Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи виявлення аномального трафіку в IoT

(тема)

Виконав:

студент II курсу, групи СПм-22-3

Марченко Р.М.
(прізвище, ініціали)

Спеціальність

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

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

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: Коваленко А.А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

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

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

Освітня програма Системне програмування
(повна назва)

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

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Марченку Роману Михайловичу
(прізвище, ім’я, по батькові)

1. Тема роботи Методи виявлення аномального трафіку в IoT

затверджена наказом по університету від “ 01 ” квітня 2024 р. № 257 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15 червня 2024 р.

3. Вхідні дані до роботи
IoT;
методи машинного навчання для класифікації даних;
програмне середовище Jupyter Notebook;
Мова програмування Python.

4. Перелік питань, що потрібно опрацювати у роботі
Вступ.
1. Аналіз предметної області та постановка задач дослідження.
2. Методи машинного навчання для виявлення аномального трафіку.
3. Реалізація і дослідження методів виявлення аномального трафіку в IoT.
Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційна презентація. Слайди – 14 штук.

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	02.04.2024–11.04.2024	
2	Аналіз завдання, підбір літератури	12.04.2024–18.04.2024	
3	Огляд існуючих методів	19.04.2024–26.04.2024	
4	Аналіз технічних засобів реалізації	27.04.2024–10.05.2024	
5	Програмна реалізація	11.05.2024–15.05.2024	
6	Отримання результатів	16.05.2024–18.05.2024	
7	Оформлення ПЗ	19.05.2024–26.05.2024	

Дата видачі завдання 01 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____ Коваленко А.А..
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 73 с., 15 рис., 2 табл., 1 дод., 15 джерел.

IOT, МАШИННЕ НАВЧАННЯ, АНОМАЛЬНИЙ ТРАФІК, КОМП'ЮТЕРНА МЕРЕЖА, МЕТОД ОПОРНИХ ВЕКТОРІВ, ДЕРЕВО РІШЕНЬ, ВИПАДКОВИЙ ЛІС.

Об'єктом дослідження є процес виявлення аномального трафіку в IoT. Предметом дослідження є методи виявлення аномального трафіку в IoT. Метою кваліфікаційної роботи є підвищення точності виявлення аномального трафіку в IoT за рахунок підбору гіперпараметів для моделей машинного навчання.

У ході виконання кваліфікаційної роботи було проведено дослідження методів виявлення аномального трафіку в IoT. Використано наступні методи, такі як: метод випадкового лісу, метод опорних векторві та метод дерева прийняття рішень. Розглянуто їх математичні моделі. Визначено переваги та недоліки використання для виявлення аномального трафіку в IoT.

У результаті проведеного дослідження здійснено підбір гіперпараметрів для реалізованих моделей для покращення виявлення аномального трафіку в IoT.

ABSTRACT

Master’s thesis: 73 pages, 15 figures, 2 tables, 1 appendices, 15 sources.

IOT, MACHINE LEARNING, ANOMALOUS TRAFFIC, COMPUTER NETWORK, SUPPORT VECTOR METHOD, DECISION TREE, RANDOM FOREST.

The object of research is the process of detecting anomalous traffic in IoT. The subject of the study is methods of detecting abnormal traffic in IoT. The purpose of the qualification work is to increase the accuracy of detecting anomalous traffic in IoT due to the selection of hyperparameters for machine learning models.

In the course of the qualification work, a study of methods for detecting anomalous traffic in IoT was conducted. The following methods were used, such as: the random forest method, the support vector method, and the decision tree method. Their mathematical models are considered. Advantages and disadvantages of using it to detect anomalous traffic in IoT are identified.

As a result of the conducted research, the selection of hyperparameters for the implemented models was carried out to improve the detection of anomalous traffic in IoT.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AMQP – протокол черг повідомлень з розширеними можливостями (англ., Advanced Message Queuing Protocol)

CoAP – протокол обмежених додатків (англ., Constrained Application Protocol)

C5.0 – алгоритм класифікаційних дерев (англ., C5.0 Algorithm)

DDS – служба поширення даних (англ., Data Distribution Service)

DT – дерево прийняття рішення (англ., Decision Tree)

GPS – глобальна система позиціонування (англ., Global Positioning System)

HTTP – протокол передачі гіпертексту (англ., Hypertext Transfer Protocol)

IOT – інтернет речей (англ., Internet of Things)

MQTT – протокол телеметрії транспорту (англ., Message Queuing Telemetry Transport)

RBF – радіальні базисні функції (англ., Radial Basis Function)

RF – випадковий ліс (англ., Random Forest)

RFID – радіочастотна ідентифікація (англ., Radio Frequency Identification)

REST – архітектурний стиль для створення веб-сервісів (англ., Representational State Transfer)

SOAP – простий протокол доступу до об'єктів (англ., Simple Object Access Protocol)

SVM – метод опорних векторів (англ., Support Vector Machines)

TCP – протокол керування передаванням (англ., Transmission Control Protocol)

UDP – протокол користувацьких датаграм (англ., User Datagram Protocol)

ВСТУП

IoT є однією з найбільш революційних технологій нашого часу. Вона об'єднує фізичні пристрої, транспортні засоби, будівлі та інші об'єкти, що оснащені електронікою, програмним забезпеченням, датчиками та мережевими підключеннями.

Використання IoT призводить до значного покращення якості життя людей, пропонуючи нові можливості доступу до даних та спеціальних послуг у сферах освіти, безпеки, охорони здоров'я та транспорту, серед інших. Також IoT стане ключовим чинником для підвищення продуктивності підприємств, забезпечуючи розподілену мережу "розумних" пристроїв та нових послуг, які можуть бути персоналізовані відповідно до потреб клієнтів.

IoT приносить численні переваги, серед яких покращене управління та відстеження активів і продуктів, збільшення обсягу інформаційних даних, а також покразобладнання та використання ресурсів, що може призвести до значної економії коштів. Крім того, IoT відкриває можливості для створення нових "розумних" взаємопов'язаних пристроїв та дослідження нових бізнес-моделей, сприяючи інноваціям та розвиткові у різних галузях.

Тим не менш, є проблеми, пов’язані з безпекою та захистом мережевих інфраструктур IoT. У нашому світі, де мільйони пристроїв IoT неперервно генерують величезні обсяги даних, методи аналізу аномального трафіку стають особливо цінними. Вони дозволяють не тільки реагувати на вже відомі типи аномалій, а й адаптуватися до нових викликів, ефективно виявляючи незвичайні патерни поведінки, які можуть вказувати на спроби вторгнення, помилки у пристроях або інші ризики.

Аномальна активність, яка може вказувати на вторгнення, атаки або непередбачені випадки, є однією з основних загроз для мереж IoT. Аномалії можуть спричинити ненормальну роботу системи, втрату даних або навіть загрозу безпеці. Аномалії в IoT можуть вказувати на різні загрози, від

кіберзлочинців до збоїв обладнання, що може призвести до серйозних наслідків, таких як втрата даних, порушення роботи систем і загроза безпеці.

Таким чином, виявлення аномалій є надзвичайно важливими для забезпечення безпеки та стабільності в IoT.

