НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

До захисту допущено
Завідувач кафедри

_____ Дмитро ЛАНДЕ

(підпис)

«_____»_____2024 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системи, технології та математичні методи кібербезпеки»
спеціальності 125 «Кібербезпека»

на тему: ПІДСИСТЕМА ПОШУКУ АНОМАЛІЙ В СИСТЕМІ АНАЛУЗУ ПОВЕДІНКИ КОРИСТУВАЧІВ

Виконав (-ла): здобувач вищої освіти **IV** курсу, групи ФБ-04
(шифр групи)

Сербіненко Олексій Григорович
(прізвище, ім'я, по батькові) (підпис)

Керівник к.т.н., доцент каф. ІБ Коломицев М.В.
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові) (підпис)

Рецензент
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по батькові) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Здобувач вищої освіти _____
(підпис)

Київ – 2024 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 125 «Кібербезпека»
Освітньо-професійна програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ Дмитро ЛАНДЕ
(підпис)

«___»_____2024 р.

ЗАВДАННЯ
на дипломну роботу здобувачу вищої освіти

(прізвище, ім’я, по батькові)

1. Тема роботи “Підсистема пошуку аномалій в системі аналізу поведінки користувачів”,

керівник роботи Коломицев Михайло Володимирович, доцент кафедри ІБ.,
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від _____ _____ 2024 р. №

2. Термін подання здобувачем вищої освіти роботи ___ _____ 2024 р.

3. Вихідні дані до роботи: вихідні дані для моєї дипломної роботи включають логи активності користувачів, зібрані з інформаційних систем для аналізу поведінки користувачів і виявлення аномалій.

4. Зміст роботи: розробка підсистеми пошуку аномалій для аналізу поведінки користувачів в інформаційних системах, з використанням алгоритму Isolation Forest для ефективного виявлення аномальних дій.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація

6. _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Отримання завдання	20.03.2024	виконано
2	Пошук інформаційних джерел	20.03.2024 – 20.04.2024	виконано
3	Огляд внутрішніх загроз та методів захисту	20.04.2024 – 30.04.2024	виконано
4	Огляд алгоритмів пошуку аномалій	30.04.2024 – 25.04.2024	виконано
5	Порівняння алгоритмів	25.04.2024 - 25.05.2024	виконано
6	Розробка підсистеми пошуку аномалій	25.05.2024 – 12.06.2024	виконано
7	Предзахист дипломної роботи	10.06.2024	виконано
8	Захист дипломної роботи	17.06.2024	

Здобувач вищої освіти _____ Олексій Сербіненко
(підпис) **(Власне ім'я, ПРІЗВИЩЕ)**

Керівник роботи _____ Михайло Коломицев
(підпис) **(Власне ім'я, ПРІЗВИЩЕ)**

РЕФЕРАТ

Дана робота обсягом 45 сторінок, включає 15 рисунків, 1 таблицю, 2 додатки та містить посилання на 5 джерел літератури.

Об’єктом дослідження є діяльність користувачів в інформаційних системах.

Предметом дослідження є алгоритми машинного навчання для виявлення аномальної поведінки серед великих наборів даних, зокрема, Isolation Forest.

Метою роботи є розробка підсистеми виявлення аномалій у поведінці користувачів корпоративних систем на основі алгоритмів машинного навчання.

В ході виконання роботи були розглянуті існуючі рішення в виявленні внутрішніх загроз та області виявлення аномалій. Було розглянуто різні алгоритми пошуку аномалій та порівняно їх та обрано найпідходящий для розробки підсистеми пошуку аномалій. Також було розроблено підсистему пошуку аномалій.

ABSTRACT

This work is 45 pages long, including 15 figures, 1 table, 2 appendices and contains references to 5 literature sources.

The object of research is the activity of users in information systems.

The subject of research is machine learning algorithms for detecting anomalous behavior among large data sets, in particular Isolation Forest.

The purpose of the work is to develop a subsystem for detecting anomalies in the behavior of users of corporate systems based on machine learning algorithms.

In the course of the work, existing solutions in the detection of internal threats and the area of anomaly detection were considered. Various anomaly detection algorithms were considered and compared, and the most suitable one was chosen for the development of the anomaly detection subsystem. An anomaly detection subsystem was also developed.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

UEBA - User and Entity Behavior Analytics

IDS - Intrusion Detection System

IPS - Intrusion Prevention System

SIEM - Security information and event management

One-Class SVM - One Class Support Vector Machines

ВСТУП

У звіті про інсайдерські загрози за 2020 рік від Cybersecurity-Insiders повідомляється, що 68% організацій помітили, що інсайдерські загрози почастішали протягом останнього року. Згідно зі звітом Verizon Data breach report за 2020 рік, для виявлення понад 25% порушень знадобилося 12 місяців, а то й більше, щоб їх помітити і виявити [4]. Тому я хочу зосередитись на внутрішніх загрозах та методах захисту.

У сучасному світі, де обсяг даних і складність інформаційних систем зростають з кожним днем, забезпечення безпеки цих систем як ніколи важливо. Особливо важливим є виявлення аномалій у поведінці користувачів, які можуть вказувати на внутрішні загрози, зловмисне використання або операційні помилки. Виявлення таких аномалій важливо для захисту від внутрішніх загроз і забезпечення надійності системи. Метою роботи є розробка підсистеми виявлення аномалій у поведінці користувачів корпоративних систем на основі алгоритмів машинного навчання.

Об'єктом дослідження є поведінка користувачів в інформаційних системах організацій.

Предметом дослідження є алгоритми машинного навчання для виявлення аномальної поведінки серед великих наборів даних, зокрема, Isolation Forest.

Наукова новизна моєї дипломної роботи полягає у розробці підсистеми пошуку аномалій, яка буде використовувати алгоритм машинного навчання Isolation Forest. Відмінністю цієї підсистеми буде її здатність адаптуватися до будови корпоративних мереж і поведінки користувачів, що дозволяє виявляти не лише явні відхилення в поведінці, а й менш помітні аномалії.

1 ОГЛЯД ВНУТРІШНІХ ЗАГРОЗ ТА МЕТОДІВ ЗАХИСТУ

1.1 Огляд внутрішніх загроз

Внутрішні загрози часто залишаються непоміченими довгий час і можуть завдати значної шкоди організації. На відміну від зовнішніх атак, внутрішні загрози важче ідентифікувати, оскільки вони можуть виникати у звичайному процесі роботи і можуть бути замасковані під звичайну діяльність. Крім того, внутрішні загрози можуть бути викликані як навмисними, так і ненавмисними діями співробітників, що ускладнює їх виявлення та запобігання.

Типи внутрішніх загроз від користувачів включають в себе різноманітні дії, які можуть негативно вплинути на бізнес. Серед них саботаж, який відбувається в умисних діях на шкоду компанії, наприклад, видалення важливих файлів. Шахрайство передбачає маніпулювання фінансовою інформацією або ресурсами. Крадіжка даних — це незаконне копіювання або передача конфіденційної інформації. Шкідливі дії після компроментації облікових записів в системі. Також людський фактор включає помилки при введенні даних і випадкове видалення важливої інформації, що також може завдати значної шкоди.

У 2008 році Жером Кервіль, трейдер французького банку Société Générale, завдяки свєму доступу до торгових систем зміг приховати збиткові операції, що призвело до втрати приблизно 4.9 мільярда євро. Кервіель зловживав довірою та слабкостями в системі контролю банку, створюючи фальшиві документи і вводячи в оману колег та керівництво. Цей інцидент викликав серйозні питання щодо рівня нагляду та контролю у фінансових інститутах та показав, як одна особа може викликати колосальні фінансові втрати використовуючі внутрішні загрози.

Цей випадок підкреслює важливість належного управління доступом, моніторингу та аналітики поведінки користувачів та вдосконалення процедур внутрішнього контролю, щоб мінімізувати ризики внутрішніх загроз.

1.2 Методи захисту від внутрішніх загроз

Захист від внутрішніх загроз вимагає комплексного підходу, який поєднує технологічні рішення, організаційну політику та постійну програму навчання співробітників. Одним із способів боротьби з інсайдерськими загрозами це реалізація принципу найменших привілеїв, котра забезпечує кожному працівнику доступ лише до тих ресурсів, які необхідні для виконання його обов'язків. Також важливе регулярне навчання персоналу для здобуття нових знань про існуючі загрози та навчання розпізнавати кіберзагрози та правильно на них реагувати.

