні ризики. Саме така відсутність

заходів може стати причиною того, що шахраї дуже легко зможуть заволодіти конфіденційною інформацією та доступом до всіх даних клієнта. Тому важливо не лише помічати та аналізувати такі ризики, але й знати методи їх усунення.

У випадку онлайн-магазинів, шахрайство може проявлятися в різних формах: фішинг, несанкціонований доступ, DDos-атаки, підроблені сайти та інші способи обману.

З огляду на те, що така сфера продовжує збільшуватися - підприємствам необхідно вжити активних заходів для запобігання шахрайству. Це може бути впровадження сучасних технологій шифрування, моніторинг транзакцій у реальному часі, а також навчання співробітників та споживачів щодо безпеки в Інтернеті. Системи виявлення шахрайства повинні бути гнучкими та адаптивними до нових загроз, що з’являються на ринку.

У цьому дослідженні буде проведений аналіз ризиків шахрайства в онлайн-торгівлі, а також розроблені рекомендації щодо їх виявлення та усунення. Це дозволить не лише покращити безпеку онлайн-магазинів, але й підвищити загальний рівень довіри споживачів до електронної комерції. Отже, метою даної роботи є дослідження існуючих загроз та розробка ефективних методів для їх подолання, що сприятиме безпеці в онлайн-торгівлі.

РОЗДІЛ 1. АНАЛІЗ ОНЛАЙН-ТОРГІВЛІ ТА ШАХРАЙСТВА

1.1. Визначення онлайн-торгівлі

Онлайн-торгівля – це купівля або продаж товарів, через мережу Інтернет. Такий процес дає можливість придбати будь-яку річ, не виходячи з дому, використовуючи смартфон або комп'ютер. Споживачі мають змогу переглядати асортименти товарів, порівнювати ціни, застосовувати промокоди на знижки та замовляти товари до більш зручного місця. Продавці в свою чергу можуть досягти більшої аудиторії, без обмежень фізичного розташування. Такий процес забезпечує зручність, швидкість, як для клієнтів, так і для робітників.

Серед найвідоміших платформ онлайн-торгівлі можна виділити Amazon, eBay, Alibaba та українську Rozetka. Ці компанії забезпечують широкий асортимент товарів і послуг, що робить їх конкурентоспроможними на глобальному ринку. Однак не лише великі гравці користуються перевагами онлайн-продажів, багато малих підприємств також адаптують свої моделі бізнесу до електронної комерції, використовуючи соціальні мережі та власні веб-сайти, щоб досягати нових клієнтів (Рис. 1. 1).

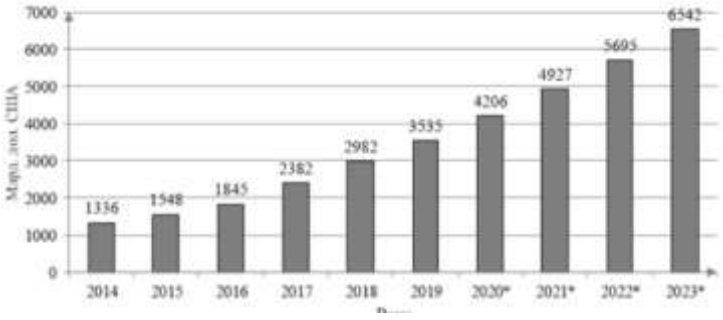

Рис. 1. 1. Динаміка світових обсягів електронного продажу за період 2020-2023 р.

Онлайн-торгівля стала важливою частиною глобальної економіки, особливо в умовах війни, коли споживачі масово переходили на дистанційні

покупки. Такі умови супроводжуються новими викликами, зокрема, загрозами безпеки даних та шахрайством, що потребує розробки ефективних методів виявлення та запобігання таким діям. Тому, розуміння основ онлайн-торгівлі є необхідним для подальшого аналізу ризиків, пов’язаних із цим процесом.

1.2. Основні типи шахрайства в онлайн-торгівлі

Незважаючи на свою популярність і зручність, онлайн-торгівля має значні ризики, пов'язані з шахрайством. Основні типи ризиків можна поділити на кілька категорій, кожна з яких має свій характер, особливості та методи здійснення. Перший тип - фішинг, який застосовують для підробки веб-сайтів або електронних листів з метою викрадення конфіденційної інформації, такої як паролі або дані кредитних карток. Зловмисники часто маскуються під відомі бренди, щоб обманути споживачів і змусити їх надати особисту інформацію. Це порушує принципи конфіденційності, оскільки особисті дані опиняються у руках шахраїв.

Другий тип шахрайства - підробка товарів. У цьому випадку шахраї пропонують неякісні або навіть небезпечні товари, видаючи їх за оригінальні. Це може призвести до серйозних фінансових втрат для покупців, а також негативно вплинути на репутацію законних бізнесів. Порушення цілісності товарів може завдати шкоди клієнтам, які очікують отримати продукт відповідної якості.

Шахрайство з кредитними картками - ще один розповсюджений метод, через який викрадають дані картки для здійснення покупок без відома власника. Зловмисники можуть отримати ці дані через фішинг або через злом веб-сайтів. Цей вид шахрайства завдає значних збитків як споживачам, так і фінансовим установам, це впливає на доступність коштів для законних користувачів [5].

Серйозною загрозою є несанкціонований доступ до облікових записів

клієнтів. Це може відбуватися через злом паролів або використання слабких паролів, що дозволяє зловмисникам отримати контроль над акаунтами. Вони можуть змінювати дані користувачів, здійснювати покупки від їх імені або навіть викрадати конфіденційну інформацію. Таке шахрайство підкреслює важливість надійних методів аутентифікації для забезпечення конфіденційності та цілісності даних.

Також варто згадати про «псевдопродажі», коли шахраї створюють фальшиві оголошення на платформах, пропонуючи товари за зниженою ціною. Після отримання передоплати вони зникають, і споживачі не отримують нічого, окрім поганого настрою. Це актуально для товарів, які користуються великим попитом, таких як електроніка.

Існує шахрайство з поверненням товарів, яке виникає, коли покупець отримує товар, а потім повертає його, стверджуючи, що він не відповідає опису, і отримує назад кошти. У таких випадках зловмисники можуть використовувати це для отримання безкоштовних товарів, що порушує цілісність процесу повернення.

Останній тип злочину - DDoS-атаки, які представляють собою спроби перевантажити сервери онлайн-магазинів. Внаслідок таких атак платформи стають недоступними для легітимних користувачів, що може призвести не лише до фінансових втрат, але й до негативного впливу на довіру споживачів певної платформи, порушуючи доступність сервісів.

Всі типи шахрайства мають значний вплив на конфіденційність, цілісність і доступність (Рис. 1. 2). Шахрайські дії, такі як фішинг, безпосередньо загрожують конфіденційності, оскільки призводять до викрадення особистих даних. Порушення цілісності спостерігається, коли споживачі отримують підроблені товари, що не відповідають заявленим характеристикам. Водночас цілісність інформації страждає не тільки через підроблені товари, але й через маніпуляції з відгуками чи рейтингами магазинів, які можуть ввести споживачів в оману. DDoS-атаки ставлять під загрозу доступність сервісів, роблячи їх недоступними. Конфіденційність

також може бути поставлена під загрозу в результаті витоку даних унаслідок кібератак, таких як використання шкідливих програм. Це може статися через недостатній захист інформаційних систем, що відкриває доступ зловмисникам до конфіденційних даних користувачів.

Рис. 1. 2. «Тріада CIA»

1.3. Кібербезпека як захист від шахрайства

У сучасному цифровому середовищі, кібербезпека відіграє критичну роль у захисті бізнесу та споживачів від шахрайства. З кожним роком зростає кількість злочинних схем, які намагаються використати вразливості електронних платформ. У контексті онлайн-магазинів ефективні механізми кібербезпеки стають не лише засобом захисту, але й важливим елементом довіри між компаніями та їхніми клієнтами.

Забезпечення кібербезпеки стало важливим завданням для всіх рівнів публічного управління - від місцевих органів, які здійснюють онлайн-транзакції, до центральних структур, відповідальних за національну безпеку. У світі, де технології швидко розвиваються, а кіберзагрози стають поширенішими, захист від атак є нагальною необхідністю для всіх сфер суспільного життя. Постійно зростаючі загрози підкреслюють важливість постійного вдосконалення та адаптації стратегій кібербезпеки. Необхідно не лише реагувати на існуючі загрози, а й прогнозувати їх та вжити запобіжних

заходів.

Важливо не тільки реагувати на актуальні кіберзагрози, але й передбачати їх та вживати запобіжних заходів. Лише завдяки спільній роботі та постійному вдосконаленню можна ефективно захиститися від кіберзагроз і підтримувати надійність онлайн-магазинів у нашому світі.

Економічні наслідки шахрайства. Один із найбільших ризиків шахрайства полягає в прямих фінансових втратах. Онлайн-магазини, які стають жертвами шахрайських схем, можуть зазнавати значних економічних збитків через повернення товарів, компенсації клієнтам і витрати на розслідування. Фальшиві замовлення можуть призвести до втрати не лише коштів, але й ресурсів, витрачених на доставку та обслуговування.

Шахрайство обмежує також можливості для інвестицій у розвиток онлайн-магазинів. Коли підприємства витрачають кошти на боротьбу з шахрайством, це зменшує їх здатність інвестувати в нові технології, маркетинг або покращення сервісу. Це призводить до зниження конкурентоспроможності, тому що інші компанії, які не стикаються з такими проблемами, можуть швидше розвиватися.

Зниження довіри клієнтів - ще один важливий наслідок шахрайства. Коли покупці стають жертвами кіберзлочинців, вони починають уникати таких платформ або взагалі онлайн-покупок. В результаті відбувається зниження обсягів продаж, що може суттєво вплинути на прибутковість бізнесу. Споживачі можуть почати шукати більш безпечні альтернативи, що негативно вплине на лояльність до бренду.

Збільшення злочинів може призвести до посилення регуляторних вимог щодо захисту споживачів. Уряди можуть запроваджувати нові закони, які зобов'язують онлайн-магазини розробляти додаткові заходи безпеки. Це призведе до збільшення адміністративних витрат і потреби у додаткових ресурсах для дотримання нових норм.

Розуміння цих наслідків – важливий крок для розробки ефективних стратегій виявлення і запобігання шахрайству. Інвестиції в технології,

навчання персоналу та впровадження нових практик можуть суттєво зменшити негативний вплив шахрайства на бізнес і споживачів, сприяючи безпеці та стабільності онлайн-торгівлі.

Соціальні наслідки шахрайства. Вплив кіберзлочинців має не лише економічні, але й суттєві соціальні наслідки, які впливають на різні верстви населення та суспільство в цілому. Шахрайство може мати негативний психологічний вплив на жертв. Люди, які стали жертвами шахрайських дій, можуть відчувати стрес, тривогу та розчарування. Це може призвести до зниження їхнього загального рівня життя і вплинути на їхню готовність здійснювати покупки в майбутньому. Такі дії можуть загострювати соціальну нерівність. Найвразливіші верстви населення, такі як літні люди або особи з обмеженими можливостями, можуть стати легкими мішенями для шахраїв. Втрата грошей або особистих даних може суттєво вплинути на їхнє фінансове становище, що загострює вже існуючі соціальні проблеми.

Коли, через шахрайство, робітникам не вистачає заробітньої плати - це може призвести до скорочення робочих місць або зменшення інвестицій у розвиток. Такий випадок, може негативно вплинути на економічне становище місцевих жителів і призвести до зниження якості життя в громадах.