Використання методів аналізу аномального трафіку в IoT покращує безпеку систем IoT, дозволяючи проводити превентивне виявлення аномалій перш, ніж вони стануть критичними. Враховуючи широкий спектр застосувань IoT, від побутових пристроїв до промислових систем, здатність до точного та своєчасного виявлення аномалій є ключовим аспектом для підтримання не тільки функціональності, а й загальної безпеки цих складних мереж.

Використання машинного навчання для аналізу даних у значно підвищує надійність та безпеку систем IoT. Машинне навчання дозволяє створювати моделі, які здатні аналізувати великі обсяги даних та виявляти аномалії у мережевому трафіку на основі характерних ознак. Це дозволяє своєчасно ідентифікувати потенційні загрози та вживати необхідних заходів.

Серед методів машинного навчання, які застосовуються для виявлення аномального трафіку в IoT, є метод опорних векторів, метод дерева рішень та метод випадкового лісу. Кожен із цих методів має свої переваги та недоліки, проте всі вони сприяють підвищенню ефективності захисту мереж IoT від кіберзагроз.

Отже, забезпечення безпеки мереж IoT є надзвичайно важливим завданням у сучасному світі. Використання методів для виявлення аномального трафіку сприяє своєчасній ідентифікації потенційних загроз та мінімізації ризиків. Це дозволяє не лише захистити дані та забезпечити безперебійну роботу систем, але й підтримати подальший розвиток та впровадження IoT у різних сферах. Стабільні та надійні системи IoT відкривають нові можливості для інновацій та покращення якості життя, забезпечуючи сталий розвиток суспільства вцілому.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

Сьогодні IoT є доступним практично з кожного куточка Землі і, безсумнівно, має значний вплив на життя людей. IoT можна розглядати як динамічну глобальну мережеву інфраструктуру, яка керує об’єктами, що самоналаштовуються.

Згідно з Вермесаном [2], IoT є взаємодією цифрового і фізичного світів через приводи і датчики. Пенья-Лопес же визначає IoT як парадигму, що включає обчислювальні та мережеві функції у будь-який розумний об’єкт, дозволяючи отримувати інформацію про стан об’єкта [3].

Незалежно від визначення, безсімнівно, IoT створений для покращення світу і якості життя людей, роблячи всі пристрої та прилади більш «розумними» і взаємопов’язаними.

IoT використовується у багатьох сферах, таких як охорона здоров’я, фізична культура, освіта, розваги, соціальне життя, збереження енергії, моніторинг навколишнього середовища, автоматизація домашнього господарства та транспортні системи. На рисунку 1.1 показані сфери використання IoT [3].

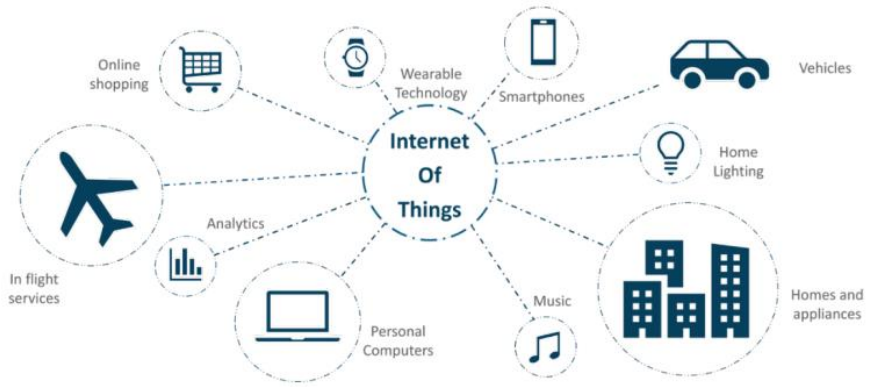

Рисунок 1.1 – Сфери використання IoT

Концепцію IoT вперше представив Кевін Ештон у 1999 році. Він сказав: «Інтернет речей має потенціал змінити світ, як це зробив Інтернет. Можливо, навіть більше». Пізніше IoT був офіційно представлений Міжнародною спілкою електрозв'язку у 2005 році.

IoT – це перспективна технологія, яка починає значно розвиватися: дослідники роблят прогнози, що в в 2030 кількість підключених до Інтернету пристроїв досягне або навіть перевищить 50 мільярдів пристроїв.[3]

Варто зазначити, що IoT – це комбінація різних технологій, а не однієї. Для отримання корисних результатів із даних, зібраних датчиками, необхідно їх зберігати та обробляти, тому пристрої IoT є оснащені вбудованими датчиками, приводами, процесорами та передавачами для забезпечення комунікації та взаємодії між ними.

IoT може зберігати та обробляти дані у самій мережі IoT або ж на віддаленому сервері. Попередня обробка даних зазвичай виконується на датчику або іншому пристрої, але дані для аналізу надсилаються на віддалений сервер. Можливості обробки та зберігання даних є обмежені ресурсами, які часто залежать від розміру, потужності та обчислювальної здатності пристрою.

Бездротові канали часто мають високі рівні спотворень, що створює серйозну проблему для надійного передавання даних [11].

1.1 Архітектура IoT

Архітектура IoT є багаторівневою системою, яка забезпечує взаємодію між фізичними пристроями, мережевою інфраструктурою та програмними додатками для обробки та аналізу даних. Єдиного представлення еталонної архітектури IoT не існує.

На рисунку 1.2 показано трирівнуву архітектуру IoT. Вона була введена вперше на ранніх стадіях досліджень у цій галузі. Вона складається з трьох рівнів:

- фізичний рівень;
- мережевий рівень;
- прикладний рівень.

Фізичний рівень містить датчики та фізичні пристрої, які збирають дані з реального середовища. До них можуть входити датчики температури, вологості, руху та відеокамер.

Мережевий рівень забезпечує підключення та спілкування пристроїв Інтернету речей з іншими мережевими пристроями або серверами. Він підтримує як провідні, так і бездротові технології, такі як: Wi-Fi, Bluetooth і Zigbee [1].

Прикладний рівень дозволяє керувати засобами обробки даних і управління пристроями IoT.

Рисунок 1.2 – Трирівнева архітектура IoT

Трирівнева архітектура є основною концепцією IoT, але її недостатньо для досліджень, оскільки вони часто зосереджуються на тонких аспектах IoT. Ось чому дослідниками було запропоновано значно більшу кількість архітектур, які мають більше архітектурних рівнів [5].

Такою є п’ятирівнева архітектура, яка додатово також включає рівень бізнесу та обробки. На рисунку 1.3 продемонстровані наступні п’ять рівнів:

- фізичний рівень;
- мережевий рівень;
- рівень обробки;
- прикладний рівень;
- бізнес рівень.

Рівень обробки складається з баз даних, які зберігають інформацію, що передається нижчими рівнями, де він виконує обробку інформації та використовує результати для прийняття подальших рішень.

Бізнес рівень визначає майбутні або подальші дії, необхідні на основі даних, наданих нижчими рівнями.

Рисунок 1.3 – П’ятирівнева архітектура IoT

1.2 Протоколи в IoT

Протоколи IoT регулюють, як різні пристрої та системи взаємодіють і спілкуються між собою. Вони є важливими для стандартизації, сумісності та

ефективного обміну даними в екосистемі IoT [14]. Протоколи в IoT відіграють ключову роль, забезпечуючи зв'язок між пристроями, обмін даними та їх обробку. Вибір протоколів залежить від конкретних вимог системи, таких як споживання енергії, діапазон дії, безпека, масштабованість та інші фактори. На рисунку 1.4 наведені основні протоколи, які використовуються в IoT.