Серед комплексних методів використовують рішення безпеки інформації та управління подіями (SIEM), які пов’язані з аналітикою поведінки користувачів і об’єктів (UEBA). Таким чином, вживаються заходи для моніторингу та аналізу подій через мережі та системи. Ці системи дозволяють постійно контролювати поведінку всіх користувачів, щоб перевірити, чи події відповідають політикам безпеки корпорацій, щоб сприяти посиленню запобігання підозрілим діям проти інформаційних систем. Наприклад, багаторазове введення неправильного пароля під час доступу до головної системи будь-якої компанії може призвести до блокування системи на хвилини та видачі відповідних сповіщень. Основна функція UEBA полягає у відстеженні поведінки користувачів протягом певного періоду часу, і такі дії будуть вважатись нормальним зразком поведінки користувача. Будь-яке відхилення від звичайної поведінки може вважатися ознакою потенційної внутрішньої загрози, тобто аномалією.
Як правило, система UEBA збирає дані про користувачів і дії суб’єктів із журналів систем. UEBA визначає базові моделі поведінки користувачів після цього, постійно відстежує поведінку суб’єкта порівнює її з нормальною,

базовою поведінкою, створеною раніше для того самого суб’єкта чи подібних суб’єктів, щоб виявити аномальну поведінку, використовуючи підсистему пошуку аномалій. Тим самим, завчасне ідентифікування аномалій зменшить ризики та кінцеву шкоду для організації.

1.3 Системи UEBA

1.3.1 Визначення UEBA

UEBA можна визначити як аналітику поведінки користувачів та організацій, що є процесом безпеки.

Основною функцією цієї системи є відстеження нормальної поведінки користувачів. Використовуючи машинне навчання знаходить аномалії в поведінці користувача і поведінках в інших компонентах, таких як мобільний, ПК, мережа, локальні додатки та загрози, які надходять ззовні організації. UEBA встановлює базові патерни поведінки для всіх і кожного об'єкта в середовищі, а подальші дії цієї системи будуть полягати в оцінці та вивчення шляхом порівняння з попередньо визначеними базовими показниками.

Основна концепція функції UEBA полягає в тому, щоб відповісти на питання яка поведінка користувача відповідає базовій лінії, а яка ні. Це допоможе виявити будь-які підозрілі інсайдерські загрози.

1.3.2 Переваги та недоліки UEBA

Використання системи UEBA з активами корпорації може покращити та посилити функцію безпеки, завчасно протидіяти внутрішнім та зовнішнім загрозам, підтримати моніторинг діяльності організації.

Основні переваги використання UEBA включають наступне:

- UEBA використовується для скорочення середнього часу реагування: однією з основних переваг UEBA є надання достатнього час для команди безпеки, щоб відреагувати на внутрішню загрозу, оскільки UEBA має

високий показник лояльності мінімізувати ризики та скоротити час, необхідний для реагування на атаки.

- Мінімізація ризику безпеки: раннє виявлення порушення даних зменшить ризик, втрату даних та збитки від загроз.

- Автоматизоване виявлення загроз: ніхто не має повного досвіду та знань у сфері безпеки, щоб відстежувати усі події в системі, використовуючи машинне навчання та поведінкову аналітику, організація може врахувати брак досвіду у аналітиків з кібербезпеки, і вони можуть покращити ресурси, які вже існують, для покращення виявлення загроз.

- Зменшення кількості хибнопозитивних сповіщень: допомогти адміністраторам зосередитися на реальних випадках порушення нормальної активності та надавати більш пріоритетне реагування на інциденти, які є найбільш важливими для організації.

Використання методів машинного навчання пов'язане з багатьма недоліками, такими як обмеження роботи з привілеями користувачів і розробників. Ці користувачі є особливим випадком, оскільки їхні робочі функції часто вимагають нестандартної поведінки, і це викликає труднощі при створенні базової лінії алгоритмів. Інший недолік UEBA полягає в тому, що вони не можуть вказати на довгострокові складні та повільні атаки, оскільки вони не мають щоденного впливу і стають немовби неіснуючими.

Техніки машинного навчання вже давно використовуються для виявлення аномалій у системах на базі Windows, дозволяючи створювати чіткі правила для виявлення небажаних дій у мережі. Сучасні дослідження [5] підкреслюють важливість технології UEBA у сфері кібербезпеки, особливо для зменшення помилкових сповіщень і нормалізації даних, що дозволяє робити прогнози на основі даних системних журналів. Ці методи допомагають вирішувати недоліки і підтримувати ключові компоненти безпеки, хоча людська участь залишається необхідною для аналізу подій та підтвердження їх небезпеки. Машинне навчання показує перспективи для майбутнього у виявленні та реагуванні на загрози, особливо для відстеження

поведінки користувачів і розслідування інцидентів, допомагаючи розробляти методи виявлення внутрішніх загроз.

Технології кібербезпеки та аналітика поведінки користувачів (UEBA) мають за мету захистити комп'ютерні системи від витоків даних і фізичних пошкоджень. Вони допомагають виявляти та запобігати загрозам, зокрема шахрайству. UEBA зосереджується на поведінці користувачів, аналізуючи їхні дії та виявляючи будь-які відхилення від звичайної активності, що може вказувати на внутрішні загрози.

UEBA збирає дані з журналів, пакетів та інших джерел всередині організації, включно з фаєрволами, антивірусними системами, та іншими компонентами мережевого управління. Дані нормалізуються та зберігаються для подальшого аналізу. Методи машинного навчання, які використовуються в UEBA, дозволяють точно аналізувати поведінку користувачів і виявляти аномалії. Це сприяє зниженню хибнопозитивних сповіщень і збільшує точність ідентифікації реальних загроз.

Поведінка користувачів постійно змінюється, і навколишнє середовище також еволюціонує, що вимагає від UEBA адаптації та постійного оновлення для ефективного виявлення і відслідковування аномальних дій в різноманітних джерелах даних.

Системи UEBA, що аналізують поведінку користувачів та організацій, використовують контекстуалізацію подій для більш точного ідентифікування аномалій, враховуючи такі фактори, як час доби та місцеположення. Ці системи ефективно інтегруються з іншими інструментами кібербезпеки, включаючи IDS/IPS, системи управління ідентичністю та доступом, та SIEM. Це дозволяє створювати комплексний захист і більш точно реагувати на потенційні загрози.

У сфері технологій, UEBA використовує розширені можливості машинного навчання, які охоплюють як наглядове, так і ненаглядове навчання для вдосконалення виявлення та реагування на аномалії. Однак існують виклики з точки зору приватності та етики, зокрема баланс між захистом організації

від внутрішніх загроз та забезпеченням приватності користувачів, що вимагає етичного збору та обробки даних.

1.3.3 Функції UEBA

1. Виявлення внутрішніх загроз: для виявлення співробітників в організації, які можуть викрасти дані та інформацію.
2. Виявлення скомпроментованих облікових записів.
3. Виявлення порушення захищених даних
4. Виявляйте кібер-зловмисників, виявляючи відхилення в безпеціє

1.3.4 Компоненти системи UEBA включають:

1. Збір даних: UEBA збирає дані з журналів, пакетів, і інших джерел всередині організації, включно з фаєрволами, антивірусними системами, і іншими компонентами мережевого управління.
2. Нормалізація та зберігання даних: Після збору дані нормалізуються та зберігаються в централізованій системі для подальшого аналізу.
3. Аналіз даних: Використовуючи алгоритми машинного навчання, система аналізує поведінку користувачів та виявляє аномалії, що можуть вказувати на внутрішні загрози.
4. Звітування та сповіщення: Система генерує звіти та сповіщення, які надсилаються співробітникам служби безпеки для подальшого реагування.

Висновки до першого розділу

У цьому розділі ми детально розглянули внутрішні загрози, ризики та втрати котрі вони несуть. Визначили різні типи внутрішніх загроз, які можуть включати саботаж, шахрайство, крадіжку даних і людські помилки.

Дослідили методи боротьби з внутрішніми загрозами серед яких, наприклад, реалізація принципу найменших привілеїв, регулярне навчання персоналу. Також оглянули комплексні рішення, серед яких такі системи як SIEM та UEBA. Також було описано як системи аналітики поведінки користувачів (UEBA) і алгоритми виявлення аномалій допомагають ідентифікувати внутрішні загрози.

За допомогою таких систем ми маємо здатність не тільки реагувати на інциденти, але й передбачати виявлені загрози, зменшуючи можливість збитків для організації.

2 ПІДСИСТЕМА ПОШУКУ АНОМАЛІЙ ТА АЛГОРИТМИ

2.1 Існуючі алгоритми пошуку аномалій

Системи UEBA використовують підсистеми пошуку аномалій, реалізовані за допомогою алгоритмів машинного навчання. Далі розгляну декілька алгоритмів.

2.1.1 Isolation Forest

Опис та принцип дії: Isolation Forest (iForest) є алгоритмом, який працює методом ізоляції точок даних в дереві рішень. На відміну від традиційних методів, які шукають нормальність, iForest концентрується на швидкому виявленні аномалій, використовуючи механізм випадкових лісів для ізоляції аномальних точок. Цей алгоритм працює за принципом, що аномалії легше ізолювати від нормальних спостережень, тому що вони менш чисельні і зазвичай мають атрибути, що сильно відрізняються від нормальних даних.

Типи аномалій, на які націлен: Isolation Forest ефективний для виявлення різноманітних аномалій, зокрема зловживань з даними, несанкціонованого доступу та інших нетипових дій у системах, де аномалії не вписуються в загальний розподіл нормальних даних. Це робить його особливо цінним для виявлення внутрішніх загроз, таких як інсайдерські атаки чи несанкціоновані дії з боку користувачів системи.