Соціальні наслідки шахрайства підкреслюють необхідність підвищення обізнаності та освіти споживачів, особливо старшого віку. Саме такі люди не знають про ризики, пов'язані з онлайн-торгівлею, і не мають достатніх знань для захисту своїх даних. Це створює потребу в освітніх програмах, які б навчали споживачів безпечним практикам в інтернеті.

1.4. Актуальність проблеми шахрайства в Україні

Проблема шахрайства в Україні, особливо в умовах війни, стає дедалі більш актуальною. Конфлікт, який триває з 2014 року, а повномасштабне вторгнення в 2022 році, суттєво змінив соціально-економічну ситуацію та

вплинув на динаміку шахрайських схем.

Війна призвела до значних економічних потрясінь, що збільшило вразливість населення. Багато людей втратили джерела доходу, що спонукало їх шукати альтернативні способи заробітку, іноді включаючи шахрайство. Водночас, економічна нестабільність створює підґрунтя для шахраїв, які експлуатують страх і невизначеність населення.

Шахраї використовують військову ситуацію для обману громадян. Наприклад, вони можуть пропонувати "допомогу" у зборі коштів для армії або постраждалих, але фактично заволодівають цими грошима. Це не лише завдає фінансових збитків, але й підриває довіру до благодійних організацій та ініціатив.

В умовах війни увага населення зосереджена на виживанні та безпеці. Люди стають менш уважними до деталей і можуть легше потрапити в пастки шахраїв. Важливою є також роль інформаційної безпеки, адже в умовах інформаційної війни шахраї можуть використовувати дезінформацію для своїх цілей [1;9;12].

Наприклад, на платформі Rozetka.ua шахраї створювали підроблені акаунти, імітуючи відомі магазини, та пропонували товари за зниженими цінами. Споживачі, замовляючи продукцію, не отримували її, а шахраї зникали з отриманими коштами.

На Prom.ua також було зафіксовано випадки шахрайства через фальшиві оголошення, коли шахраї публікували пропозиції на неіснуючі товари, такі як електроніка, та просили передплату. Після отримання грошей вони зникали, залишаючи покупців без товару.

На платформі OLX.ua шахраї часто розміщують фальшиві оголошення на продаж популярних товарів, таких як мобільні телефони, просячи оплату наперед. Після отримання грошей вони зникають, не надаючи товар.

Користувачі Aliexpress також стикаються з шахрайськими продавцями, які пропонують товари за надзвичайно низькими цінами, але після оформлення замовлення покупці можуть отримати не те, що очікували, або

взагалі нічого не отримати.

Крім того, на Moyo.ua сталися випадки, коли шахраї використовували фальшиві сайти, що імітують Moyo, пропонуючи товари за знижками. Споживачі, переходячи за посиланнями, надавали свої дані, після чого їхні рахунки могли бути зламані.

Citrus.ua також став об'єктом шахрайства, коли шахраї розсилали фальшиві акції з "ексклюзивними" знижками. Клієнти, натискаючи на посилання, потрапляли на підроблені сайти, де їхні дані могли бути вкрадені [8].

Ці приклади ілюструють різноманітність шахрайських схем в онлайн-магазинах України, споживачі повинні бути обережними, перевіряючи відгуки про продавців та уникати підозрілих пропозицій. Держава та бізнес повинні спільно працювати над створенням безпечніших умов для онлайн-торгівлі. Це надає нові виклики для спеціалістів кібербезпеки. Потрібно впроваджувати нові технології захисту, проведення та зміцнення правової бази для покарання шахраїв.

РОЗДІЛ 2. АНАЛІЗ РИЗИКІВ ШАХРАЙСТВА В ОНЛАЙН-ТОРГІВЛІ

2.1. Методи виявлення ризиків шахрайства

Метод виявлення ризиків – це можливий інструмент, який використовують для аналізу, ідентифікації та оцінки потенційний атак в бізнесі або онлайн-торгівлі. Такі методи допомагають організаціям виявляти аномальні або підозрілі дії, що можуть свідчити про шахрайство. Ці дії допомагають своєчасно вжити відповідні заходи для запобігання кіберзлочину. В умовах зростаючої популярності електронної комерції та збільшення випадків шахрайства, ефективні методи виявлення ризиків стають критично важливими для захисту фінансових інтересів компаній та їхніх споживачів.

Виявлення ризиків шахрайства є складним процесом, який включає кілька етапів:

1. Збір даних про транзакції (інформація про транзакції, поведінку користувачів, профілі споживачів та інші релевантні показники. Це можуть бути дані про місцезнаходження, типи товарів, суми покупок, час здійснення транзакцій та використані платіжні методи);
2. Аналіз даних (використання статистичних методів для виявлення аномалій, а також побудову профілів ризику для різних категорій клієнтів. Наприклад, якщо певний користувач здійснює покупку за незвичною великою сумою або з незвичної географічної локації, це може бути сигналом для подальшого розслідування);
3. Верифікація користувачів (на цьому етапі важливо перевірити особу клієнта, особливо під час реєстрації нових облікових записів. Можуть включатися процедури, такі як двофакторна аутентифікація, перевірка документів та інші заходи, що підтверджують особу користувача. Саме такий етап допомагає запобігти створенню підроблених облікових записів і зменшує ризики шахрайства);

4. Моніторинг транзакцій (використання спеціалізованих алгоритмів для автоматичного виявлення підозрілих дій допомагає швидко реагувати на потенційні загрози. Наприклад, якщо система виявляє, що один і той же обліковий запис здійснює багато запитів на повернення товару, це може свідчити про шахрайство);

5. Оцінка ризиків (організації повинні регулярно аналізувати результати виявлення шахрайства, враховуючи нові загрози та зміни в поведінці споживачів, що дозволить адаптувати стратегії захисту та вжити відповідних заходів для зменшення ризиків);

Правильне впровадження цих етапів допомагає зменшити ризики обману та забезпечити безпеку фінансових операцій (Рис. 2. 1).

Фахівці з кібербезпеки використовують різноманітні технології та платформи для виявлення та запобігання шахрайству в онлайн-магазинах. Один із ключових інструментів - це Python, який забезпечує потужні можливості для аналізу даних. За допомогою бібліотек, таких як Pandas, фахівці можуть обробляти великі обсяги інформації про транзакції та виявляти аномалії. Бібліотека Scikit-learn дозволяє створювати моделі машинного навчання, які ідентифікують шахрайські транзакції на основі історичних даних [7].

Інша важлива технологія - JavaScript, яка використовується для впровадження захисних механізмів на стороні клієнта. Наприклад, за допомогою бібліотек, таких як Authy або Google Authenticator, можна реалізувати двофакторну аутентифікацію, що значно підвищує безпеку облікових записів користувачів. Моніторинг активності користувачів також здійснюється за допомогою JavaScript, що дозволяє виявляти підозрілі дії, такі як численні спроби входу з однієї IP-адреси.

Системи управління безпекою (SIEM), наприклад, Splunk або IBM QRadar, забезпечують більш глибокий аналіз даних безпеки в реальному часі, збираючи журнали подій з різних систем і використовуючи алгоритми машинного навчання для автоматичного виявлення загроз на основі

історичних даних.

Node.js також відіграє важливу роль у захисті API, дозволяючи реалізувати валідацію запитів за допомогою бібліотек, таких як express-validator, і використовувати токенізацію через JSON Web Tokens (JWT) для автентифікації користувачів. Щоб захистити дані в транзакціях, фахівці застосовують шифрування, зокрема SSL/TLS для захисту інформації, що передається між браузером і сервером, а також алгоритм AES (Advanced Encryption Standard) для шифрування чутливих даних, зокрема інформації про кредитні картки.

Тестування на проникнення є ще одним важливим аспектом, такі платформи як Burp Suite або OWASP ZAP, дозволяють фахівцям проводити такі перевірки для виявлення вразливостей в онлайн-магазинах. Це включає автоматизоване сканування на наявність вразливостей, таких як SQL-ін'єкції або XSS-атаки, а також створення сценаріїв для перевірки безпеки системи під час реальних атак.

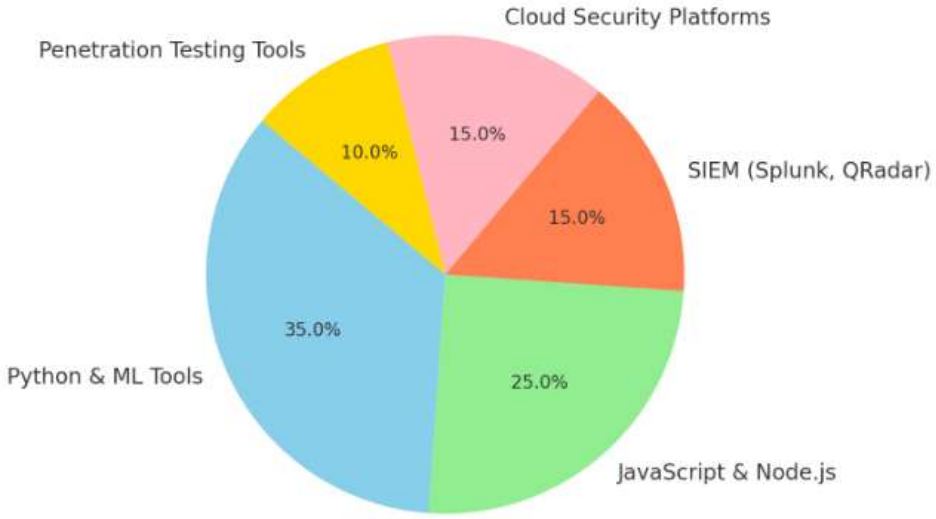

Рис. 2. 1. Найчастіше використовувані платформи в кібербезпеці 2024 р.

Таким чином, фахівці з кібербезпеки використовують різноманітні технології для безпосередньої роботи фінансових операцій в онлайн-магазинах і підтримки довіри споживачів. А споживачі в свою чергу

обирають перевірені магазини в інтернеті.

2.2. Огляд існуючих систем виявлення шахрайства

Онлай-магазини, бізнеси та інші підприємства мають потребу та виконують усі можливі кроки для впровадження системи виявлення шахрайства, які використовують передові технології що основані на моніторингу та аналізі транзакцій у реальному часі. Такі системи виявляють підозрілі дії, знижують ризики та захищають як фінансові ресурси компанії, так і довіру споживачів.

SAS Fraud Management - потужне рішенням, яке використовують для аналітики в реальному часі та виявлення шахрайських дій. Ця система аналізує великі обсяги даних про транзакції, дозволяючи виявляти аномалії та підозрілі дії. SAS також пропонує гнучкі правила для налаштування виявлення шахрайства, що робить її ідеальною для онлайн-магазинів, які потребують адаптивного підходу до безпеки.

FICO Falcon - одна з найвідоміших систем у сфері виявлення шахрайства, що активно використовується в банківському секторі та електронній комерції. Така система дозволяє застосовувати алгоритми машинного навчання для перегляду транзакцій, а це дозволить виявляти шахрайство. Завдяки можливості адаптації до нових загроз, FICO Falcon забезпечує високий рівень захисту для онлайн-магазинів.