Рисунок 1.4 – Протоколи в IoT

Протокол MQTT призначений для обміну повідомлень через мережу між пристроями Інтернету речей. Особливо корисним він є в умовах обмеженого ресурсу трафіку та енергії. MQTT побудований на основі протоколу TCP і підходить для пристроїв з низькою доступністю ресурсів, ненадійною або низькою пропускною спроможністю. MQTT використовується в багатьох програмах, таких як охорона здоров’я, моніторинг, лічильники енергії, сповіщення Facebook, тощо.

Протокол MQTT є ідеальним протоколом обміну повідомленнями для IOT, адже він здатний забезпечити маршрутизацію для малих, дешевих, малопотужних пристроїв пристрої пам'яті та для мереж з низькою пропускною здатністю.

Іншим протоколом, який використовується в IoT, є CoAP. Він ризначений для пристроїв, які мають обмежені ресурси, такі як енергія та

обчислювальна потужність. Також він полегшує керування пристроями та обмін даними.

Однією з ключових особливостей CoAP є підтримка групової передачі даних, що дозволяє ефективно передавати дані від одного пристрою до багатьох інших одночасно. Це особливо корисно в сенсорних мережах, де потрібно розповсюджувати інформацію серед великої кількості пристроїв.

REST – це кешований протокол з’єднання, який базується на клієнт-серверерній архітектурі без збереження стану. Він дозволяє клієнтам і серверам надавати та використовувати веб-сервіси, такі як «Simple Object».

Протокол SOAP працює у простіший спосіб, ніж REST, використовуючи уніфіковані ідентифікатори ресурсів, як іменники та HTTP методи, такі як: «get», «post», «put» і «delete». На відміну від протоколу REST, SOAP за замовчуванням прив’язано до UDP, а не до TCP, що робить його більш придатним для пристроїв IoT.

Також в IoT використовується протокол AMQP. Він є протоколом передачі повідомлень, який дозволяє пристроям IoT та хмарним сервісам спілкуватися надійним та масштабованим способом. AMQP є відкритим стандартним протоколом прикладного рівня. Він забезпечує надійний зв'язок через примітиви гарантії доставки повідомлень, які включають «at-most-once», «at-aleast-once» і одноразову доставка. Для обміну повідомленнями AMQP використовує надійний транспортний протокол TCP.

В IoT не обходиться без протоколу DDS, який забезпечує розподілену архітектуру для обміну даними у реальному часі між пристроями IoT. Він часто використовується, щоб забезпечити високу надійність та швидкість передачі повідомлень. DDS відповідає різноманітним критеріям зв’язку, як-от: безпека, терміновість, пріоритет, довговічність, надійність.

Варто також згадати протокол HTTP – це протокол передачі гіпертексту та здебільшого використовується веб-браузерами та веб-серверами, але він також може використовуватися для зв’язку між пристроями IoT та хмарними сервісами.

Протокол Bluetooth є стандартом бездротового зв'язку, розробленим для передачі даних на короткі відстані. Він використовується для створення персональних мереж і забезпечує зручний спосіб з'єднання різноманітних пристроїв без використання кабелів. Bluetooth і його варіант з низьким споживанням енергії, BLE, широко застосовуються в IoT. Зокрема, вони з’єднують смартфони, планшети, переносні пристрої та пристрої розумного дому. Обидві технології енергоефективні та прості у застосуванні.

Протокол Zigbee – сітчастий мережевий протокол, розроблений для створення мереж з низьким енергоспоживанням, низькою швидкістю передачі даних і великою кількістю підключених пристроїв. Це поширений протокол у домашній автоматизації та промислових підприємствах.

Використання відповідних протоколів допомагає забезпечити ефективний, надійний і безпечний обмін даними в системах IoT.

1.3 Датчики в IoT

Датчики відіграють ключову роль у системах IoT, забезпечуючи збір даних з фізичного світу і їх передачу для подальшої обробки та аналізу. Вони є основними компонентами IoT, оскільки надають інформацію про стан навколишнього середовища, що дозволяє здійснювати моніторинг, контроль і автоматизацію процесів.

Датчик – це, як правило, пристрій, здатний виявляти зміни в навколишньому середовищі. Датчик здатний вимірювати фізичне явище і перетворювати його в електричний або аналоговий сигнал [13]. Варто зазначити, що датчики в IoT класифікуються на декілька класів, як от:

- активні;
- пасивні;
- цифрові;
- аналогові.

Активні датчики, також відомі як параметричні датчики – це датчики,

які потребують зовнішнього джерела живлення для роботи. Приклади активних датчиків включають датчики GPS та радари.

Пасивні датчики, також відомі самогенерованими датчиками, генерують власний електричний сигнал і не потребують зовнішнього джерела живлення. Приклади пасивних датчиків включають термодатчики, датчики електричного поля, зондування та виявлення металів.

Аналогові датчики виробляють безперервні аналогові вихідні сигнали, пропорційні вимірюванню. До прикладів аналогових датчиків відносяться акселерометри, датчики тиску, датчики світла та звуку.

Цифрові датчики, також відомі як електронні або електрохімічні датчики, перетворюють дані у цифровому вигляді. До прикладів включають цифрові акселерометри, датчики тиску та температури. Основними типами датчиків в IoT є наступні:

- датчик температури;
- датчик наближення;
- акселерометр;
- датчик світла;
- ультразвуковий датчик;
- датчик диму, газу та алкоголю;
- датчик руху;
- датчик вологості.

Усі ці датчики використовуються для вимірювання однієї з фізичних властивостей: температури, опору, ємності, проведення, перетворення тепла тощо.

1.4 Аномалії в IoT

Аномалія в контексті IoT – це дані, які виходять за межі очікуваної поведінки в системі [1]. Це може бути рідкісна подія або відхилення від типового шаблону у конкретний момент часу або для певного контексту.

Аномалії можуть бути спричинені зовнішніми факторами, такими як помилки датчиків або кібератаки. Задача методу виявлення аномалій – виявити ці відхилення і, за можливості, визначити їхні причини [7].

Методи виявлення аномалій можна розділити на чотири категорії в залежності від підходу до вирішення задачі, способу застосування, типу методу та затримки.

Виявлення аномалій даних у парадигмі IoT є складним завданням, оскільки важко визначити нормальний шаблон даних, оскільки дані в IoT залежать від домену та, окрім того, дані надходять від різнорідних датчиків різного формату.

1.4.1 Типи аномалій

Різні джерела показали, що існує три типи аномалій [1, 3, 4]:

- точкова аномалія, яка є окремою точкою даних, яка відрізняється від інших точок даних;
- контекстна аномалія, яка з однієї точки зору може бути нормальною, але з іншої точки зору може бути ненормальною;
- колективна аномалія, яка вимагає більше знань про самі дані, оскільки аномалія може бути в послідовності.

Ця класифікація поділяє аномалії на різні категорії на основі характеристик аномальних даних.

Точкова аномалія – це аномальна точка даних, яка знаходиться далеко від інших точок даних.

Цей тип може виникнути у вигляді раптового піку на графіках, і зазвичай це спотворює дані, особливо під час їх агрегування. Такі винятки потрібно нормалізувати, перш ніж продовжувати аналізувати дані. Аномалія точки може бути спричинена тим, що щось не так із датчиком або під час зчитування датчика.