Механіка роботи: Процес роботи Isolation Forest полягає в створенні дерев ізоляції. Випадковим чином вибирається атрибут і розбивається випадковим чином на діапазони, до тих пір, поки кожна точка даних не буде "ізольована" у власному "листку" дерева. Аномалії зазвичай ізолюються швидше через їхні рідкісні або відмінні характеристики, тому шляхи до таких "листків" коротші порівняно з нормальними точками.

Плюси Isolation Forest:

Швидкість виявлення аномалій: Isolation Forest ефективно виявляє аномалії, навіть у великих наборах даних, завдяки своєму механізму випадкового розділення, який ізолює точки даних без потреби проходити через всі дані.

Мала чутливість до шуму: На відміну від багатьох інших алгоритмів, Isolation Forest може ефективно розрізняти шум і реальні аномалії, знижуючи кількість помилкових спрацювань.

Не потребує визначення порогів: Алгоритм не вимагає від користувача заздалегідь визначати порогові значення для виявлення аномалій, що робить його ідеальним для застосувань, де важко визначити, що є "нормальним".

Скалярність і ефективність: Isolation Forest ефективно обробляє велику кількість записів, оскільки його алгоритмічна структура забезпечує швидке розділення даних на деревах, навіть якщо цих записів дуже багато.

Мінуси Isolation Forest:

Обмежена ефективність у високорозмірних даних: зі збільшенням кількості атрибутів в аналізі Isolation Forest стає все складніше визначити, які з них є важливими для розпізнавання аномалій, а дані стають більш розрідженими у просторі атрибутів. Це може призвести до того, що ізоляція аномалій займає більше часу і потребує більше обчислень.

Залежність від параметрів: Хоча алгоритм не вимагає визначення порогів, правильний вибір інших параметрів, таких як кількість дерев у лісі, все ж важливий для оптимізації його продуктивності.

Адаптація до нових даних: Isolation Forest може вимагати перенавчання для адаптації до змін у розподілі даних, що може бути ресурсозатратним у динамічних середовищах.

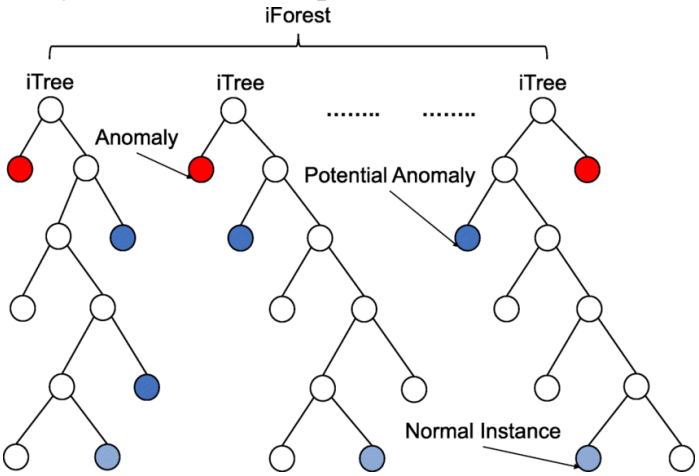

Рисунок 1.1 — Візуалізація роботи алгоритму **Isolation forest**

2.1.2 K-means Clustering

Опис та принцип дії: K-means Clustering — це алгоритм кластеризації, який групує дані на основі відстані між точками. Алгоритм намагається мінімізувати суму відстаней від кожної точки даних до центру найближчого їй кластеру. В контексті виявлення аномалій, аномалії часто з'являються як ізольовані точки, які не належать до основних кластерів або формують дуже маленькі кластери.

Типи аномалій, на які націлені: K-means часто використовується для виявлення групових аномалій, де аномальні дані можуть утворювати власні кластери, або точкових аномалій, які є далекими від будь-яких основних кластерів.

Механіка роботи: K-means починає з вибору кількості кластерів, а потім ітеративно оновлює їх положення, вираховуючи середні значення точок у кожному кластері, до тих пір, поки не буде досягнута конвергенція.

Плюси K-means Clustering:

Простота імплементації: Алгоритм легко реалізувати та інтегрувати в більшість систем аналітики.

Ефективність на помірно розмірних даних: Працює добре, коли розмірність даних і кількість кластерів не є надто великими.

Мінуси K-means Clustering:

Чутливість до вибору кількості кластерів: Неправильний вибір кількості кластерів може значно вплинути на якість кластеризації.

Чутливість до викидів: Викиди можуть значно впливати на розрахунок центрів кластерів, що веде до помилкових групувань.

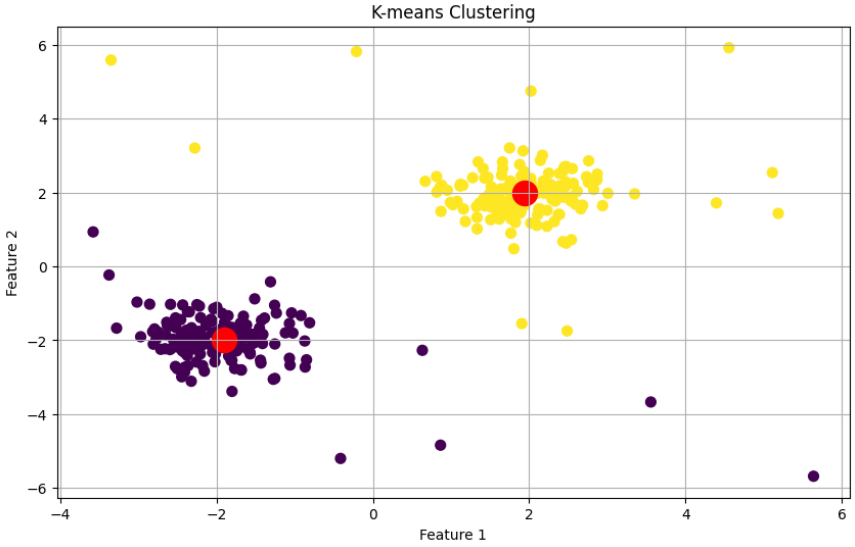

Рисунок 1.2 — Візуалізація роботи алгоритму **K-means Clustering**

2.1.3 One-Class SVM

Опис та принцип дії: One-Class SVM (Однокласова машина опорних векторів) — це спеціалізована версія алгоритму машини опорних векторів, яка використовується для виявлення аномалій. Вона тренується на даних, які представляють лише один клас (нормальні спостереження), і має за мету визначити межу цього класу. Якщо новий приклад даних падає за межу

встановлену моделлю, він вважається аномалією. Алгоритм ефективно використовує високорозмірний простір, куди проектують дані за допомогою ядрових функцій, щоб з легкістю відокремлювати аномалії від нормальних даних.

Типи аномалій, на які націлені: One-Class SVM особливо корисний для виявлення складних аномалій у високорозмірних даних, де інші алгоритми можуть зазнавати невдач через обмеження розмірності. Це робить його ідеальним для застосувань, де важливо виявляти нові, раніше невідомі типи аномалій.

Механіка роботи: One-Class SVM спочатку перетворює вхідні дані в більш високий вимірний простір за допомогою ядрової функції. Потім, використовуючи ці трансформовані дані, алгоритм шукає гіперплощину, яка максимально відділяє всі трансформовані точки від початку координат. Точки, які віддалені від цієї гіперплощини, визначаються як аномалії.

Плюси One-Class SVM:

Ефективність у високорозмірних даних: Може ефективно виявляти аномалії у великих і складних датасетах, де інші методи можуть бути неефективними.

Гнучкість у виборі ядра: Вибір різних ядерних функцій дозволяє адаптувати алгоритм до специфіки даних, підвищуючи його точність і чутливість до аномалій.

Мінуси One-Class SVM:

Висока обчислювальна складність: Необхідність обробки великих обсягів даних у високорозмірному просторі може бути дуже ресурсозатратною.

Чутливість до налаштування параметрів: Точність моделі значно залежить від вибору параметрів, таких як тип ядра і його налаштування, що вимагає глибокого розуміння як даних, так і алгоритму.

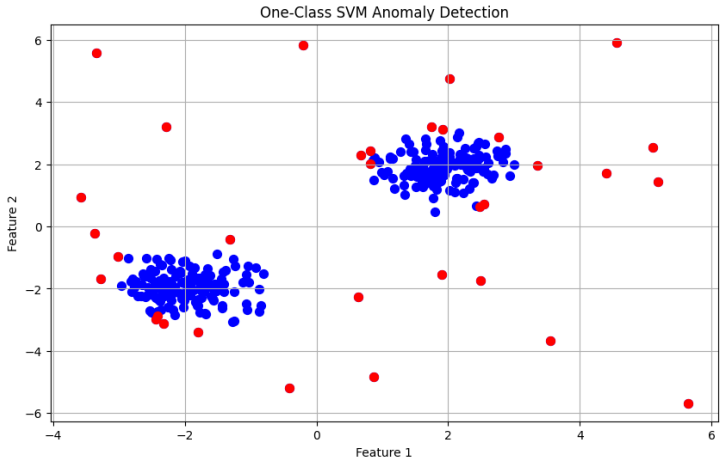

Рисунок 1.3 — Візуалізація роботи алгоритму **One-Class SVM**

2.2 Порівняння алгоритмів

Цей синтетичний набір даних було створено для оцінки продуктивності алгоритмів виявлення аномалій, включаючи Isolation Forest, One-Class SVM і K-Means Clustering. За основу було взято структуру схожу на поведінку користувачів в системах UEBA, тобто нормальні дані та аномалії можуть сильно відризнятись, поведінка користувачів може бути складною і мультиваріантною, Набір даних включає 1000 точок даних, з яких 10% (100 точок) є аномаліями. Аномалії менш щільно розподілені та мають більший розкид по обох осях, від -3 до 6, що робить їх більш різноманітними та непередбачуваними порівняно зі звичайними даними, які згруповані навколо точки [2, 2] і мають стандартне відхилення 0,5.