IBM Safer Payments - спеціалісти використовують її для безпечних транзакцій в режимі реального часу. Система має потужні аналітичні інструменти, які виявляють підозрілі дії. Ця система також інтегрується з іншими бізнес-процесами. Вона може використовувати штучний інтелект, що дозволяє системі адаптуватися до нових шахрайських схем, а це робить її ефективним інструментом для онлайн-торгівлі.

ThreatMetrix – надає рішення, щодо виявлення шахрайства на результатах ризику, що аналізують транзакції в реальному часі. Така система

використовує дані поведінки для ідентифікації користувачів і може виявляти шахрайство, базуючись на зібраних даних з різних джерел. Інтеграція з іншими системами безпеки робить ThreatMetrix надійним партнером.

Kount - це платформа, яка також використовує штучний інтелект для виявлення шахрайства. Вона теж має змогу оцінювати ризики у реальному часі та дозволяє налаштовувати правила для виявлення злочинних дій.

Riskified має певну спеціалізацію, пропонуючи рішення, для виявлення та управління ризиками. Система аналізує транзакції за допомогою машинного навчання, виявляє шахрайські замовлення та гарантує захист від збитків. Саме такий аналіз даних допомагає онлайн-магазинам покращувати свої рішення в майбутньому.

Zelle – одна з платіжних систем, у якої є вбудовані механізми для виявлення шахрайства. Вона спостерігає за транзакціями в реальному часі та сповіщає користувачів про підозрілі дії, тим самим забезпечує додатковий рівень захисту для онлайн-магазинів, які приймають платежі через цю платформу.

Серед нових технологій, що активно впроваджуються в системи виявлення шахрайства, важливі місця займають блокчейн і біометричні методи аутентифікації.

Блокчейн – технологія, яка має змогу забезпечити високий рівень безпеки та прозорості транзакцій. Вона особлива тим, що має функції на основі дистрибутивних даних серед численних учасників мережі, а це значно ускладнює можливість підробити ту чи іншу інформацію. У напрямку онлайн-магазинів блокчейн використовують для створення незмінних записів про всі транзакції, що дозволяє легко відстежувати підозрілі дії та запобігати шахрайству. Додатково, завдяки механізмам консенсусу, всі учасники можуть перевіряти достовірність інформації, це значно підвищує довіру до системи.

Біометричні методи аутентифікації, такі як розпізнавання обличчя, відбитків пальців або райдужної оболонки ока, мають попит та стають дедалі

популярнішими. Вони мають змогу забезпечити високий рівень захисту, бо особливістю є те, що біометричні дані унікальні для кожної особи і важко підробити. У онлайн-магазинах біометрична аутентифікація може бути інтегрована, коли клієнт хоче увійти або зареєструвати свій обліковий запис, або підтвердити платіж, що значно ускладнює доступ шахраїв до облікових записів користувачів.

Отже, впровадження блокчейн - технологій і біометричних методів аутентифікації відкриває нові можливості для підвищення безпеки онлайн-магазинів і захисту від шахрайства, створюючи більш надійні та прозорі системи для споживачів.

Успішні кейси впровадження систем виявлення шахрайства. Одним із найкращих прикладів успішного встановлення системи виявлення шахрайства - компанія Amazon. Ця компанія стикалася з численними проблемами, пов'язаними з шахрайськими транзакціями, які призводили до значних фінансових втрат і зниження довіри споживачів. Кіберзлочинці використовували різноманітні методи, такі як фішинг, підробка платіжних даних і повернення товарів, що негативно впливало на бізнес. Після впровадження системи на базі машинного навчання, Amazon змогла набагато покращити свої процеси безпеки. Було використано алгоритми, які аналізують поведінку покупців у режимі реального часу, компанія змогла виявляти підозрілі транзакції ще до їх завершення. Результатом стало те, що Amazon повідомила про зменшення рівня шахрайства на 20% протягом перших шести місяців після встановлення нової системи, а це в свою чергу дозволило зекономити мільйони доларів.

Одна з провідних платіжних систем у світі, компанія PayPal, стикалася з серйозними проблемами, пов’язаними із шахрайськими платежами. Високий рівень загрози був не лише фінансовим ресурсам компанії, а й її репутації. Кіберзлочинці використовували різноманітні методи, такі як фальшиві платежі, підробка облікових записів і використання викрадених кредитних карток. Такі дії призводили до значних втрат для компанії,

знижували довіру користувачів і погіршували загальний досвід роботи з платформою. Щоб вирішити такі проблеми PayPal інтегрували потужну систему виявлення шахрайства, яка базується на штучному інтелекті та машинному навчанні. Основні етапи такої системи включали:

1. Аналіз даних у реальному часі;
2. Моделі прогнозування;
3. Постійне навчання;
4. Сповіщення про підозрілі транзакції;
5. Аналіз поведінки користувачів.

Вже за перший рік компанія спостерігала зниження рівня шахрайства на 30%. Це призвело до: економії мільйонів доларів, покращення репутації, зростання кількості активних користувачів. Можна сказати, що такі методи позитивно вплинули на загальний бізнес компанії.

Shopify, одна з провідних платформ для створення онлайн-магазинів, яка також зіштовхнулася з проблемами у сфері безпеки. Через те, що це популярна платформа серед малих і середніх бізнесів, вона стала мішенню для шахраїв, які використовували різноманітні методи для обману продавців. Дуже високий рівень шахрайських замовлень, такі як фальшиві платіжні дані і повернення товарів, призвело до значних фінансових втрат для магазинів, які користуються сервісами Shopify.

Для подолання таких проблем, було вирішено впровадити інтегровану систему виявлення шахрайства, яка мала змогу поєднувати в собі сучасні технології аналізу даних, машинного навчання та автоматизації. Система враховувала різні параметри, такі як геолокація, сума замовлення, тип платіжних карток та історія покупок. На цьому не зупинилися, далі впровадили алгоритми машинного навчання, які навчалися на історичних даних про транзакції, дозволяючи системі ідентифікувати підозрілу поведінку, притаманну як законним, так і шахрайським замовленням. Завдяки цьому, система могла автоматично адаптуватися до нових загроз, що робило її більш ефективною у виявленні шахрайства. Коли було виявлено

аномалію, система автоматично сповіщала продавця та могла блокувати такі транзакції. Це дозволяло онлайн-магазинам швидко реагувати на можливі шахрайські дії та запобігати фінансовим втратам. Окрім цього, продавцям надавалися рекомендації, щодо покращення безпеки їхніх магазинів, звісно це підвищувало загальний рівень захисту. Shopify також організувала навчальні програми для власників магазинів, щоб навчити їх розпізнавати потенційно шахрайські дії та використовувати інструменти платформи для захисту своїх бізнесів. Встановлення нової системи виявлення шахрайства призвело до кращих змін для Shopify та її користувачів. Вже протягом першого року після впровадження системи, Shopify зафіксувала зниження рівня шахрайських замовлень на 40%. Такий позитивний вплив допоміг економити значні суми для онлайн-магазинів, що особливо важливо малим та середнім підприємствам, які мають обмежені фінансові ресурси. Зростання довіри до платформи Shopify відзначалося як з боку продавців, так і покупців, оскільки споживачі почали відчувати себе безпечніше, здійснюючи покупки через онлайн-магазини на базі Shopify. Такі заходи безпеки не тільки зміцнили позиції Shopify на ринку, але й зробили платформу набагато привабливою для нових бізнесів, що прагнуть безпечно вести електронну комерцію.

2.3. Аналіз ризиків на прикладі конкретного онлайн-магазину

OLX - це популярна онлайн-платформа для покупки та продажу товарів або послуг, яка функціонує в багатьох країнах, зокрема в Україні. Така платформа була заснована в 2006 році, вона стала одним із провідних сайтів оголошень, де люди можуть взаємодіяти безпосередньо один з одним. Можливості, які має платформа OLX – це класифіковані оголошення, безкоштовна публікація товарів, локалізація для зручності користувачів, інтуїтивно зрозумілий інтерфейс, мобільний додаток, різноманітність товарів, а також комунікація між користувачами, через вбудовану систему

повідомлень. Додатково, така система має фільтри, пошуки по місцю знаходження, рейтинги користувачів, відгуки. Але існують і численні ризики з боку шахраїв. Користувачі можуть натрапити на підроблені оголошення або злочинців, які створюють фальшиві профілі для обману інших. Це може призвести до фінансових втрат для покупців і зниження довіри до платформи.

OLX також стикається з численними ризиками в сфері кібербезпеки, що пов'язано з обробкою великої кількості особистих даних користувачів і фінансовими транзакціями (Рис. 2. 2). По-перше, ризик неавторизованого доступу найактуальніший, оскільки зловмисники можуть намагатися отримати доступ до акаунтів користувачів через фішинг або злом паролів, що призводить до крадіжки особистих даних та фінансових втрат. По-друге, фішинг, який залишається серйозною загрозою: користувачі можуть отримувати фальшиві електронні листи або повідомлення, які виглядають, як офіційні від OLX, з метою збору їхніх особистих даних або реквізитів карток.

Зловмисне програмне забезпечення може атакувати платформу, що призведе до компрометації системи та крадіжки даних. Атаки типу DDoS також становлять загрозу, тому що кіберзлочинці намагаються перевантажити сервери OLX, що призведе до недоступності платформи для користувачів і може негативно вплинути на репутацію компанії. Вразливості в коді платформи можуть використовуватися злочинцями для отримання доступу до системи або даних користувачів [2].

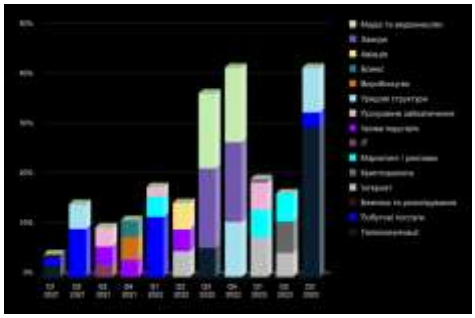

Рис. 2. 2. Прогноз кіберзагроз 2024 р.

Важливим питанням постає конфіденційність даних, адже зберігання та обробка особистої інформації користувачів створює ризики витоку даних, а це може вдарити по репутації. Неправильна конфігурація систем безпеки призведе до вразливостей, які зловмисники можуть експлуатувати, включаючи недостатню аутентифікацію або шифрування. Нарешті, ризики іншої сторони, пов'язані з партнерами або постачальниками, які також мають доступ до даних, саме таким чином можуть відбутися загрози, якщо працюють ненадійні практики кібербезпеки.

Рекомендації щодо управління ризиками. Проаналізувавши ризики, які можуть бути на прикладах будь-якого онлайн-магазину, розробимо кроки рекомендацій, які тим чи іншим чином допоможуть у нейтралізації таких проблем (Рис. 2. 3).