У контекстній аномалії дані здаються нормальними з однієї точки зору,

але за іншого сценарію вони можуть бути ненормальними. Наприклад, якщо рано вранці на вулицях є певна кількість транспорту, опівдні трафік може здаватися надзвичайно великим. Тому, контекстна аномалія вимагає знання контексту, щоб розпізнавати аномальні дані.

У колективних аномаліях дані потрібно проаналізувати перед тим, як класифікувати їх на нормальні та аномальні, оскільки аномалія виникає в послідовності, а не в точках. Будь-який виняток може призвести до колективної аномалії. Тільки після аналізу нормальної моделі, аномальну поведінку можна розпізнати.

Окрім наведених трьох типів аномалій, існує ще декілька типів аномалій, які зазвичай зустрічаються у середовищах IoT:

- аномалії несправності пристрою виникають, коли пристрої IoT поводяться ненормально через апаратні збої, помилки програмного забезпечення або пошкоджену передачу даних та можуть проявлятися у вигляді незвичайних вихідних даних або збою в роботі;

- аномалії порушення безпеки вказують на інциденти безпеки, такі як несанкціонований доступ, порушення даних або інші форми кібератак та можуть бути виявлені через незвичайний мережевий трафік, неочікувану поведінку пристрою або неавторизоване виконання команд;

- екологічні аномалії виникають, коли зовнішні фактори навколишнього середовища впливають на пристрої або ж коли датчики можуть реєструвати ненормальні показання через коливання температури, фізичне втручання або інші впливи навколишнього середовища;

- аномалії зв’язку трапляються, коли виникають збої або нерегулярності в моделях зв’язку між пристроями, що може статися через проблеми з мережею, перешкоди сигналу або зловмисні атаки, спрямовані на порушення нормальної роботи;

- аномалії зниження продуктивності, виникають коли пристрої або мережа демонструють зниження показників продуктивності, наприклад уповільнення часу відгуку або зниження пропускної здатності, та це може

сигналізувати про основну проблему, яка відхиляється від стандартів;

- аномалії цілісності даних виникають, коли дані, створені пристроями IoT, є суперечливими, неповними або явно пошкодженими, що може вплинути на процеси прийняття рішень і ефективність роботи;

- аномалії споживання енергії можуть призвезти до стрибків або падіння споживання енергії та можуть вказувати на проблеми чи неефективність у роботі пристрою, що може бути спричинено проблемами з апаратним забезпеченням або зловмисними діями.

Такі аномалії потрібно не тільки розпізнавати, але й усувати, або ж якимось чином ними маніпулювати.

1.4.2 Проблеми при виявленні аномалій

Виявлення аномального трафіку в IoT є передусім комплексною задачею, адже існує ряд проблем у цьому процесі. Першою проблемою є складність багатовимірних часових рядів. На відміну від одновимірних часових рядів, багатовимірні часові ряди включають дані з кількох джерел, що ускладнює виявлення аномалій. Взаємозалежность між різними потоками даних вимагає більш складних методів аналізу даних.

Другою складністю є сегментовані аномалії. У часових рядах аномалії можуть не з'являтися як ізольовані точки, а як незвичайні патерни сегментів. Це ставить під сумнів традиційні методи виявлення аномалій, які зазвичай призначені для ідентифікації окремих викидів, а не патернів аномалій.

Наступною складністю є висока кількість хибнопозитивних результатів у сфері безпеки. Ефективні системи виявлення аномалій є життєво важливими для ідентифікації порушень безпеки. Однак ці системи часто страждають від високої кількості хибнопозитивних результатів, що частково пов'язано з потребою відрізнити нормальну та аномальну поведінку з великих потоків даних.

Іншою проблемою є проблеми з адаптацією, адже IoT системи є

динамічними та з постійно змінюваними даними, тому багатьом існуючим методам аналізу даних важко адаптуватися до нових патернів аномалій, що призводить до помилок. Окрім вищенаведених проблем, існує складність виявлення аномалії на рівні пристрою, адже багато IoT пристроїв мають обмеження у доступних ресурсах, що знижує можливість використання складних методів виявлення аномалій.

Також варто наголосити на проблемі винятковості аномальних даних. Аномальні події, як правило, є рідкісними порівняно з нормальними даними, що призводить до незбалансованих наборів даних. Цей дисбаланс може спотворити ефективність моделей виявлення аномалій, роблячи їх менш ефективними в ідентифікації справжніх аномалій.

Ці проблеми підкреслюють необхідність удосконалення методів виявлення аномалій, які могли б впоратися з проблемними фактормаи систем IoT, таких як розробка більш адаптивних методів, зниження кількості хибнопозитивних результатів і пошук ефективних способів обробки енергетичних і обчислювальних обмежень.

1.5 Постановка задач дослідження

Таким чином, можна виділити наступні задачі дослідження, що повинні бути вирішені в рамках даної роботи:

- провести аналіз архітектури та протоколів в IoT;
- провести аналіз класифікації методів виявлення аномалій в IoT;
- обрати групу методів, що доцільно використовувати для виявлення аномального трафіку в IoT;
- реалізувати та провести навчання моделей машинного навчання для виявлення аномального трафіку в IoT;
- підібрати гіперпараметри моделей для покращенння точності виявлення аномалій;
- провести аналіз отриманих результатів.

2 МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ АНОМАЛЬНОГО ТРАФІКУ

2.1 Методи виявлення аномального трафіку в IoT

Безпека даних в IoT є критично важливою, оскільки аномалії в мережевому трафіку можуть вказувати на спроби зловмисників проникнути в систему, викрасти дані або здійснити інші шкідливі дії.

Виявлення аномального трафіку в IoT є складним завданням через високу різноманітність та обсяг даних, а також обмежені обчислювальні ресурси багатьох IoT пристроїв. Однак, існують ефективні методи, які можуть допомогти у вирішенні цієї проблеми.

Наразі більшість методів виявлення аномалій у IoT вимагають значної участі людини для локальних рішень. Теоретично, аномалію легко зрозуміти, і експерт у цій галузі зможе виявити аномальні дані, якщо матиме достатньо часу.

Однак існує кілька труднощів у розробці автоматизованої моделі в IoT середовищі. Складно і не завжди можливо правильно визначити та класифікувати всі типи аномальних даних, особливо коли марковані тренувальні дані лише частково доступні або взагалі відсутні.

Крім того, дані часто містять шум, і коли співвідношення сигналу та шуму низьке, величина шуму нагадує справжні аномалії. Складність збільшується з ростом кількості взаємозв'язаних систем та різноманітністю типів вхідних даних.

2.1.1 Класифікація методів виявлення аномального трафіку

Методи виявлення аномального трафіку можна класифікувати за різними критеріями, такими як підхід до навчання, тип аналізу даних та

використання моделі.

Наприклад, один метод може бути кращим для виявлення аномалій у вимірах датчиків, а інший – для виявлення відхилень у мережному трафіку. У задачі бінарної класифікації аномалій велике значення має вибір моделі наближення, яка найкраще відображає очікувану поведінку даних. Точність цієї моделі визначає, наскільки ефективно будуть виявлені аномалії.