Використання такого набору даних у порівняльному аналізі алгоритмів виявлення аномалій є важливим для ілюстрації того, як різні методи можуть реагувати на змішані типи даних і як вони справляються з виявленням реальних і потенційних загроз у складних середовищах. Аномалії тут представляють потенційні загрози або винятки, які не вписуються в загальну

модель поведінки чи діяльності, і їх виявлення може допомогти запобігти можливим системним збоям або зловживанням.

Цей підхід дає можливість не тільки перевірити точність виявлення аномалій, але й оцінити, як алгоритми можуть відрізнити нормальну поведінку від потенційно шкідливих аномалій, що є ключовим для додатків у системах реального часу, де швидкість і точність відіграють вирішальну роль.

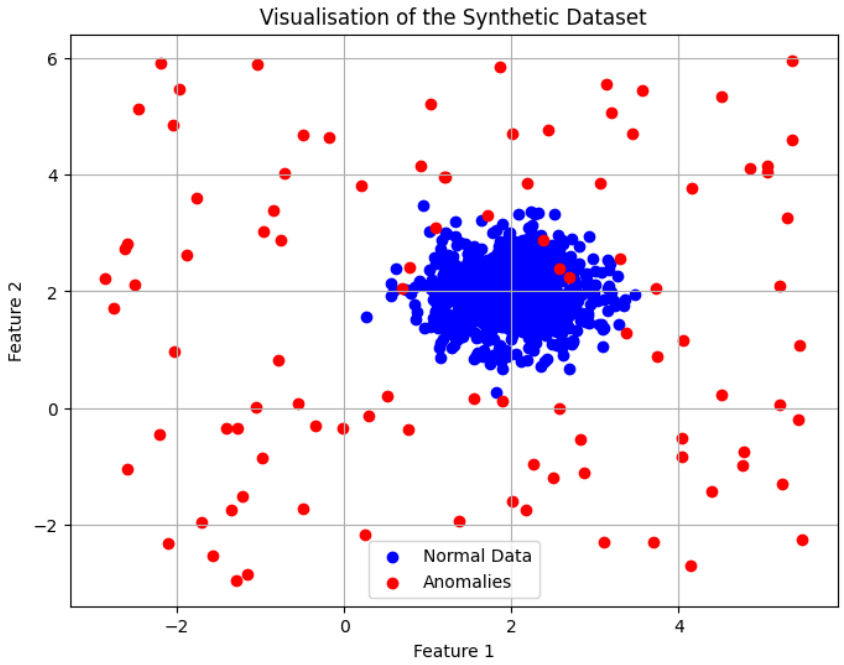

Рисунок 2.1 — Візуалізація нормальних та аномальних даних в датасеті

2.2.1 Тести алгоритмів на синтетичному датасеті

Використовуючи згенерований датасет та реалізації цих алгоритмів з бібліотеки **sklearn** було проведено тест цих алгоритмів для порівняння їх на практиці.

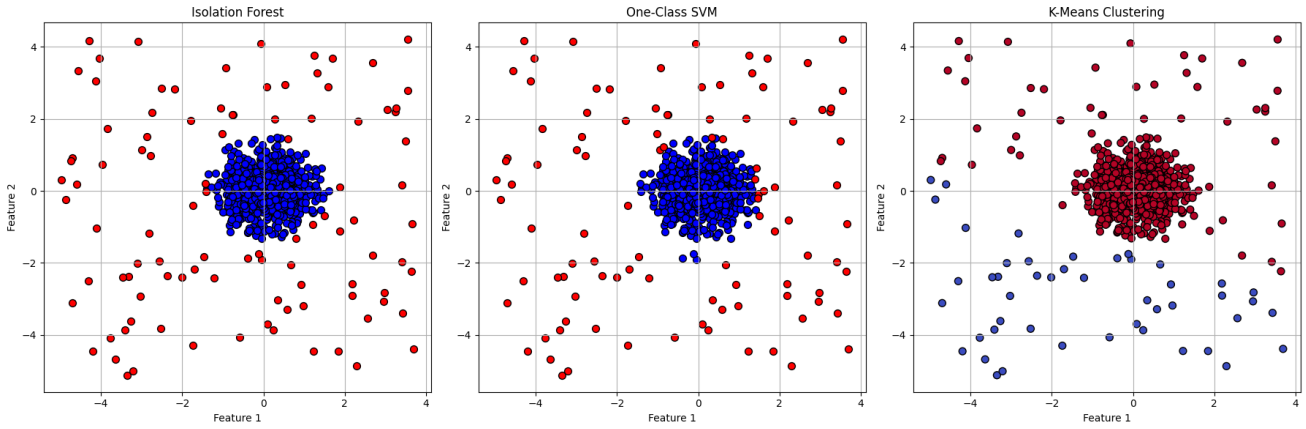

Рисунок 2.2 — Візуалізація знаходження аномалій алгоритмами

На візуалізації результатів алгоритмів виявлення аномалій (Рис 2.2.1) можемо спостерігати як Isolation Forest, One-Class SVM, та K-Means Clustering впоралися з пошуком аномалій на синтетичному датасеті.

Isolation Forest відмінно підходить для цього датасету, оскільки алгоритм спеціалізується на швидкому виявленні аномалій в наборах даних, де присутні рідкісні точки, що сильно відрізняються від більшості. На графіку видно, що Isolation Forest ефективно визначив більшість аномалій, розташованих далеко від основної групи нормальних даних.

One-Class SVM також показав гарні результати, але цей алгоритм може бути чутливим до вибору параметра nu та gamma, що впливає на кількість визначених аномалій. На графіку видно, що One-Class SVM позначив декілька нормальних точок як аномалії (false positives), що є типовим для цього методу при неідеальному підборі параметрів.

K-Means Clustering, використаний для визначення аномалій, виявився менш ефективним порівняно з іншими алгоритмами. K-Means припускає, що кластери мають відносно симетричну форму, і алгоритм працює на основі відстані до центру кластеру, що може не завжди коректно відображати реальну природу розподілу даних. Тому, незважаючи на те що K-Means виділив центральну групу як нормальні дані, він також помилково визначив кілька аномалій як нормальні точки.

За результатами можна сказати, що, кластеризація методом K-Means котра призначена для групування схожих елементів, а не для ідентифікації аномалій, не підійде для пошуку аномалій у контексті систем UEBA тому що, в системах аналітики поведінки користувачі ми шукаємо аномалії, які зазвичай розріненні та мають непередбачуваний характер і можуть бути схожими на звичайну поведінку.

2.2.2 детальніше порівняння Isolation Forest та One-Class SVM

Розглянемо детальніше алгоритми Isolation Forest та One-Class SVM, адже вони мають схожі результати. Проаналізуємо матриці кореляції цих алгоритмів.

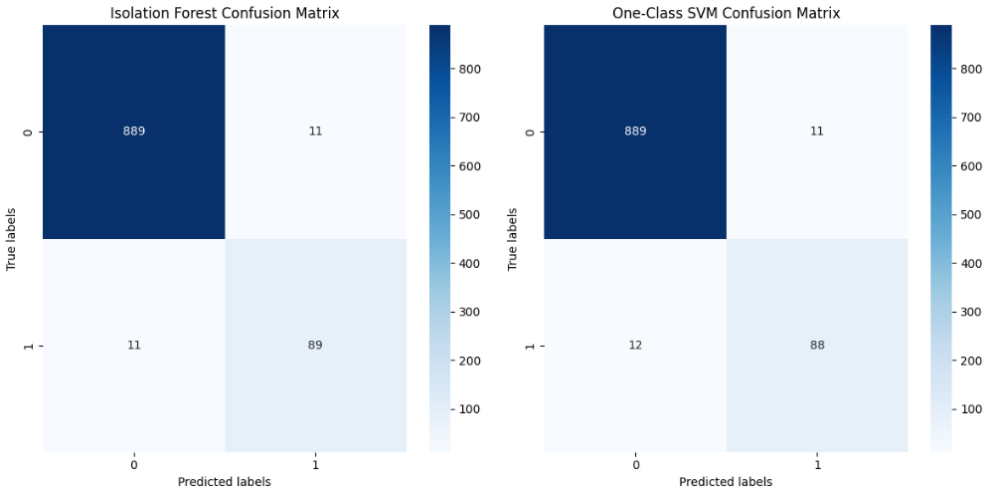

Рисунок 2.3 — матриці кореляцій **Isolation Forest та One-Class SVM**

Isolation Forest виявив 89 з 100 аномалій і мав 11 помилково позитивних результатів (false positives), де нормальні дані помилково були визначені як аномалії. Цей алгоритм використовує механізм випадкового розділення даних, що дозволяє йому ефективно ізолювати аномалії, особливо

в даних з розрізненими шаблонами поведінки, що робить його відмінним вибором для різноманітних сценаріїв, включаючи UEBA системи.