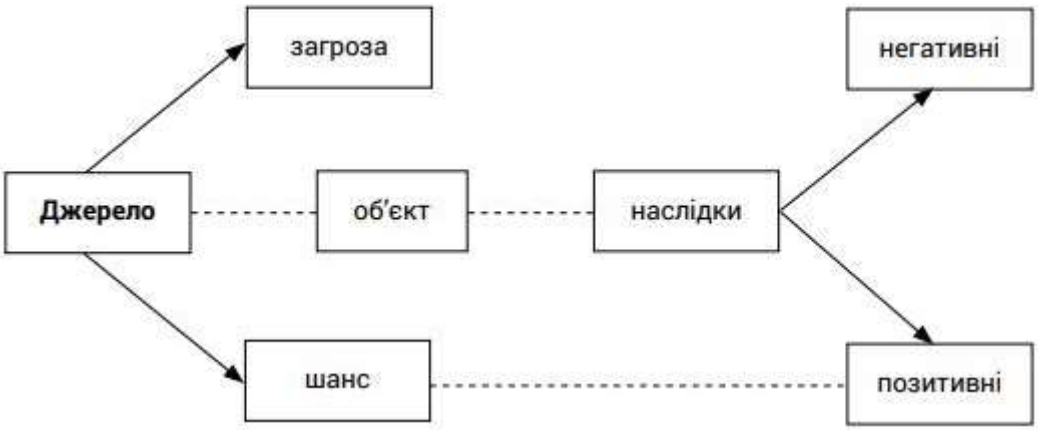

Рис. 2. 3. Структура ризику

Перш за все, важливо покращити навчання користувачів. Регулярні курси, веб-сайти, вебінари, воркшопи з підвищення обізнаності про кіберзагрози, можуть допомогти користувачам розпізнавати потенційні загрози. Впровадження двофакторної аутентифікації також особливо-важливим для підвищення рівня безпеки акаунтів. Регулярні аудити безпеки, проведені незалежними експертами, зможуть виявити та усунути потенційні загрози.

Удосконалення системи шифрування для захисту персональних даних

користувачів суттєво знизить ризики витоку інформації. Впровадження системи моніторингу активності користувачів дозволить виявляти підозрілу діяльність на платформі та швидко реагувати на можливі загрози. Це може включати автоматичні оповіщення про підозрілі дії, такі як спроби входу з незнайомих пристроїв.

І останнє, регулярне оновлення програмного забезпечення для усунення вразливостей і покращення захисту є необхідним заходом. Це стосується не лише операційної системи, а й бібліотеки та інших залежностей.

Такі рекомендації допоможуть підсумувати аналіз ризиків і вказати на шляхи їх зменшення, що є важливим етапом у забезпеченні кібербезпеки платформи OLX [13].

РОЗДІЛ 3. РОЗРОБКА МЕТОДІВ ВИЯВЛЕННЯ ШАХРАЙСТВА

3.1. Алгоритми виявлення шахрайства

Виявлення шахрайства є найважливішим аспектом безпеки онлайн-платформ, і для цього використовуються різноманітні алгоритми, які аналізують дані та виявляють підозрілі дії. Розглянемо основні алгоритми, які можуть бути застосовані для виявлення шахрайства:

1. **Logistic Regression** - один із найпростіших методів класифікації. Він використовується для прогнозування ймовірності шахрайства на основі різних характеристик користувачів та їх дій. Logistic Regression аналізує, як зміна однієї або кількох незалежних змінних впливає на ймовірність виникнення шахрайства. Це дозволяє виявити користувачів, чия поведінка відхиляється від норми.

2. **Decision Trees** - алгоритм дерев рішень розбиває дані на категорії, одночасно визначає ключові ознаки, які найкраще відокремлюють шахрайські дії від звичайних. Кожен внутрішній вузол дерева представляє тест на певну характеристику, а кожен лист – результат, де показується шахрайство чи ні. Decision Trees є зрозумілими для інтерпретації, що дозволяє легко візуалізувати процес прийняття рішень.

3. **Random Forest** - такий метод є вдосконаленням дерева рішень. Він комбінує кілька дерев для підвищення точності виявлення шахрайства. Кожне дерево в лісі робить окремий прогноз, а фінальне рішення визначається за допомогою голосування. Це значно зменшує ризик переобучення, оскільки результати кількох моделей більш надійні, ніж результати однієї.

4. **Isolation Forest -** алгоритм для виявлення аномалій. Він створює випадкові дерева для ізоляції аномальних даних. Чим швидше дані ізолюються, тим більш імовірно, що вони є аномальними. Цей підхід особливо ефективний для виявлення шахрайських подій.

5. **One-Class SVM -** однокласовий метод опорних векторів виявляє аномалії, навчаючись на нормальних даних. Алгоритм визначає межу, яка максимально відокремлює нормальні дані від всього іншого. Якщо нові дані потрапляють за межі цієї області, вони класифікуються як аномальні.

6. **Neural Networks -** глибинні нейронні мережі здатні аналізувати великі обсяги даних і виявляти складні патерни шахрайства. Вони ефективні для обробки неструктурованих даних, таких як тексти оголошень або зображення товарів. Нейронні мережі можуть навчатися на великих наборах даних, що дозволяє їм адаптуватися до нових шахрайських схем.

7. **Gradient Boosting Machines (GBM)** - створює потужні моделі, враховуючи взаємозв'язки між різними ознаками. GBM працює шляхом послідовного навчання нових моделей, які намагаються виправити помилки попередніх. Це дозволяє досягти високої точності при виявленні шахрайства.

8. **K-means Clustering -** класифікує користувачів або транзакції за схожістю, що допомагає виявити аномалії. Кластеризація дозволяє групувати подібні дані, і якщо новий запис не вписується в жоден з кластерів, це може служити сигналом про можливе шахрайство.

9. **DBSCAN -** алгоритм виявляє аномальні кластери без попереднього визначення кількості кластерів. DBSCAN ідеально підходить для виявлення шуму в даних та ідентифікації аномальних точок, що може бути корисно для визначення шахрайських дій.

10. **Natural Language Processing (NLP) -** методи обробки природної мови можуть бути використані для аналізу тексту оголошень, виявляючи шахрайські патерни, такі як повторювані фрази або специфічні ключові слова. NLP допомагає автоматизувати процес виявлення шахрайства в текстових даних.

11. **Graph-Based Algorithms -** алгоритми використовують графи для моделювання зв'язків між користувачами. Аналізуючи графи, можна виявити шахрайські схеми, які можуть включати мережі шахраїв, що взаємодіють між собою.

12. **Time Series Analysis** - аналіз часових рядів дозволяє вивчати дані по часі, виявляючи шахрайську активність, що відбувається в певні періоди. Наприклад, якщо велика кількість шахрайських транзакцій відбувається в один і той же час, це може вказувати на організовану діяльність.

Є дуже багато різних та ефективних алгоритмів, які допоможуть виявити шахрайство. Якщо почати комбінувати їх, то це може суттєво підвищити рівень безпеки, що дозволить оперативно реагувати на нові загрози.

3.2. Використання машинного навчання

Машинне навчання – це засоби штучного інтелекту, які спираються на розробці алгоритмів і моделей статистики. Це дозволяє комп’ютерам навчатися, спираючись на дані, робити деякі прогнози або приймати рішення для подальшої ідентифікації загроз.

Така специфіка має великий потенціал для онлайн-магазинів, що може допогти покращити обслуговування клієнтів, оптимізувати бізнес-процеси та підвищити прибутковість. Найосновніші застосування - це рекомендаційні системи, які аналізують поведінку користувачів, історію переглядів та покупки, щоб створити найактуальніші рекомендації. Це підвищує ймовірність покупки, оскільки клієнти отримують пропозиції, які відповідають їхнім уподобанням.

Таке навчання дозволяє розподілити клієнтів на основі їхньої поведінки, демографічних даних та історії покупок, а це допомагає в розробці цільових маркетингових кампаній, адаптованих для конкретних груп. Алгоритми машинного навчання можуть виявляти шахрайські дії: аналізувати транзакції, виявляючи аномалії, що в свою чергу дозволяє запобігти зловмисним діям, наприклад, виявляючи підозрілі покупки або фальшиві облікові записи.

Прогнозування попиту є ще одним плюсом для онлайн-магазинів.

Машинне навчання може аналізувати історичні дані про продажі, сезонність та інші фактори для того, щоб працівники мали змогу прогнозувати попит на товари. Такі дії дозволять оптимізувати запаси, зменшити витрати на зберігання та уникнути дефіциту товарів. В такому випадку на допомогу може прийти динамічне ціноутворення - алгоритми можуть автоматично коригувати ціни на основі ринкових умов, конкурентів та поведінки споживачів, саме це дозволить максимізувати прибуток.

Автоматизація обслуговування клієнтів за допомогою чат-ботів на базі машинного навчання стала надзвичайно популярною у сучасних умовах (Рис. 3. 1). Вони можуть автоматично відповідати на запитання клієнтів, обробляти замовлення та надавати рекомендації, що знижує навантаження на службу підтримки та покращує обслуговування клієнтів. Аналіз відгуків за допомогою обробки природної мови (NLP) дозволяє виявляти загальні проблеми, позитивні та негативні аспекти товарів чи послуг, що допомагає підприємствам покращувати якість своїх продуктів [4].

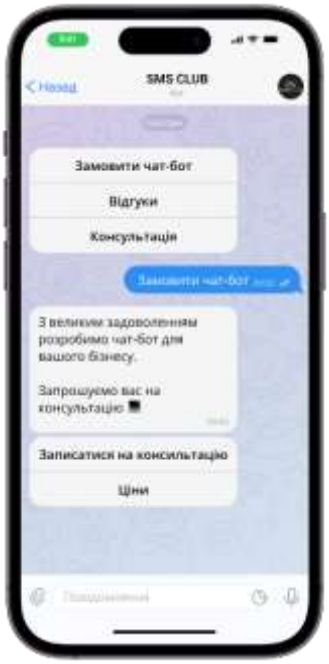

Рис. 3. 1 Приклад чат-боту

Візуальний пошук набирає популярності серед онлайн-магазинів. Вони можуть впроваджувати функції, які дозволяють користувачам завантажувати

зображення товарів для пошуку схожих предметів, що підвищує зручність користування та збільшує конверсію (Рис. 3. 2). Впровадження машинного навчання в онлайн-магазинах не тільки покращує досвід клієнтів, але й оптимізує внутрішні процеси, що сприяє підвищенню ефективності та прибутковості бізнесу. Ці технології дозволяють магазинам залишатися конкурентоспроможними в умовах стрімко змінюваного ринку.

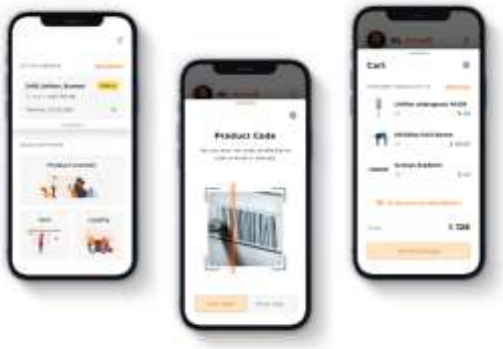

Рис. 3. 2 Технологія візуального пошуку товарів

Навчання на реальних даних - важливий етап у розробці алгоритмів машинного навчання, оскільки це дозволяє моделі адаптуватися до реальних умов і підвищити точність. Використання реальних даних допомагає вивчати шаблони поведінки, які можуть вказувати на певні результати, що особливо важливо в змісті виявлення шахрайства в онлайн-торгівлі.

По-перше, такі дані містять різноманітність, яка відображає реальні умови. Це може означати, що моделі, навчені на таких даних, можуть краще справлятися з новими, невідомими прикладами, оскільки вони вже "знайомі" з різними варіантами поведінки споживачів. Наприклад, якщо модель навчена на обмеженому наборі синтетичних даних, вона може не враховувати зразки аномальної поведінки, які можуть виникнути в реальному житті.

По-друге, навчання на реальних даних дозволяє виявити важливі кореляції та залежності, які можуть бути неочевидними. Наприклад, дані

можуть показати, що певні часові проміжки або комбінації товарів часто супроводжують шахрайські транзакції. Такі знання використовують для покращення алгоритмів виявлення шахрайства, допомагаючи зосередитися на найбільш ризикованих транзакціях.