Оскільки IoT включає в себе різноманітні застосунки та типи даних, часто потрібно використовувати різні стратегії для виявлення аномалій для конкретних сценаріїв. На рисунку 2.1 показана діаграма розсіювання з прикладом аномалії [7].

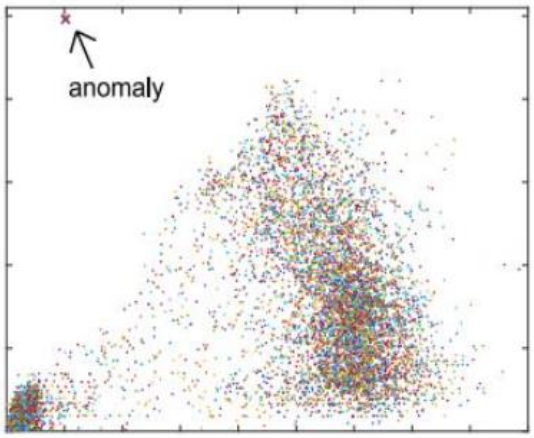

Рисунок 2.1 – Діаграма розсіювання з прикладом аномалії

Методи виявлення аномалій в IoT поділяються на чотири категорії, комбінуючи класифікації з опублікованих результатів досліджень [8]. Їх класифікують за способом підходу до проблеми, застосуванням, типом методу та затримкою алгоритму.

Нижче наведено короткий огляд цих методів та деяких традиційних підходів, що використовуються в IoT.

Геометричні методи ґрунтуються на ідеї, що при стратегіях, основаних на відстані та щільності даних, очікувані та аномальні дані розділені. Зазвичай вони використовують статичний або динамічний поріг для класифікації даних

як нормальних або аномальних. Декілька прикладів геометричних методів включають в себе методи на основі відстані та щільності.

Статистичні методи намагаються моделювати нормальні дані за допомогою математичних моделей та розподілів. Один із прикладів є метод мінімального об'єму, який намагається створити *n*-вимірний симплекс навколо заданої області даних.

Методи машинного навчання та глибокого навчання базуються на вибірі моделі, яка залежить від характеру наданих даних. З іншого боку, моделі типу Convolutional Neural Network (CNN) та Autoencoder (AE) підходять для непослідовних даних, таких як зображення.

Далі наведена класифікація методів за застосуванням. Наприклад, конструктивні застосування спрямовані на позитивну діяльність та надають користь, таку як моніторинг щоденної активності літніх людей для попередження падінь.

У той час як, деструктивні застосування спрямовані на завдання шкоди, такі як атаки на мережу IoT або намагання завдати шкоди даним та застосункам. Застосування для очищення даних спрямовані на видалення непотрібних даних або шуму з вхідного сигналу.

Окрім цього, існує класифікація методів за типом аномалії. Пунктові аномалії виникають, коли одна точка даних відхиляється від очікуваної поведінки. Прикладом може бути виявлення шахрайства з бан-ківськими картками.

Контекстуальні аномалії, які можуть вважатися такими лише в певному контекстіі виявляються, коли розглядаються як контекстуальні, так і поведінкові характеристики.

Колективні аномалії визначаються на основі всього набору даних та не пов'язані з окремими точками даних.

Класифікація за затримкою включає наступні підходи. Onlin – методи, які обробляють дані під час їх збору і можуть аналізувати одну точку даних або вікно даних без повного доступу до всіх даних.

Offline-методи мають доступ до всіх даних і використовують більш складні обчислювальні методи для розв'язання задачі.

Ця категоризація вказує на різноманітність методів виявлення аномалій в IoT та їх застосування в залежності від конкретного сценарію та потреби. Основні переваги та проблеми методів виявлення аномалій в IoT продемонстровані у таблиці 2.1.

Наведена таблиця результатів аналізу надає загальний огляд переваг та недоліків кожної категорії методів виявлення аномалій в Інтернеті речей відповідно до різних аспектів їх використання та застосування.

Таблиця 2.1 – Основні переваги та недоліки методів

Категорія методу	Переваги	Недоліки
За методом		
Геометричні методи	Добре підходять для даних з чітко визначеними структурами	Можуть бути неефективними для даних із складними структурами або часово залежними даними
Статистичні методи	Можуть моделювати різноманітні розподіли даних	Вимагають чіткого розуміння розподілу даних, що моделюються, і можуть бути неефективними для даних зі складними структурами або змінами з часом

Продовження таблиці 2.1

Категорія методу	Переваги	Недоліки
За застосуванням		
Методи машинного навчання та глибокого навчання	Можуть виявляти складні аномалії та залежності між даними	Вимагають великої кількості даних для тренування. Можуть бути складними для налаштування та оптимізації
За застосуванням		
Конструктивні застосування	Надають користь та вирішують практичні завдання	Вимагають розробки специфічних застосунків для кожного випадку
Деструктивні застосування	Допомагають виявляти та запобігати шкідливим діям та атакам	Зазвичай потребують додаткових заходів для захисту систем. Можуть призводити до фальсифікації або неправильного реагування
Застосування для очищення даних	Допомагають видалити непотрібні дані та шум з даних	Можуть втрачати корисну інформацію. Вимагають заздалегідь відомих шаблонів для очищення
За типом аномалії		
Пунктові аномалії	Відокремлюють аномалії, які виникають в окремих точках даних	Можуть пропустити аномалії, які виникають лише в контексті

Продовження таблиці 2.1

Категорія методу	Переваги	Недоліки
За типом аномалії		
Контекстні аномалії	Враховують контекст та поведінкові характеристики для виявлення аномалій	Вимагають складних аналітичних методів та більше обчислювальних ресурсів
Колективні аномалії	Визначають аномалії на основі всього набору даних та структури взаємозв'язків між даними	Можуть бути обчислювально витратними та вимагати великої кількості даних для навчання
За затримкою		
Online алгоритми	Здатні обробляти дані під час їх збору та аналізувати їх в реальному часі	Можуть бути обмеженими за ресурсами та вимагати високої затримки
Offline алгоритми	Мають доступ до всього набору даних і можуть використовувати більш складні обчислювальні методи	Зазвичай вимагають більше обчислювальних ресурсів та можуть бути повільнішими в роботі

2.1.2 Обмеження та вимоги до методів виявлення аномалій в IoT

Методи виявлення аномалій включають в себе етап попередньої обробки для визначення нормального діапазону значень, де будь-яке значення в межах визначеного діапазону вважається нормальним. Натомість будь-яке інше значення є аномалією.

Для потоку даних, залежного від часу, стандартний діапазон значень

може змінюватися в залежності від повторюваного циклу.

Тому правильне визначення повторюваного циклу має вирішальне значення для точності процесу виявлення аномалій. Тому, слід зазначити наступні важливі обмеження та вимоги:

- визначення довжини повторюваного циклу є найважливішим кроком в аналізі даних IoT, адже неправильна довжина циклу призводить до невірного виявлення аномалій;
- виявлення аномалій на початкута в кінці кожного циклу є більш складним, оскільки різниця між нормальним станом та аномальним станом є незначною, тому, ймовірність помилки є значною;
- підтримки точності у стандартних показниках, вимагається постійно перевіряти правильність значень оболонок та їх адаптацію до визначеного циклу і передбачати природні та обґрунтовані зміни в циклі та відповідні значення, що використовуються для перевірки аномалій з плином часу [1].