One-Class SVM також показав схожі результати, виявивши 88 аномалій та маючи 12 помилково позитивних результатів. One-Class SVM використовує ядерне перетворення для визначення гіперплощини, що максимально відділяє нормальні дані від аномалій у високовимірному просторі. Цей метод чутливий до вибору параметра nu і gamma, які визначають, наскільки часто об'єкти вважаються аномаліями та як розтягується простір даних відповідно.

Оскільки, ці 2 алгоритми схожі за результатами, роздивимось їх особливості.

One-Class SVM

One-Class SVM є методом машинного навчання, який адаптований для виявлення аномалій, відомий також як детектор новизни. У контексті бінарної класифікації, цей алгоритм використовується, коли всі дані вважаються належними до одного класу, і відповідно, викиди відсутні.

One-Class SVM працює, перетворюючи точки в простір з вищою вимірністю, де він спробує знайти гіперплощину, яка найкраще розділяє дані від початку координат. Головне завдання полягає у тому, щоб розташувати цю гіперплощину так, щоб вона проходила через початок координат і максимально відштовхувала дані від цього початку у просторі ознак. Алгоритм має збалансувати між виразною розділовою здатністю та мінімізацією помилок у тренувальному наборі.

Isolation forest

Ліс ізоляцій — це алгоритм виявлення викидів, який використовує дерева рішень. Принцип цього алгоритму полягає в тому, що викиди – це точки з характеристиками, які значно відрізняються від решти нормальних

даних. Розглянемо набір даних у просторі. Внутрішні елементи будуть ближче один до одного, а викидні – далі один від одного. Як ми знаємо, дерева рішень ділять p-вимірний простір за допомогою ортогональних розрізів.

Розглянемо дерево рішень, яке будується шляхом випадкових розрізів із довільно вибраними функціями. Дереву дозволяють рости, доки всі точки не будуть ізольовані. Буде легше ізолювати викиди, ніж у нормальні дані. Іншими словами, для ізоляції викидів потрібно менше розділень, ніж внутрішніх, і викидні вузли будуть розташовані ближче до кореневого вузла. Процес побудови дерева з випадковими розрізами повторюється для створення «лісу». Незважаючи на те, що алгоритм міг застосувати більш складний спосіб ізоляції точок, створення випадкових поділів обчислювально дешевше, а усереднення по всіх деревах враховує кілька способів ізоляції даних. Двома ключовими гіперпараметрами є оцінки, які дають нам кількість використаних дерев у лісі, і забруднення(очікувана кількість аномалій), яке дає частку викидів у наборі даних.

Таблиця 2.1 – Порівняння **One-Class SVM** та **Isolation Forest**

One-Class SVM	**Isolation Forest**
One-class SVM більш чутливий до викидів.	Менш чутливий порівняно з One-class SVM
One-class SVM тренується повільніше порівняно з Isolation forest.	Isolation forest швидше тренується
SVM може вимагати додаткових налаштувань та оптимізації ядра для роботи з багатовимірними даними	Isolation Forest ефективніше обробляє датасети з великою кількістю атрибутів. Він не вимагає складних передопрацювань даних

SVM менш адаптивний до даних, що змінюються з часом	Isolation Forest може ефективніше адаптуватися до даних, що змінюються з часом, що є типовим у системах UEBA
One-class SVM потребує більших обчислювальних витрат	Isolation Forest вимагає менших обчислювальних витрат, що дозволяє швидше обробляти дані та забезпечує більшу скалярність у великих системах.

У контексті систем UEBA, де важлива швидка і точна ідентифікація внутрішніх загроз з мінімальною кількістю помилкових спрацьовувань, серед даних які можуть включати в себе логи активностей всередині системи, наприклад:

- логи аутентифікації – включають в себе інформацію про успішні та невдалі спроби входу, методи аутентифікації, час і місцезнаходження спроб входу;
- логи доступу до файлів – події відкриття, зміни, видалення або копіювання файлів. Включають інформацію про взаємодію користувачів з чутливими або важливими файлами;
- мережеві логи – відображають вхідний і вихідний мережевий трафік, включаючи IP-адреси, порти, протоколи, а також об'єми переданих даних. Вони можуть також включати спроби доступу до заборонених ресурсів.
- логи адміністративних операцій - записують дії, виконані користувачами з адміністративними правами, такі як налаштування системних конфігурацій або управління користувацькими обліковими записами.

Isolation Forest виявляється кращим вибором завдяки його здатності ефективно працювати з великими обсягами даних та різноманітністю

атрибутів, а також високій скалярності та адаптивності до змін у даних. Це робить його ідеальним рішенням для систем, де дані часто є розрізненими та динамічними, як це часто буває в системах UEBA.

2.3 Використання алгоритму Isolation forest в підсистемі пошуку аномалій.

У практичній частині моєї роботи зосереджено увагу на створенні та тестуванні підсистеми пошуку аномалій, яка використовує алгоритм Isolation Forest для аналізу поведінки користувачів в контексті системи UEBA (User and Entity Behavior Analytics). Ця підсистема призначена для ефективного виявлення потенційних внутрішніх загроз і несанкціонованих дій, базуючись на аналізі поведінкових даних користувачів.

Першим етапом реалізації підсистеми пошуку аномалій буде вибір даних для тестування. Оскільки в мене немає доступу до будь-якої системи та немає змоги самому зібрати дані у цій системі, буде обрано датасет, який буде відображати типові моделі поведінки користувачів у корпоративних системах. Датасет може включати в себе логи системних операцій, логи доступу до файлів, мережеві логи та логи аутентифікації. Цей набір даних міститиме типові сценарії внутрішніх загроз, котрі будуть аномаліями в поведінці користувачів, ці аномалії мають бути ідентифіковані за допомогою підсистеми пошуку аномалій.

Далі, оскільки, підсистема пошуку аномалій буде реалізована в контексті системи UEBA, для покращення результатів виявлення аномалій і зменшення кількості хибнопозитивних спрацювань, буде реалізовано профілювання нормальних патернів поведінки користувачів. Профілі допомагають системі краще розуміти, що вважати "нормальним" в контексті конкретного користувача чи групи.

Третім кроком буде навчання Isolation Forest на навчальному наборі даних, з урахуванням профілів користувачів, що містить як нормальні, так і аномальні поведінкові записи. Навчання із використанням цих профілів як додаткових ознак, дозволить точніше ідентифікувати аномалії відносно звичайної поведінки користувача.

Після створення профілів та навчання буде проведено тестування, де Isolation Forest використовується на тестовому наборі даних для аналізу поведінки з урахуванням профілів користувачі. Це дозволяє перевірити як алгоритм ідентифікує відхилення від "норми" та виявти справжні аномалії що і буде являти собою підсистему пошуку аномалій в контексті систем аналітики поведінки користувачів.

В кінці буде проаналізовано результати. Спершу буде проаналізовано результати пошуку аномалій без профілів та з профілями, тобто можна буде побачити як збільшується точність підсистеми пошуку аномалій з урахуванням нормальної поведінки користувачів. Далі буде розглянуто детально результат знаходження аномалій вибором юзера і порівняння його звичайної поведінки зі знайденою аномалію.

Ця підсистема пошуку аномалій важлива для забезпечення безпеки інформаційних систем, оскільки вона дозволяє оперативно виявляти потенційні внутрішні загрози та незвичайну поведінку користувачів. Завдяки високій автоматизації процесу, знижується ризик людської помилки, а швидкість реагування на інциденти збільшується, що критично для підтримки високого рівня корпоративної безпеки.

Цей підхід не лише підвищує загальну безпеку системи, але і сприяє більшій зрозумілості та прозорості в поведінці користувачів, що є ключовим аспектом управління ризиками в сучасних IT-інфраструктурах.

Висновок до другого розділу

Було детально описано різні алгоритми пошуку аномалій, такі як Isolation Forest, K-means Clustering та One-Class SVM. Визначено та проаналізовано переваги та обмеження цих алгоритмів у виявленні аномальній у поведінці користувачів.

Шляхом практичних тестів алгоритмів машинного навчання для виявлення аномалій було виявлено, що Isolation Forest є кращим вибором для створення підсистеми пошуку аномалій в системі аналітики поведінки користувачів, завдяки своїй здатності швидко ідентифікувати аномалії з високою точністю, знижуючи кількість помилкових спрацювань і ефективно обробляючи великі обсяги даних. Це робить його ідеальним вибором для застосувань у динамічних середовищах, де дані часто змінюються. У порівняльних тестах також було виявлено недоліки алгоритмів K-means Clustering і One-Class SVM у контексті систем аналітики поведінки користувачів, а саме, алгоритм K-means Clustering має значну чутливість до вибору кількості кластерів та чутливість до викидів. В тей саме час One-Class SVM: має високу обчислювальну складність, обробка великих обсягів даних у високорозмірному просторі може бути дуже ресурсозатратною, та чутливість до налаштування параметрів, що може бути проблемою в умовах постійної зміни поведінкових патернів користувачів.