Одним із таких наборів даних є Kaggle Credit Card Fraud Detection Dataset. Цей набір даних містить транзакційні дані кредитних карток, де лише близько 0,17% транзакцій є шахрайськими. Набір включає такі характеристики: час транзакції (час, в який була здійснена транзакція), сума транзакції (сума, яка була витрачена), анонімні ознаки (дані, отримані в результаті PCA, методу зменшення розмірності, які представляють різні аспекти транзакцій без розкриття особистої інформації).

Такий набір дозволяє навчати моделі виявлення шахрайства на реальних транзакційних шаблонах, це підвищує їх точність у виявленні шахрайських дій. Використання реальних даних з набору Kaggle дозволяє алгоритмам не лише вивчати зразки шахрайських дій, але й адаптуватися до змін у поведінці споживачів, що постійно еволюціонують.

3.3. Огляд мов програмування

У нашому світі існує велика кількість мов програмування, якими спеціалісти користуються кожен день. Кожна з них має свої унікальні властивості, що дозволяє вирішувати широкий спектр завдань у галузі кібербезпеки. Вибір мови програмування залежить від конкретних вимог проекту, а також від особистих переваг та навичок фахівця. Основними критеріями при виборі мови є її продуктивність, зручність у використанні та можливості інтеграції з іншими інструментами. Також важливо враховувати підтримку спільноти та наявність документації, що значно полегшує процес розробки.

Python є однією з найбільш поширених мов програмування, завдяки своїй простоті та зрозумілому синтаксису. Її легко вивчити, що робить її

популярною серед новачків у програмуванні. Python широко використовується у наукових обчисленнях, веб-розробці та машинному навчанні завдяки великій кількості бібліотек та фреймворків, таких як NumPy, Django та TensorFlow. Однією з головних переваг Python є велика та активна спільнота користувачів, яка забезпечує багату документацію та підтримку. Це дозволяє швидко знаходити відповіді на питання та вирішувати проблеми, що виникають під час розробки. Однак, Python має деякі недоліки, такі як повільніша робота порівняно з компільованими мовами, що обмежує його використання для мобільної розробки та високопродуктивних програм. Це означає, що для задач, які вимагають великої продуктивності, така мова програмування може бути не найкращим вибором.

JavaScript - основна мова для веб-розробки, як на клієнтській, так і на серверній стороні. Завдяки своїй здатності підтримувати асинхронне програмування, вона дозволяє обробляти події та працювати з мережею ефективно. Ця мова має велику екосистему бібліотек та фреймворків, таких як React, Angular та Vue, які роблять процес розробки швидким та ефективним. JavaScript також є основною мовою для розробки динамічних веб-сайтів та веб-додатків, що робить її незамінною у цій галузі. Проте, JavaScript має деякі проблеми з безпекою через відкритий характер веб-середовища. Це означає, що розробники повинні бути особливо обережними при написанні коду, щоб уникнути вразливостей. Крім того, деякі особливості мови можуть бути складними для розуміння, наприклад, область видимості змінних та управління контекстом виконання.

Java є однією з найпоширеніших мов програмування, завдяки своїй незалежності від платформи, що досягається за допомогою віртуальної машини Java (JVM). Це дозволяє розробникам писати код один раз та запускати його на будь-якій платформі, що підтримує JVM. Java використовується для розробки великих корпоративних додатків, мобільних додатків для Android та вбудованих систем. Мова характеризується сильною

типізацією та об’єктно-орієнтованим підходом, що забезпечує високу надійність та зрозумілість коду. Однак, Java є складнішою у вивченні порівняно з деякими іншими мовами, такими як Python. Вона також вимагає більше написання коду для виконання завдань, що може збільшити час розробки.

C++ є мовою високої продуктивності завдяки компіляції в машинний код. Вона підтримує низькорівневе програмування та доступ до апаратного забезпечення, що робить її ідеальною для розробки системного програмного забезпечення, ігор та програм для вбудованих систем. C++ надає розробникам повний контроль над управлінням пам'яттю, що дозволяє оптимізувати продуктивність програм. Проте, C++ має складний синтаксис та вимагає глибокого розуміння управління пам'яттю, що створює високий поріг входу для новачків у програмуванні. Це означає, що розробникам потрібно багато часу для опанування мови та ефективного використання її можливостей.

C# є добре інтегрованою мовою з платформою .NET, що робить її ідеальною для розробки додатків для Windows. Вона також використовується для розробки ігор за допомогою Unity та веб-додатків. C# є об'єктно-орієнтованою мовою з сильною типізацією, що забезпечує високу надійність коду. Мова надає багатий набір інструментів для розробки, що робить процес створення додатків швидким та ефективним. Проте, C# переважно орієнтована на платформи Microsoft, що обмежує її використання в інших областях розробки. Це може бути недоліком для розробників, які працюють з різними платформами.

PHP є основною мовою для веб-розробки на серверній стороні. Вона легко інтегрується з HTML та базами даних, що робить її популярною серед веб-розробників. PHP має велику кількість бібліотек та фреймворків, таких як Laravel та Symfony, які допомагають створювати масштабовані та безпечні веб-додатки. Проте, PHP має повільнішу продуктивність у порівнянні з іншими мовами серверної сторони, такими як Node.js або Python. З іншої

сторони, PHP може мати проблеми з безпекою, якщо не правильно налаштовано, що вимагає від розробників додаткової обережності при написанні коду.

3.4. Опис коду та функціональних компонентів

Першим кроком у розробці чат-бота для виявлення шахрайства було встановлення мови програмування Python. Python обраний завдяки своїй простоті у вивченні, гнучкості та великій кількості бібліотек, які підтримують машинне навчання та роботу з даними.

Для встановлення Python потрібно завантажити інсталяційний файл з офіційного сайту python.org. Після завантаження слід запустити інсталяційний файл і слідувати інструкціям інсталятора. На етапі налаштувань необхідно обрати опцію "Add Python to PATH", що дозволить використовувати Python з командного рядка (Рис. 3. 1).

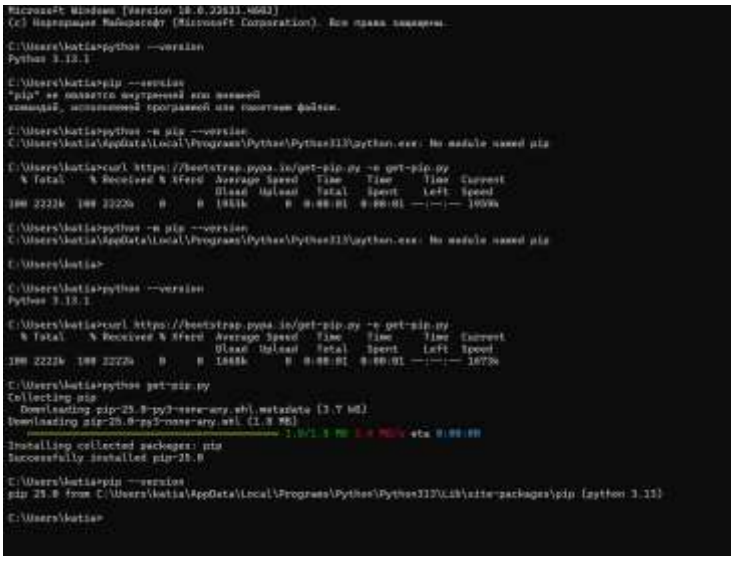

Рис. 3. 1. Встановлення Python

Наступним кроком було створення самого бота в Telegram. Цей процес включає кілька етапів. Відкриття додатку Telegram на своєму пристрої (мобільному телефоні або комп'ютері). Знайти у пошуку Telegram бота під

назвою "BotFather" [6]. Він є офіційним ботом Telegram, який дозволяє створювати та керувати іншими ботами. Відкриваємо діалог з BotFather і вводимо команду /start, щоб почати взаємодію з ним (Рис 3. 2). Вводимо команду /newbot, щоб створити нового бота. BotFather запросить ім'я для вашого нового бота. Вводимо бажане ім'я. Після цього BotFather запросить ввести юзернейм для бота. Юзернейм повинен закінчуватися на "bot" (наприклад, MyNewBot). Після успішного створення бота BotFather надає токен API, який буде використовуватися для інтеграції бота з проєктом. Зберігаємо цей токен у безпечному місці, оскільки він необхідний для налаштування бота. Вказуємо короткий опис, який буде відображатися користувачам. Також можемо налаштувати фотографію профілю для цього бота за допомогою команди /setuserpic (Рис 3. 3).

Рис 3. 2. Початок створення Telegram - бота

Рис 3. 3. Команди для реалізації

Після таких налаштувань, було надано інформацію про створення бота,

автоматичне створення посилання на нього, та токен з яким далі можна працювати (Рис 3. 4).

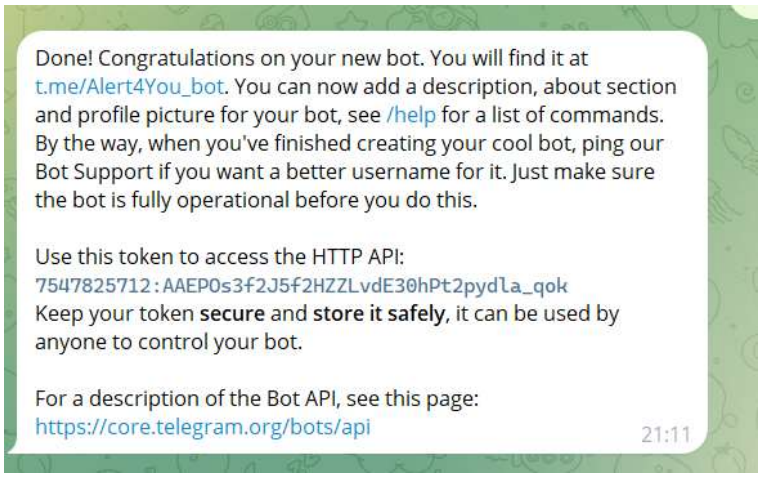

Рис 3. 4. Інформація про створення

Перейшовши за посиланням ми маємо базу, з якою надалі будемо працювати, наразі бот «чистий», він не виконує команди, і не відповідає користувачам (Рис. 3. 1).

Рис 3. 5. Інтерфейс бота

Після створення базової частини ми переходимо до написання

основного коду чат-бота. Спочатку ми імпортуємо всі необхідні бібліотеки, які забезпечують інтеграцію з Telegram API, роботу з базою даних, логування та обробку дат (Рис 3. 6).

```
import telebot
import logging
from telebot import types
import sqlite3
from datetime import datetime
```

Рис 3. 6. Імпортування необхідних бібліотек

Бібліотека telebot використовується для роботи з Telegram API, що дозволяє створювати та керувати ботами. Бібліотека logging дозволяє вести логи подій, що відбуваються під час роботи бота, а модуль types містить різні типи повідомлень та кнопок для взаємодії з користувачами. Бібліотека sqlite3 використовується для роботи з базою даних SQLite, а бібліотека datetime допомагає працювати з датами та часом.