Окрім, того з огляду на результати аналізу, що наведено у попередніх підрозділах, можна сформулювати додаткові обмеження та вимоги до методів виявлення аномалій в IoT.

Передусім – це висока точність, адже методи виявлення аномалій повинні бути досить точними у виявленні незвичайних подій або аномалій. Особливо важливо виявляти аномалії в реальному часі для запобігання можливим проблемам.

Не менш важливим фактором є адаптованість до змін. Середовище IoT може змінюватися, і тому методи мають бути адаптованими до нових умов та типів даних та мають бути здатними навчатися на нових даних та оновлювати моделі.

Окрім вище описаного, для методів необхідна низька обчислювальна складність. Оскільки IoT може включати велику кількість пристроїв з обмеженими ресурсами, методи повинні бути ефективними з точки зору обчислень і споживання енергії.

Варто згадати про здатність до роботи у режимі реального часу. Деякі

випадки виявлення аномалій вимагають негайного реагування. Методи повинні бути здатними працювати в режимі реального часу та виявляти аномалії негайно.

Також методи повинні підтримувати роботу з різними типами даних. IoT може генерувати різноманітні типи даних, від сенсорних даних до великих обсягів текстової інформації. Методи повинні бути придатними для роботи з різними видами даних.

Також методи повинні мати захист від фальсифікації та атак. Методи повинні бути відповідними до заходів з безпеки, оскільки IoT може бути піддана атакам та фальсифікаціям даних.

І нарешті, методи повинні підтримувати масштабованість, тобто повинні бути придатними для роботи у масштабах, що відповідають маштабам використання IoT, де кількість пристроїв і обсяги даних можуть бути дуже великими.

2.2 Методи машинного навчання для виявлення аномалій

Машинне навчання відіграє ключову роль у виявленні аномалій IoT. Завдяки здатності аналізувати великі обсяги даних та виявляти приховані закономірності, методи машинного навчання забезпечують ефективні інструменти для ідентифікації аномалій, які можуть вказувати на можливі загрози або несправності в системі.

Виявлення аномалій є складним завданням через різноманітність і динамічність трафіку IoT, а також обмежені обчислювальні ресурси багатьох пристроїв IoT. Однак сучасні методи машинного навчання дозволяють ефективно вирішувати ці проблеми. Наприклад, методи класифікації, які дозволяють розділяти дані на нормальні та аномальні. Інші методи кластеризації допомагають групувати схожі дані, виділяючи ті, що відрізняються від інших, як потенційні аномалії.

Кластеризація є особливо корисною, коли немає чітких даних про

нормальну поведінку, оскільки вона дозволяє ідентифікувати групи даних, що відрізняються від основної маси.

Метод часових рядів також широко використовується для виявлення аномалій у даних IoT. Він дозволяє відслідковувати зміни в даних протягом часу, ідентифікуючи відхилення від звичайної поведінки. Метод часових рядів є особливо корисним для виявлення поступових змін або трендів, що можуть свідчити про повільний розвиток проблеми.

Використання гібридних методів, які поєднують кілька методів машинного навчання, дозволяє підвищити точність та надійність виявлення аномалій.

Наприклад, поєднання класифікації та кластеризації може забезпечити більш точне виявлення аномалій, ніж використання одного методу. Гібридні моделі можуть також використовуватися для обробки різних типів даних або для комбінування результатів різних алгоритмів, підвищуючи їх загальну ефективність.

Крім того, важливим аспектом є використання методів автоматичного навчання, які дозволяють автоматизувати процес створення моделей машинного навчання. Це є особливо корисним для IoT, де часто потрібно швидко реагувати на нові загрози або зміни в трафіку.

Важливою частиною системи виявлення аномалій є також постійний моніторинг та оновлення моделей.

Оскільки характер трафіку IoT може змінюватися з часом, методи та моделі машинного навчання повинні регулярно оновлюватися на основі нових даних, щоб залишатися ефективними. Постійний моніторинг дозволяє швидко виявляти нові типи загроз та реагувати на них у реальному часі.

Тому методи машинного навчання є потужними інструментами для виявлення аномалій в IoT.

Вони дозволяють аналізувати великі обсяги даних, виявляти приховані закономірності та прогнозувати потенційні загрози. Комбінація різних методів, гібридні моделі та постійний моніторинг є ключовими факторами для

забезпечення безпеки та ефективної роботи IoT систем.

2.2.1 Методи машинного навчання з учителем

Методи навчання з учителем використовують помічені навчальні набори, що містять як нормальні, так і аномальні зразки для побудови моделей прогнозування. Ці моделі проходять ретельну оцінку, враховуючи такі показники, як точність, запам’ятовування та оцінка F1, щоб забезпечити точне визначення аномалії.

Істотний виклик полягає в отриманні мічених даних, особливо для рідкісних або нових аномалій, які можуть вимагати складних методів збору даних або синтезу.

Метод випадкового лісу – це метод машинного навчання, що складається з кількох дерев рішень. Він стійкий до перенавчання, добре працює з даними, що містять шуми та з аномаліями. Також цей метод може обробляти дані великої розмірності, але через це може мати дорогу обчислювальну вартість через велику кількість дерев у лісі. Метод випадкового лісу вирізняється тим, що надає сильні та надійні класифікації, поєднуючи рішення з кількох дерев.

Метод опорних векторів є ефективним методом машинного навчання, який спрямований на пошук гіперплощини, яка найкраще розділяє класи шляхом максимізації простору між ними.

Незважаючи на свою ефективність, метод опорних векторів може зіткнутися з обчислювальними проблемами при аналізу великих даних. Також налаштування гіперпараметрів методу опорних векторів, таких як вибір ядра та параметра регуляризації, може мати вирішальне значення для його продуктивності. Цей метод є особливо ефективний, коли межа між нормальними та аномальними випадками є чітко визначеною.

Метод дерева рішень поділяє дані шляхом рекурсивного розбиття на основі атрибутів функції. Він високо інтерпретується, що дозволяє легко

візуалізувати правила прийняття рішень. Однак він, як правило, переповнюється, коли дерево росте занадто глибоко, і йому може бути важко вловити складні зв’язки в даних.

Метод *k* найближчих сусідів класифікує точки даних на основі класу більшості серед їхніх *k* найближчих сусідів. Його можна адаптувати до нестандартних типів даних, таких як текст або зображення. Однак його продуктивність значною мірою залежить від вибору метрики відстані та значення *k*. Крім того, це може бути дорогим з точки зору обчислень, особливо з великими наборами даних, оскільки вимагає обчислення відстані між точкою запиту та всіма точками навчання.

2.2.2 Методи машинного навчання без учителя

Методи машинного навчання без учителя працюють без помічених даних, сприпаючись на припущення щодо статистичних відмінностей між нормальними та ненормальними випадками. Ці методи мають вирішальне значення, коли позначені дані є дефіцитними або дані є новими або еволюційними.

Метод кластеризації поділяє дані на *k* кластерів на основі показників подібності, щоб мінімізувати відстані між кластерами. Він є ефективним і добре працює з великими наборами даних. Однак він чутливий до початкового розміщення центроїдів, бореться з несферичними кластерами та вимагає попереднього знання кількості кластерів *k*.