Також у цьому розділі була представлена реалізація підсистеми пошуку аномалій на основі Isolation Forest, що включає вибір датасету, тестування алгоритму, профілювання поведінки користувачів і повторне тестування з урахуванням профілів. Ця підсистема не тільки забезпечує ефективне виявлення внутрішніх загроз, шляхом пошуку аномалій з урахуванням нормальної поведінки кожного з користувачів, але й підвищує загальну безпеку інформаційних систем, сприяючи зменшенню ризиків і поліпшенню управління інформаційною безпекою на підприємствах, автоматизуючи

процес моніторингу системи, даючи змогу приділяти більше уваги аномаліям та мати більше часу на аналіз аномалії та на реагування.

3 Розробка підсистеми пошуку аномалій

У цьому розділі буде представлено процес розробки підсистеми пошуку аномалій в системи аналітики поведінки користувачів з використанням алгоритму машинного навчання для пошуку аномалій Isolation forest. На Рис 3.1 показано місце моєї підсистеми пошуку аномалій в системі UEBA. Вона буде реалізована в рамках компоненту системи аналітики поведінки користувачів который відповідає за пошук аномалій та їх аналіз.

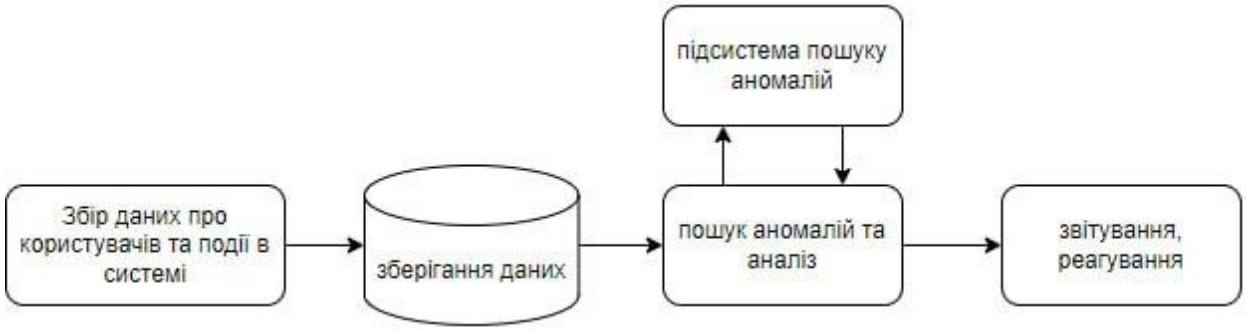

Рисунок 3.1 — Структура зв'язоку системи UEBA та підсистеми пошуку аномалій

3.1 Механізми виявлення аномалій в поведінці користувачів

Механізм виявлення аномалій в поведінці користувачів базується на використанні алгоритму Isolation Forest, який аналізує події в системі та класифікує їх як нормальну поведінку або аномальну. Основні кроки реалізації включають:

1. **Підготовка даних:** очищення та нормалізації для забезпечення консистентності та точності аналізу.
2. **Профілізація користувачів:** створення профілів поведінки користувачів для визначення іх нормальної поведінки.
3. **Навчання алгоритму:** навчання Isolation forest на тренувальних даних з урахуванням профілів користувачів для підвищення точності знаходження аномалій.

4. **Використання навченого алгоритму з урахуванням профілів користувачів:** використання Isolation forest разом з профілями користувачів для визначення аномальної поведінки користувачів в системі.

Нижче на Рис 3.2 показана схема роботи моєї підсистеми пошуку аномалій. На вхід вона отримує дані користувачів. Потім відбуваються дії описані вище і після роботи підсистеми пошуку аномалій отримуються оброблені дані про користувачів та знайдені аномалії.

Рисунок 3.2 — Структура моєї підсистеми пошуку аномалій

3.2 Підготовка даних для навчання алгоритму

Для ефективного навчання алгоритму пошуку аномалій необхідно підготувати відповідний датасет, який містить журнал подій в системі з користувачами. Підготовка даних включає кілька основних етапів. Перш за все, потрібно завантажити датасет "CERT Insider Threat".

id	date	user	pc	to	cc	bcc	from	size	attachments
2629979 unique values	2010-01-02 2011-05-17	1000 unique values	1000 unique values	659170 unique values	[null] 61% Winter.Veda.Burks... 0% Other (1008583) 38%	[null] 84% Winter.Veda.Burks... 0% Other (411965) 16%	2678 unique values	6182 142k	0 9
{R3I7-S4TX96FG-8219JWFF}	01/02/2010 07:11:45	LAP0338	PC-5758	Dean.Flynn.Hines@dtaa.com;Wade_Harrison@lockheedmartin.com	Nathaniel.Hunter.Heath@dtaa.com		Lynn.Adena.Pratt@dtaa.com	25830	0
{R0R9-E4GL59IK-2907OSWJ}	01/02/2010 07:12:16	MOH0273	PC-6699	Odonnell-Gage@bellsouth.net			MOH68@optonline.net	29942	0
{G2B2-A8XY58CP-2847ZJZL}	01/02/2010 07:13:00	LAP0338	PC-5758	Penelope_Colon@netzero.com			Lynn_A_Pratt@earthlink.net	28780	0
{A3A9-F4TH89AA-831BGFGK}	01/02/2010 07:13:17	LAP0338	PC-5758	Judith_Hayden@comcast.net			Lynn_A_Pratt@earthlink.net	21907	0
{E8B7-C8FZ88UF-2946RUQQ}	01/02/2010 07:13:28	MOH0273	PC-6699	Bond-Raymond@verizon.net;Alea_Ferrell@msn.com;Jane_Mcdonald@juno.com		Odonnell-Gage@bellsouth.net	MOH68@optonline.net	17319	0
{X8T7-A6BT54FP-7241DLBV}	01/02/2010 07:36:03	HVB0037	PC-7979	Gaines-Joseph@msn.com	Hollee_Becker@hotmail.com		Hollee_Becker@hotmail.com	44345	0

Рисунок 3.3 — приклад даних в датасеті.

Далі йде очищення та нормалізація даних, створення бінарних ознак, вибір ознак для моделювання і поділ на навчальний та тестовий набори.

```
df = data
df.fillna(value='Unknown', inplace=True)

# Перетворення стовпця 'date' у формат datetime
df['date'] = pd.to_datetime(df['date'])

# Створення нових часових ознак
df['hour'] = df['date'].dt.hour
df['day_of_week'] = df['date'].dt.dayofweek
df['month'] = df['date'].dt.month

# Створення бінарної ознаки для вкладень
df['has_attachments'] = df['attachments'].apply(lambda x: 1 if x > 0 else 0)

# Нормалізація числових ознак
scaler = StandardScaler()
df['size'] = scaler.fit_transform(df[['size']])
# Вибір ознак для моделювання і поділ на навчальний та тестовий набори
features = ['hour', 'day_of_week', 'size', 'has_attachments']
X = df[features]
X_train, X_test = train_test_split(X, test_size=0.2, random_state=42)

```

Рисунок 3.4 — приклад коду для обробки даних та поділу датасету

3.3 Профілізація користувачів

Процес профілювання користувачів починається з агрегування історичних даних за кожним користувачем. Для кожного користувача обчислюються статистичні показники по важливим параметрам активності: середній розмір переданих файлів, частота активності в різні години доби, а також середній день тижня активності. Ці параметри дозволяють виявити приклади поведінки, які будуть вважатись нормою для даного користувача.

Після обрахунку основних статистик, проводиться процедура нормалізації. Нормалізація допомагає уникнути ситуації, коли одні змінні через свої великі масштаби набувають непропорційного впливу на результати аналізу.

Після нормалізації, створений профіль кожного користувача включає нормалізовані значення середнього розміру файлів, середньої години активності, та середнього дня тижня, коли користувач був активний. Ці профілі служать основою для подальшого аналізу поведінки користувачів, дозволяючи системі ефективно ідентифікувати відхилення від звичної поведінки, які можуть сигналізувати про потенційні внутрішні загрози або аномалії.

```python
# Створення профілів користувачів
user_profiles = data.groupby('from').agg({
    'size': ['mean', 'std'],
    'hour': ['mean', 'std'],
    'day_of_week': ['mean', 'std']
}).reset_index()

user_profiles.columns = ['from', 'size_mean', 'size_std', 'hour_mean', 'hour_std', 'day_mean', 'day_std']

# Нормалізація
scaler = StandardScaler()
features = ['size', 'hour', 'day_of_week']
user_profiles_scaled = scaler.fit_transform(user_profiles[['size_mean', 'hour_mean', 'day_mean']])
user_profiles_scaled = pd.DataFrame(user_profiles_scaled, columns=features)
```

Рисунок 3.5 — приклад коду для профілізації

3.4 Навчання алгоритму

Тренування алгоритму на основі профілів користувачів включає застосування алгоритму ізоляційного лісу до нормалізованих профілів. Це дозволяє алгоритму вчитися на заздалегідь визначених показниках нормальної поведінки для кожного користувача. В процесі навчання алгоритм аналізує, як часто і як сильно поведінка кожного користувача відрізняється від його типової поведінки. Це дозволяє системі ефективно

```python
# Тренування Isolation Forest
clf = IsolationForest(contamination=0.1)
clf.fit(user_profiles_scaled)
```

ідентифікувати випадки, коли поведінка користувача виходить за рамки норми, що може вказувати на аномальні або підозрілі дії.