Далі відбувається ініціалізація бота за допомогою API токену, який отриманий від BotFather під час створення бота, а також налаштування логування (Рис 3. 7).

```
API_TOKEN = '7547825712:AAEPOs3f2J5f2HZZLvdE30hPt2pydla_qok'
bot = telebot.TeleBot(API_TOKEN)

# Логування
logging.basicConfig(format='%(asctime)s - %(name)s - %(levelname)s - %(message)s', level=logging.INFO)
```

Рис 3. 7. Ініціалізація та логування

На цьому етапі створюється об'єкт bot, який дозволяє взаємодіяти з Telegram API, використовуючи API токен. Логування налаштовується для зберігання інформації про події та помилки, що виникають під час роботи бота.

Після цього відбувається створення бази даних SQLite і дві таблиці: users та transactions. Таблиця users зберігає інформацію про користувачів,

включаючи їх ідентифікатори, імена, електронні адреси та номери телефонів, а таблиця transactions зберігає дані про транзакції, такі як тип товару, сума транзакції, час транзакції та статус шахрайства (Рис 3. 8).

```
# Створення бази даних
conn = sqlite3.connect('transactions.db', check_same_thread=False)
cursor = conn.cursor()

cursor.execute('''CREATE TABLE IF NOT EXISTS users
                  (user_id TEXT PRIMARY KEY, name TEXT, email TEXT, phone TEXT)''')
cursor.execute('''CREATE TABLE IF NOT EXISTS transactions
                  (id INTEGER PRIMARY KEY, user_id TEXT, transaction_time TEXT, item_type TEXT, amount REAL, is_fraud BOOLEAN,
                  FOREIGN KEY (user_id) REFERENCES users (user_id))''')
conn.commit()

user_data = {}
user_transactions = {}
```

Рис 3. 8. Створення бази даних

Після створення бази даних йде визначення словника acceptable_price_ranges, який містить діапазони допустимих цін для різних типів товарів. Цей словник використовується для перевірки підозрілих транзакцій, і у кожному магазині він буде різний, взалежності від кількості та цін товарів. Наприклад: якщо онлайн-магазин продає побутові речі і в наявності є тільки телефон та комп’ютер, то магазин додає такі товари та діапазон цін, які можуть бути. Якщо ж цей магазин оновив асортимент, то відповідно оновлюється й словник (Рис 3. 9).

```
# Діапазони допустимих цін
acceptable_price_ranges = {
    'футболка': (100, 1000),
    'холодильник': (5000, 50000),
    'телевізор': (3000, 50000),
    'ноутбук': (10000, 100000),
    'смартфон': (5000, 50000)
}
```

Рис 3. 9. Словник

Далі створюється функція is_suspicious_transaction, яка аналізує транзакції і визначає, чи є вони підозрілими. Ця функція порівнює суму транзакції з допустимим діапазоном цін для певного типу товару. Якщо сума транзакції виходить за межі встановленого діапазону цін, транзакція

позначається як підозріла(Рис 3. 10).

```
# Функція перевірки підозрілих транзакцій
def is_suspicious_transaction(transaction):
    item_type = transaction['item_type']
    amount = transaction['amount']

    if item_type in acceptable_price_ranges:
        min_price, max_price = acceptable_price_ranges[item_type]
        return amount < min_price or amount > max_price
    return False
```

Рис 3. 10. Функція перевірки підозрілих транзакцій

Ця функція повертає True, якщо транзакція є підозрілою, і False, якщо ні.

Для забезпечення взаємодії з користувачами створюється головне меню бота, яке містить кнопки для реєстрації, перевірки транзакцій та повідомлення про шахрайство. Використовується метод ReplyKeyboardMarkup, що дозволяє створити інтерактивні кнопки в чаті Telegram. Головне меню відправляється користувачеві після команд /start або /help (Рис 3. 11).

```
# Головне меню
def main_menu(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    markup.add("/register", "/check_transaction", "/report_fraud")
    bot.send_message(message.chat.id, "Виберіть дію:", reply_markup=markup)
```

Рис 3. 11. Реалізація головного меню

Процес реєстрації користувачів включає збирання інформації про них та збереження в базу даних. Після запуску команди /register бот запитує у користувача ім'я, електронну адресу та номер телефону. Введена інформація зберігається в базу даних. Реєстрація користувачів реалізована через серію послідовних кроків, де кожен крок викликає наступну функцію для збору наступного поля даних (Рис 3. 12). Збереження даних відбувається у

відповідності до структури бази, забезпечуючи їх доступність для подальшої обробки. При успішній реєстрації користувач отримує підтверджуюче повідомлення з деталями внесених даних.

```
@bot.message_handler(commands=['start', 'help'])
def send_welcome(message):
    main_menu(message)

@bot.message_handler(commands=['register'])
def register(message):
    msg = bot.reply_to(message, "Введіть ваше ім'я")
    bot.register_next_step_handler(msg, get_name)

def get_name(message):
    user_data['name'] = message.text
    msg = bot.reply_to(message, "Введіть ваш email")
    bot.register_next_step_handler(msg, get_email)

def get_email(message):
    user_data['email'] = message.text
    msg = bot.reply_to(message, "Введіть ваш номер телефону")
    bot.register_next_step_handler(msg, get_phone)

def get_phone(message):
    user_id = str(message.from_user.id)
    user_data['phone'] = message.text
    user_data['user_id'] = user_id
    cursor.execute("INSERT OR REPLACE INTO users (user_id, name, email, phone) VALUES (?, ?, ?, ?)",
                   (user_id, user_data['name'], user_data['email'], user_data['phone']))
    conn.commit()
    bot.reply_to(message, "Реєстрація успішна!")
    main_menu(message)
```

Рис 3. 12. Реалізація збору даних користувачів

Коли користувач вводить команду /check_transaction, ця функція перевіряє, чи зареєстрований користувач. Для цього виконується SQL-запит до бази даних, щоб знайти користувача за його user_id. Якщо користувач не зареєстрований, бот надсилає повідомлення з пропозицією зареєструватися (Рис 3. 13).

```
@bot.message_handler(commands=['check_transaction'])
def check_transaction(message):
    user_id = str(message.from_user.id)
    cursor.execute("SELECT * FROM users WHERE user_id = ?", (user_id,))
    user = cursor.fetchone()
    if not user:
        bot.reply_to(message, "Ви не зареєстровані. Спочатку виконайте /register.")
        return
```

Рис 3. 13. Перевірка користувача

Далі, якщо користувач зареєстрований, створюється запис у user_transactions для зберігання даних про транзакцію (Рис 3. 14).

```
user_transactions[user_id] = {'user_id': user_id}
msg = bot.reply_to(message, "Введіть тип товару (футболка, холодильник, телевізор, ноутбук, смартфон)")
bot.register_next_step_handler(msg, get_item_type)
```

Рис 3. 14. Функція user_transaction

Ця частина коду ініціалізує запис у словнику user_transactions для зберігання інформації про транзакцію та запитує у користувача тип товару. Далі викликається функція get_item_type, яка обробляє введення користувача. Функція get_item_type зберігає введений користувачем тип товару і запитує суму транзакції (Рис 3. 15).

```
def get_item_type(message):
    user_id = str(message.from_user.id)
    user_transactions[user_id]['item_type'] = message.text.lower()
    msg = bot.reply_to(message, "Введіть суму транзакції у гривнях")
    bot.register_next_step_handler(msg, get_transaction_amount)
```

Рис 3. 15. Функція get_item_type

Ця функція зберігає тип товару в словник user_transactions та запитує у користувача суму транзакції. Далі викликається функція get_transaction_amount, яка обробляє введену суму. Функція get_transaction_amount зберігає суму транзакції, час транзакції та перевіряє транзакцію на наявність шахрайства (Рис 3. 16).

```
def get_transaction_amount(message):
    user_id = str(message.from_user.id)
    try:
        amount_text = message.text.replace(",", ".")
        user_transactions[user_id]['amount'] = float(amount_text)
        user_transactions[user_id]['transaction_time'] = datetime.now().strftime('%Y-%m-%d %H:%M:%S')

        is_fraud = is_suspicious_transaction(user_transactions[user_id])

        cursor.execute("INSERT INTO transactions (user_id, transaction_time, item_type, amount, is_fraud) VALUES (?, ?, ?, ?, ?)",
                       (user_transactions[user_id]['user_id'], user_transactions[user_id]['transaction_time'],
                        user_transactions[user_id]['item_type'], user_transactions[user_id]['amount'], is_fraud))
        conn.commit()
```

Рис 3. 16. Функція get_transaction_amount

Ця функція зберігає введену суму транзакції, час транзакції та викликає

функцію is_suspicious_transaction для перевірки транзакції на шахрайство. Якщо сума введена некоректно, функція видасть помилку ValueError. Далі результати перевірки зберігаються в базу даних (Рис 3. 17).

```
        if is_fraud:
            bot.reply_to(message, "⚠️ Транзакція підозріла!")
        else:
            bot.reply_to(message, f"✔ Транзакція безпечна: {user_transactions[user_id]['item_type']} за {user_transactions[user_id]
['amount']} грн")
    except ValueError:
        bot.reply_to(message, "❌ Будь ласка, введіть коректну суму (лише цифри).")
```

Рис 3. 17. Функція is_suspicious_transaction

Якщо транзакція є підозрілою, бот надсилає попередження користувачу. Якщо ж ні - бот повідомляє, що транзакція безпечна. У разі помилки введення суми бот просить користувача ввести коректну суму. Наступною йде команда /report_fraud, яка дозволяє користувачам повідомляти про підозрілі транзакції. Ця функція запитує у користувача деталі про підозрілу транзакцію і передає цю інформацію для подальшої обробки функцією process_fraud_report. Функція process_fraud_report зберігає повідомлення про підозрілу транзакцію у файл. Вона зберігає введену користувачем інформацію про підозрілу транзакцію у файл fraud_reports.txt разом з ідентифікатором користувача. Після збереження інформації бот надсилає повідомлення користувачу, дякуючи за скаргу (Рис 3. 18).

```
@bot.message_handler(commands=['report_fraud'])
def report_fraud(message):
    msg = bot.reply_to(message, "Будь ласка, опишіть підозрілу транзакцію.")
    bot.register_next_step_handler(msg, process_fraud_report)

def process_fraud_report(message):
    report_info = message.text
    user_id = str(message.from_user.id)

    with open("fraud_reports.txt", "a") as file:
        file.write(f"User ID: {user_id}\nReport: {report_info}\n\n")

    bot.reply_to(message, "Дякуємо, ваша скарга зафіксована!")
```

Рис 3. 18. Функція report_fraud

Нарешті, ми запускаємо бота в режимі polling для обробки повідомлень

користувачів. Цей рядок коду забезпечує постійну роботу бота, очікуючи на нові повідомлення від користувачів і обробляючи їх без перерви (Рис 3. 19). Завдяки цьому кожне отримане повідомлення швидко аналізується та викликає відповідну функцію для реагування.

```
if __name__ == '__main__':
    bot.polling(none_stop=True)
```

Рис 3. 19. Режим polling

3.5. Результат розробки

Метод виявлення шахрайства, який було реалізовано в даному проєкті, має високу гнучкість і здатність адаптуватися до специфіки кожного окремого магазину. Це досягається завдяки можливості налаштування діапазонів цін, що відповідають асортименту та ціновій політиці конкретного продавця.

Оскільки ціни на одні й ті самі товари можуть варіюватися залежно від регіону, сезону чи інших зовнішніх факторів, запропонований метод враховує ці особливості. Для кожного магазину можуть бути створені унікальні параметри, які дозволяють врахувати коливання ринкових цін та різноманітність товарів. Наприклад, якщо в одному магазині ціновий діапазон на смартфони становить від 5 000 до 50 000 гривень, у іншому він може бути змінений відповідно до специфіки асортименту або клієнтської бази.