Метод просторової кластеризації ідентифікує кластери на основі щільності в просторі даних, розрізняючи основні точки, граничні точки та шум. Він здатний ідентифікувати кластери довільної форми і не вимагає попереднього вказівки кількості кластерів.

Однак встановлення відповідних параметрів, таких як епсилон і мінімальні точки, може бути складним завданням.

Метод ізольованого лісу виділяє аномалії шляхом випадкового

розділення простору даних і виявлення аномалій у меншій кількості розділів. Він є ефективним для великих наборів даних і не передбачає розподілу базових даних. Однак він може мати проблеми з мультимодальними даними та не є ефективним у виявленні аномалій, близьких до нормальних випадків.

Метод однокласових опорних векторів має на меті відокремити нормальні випадки від аномалій у гіперпросторі. Він підходить для виявлення аномалій, коли для навчання доступні лише нормальні дані. Однак визначення відповідного ядра та встановлення гіперпараметрів може бути складним завданням.

2.3 Метод дерева прийняття рішень

Метод дерева прийняття рішень є потужним методом машинного навчання для виявлення аномалій у IoT. Дерево рішень використовує ієрархічну структуру для прийняття рішень на основі значень ознак даних.

Метод дерева рішень використовується для прийняття рішень та класифікації даних за допомогою послідовних розщеплень. Дерево прийняття рішень складається з трьох компонентів:

- вузлів рішень;
- листових вузлів;
- кореневого вузла.

Вузли представляють точки прийняття рішень, де дані поділяються на підмножини на основі значень ознак. Гілки з'єднують вузли, показуючи шлях поділу даних. Листки є кінцевими вузлами дерева, що містять остаточне рішення або передбачення. На рисунку 2.2 представлена структура дерева прийняття рішень.

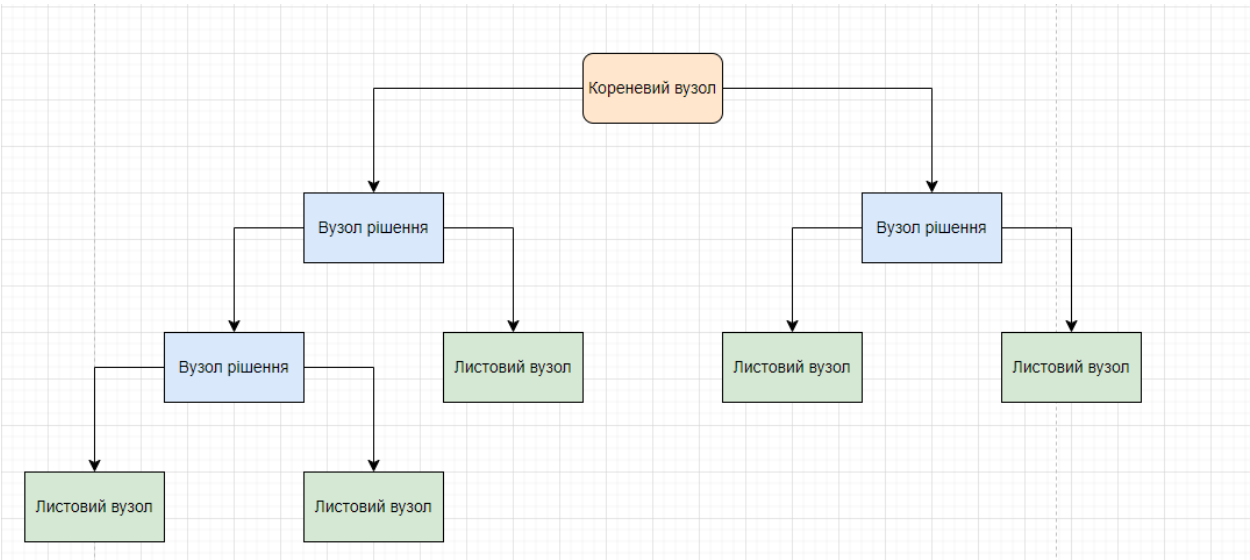

Рисунок 2.2 – Структура дерева прийняття рішень

Цей метод може бути ефективно застосований для аналізу характеристик трафіку IoT та виявлення відхилень від нормальної поведінки. Основні кроки для виявлення аномалій в IoT за допомогою метода дерева прийняття рішень:

- збір даних з пристроїв IoT;
- підготовка даних;
- маркування даних;
- побудова дерева прийняття рішень;
- навчання моделі дерева прийняття рішень;
- оцінка моделі дерева прийняття рішень;
- виявлення аномалій;
- аналіз результатів.

Першим кроком є збір даних з пристроїв IoT. Це можуть бути дані про мережевий трафік, показники роботи пристроїв, лог-файли та інша релевантна інформація. Зібрані дані є основою для подальшого аналізу.

Наступним кроком є підготовка зібраних даних. Цей етап включає очищення даних від шуму, заповнення відсутніх значень, нормалізацію та трансформацію змінних для забезпечення їх коректного вигляду та формату,

необхідного для побудови моделі.

Далі необхідно промаркувати дані. Для цього потрібно присвоїти мітки кожному зразку даних чи є він нормальним або аномальним. Марковані дані дозволяють моделі навчитися розрізняти нормальні та аномальні патерни.

На цьому етапі обирається алгоритм побудови дерева прийняття рішень та визначаються його параметри. Це включає вибір критеріїв розщеплення вузлів, глибини дерева та інших параметрів, що впливають на точність та ефективність моделі.

Після побудови дерева прийняття рішень, модель навчається на навчальних даних. Дані поділяються на навчальну та тестову вибірки для оцінки продуктивності моделі. Навчання включає побудову дерева прийняття рішень на основі навчальної вибірки та перевірку його ефективності на тестовій вибірці.

Після навчання моделі необхідно оцінити її точність та ефективність. Це здійснюється за допомогою різних метрик, таких як точність, повнота, F1-міра та інших. Оцінка дозволяє зрозуміти, наскільки добре модель справляється з виявленням аномалій.

Після успішної оцінки, модель застосовується до нових даних для виявлення аномалій у мережевому трафіку IoT. Модель аналізує дані та ідентифікує патерни, що відхиляються від норми.

Останнім кроком є аналіз виявлених аномалій. Це включає детальний розгляд аномальних випадків, розуміння причин їх виникнення та прийняття рішень для покращення безпеки та надійності IoT систем. Аналіз результатів допомагає виявити слабкі місця та вдосконалити підхід до виявлення аномалій у майбутньому.

Метод дерева прийняття рішень може бути адаптований для задач виявлення аномалій, де ціль полягає у виявленні зразків, що відрізняються від нормальних. Це досягається шляхом тренування моделі на нормальних даних та класифікації зразків, які відхиляються від нормальної поведінки, як аномальні.

Метод дерева прийняття рішень є потужним інструментом для виявлення аномалій в IoT мережах завдяки своїй здатності ефективно обробляти дані та надавати зрозумілі результати. Завдяки використанню ієрархічної структури для прийняття рішень, метод дерева прийняття рішень може забезпечити високу точність і надійність виявлення аномалій.

2.3.1 Математична модель дерева прийняття рішень

Математична модель дерева прийняття рішень базується на принципі поділу даних на менші частини, доки всі не підпадають під одну категорію.

Поступово дерево перетворює їх на дискретні значення перед побудовою дерева. Спочатку визначається ентропія цілі. Далі набір даних розбивається на атрибути та обчислюється сума ентропії для всіх класів в атрибутах. Потім отримана ентропія віднімається від цільової ентропії перед поділом.