Рисунок 3.6 — приклад коду для тренування

3.5 Використання навченого алгоритму з урахуванням профілів користувачів та аналіз результатів

Після навчання було використано навчений алгоритм Isolation Forest з урахування профілів користувачів на тестових даних для аналізу того як працює моя підсистема пошуку аномалій в системі аналітики поведінки користувачів.

В результаті отримую візуалізацію використання підсистеми пошуку аномалій:

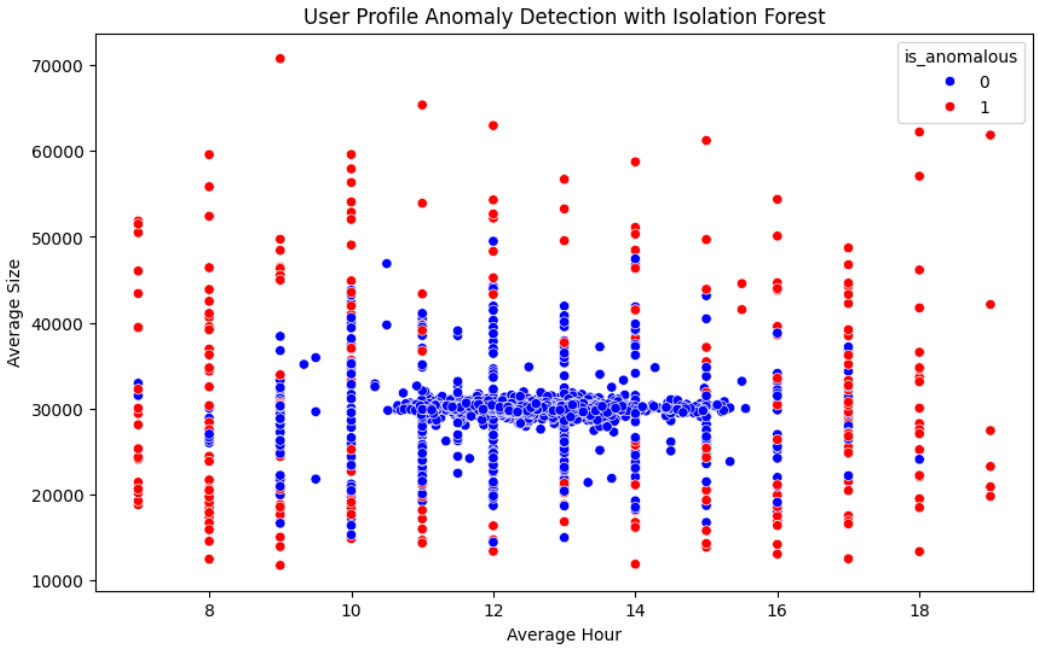

Рисунок 3.7— візуалізацію використання підсистеми пошуку аномалій

Як видно з візуалізації, більшість даних (позначені синім кольором) вписуються у типовий розподіл, що свідчить про нормальну поведінку, тоді як виокремлені точки (позначені червоним) вказують на аномалії. Ці аномалії можуть бути індикаторами незвичайних або потенційно шкідливих дій, наприклад, крадіжки даних.

Алгоритм Isolation Forest демонструє високу ефективність у виявленні аномалій, які мають суттєво відмінні характеристики від основної маси даних. Особливо це важливо у корпоративних системах, де швидке виявлення таких випадків може запобігти значним збиткам або зловживанням.

Використання нормалізованих профілів користувачів забезпечує системі глибше розуміння звичайної поведінки користувачів, що дозволяє більш точно ідентифікувати відхилення від "норми" і, як результат, покращує точність виявлення аномалій, одночасно знижуючи кількість помилкових спрацювань.

Для прикладу ефективності моєї підсистеми пошуку аномалій, нижче наведено візуалізацію пошуку аномалій без урахування профілів користувачів. На ній можна побачити, як ліс ізоляцій просто відокремлює аномалії за їх, випадаючими з норми співвідношення години дня та розміру файлу в листі, ознаками, не враховуючи контекст системи аналітики поведінки користувачів.

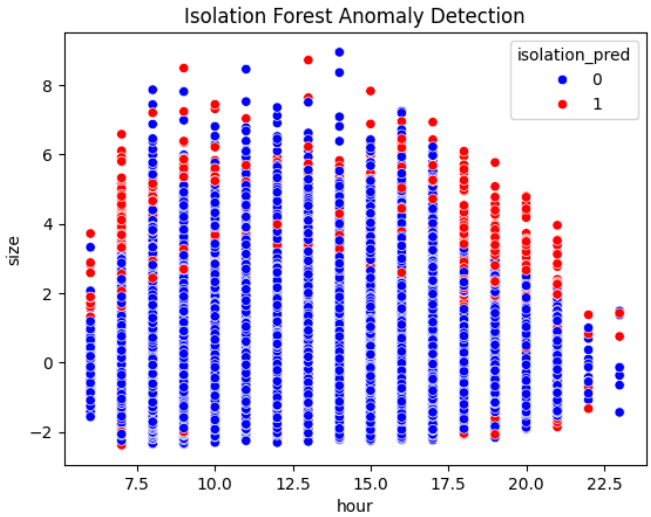

Рисунок 3.8 — візуалізація пошуку аномалій без профілювання користувачів

Для додаткової перевірки того як підсистема пошуку аномалій відокремлює нормальну поведінку та аномальну було проведено тест на конкретному користувачі.

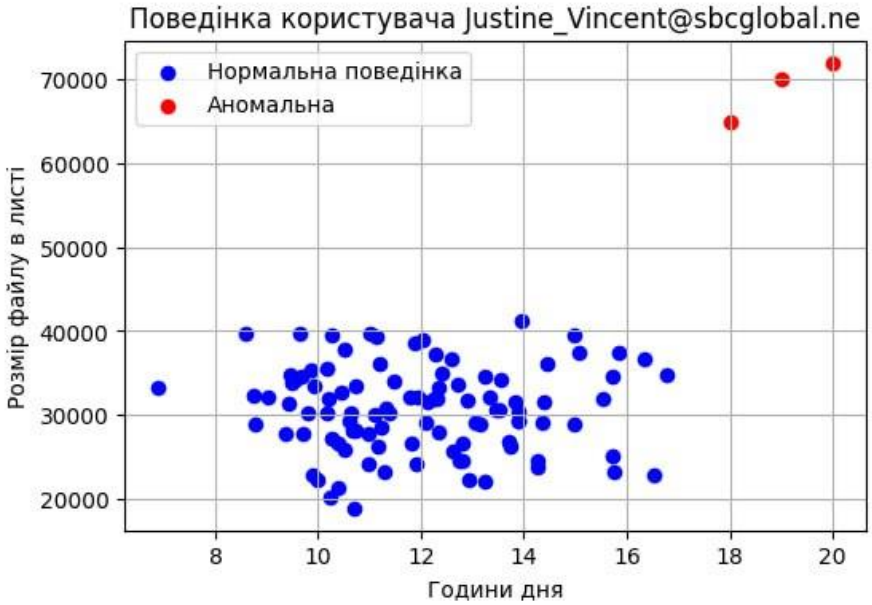

Рисунок 3.9 — візуалізація нормальної поведінки та аномальної

В аналізі поведінки конкретного користувача, Justine_Vincent@sbcglobal.ne, використання системи пошуку аномалій на базі алгоритму Isolation Forest та профілю цього користувача дозволило ідентифікувати відхилення від звичайних патернів поведінки, які можуть свідчити про потенційні внутрішні загрози або несанкціоновані дії. На візуалізації (Рис. 3.8) чітко видно, що більшість активностей користувача (позначені синім) вписуються у нормальний діапазон розмірів файлів і часу відправлення електронних повідомлень протягом дня. Проте, окремі випадки (позначені червоним) виходять за межі звичної активності, що індикує аномальну поведінку.

Ці аномалії можуть включати незвично великі файли або нехарактерні часи відправлення, що вимагають додаткового аналізу та можуть бути сигналом до глибшого розслідування. Таке виявлення допомагає забезпечувати безпеку системи, оперативно реагувати на можливі внутрішні

загрози та оптимізувати контроль за поведінкою користувачів в корпоративній мережі.

3.6 Перспективи розвитку системи

Моя підсистема пошуку аномалій в контексті UEBA має значний потенціал для розвитку і удосконалення, зокрема у сфері забезпечення корпоративної безпеки та захисту від внутрішніх загроз. Одним з напрямків є інтеграція з іншими інструментами безпеки, такими як системи виявлення вторгнень та системи управління інформаційними подіями безпеки. Це дозволить системі ефективніше обмінюватися даними та сповіщеннями, підвищуючи здатність швидко реагувати на загрози.