Ця можливість адаптації забезпечує не тільки точність у виявленні потенційно шахрайських транзакцій, але й мінімізує кількість хибних спрацювань. Таким чином, система може ефективно працювати у різних умовах, включаючи невеликі спеціалізовані магазини або великі торговельні мережі. Це робить її універсальним інструментом, який легко інтегрується у

будь-яку бізнес-модель.

Щоб почати користуватися нашим Telegram-ботом, потрібно відкрити застосунок Telegram на пристрої (мобільному чи комп'ютері) і знайти бота за його ім'ям або посиланням, яке надає адміністратор. У нашому випадку ми знаходимо посилання, клікаємо та бачимо перед собою бота з назвою «БезпечнаТорговля». Після цього просто натисніть кнопку "Start" або введіть команду /start, щоб розпочати (Рис 3. 20).

Рис 3. 20. Початок роботи Telegram-бота

Бот одразу покаже головне меню, яке пропонує кілька опцій. Якщо такі дії виконує новий користувач, спершу потрібно зареєструватися (Рис 3. 21).

Рис 3. 21. Обрання дії "Реєстрація"

Реєстрація дуже проста: бот задасть кілька запитань, як-от ваше ім'я, електронна адреса та номер телефону. Усе це потрібно для того, щоб бот міг

персоналізувати досвід і надати вам точні послуги (Рис 3. 22).

Рис 3. 22. Введення даних

Після реєстрації можна скористатися функцією перевірки транзакцій. Для цього потрібно натиснути відповідну кнопку в меню або ввести команду /check_transaction. Бот попросить вказати тип товару, пов'язаний із транзакцією (наприклад, "смартфон", "телевізор" тощо), а потім запитує суму транзакції. Як тільки така інформація була надана, бот швидко проаналізує транзакцію та повідомить, чи є вона підозрілою (Рис 3. 23).

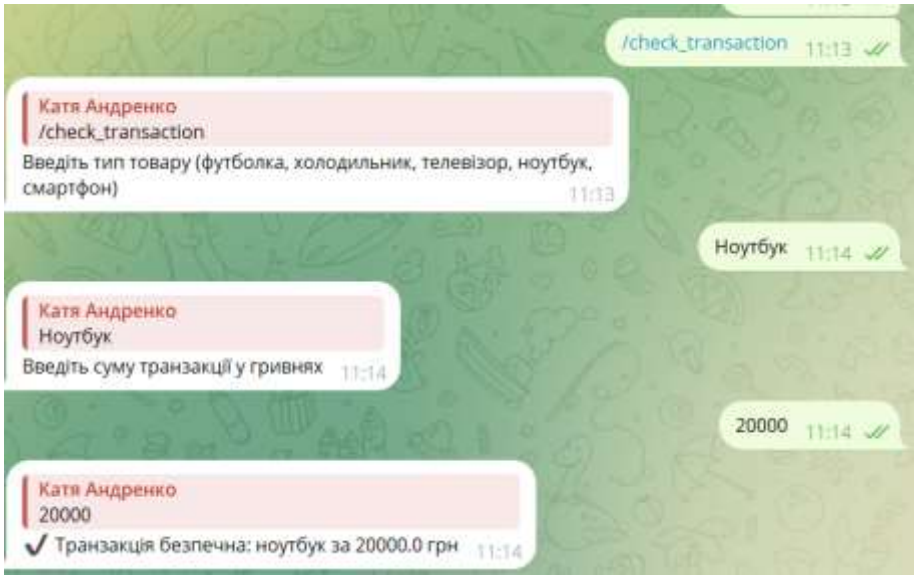

Рис 3. 23. Безпечна транзакція

На рисунку видно, що ноутбук за 20 000 грн є, тому бот видає повідомлення, що так, така транзакція може бути.

Тепер можна спробувати ввести той самий товар (ноутбук), але вже за 1000 грн (Рис 3. 24). Безпечна транзакція має значення, яке знаходиться в межах прийнятних діапазонів, і є типовою для даного товару. Це свідчить про відсутність підозрілих факторів. У свою чергу, підозріла транзакція має аномальну суму, яка виходить за рамки встановлених норм, що може бути ознакою шахрайства. Такий аналіз дозволяє оперативно визначати можливі ризики та своєчасно реагувати на них. Таке порівняння показує, як чат-бот може бути використаний для ідентифікації проблемних можливостей, що підвищує безпеку фінансових операцій.

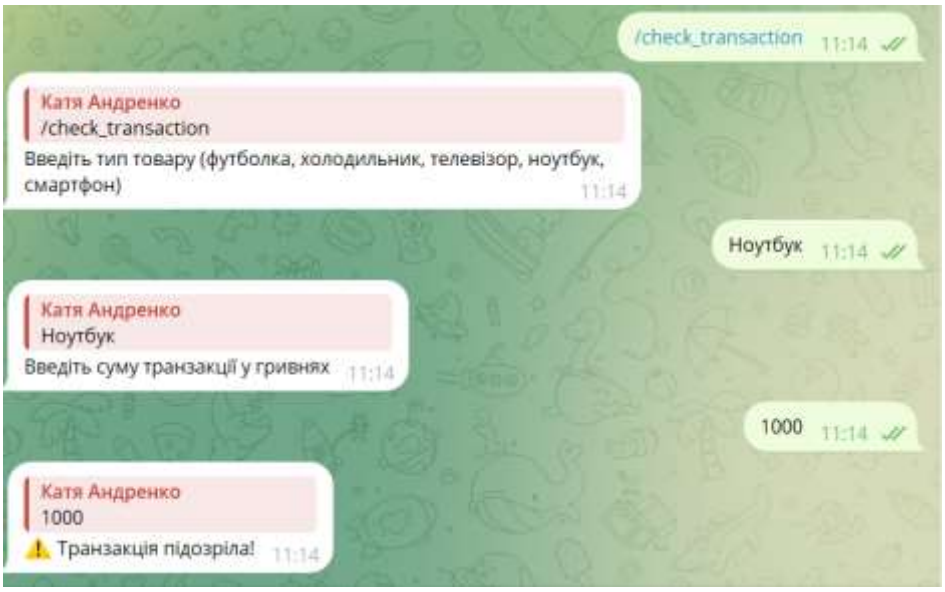

Рис 3. 24. Підозріла транзакція

Після повідомлення, що транзакція підозріла, користувач одразу може натискати кнопку /report_fraud, що означає «повідомити про пошурення». В полі, для введення інформації, можна вказувати, як і те порушення, що було зафіксовано щойно, так і те, з яким ви стикнулися та виявили підозру. Такий рядок корисний для онлайн-торгівлі з мережею магазинів, тому саме через цю кнопку можна вказати номер магазину або адресу, де було зафіксовано порушення (Рис 3. 25).

Після того, як бот фіксує порушення або отримує скаргу від користувача, вся інформація зберігається у спеціально призначеному

текстовому файлі. Цей файл слугує своєрідним блокнотом, до якого працівники магазину мають доступ для подальшого аналізу та обробки отриманих скарг.

Коли користувач повідомляє про підозрілу транзакцію, бот автоматично додає запис у цей файл. У записі міститься унікальний ідентифікатор користувача (наприклад, його Telegram ID), деталі транзакції та опис підозри, що було надано користувачем. Завдяки такій організації всі скарги структуровані та зберігаються в одному місці, що полегшує роботу працівників магазину (Рис 3. 26).

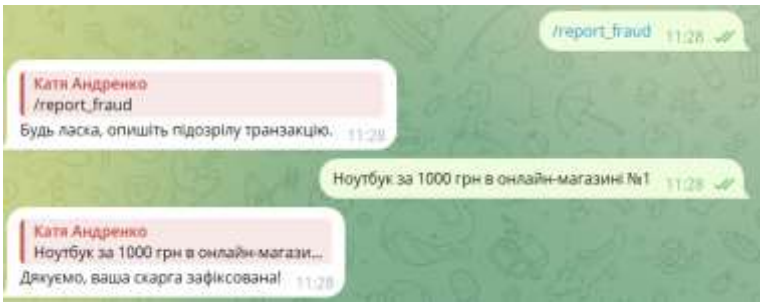

Рис 3. 25. Інформація про порушення

Працівники магазину можуть регулярно перевіряти вміст цього блокнота і фіксувати кожне порушення, а також вживати необхідних заходів - наприклад, перевіряти дані про транзакцію чи контактувати з відповідними клієнтами для уточнення деталей. Такий підхід дозволяє не лише автоматизувати збір інформації, а й оперативно реагувати на потенційні випадки шахрайства, підвищуючи загальний рівень довіри та безпеки.

Рис 3. 26. Скарги користувачів

Наразі запуск чат-бота здійснюється за допомогою командного рядка. Для цього користувач повинен відкрити термінал, перейти в каталог, де знаходиться скрипт бота, і виконати команду запуску, наприклад: python

app.py. Такий підхід є стандартним і підходить для локального тестування або роботи з ботом у невеликих масштабах. Після виконання команди бот починає працювати, підключаючись до серверів Telegram і очікує на взаємодію з користувачами (Рис 3. 27). Однак є кілька нюансів, які роблять цей спосіб менш оптимальним для постійної експлуатації.

Рис 3. 27. Запуск бота за допомогою командного рядка

Основна проблема полягає в тому, що бот припиняє свою роботу, як тільки закривається термінал або виникає непередбачена помилка, наприклад, перебої з інтернет-з'єднанням. У таких випадках бот не має вбудованого механізму автоматичного перезапуску. Це означає, що користувач повинен вручну запускати його знову, що може призводити до простоїв і знижувати ефективність роботи системи, особливо в умовах комерційної експлуатації, де потрібна постійна доступність бота.

Щоб розв'язати цю проблему і забезпечити постійну роботу бота, можна реалізувати декілька підходів до автоматизації запуску. Один із найпоширеніших підходів - використання системи управління процесами, такої як systemd, у середовищі Linux. Цей інструмент дозволяє налаштувати автоматичний запуск бота під час завантаження операційної системи, а також забезпечує автоматичний перезапуск у разі збою. Для цього створюється спеціальний файл сервісу, де вказується шлях до скрипта бота, параметри його запуску та режими роботи. Після цього бот додається до автозавантаження системи, що дозволяє забезпечити його безперебійну роботу.

Ще одним варіантом є використання менеджера процесів, такого як pm2. Хоча цей інструмент переважно використовується для Node.js

застосунків, він також підходить для скриптів на Python. pm2 дозволяє запускати бот у фоновому режимі, зберігати журнали його роботи і автоматично перезапускати скрипт у разі виникнення помилок.

Для комерційного використання або роботи з великим навантаженням можна розгорнути бот у хмарному середовищі, наприклад, на платформах AWS, Google Cloud або Heroku. Такий підхід забезпечує високу доступність і дозволяє масштабувати систему залежно від зростання навантаження. Крім того, деплой у хмарному середовищі забезпечує додаткові інструменти для моніторингу та управління ботом.

Ще одним сучасним підходом є контейнеризація за допомогою Docker. Це дозволяє ізолювати середовище бота від операційної системи, спрощуючи його розгортання на будь-якій платформі. У такому випадку створюється Docker-файл, де описується вся необхідна конфігурація для роботи бота. Після цього бот запускається у вигляді контейнера, який працює незалежно від локальних налаштувань системи.

Модернізація процесу запуску дозволить не лише автоматизувати роботу бота, але й забезпечить його надійність і доступність у будь-яких умовах. Це особливо важливо для систем, які покликані функціонувати цілодобово без простоїв і втручання з боку адміністратора.

ВИСНОВКИ

В рамках виконання роботи було створено Telegram-бот для виявлення шахрайства у сфері онлайн-торгівлі, що базується на простоті використання та інтуїтивності взаємодії. Чат-бот реалізує функції реєстрації користувачів, перевірки транзакцій на наявність аномалій та можливості повідомлення про підозрілі випадки. Під час розробки було враховано важливість автоматизації процесів аналізу транзакцій та створення адаптивної системи, яка може бути застосована для різних типів магазинів і товарів.

Основною метою роботи було забезпечення доступного інструмента для покращення фінансової безпеки користувачів та підприємств. У ході реалізації була використана бібліотека telebot, яка дозволяє інтегрувати функціонал бота із платформою Telegram. Бот взаємодіє з базою даних SQLite, де зберігається інформація про користувачів та їхні транзакції, що забезпечує структурованість та надійність системи.

Реалізований алгоритм перевірки транзакцій базується на порівнянні введеної користувачем суми із заданими діапазонами, які адаптуються до специфіки кожного магазину. Наприклад, для товарів, таких як смартфони, встановлено межі від 5 000 до 50 000 гривень, що відповідає середньо ринковим ціновим категоріям. Якщо сума транзакції значно відхиляється від цих меж, транзакція позначається як підозріла. Такий підхід є універсальним і може бути налаштований для різних умов використання залежно від коливань цін і асортименту.

Окремою перевагою системи є можливість автоматичної фіксації скарг користувачів. Усі повідомлення про шахрайство зберігаються у текстовому файлі, що виконує функцію блокнота для працівників магазину. Це дозволяє оперативно аналізувати потенційно небезпечні випадки та вживати необхідних заходів для їх усунення. Подібна організація роботи сприяє підвищенню рівня безпеки та зменшенню ризиків шахрайства.

Для запуску бота використовується команда через термінал, що

підходить для тестування, але на практиці було запропоновано методи автоматизації роботи, такі як налаштування автозапуску за допомогою systemd, використання pm2 для фонової роботи чи контейнеризація з Docker. Це дозволить інтегрувати бота в реальні бізнес-процеси та забезпечити його безперервну роботу.

Результати даної роботи є актуальними в умовах постійного зростання ризиків кіберзагроз і збільшення кількості онлайн-транзакцій. Розроблений Telegram-бот є інноваційним рішенням для забезпечення безпеки фінансових операцій, який об’єднує автоматизацію, інтуїтивність і адаптивність. Отримані результати можуть бути використані як основа для подальшого вдосконалення системи та її впровадження на практиці.

ПЕРЕЛІК ПОСИЛАНЬ

1. Андренко К. В. Вплив соціальних мереж на психічне здоров’я людини // XIV Міжнародна науково-практична конференція молодих учених та студентів «Наукові дослідження молоді з проблем європейської інтеграції», 4 квітня, м. Харків, 2025 р., С. 382-384.

2. Яненко О. С. Цифрова криміналістика як засіб ідентифікації кіберзлочинів // XIV Міжнародна науково-практична конференція молодих учених та студентів «Наукові дослідження молоді з проблем європейської інтеграції», 4 квітня, м. Харків, 2025 р., С. 423-424.

3. Фаткулін В. В. Кібербезпека: технології та тренди, що формують сучасний стан безпеки // Матеріали всеукраїнської науково-практичної студентської конференції «ІТ-простір сьогодення: тенденції, інновації та перспективи розвитку», 16 жовтня, м. Харків, 2024 р., С. 67-69.

4. Воробйов І. О. Розробка та використання телеграм боту // Матеріали всеукраїнської науково-практичної студентської конференції «ІТ-простір сьогодення: тенденції, інновації та перспективи розвитку», 16 жовтня, м. Харків, 2024 р., С. 137-138.

5. Шахрайство в інтернеті під час продажу товару. [Електронний ресурс] Режим доступу: https://torgsoft.ua/articles/safety/internet-shahrajstvo/

6. Telegram API Documentation. Офіційна документація щодо Telegram Bot API. [Електронний ресурс] Режим доступу: https://core.telegram.org/bots/api?form=MG0AV3

7. Microsoft Security. "Що таке кібербезпека?" Інформація про основні загрози та методи захисту. [Електронний ресурс] Режим доступу: https://www.microsoft.com/uk-ua/security/business/security-101/what-is-cybersecurity?form=MG0AV3

8. ТОП-5 видів інтернет-шахрайств та як від них вберегтися. [Електронний ресурс] Режим доступу: https://lutsk.rayon.in.ua/news/84924-top-5-vidiv-internet-shahraistv-ta-iak-vid-nih-vberegtisia

9. Які види інтернет-шахрайств найбільш поширені. [Електронний ресурс] Режим доступу: https://molbuk.ua/chernovtsy_news/314109-yaki-vydy-internet-shakhraistva-naibilsh-poshyreni.html

10. Інформаційні технології у сучасному суспільстві. [Електронний ресурс] Режим доступу: https://veche.kiev.ua/news/519684/

11. IT і бізнес: Як технології впливають на розвиток сучасних підприємств. [Електронний ресурс] Режим доступу: https://lemon.school/blog/it-i-biznes-yak-tehnologiyi-vplyvayut-na-rozvytok-suchasnyh-pidpryyemstv

12. Кіберзлочинність та кібершахрайство в умовах воєнного стану. [Електронний ресурс] Режим доступу: http://lsej.org.ua/11_2022/132.pdf

13. Куди звертатися при інтернет шахрайстві? [Електронний ресурс] Режим доступу: https://protocol.ua/ua/kudi_zvertatisya_pri_internet_shahraystvi/

ДОДАТОК

КОД НАПИСАННЯ ПРОГРАМИ

```
import telebot
import logging
from telebot import types
import sqlite3
from datetime import datetime
API_TOKEN = '7547825712:AAEPOs3f2J5f2HZZLvdE30hPt2pydla_qok'
bot = telebot.TeleBot(API_TOKEN)
# Логування
logging.basicConfig(format='%(asctime)s - %(name)s - %(levelname)s - %(message)s', level=logging.INFO)
# Створення бази даних
conn = sqlite3.connect('transactions.db', check_same_thread=False)
cursor = conn.cursor()
cursor.execute('''CREATE TABLE IF NOT EXISTS users
                  (user_id TEXT PRIMARY KEY, name TEXT, email TEXT, phone TEXT)''')
cursor.execute('''CREATE TABLE IF NOT EXISTS transactions
                  (id INTEGER PRIMARY KEY, user_id TEXT, transaction_time TEXT, item_type TEXT, amount REAL, is_fraud BOOLEAN,
                  FOREIGN KEY (user_id) REFERENCES users (user_id))''')
conn.commit()
user_data = {}
user_transactions = {}
# Діапазони допустимих цін
acceptable_price_ranges = {
    'футболка': (100, 1000),
    'холодильник': (5000, 50000),
    'телевізор': (3000, 50000),
    'ноутбук': (10000, 100000),
    'смартфон': (5000, 50000)
}
# Функція перевірки підозрілих транзакцій
def is_suspicious_transaction(transaction):
    item_type = transaction['item_type']
    amount = transaction['amount']
    if item_type in acceptable_price_ranges:
        min_price, max_price = acceptable_price_ranges[item_type]
        return amount < min_price or amount > max_price
    return False
```

```
# Головне меню
def main_menu(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    markup.add("/register", "/check_transaction", "/report_fraud")
    bot.send_message(message.chat.id, "Виберіть дію:", reply_markup=markup)
@bot.message_handler(commands=['start', 'help'])
def send_welcome(message):
    main_menu(message)
@bot.message_handler(commands=['register'])
def register(message):
    msg = bot.reply_to(message, "Введіть ваше ім'я")
    bot.register_next_step_handler(msg, get_name)
def get_name(message):
    user_data['name'] = message.text
    msg = bot.reply_to(message, "Введіть ваш email")
    bot.register_next_step_handler(msg, get_email)
def get_email(message):
    user_data['email'] = message.text
    msg = bot.reply_to(message, "Введіть ваш номер телефону")
    bot.register_next_step_handler(msg, get_phone)
def get_phone(message):
    user_id = str(message.from_user.id)
    user_data['phone'] = message.text
    user_data['user_id'] = user_id
    cursor.execute("INSERT OR REPLACE INTO users (user_id, name, email,
phone) VALUES (?, ?, ?, ?)",
                   (user_id, user_data['name'], user_data['email'], user_data['phone']))
    conn.commit()
    bot.reply_to(message, "Реєстрація успішна!")
    main_menu(message)
@bot.message_handler(commands=['check_transaction'])
def check_transaction(message):
    user_id = str(message.from_user.id)
    cursor.execute("SELECT * FROM users WHERE user_id = ?", (user_id,))
    user = cursor.fetchone()
    if not user:
        bot.reply_to(message, "Ви не зареєстровані. Спочатку виконайте
/register.")
        return
    user_transactions[user_id] = {'user_id': user_id}
    msg = bot.reply_to(message, "Введіть тип товару (футболка, холодильник,
```

```
телевізор, ноутбук, смартфон)")
    bot.register_next_step_handler(msg, get_item_type)
def get_item_type(message):
    user_id = str(message.from_user.id)
    user_transactions[user_id]['item_type'] = message.text.lower()
    msg = bot.reply_to(message, "Введіть суму транзакції у гривнях")
    bot.register_next_step_handler(msg, get_transaction_amount)
def get_transaction_amount(message):
    user_id = str(message.from_user.id)
    try:
        amount_text = message.text.replace(",", ".")
        user_transactions[user_id]['amount'] = float(amount_text)
        user_transactions[user_id]['transaction_time'] = datetime.now().strftime('%Y-
%m-%d %H:%M:%S')
        is_fraud = is_suspicious_transaction(user_transactions[user_id])
        cursor.execute("INSERT INTO transactions (user_id, transaction_time,
item_type, amount, is_fraud) VALUES (?, ?, ?, ?, ?)",
                       (user_transactions[user_id]['user_id'],
user_transactions[user_id]['transaction_time'],
                        user_transactions[user_id]['item_type'],
user_transactions[user_id]['amount'], is_fraud))
        conn.commit()
        if is_fraud:
            bot.reply_to(message, "⚠ Транзакція підозріла!")
        else:
            bot.reply_to(message, f"✔ Транзакція безпечна:
{user_transactions[user_id]['item_type']} за
{user_transactions[user_id]['amount']} грн")
    except ValueError:
        bot.reply_to(message, "✖ Будь ласка, введіть коректну суму (лише
цифри).")
@bot.message_handler(commands=['report_fraud'])
def report_fraud(message):
    msg = bot.reply_to(message, "Будь ласка, опишіть підозрілу транзакцію.")
    bot.register_next_step_handler(msg, process_fraud_report)
def process_fraud_report(message):
    report_info = message.text
    user_id = str(message.from_user.id)
    with open("fraud_reports.txt", "a") as file:
        file.write(f"User ID: {user_id}\nReport: {report_info}\n\n")
    bot.reply_to(message, "Дякуємо, ваша скарга зафіксована!")
```

```
if __name__ == '__main__':
    bot.polling(none_stop=True)
```