Існує також потенціал у розвитку аналітичних можливостей системи за рахунок впровадження додаткових алгоритмів машинного навчання та штучного інтелекту, зокрема глибинного навчання, для аналізу складних шаблонів поведінки.

Значні переваги можна отримати, забезпечивши систему здатністю адаптуватися до змінних умов роботи в корпоративних мережах. Автоматичне перенавчання моделей на основі нових даних та виправлення помилкових спрацьовувань дозволить системі зберігати актуальність та ефективність.

Крім того, удосконалення профілювання користувачів з метою детальнішого аналізу поведінкових патернів може значно покращити здатність системи передбачати і виявляти внутрішні загрози, що є ключовим аспектом управління ризиками в сучасних IT-інфраструктурах.

Висновок до третього розділу

Ретельно розроблена система охоплює весь процес: від підготовки даних і профілювання користувачів до навчання алгоритму та його впровадження. Ця підсистема не просто застосовує метод машинного

навчання для ідентифікації аномалій, але й забезпечує систему засобами для точного визначення звичайної та аномальної поведінки користувачів.

Особливу увагу в роботі було приділено створенню профілів користувачів, що дозволяє системі адаптуватися до індивідуальних особливостей поведінки кожного користувача. Це значно підвищує точність виявлення аномалій, оскільки алгоритм Isolation Forest навчається на даних, які відображають типові патерни поведінки, враховуючи при цьому відхилення, що можуть вказувати на потенційні загрози або зловживання.

Також було проаналізовано результати, в котрих виявлено що система дійсно виявляє аномалії та добре відрізняє їх від звичайної поведінки.

Висновки

У моїй дипломній роботі розроблено підсистему пошуку аномалій в рамках системи аналітики поведінки користувачів (UEBA) з використанням алгоритму Isolation Forest. Моїм завданням було створення ефективного механізму для виявлення потенційних внутрішніх загроз і несанкціонованих дій, базуючись на аналізі даних про поведінку користувачів. Структура моєї підсистеми включає збір і первинну обробку даних, профілювання користувачів для визначення їх нормальної поведінки, навчання алгоритму для точного виявлення аномалій. Важливим аспектом стало використання профілів користувачів, що дозволяє підсистемі точніше ідентифікувати аномальні дії, зменшуючи кількість хибнопозитивних спрацьовувань. Підсистема пошуку аномалій може забезпечити більший рівень безпеки корпоративних систем шляхом швидкого виявлення та оперативного реагування на внутрішні загрози і оптимізації контролю за діяльністю користувачів. Такий підхід не тільки покращує захист інформаційних ресурсів організації, але й сприяє підвищенню загальної прозорості та керованості IT-інфраструктури.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1 - Manya Ali Salitin, Ali Hussein Zolait, “The role of User Entity Behavior Analytics to detect network attacks in real time” 18-20 November 2018 https://ieeexplore.ieee.org /abstract/document/8855782

2. - Fei Tony Liu; Kai Ming Ting; Zhi-Hua Zhou “Isolation Forest” 15-19 December 2008 https://ieeexplore.ieee.org/abstract/document/4781136

3 -October 2021 JOURNAL OF XI AN UNIVERSITY OF ARCHITECTURE & TECHNOLOGY “ Measuring the Effectiveness of User and Entity Behavior Analytics for the Prevention of Insider Threats” https://www.researchgate.net/profile/Rasheed-Yousef/publication/355424926_Measuring_the_Effectiveness_of_User_and_Entity_Behavior_Analytics_for_the_Prevention_of_Insider_Threats/links/616fa4423d9af67ad7404388/Measuring-the-Effectiveness-of-User-and-Entity-Behavior-Analytics-for-the-Prevention-of-Insider-Threats.pdf

4 - A. Sharma, “User and Entity Behavior Analystics (UEBA) Market Size”, Growth & Industry Research Report, Feb. 2019.

5 - J. Graves, “How Machine learning is Catching Up With the Insider Threat”, Cyber Security: A Peer-Reviewed Journal. 2017 Jan 1;1(2):127-33.

ДОДАТОК А ПРОГРАМНИЙ КОД ПОРІВНЯННЯ АЛГОРИТМІВ

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs

# Generating synthetic data with more complex patterns
n_samples = 1000  # total samples
outliers_fraction = 0.1  # 10% are anomalies
n_outliers = int(outliers_fraction * n_samples)
n_inliers = n_samples - n_outliers

# Generate inlier data
inlier_data = make_blobs(n_samples=n_inliers, centers=[[2, 2]],
cluster_std=0.5)[0]

# Generate outlier data more interspersed
outlier_data = np.random.uniform(low=-3, high=6, size=(n_outliers, 2))

# Combine the data
X = np.concatenate([inlier_data, outlier_data], axis=0)

# Labels: 0 for inliers, 1 for outliers (used later for evaluation)
y = np.zeros(n_samples)
y[-n_outliers:] = 1  # Mark the last n_outliers elements as outliers

# Shuffle the data
np.random.seed(42)
shuffled_indices = np.random.permutation(n_samples)
X = X[shuffled_indices]
y = y[shuffled_indices]
```

```
# Plotting the data
plt.figure(figsize=(8, 6))
plt.scatter(X[y == 0][:, 0], X[y == 0][:, 1], color='blue', label='Normal Data')
plt.scatter(X[y == 1][:, 0], X[y == 1][:, 1], color='red', label='Anomalies')
plt.title("Visualisation of the Synthetic Dataset")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.legend()
plt.grid(True)
plt.show()


X.shape, y.shape
```

ДОДАТОК Б ПРОГРАМНИЙ КОД РОЗРОБКИ ПІДСИСТЕМИ ПОШУКУ АНОМАЛІЙ

```
import pandas as pd

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
data = pd.read_csv('email.csv')


data.head()
data.info()
data.describe()
df = data
df.fillna(value='Unknown', inplace=True)

# Перетворення стовпця 'date' у формат datetime
df['date'] = pd.to_datetime(df['date'])

# Створення нових часових ознак
df['hour'] = df['date'].dt.hour
df['day_of_week'] = df['date'].dt.dayofweek
df['month'] = df['date'].dt.month

# Створення бінарної ознаки для вкладень
df['has_attachments'] = df['attachments'].apply(lambda x: 1 if x > 0 else 0)
```

```
# Нормалізація числових ознак
scaler = StandardScaler()
df['size'] = scaler.fit_transform(df[['size']])

from sklearn.model_selection import train_test_split
from sklearn.ensemble import IsolationForest
from sklearn.neighbors import LocalOutlierFactor

# Вибір ознак для моделювання і поділ на навчальний та тестовий набори
features = ['hour', 'day_of_week', 'size', 'has_attachments']
X = df[features]
X_train, X_test = train_test_split(X, test_size=0.2, random_state=42)

# Тренування Isolation Forest
clf = IsolationForest(contamination=0.01,max_samples='auto', n_estimators=200,
random_state=42)
clf.fit(X_train)

# Прогнозування Isolation Forest
isolation_pred = clf.predict(X_test)
isolation_pred = [1 if x == -1 else 0 for x in isolation_pred]

# Візуалізація і аналіз
import matplotlib.pyplot as plt
import seaborn as sns

# Візуалізація результатів для Isolation Forest
X_test['isolation_pred'] = clf.predict(X_test[features])
X_test['isolation_pred'] = X_test['isolation_pred'].apply(lambda x: 1 if x == -1 else
0)

# Створення діаграми розсіювання для візуалізації аномалій
sns.scatterplot(x='hour', y='size', hue='isolation_pred', data=X_test)
plt.title('Isolation Forest Anomaly Detection')
plt.show()

# Створення профілів користувачів
user_profiles = data.groupby('from').agg({
    'size': ['mean', 'std'],
    'hour': ['mean', 'std'],
    'day_of_week': ['mean', 'std']
}).reset_index()

user_profiles.columns = ['from', 'size_mean', 'size_std', 'hour_mean', 'hour_std',
'day_mean', 'day_std']

# Нормалізація
```

```
scaler = StandardScaler()
features = ['size', 'hour', 'day_of_week']
user_profiles_scaled = scaler.fit_transform(user_profiles[['size_mean',
'hour_mean', 'day_mean']])
user_profiles_scaled = pd.DataFrame(user_profiles_scaled, columns=features)
# Тренування Isolation Forest
clf = IsolationForest(contamination=0.1)
clf.fit(user_profiles_scaled)

# Виявлення аномалій
user_profiles['is_anomalous'] = clf.predict(user_profiles_scaled)
user_profiles['is_anomalous'] = user_profiles['is_anomalous'].apply(lambda x: 1 if
x == -1 else 0)

# Візуалізація
plt.figure(figsize=(10, 6))
sns.scatterplot(x='hour_mean', y='size_mean', hue='is_anomalous',
data=user_profiles, palette={0: 'blue', 1: 'red'})
plt.title('User Profile Anomaly Detection with Isolation Forest')
plt.xlabel('Average Hour')
plt.ylabel('Average Size')
plt.show()
```