Для розділення набору даних вибирається атрибут із найбільшим приростом інформації. Цей процес повторюється знову і знову, поки не отримаємо чисту класифікацію. Ентропія обчислюється виразом

$$E(\mathrm{S}) = \sum_{i=1}^{\mathrm{c}} - p(i)\log(\mathrm{i}), \tag{2.1}$$

де S – множина даних;

c – кількість класів атрибута;

p – частка прикладів класу i.

Формула ентропії є ключовим поняттям у задачах класифікації та побудови дерев прийняття рішень.

Ентропія – це міра невизначеності в системі. У контексті класифікації, ентропія визначає, наскільки неоднорідною є множина даних. Вона вимірюється у бітах. Чим вище значення ентропії, тим більше випадковості або невизначеності міститься у множині даних.

Умовна ентропія – це середнє значення ентропії після розділення множини даних за певним атрибутом. Умовна ентропія вимірює невизначеність в множині даних, враховуючи певний атрибут.

Приріст інформації – це міра того, наскільки добре атрибут розділяє множину даних. Вона вимірює зменшення невизначеності після розділення множини даних за певним атрибутом. Чим вище значення приросту інформації, тим краще атрибут розділяє дані. Приріст інформації задається виразом

$$\mathrm{G(S,A)} = \mathrm{E(S)} - \mathrm{E(A)}, \tag{2.2}$$

де *S* – множина даних;

A – атрибут, за яким ми хочемо розділити множину даних.

Таким чином, обчислює приріст інформації, відображаючи різницю між ентропією множини даних до та після розділення за певним атрибутом. Використовуючи цей показник, можна вибрати атрибут для побудови дерева прийняття рішень, забезпечуючи ефективне розділення даних на класи.

2.4 Метод випадкового лісу

Метод випадкового лісу широко використовується для задач класифікації та регресії. Він використовує техніку ансамблю, відому як агрегування або пакетування, для побудови прогнозів. Ця техніка базується на створенні багатьох дерев прийняття рішень, кожне з яких навчається на різних підмножинах тренувальних даних, що сприяє різноманітності у прогнозах. Результати від кожного дерева прийняття рішень збираються, і той, що отримує найбільшу кількість голосів, вибирається як кінцевий результат.

Метод випадкового лісу використовує ансамбль дерев для забезпечення точності прогнозу. Кількість дерев у випадковому лісі зазвичай залежить від розміру і складності набору даних. Кожне дерево складається з вузлів рішень

та листових вузлів, причому листові вузли представляють кінцеві прогнози кожного дерева. Вибір кінцевого прогнозу випадкового лісу відбувається за мажоритарним принципом голосування серед дерев. На рисунку 2.3 зображена схема створення випадкового лісу.

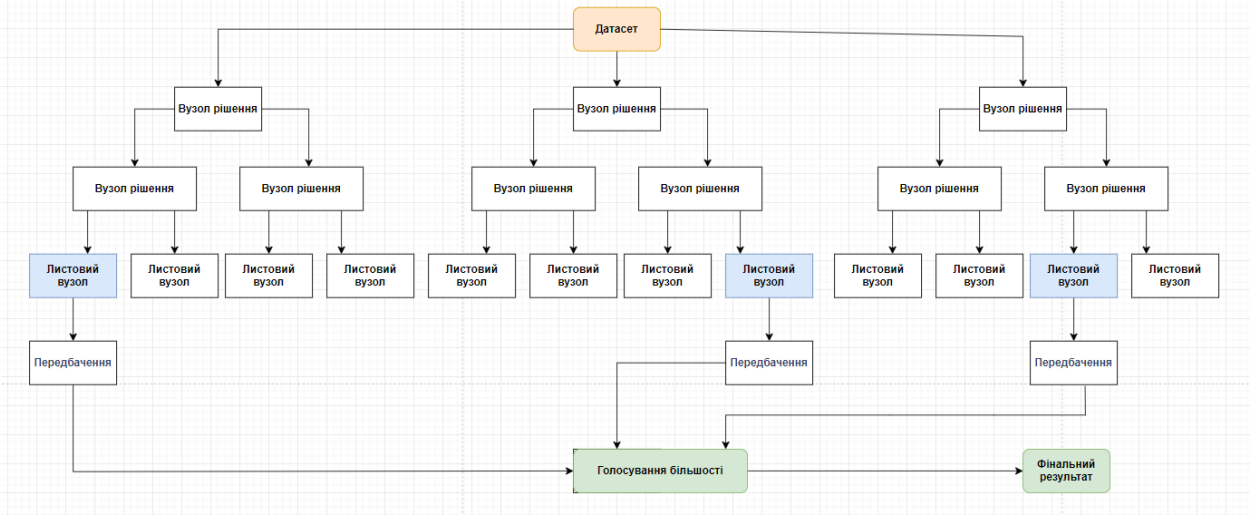

Рисунок 2.3 – Схема створення випадкового лісу

Якщо порівнювати його з попереднім методом дерева прийняття рішень, то можна стверджувати, що метод випадкового лісу дозволяє усунути обмеження використання одного дерева. Це зменшує перенавчання і також підвищує точність результатів. Метод випадкового лісу містить наступні кроки для виявлення аномалій в IoT:

- збір даних;
- маркування даних;
- навчання моделі випадкового лісу;
- оцінка моделі випадкового лісу;
- виявлення аномалій;
- аналіз результатів.

Спершу потрібно зібрати дані з пристроїв IoT, що можуть включати показники датчиків, стани експлуатації або інші відповідні метрики.

Далі необхідно перетворити отриману інформацію у набір ознак, які

можуть ефективно представляти поведінку системи IoT. Це може включати статистичні підсумки, такі як середнє значення, медіана, дисперсія або більш складні ознаки, такі як коефіцієнти Фур'є або автокореляції.

Потім потрбіно потрібно промаркувати дані як "нормальні" або "аномальні". Це може бути зроблено за допомогою правил на основі знань у галузі або через ручне маркування експертами.

Далі промарковані дані використовуються для тренування моделі випадкового лісу. Дані випадково розподіляються на численні підмножини, кожна з яких використовується для побудови дерева прийняття рішень. Дерева повинні відображати різні аспекти даних, роблячи ансамбль стійким проти перенавчання.

Для нових потоків даних вхідні дані передаються в модель випадкового лісу. Кожна порція даних оцінюється всіма деревами, і результат агрегується для визначення, чи є він нормальним або аномальним. Зазвичай використовується голосування для класифікації у випадковому лісі.

Основні концепти методу випадкового лісу включають ансамблеве навчання, беггінг і випадковий вибір ознак.

Ансамблеве навчання – це метод машинного навчання, який об'єднує передбачення кількох моделей для покращення загальної продуктивності.

Беггінг – це техніка, при якій для кожного дерева вибирається випадкова підмножина навчальних даних з поверненням. Це означає, що деякі зразки можуть бути використані кілька разів для навчання одного дерева. Беггінг допомагає зменшити варіацію моделі і покращує її стабільність та точність.

Випадковий вибір ознак означає, що на кожному вузлі дерева прийняття рішень розглядається випадкова підмножина ознак для розділення. Це зменшує кореляцію між деревами та підвищує продуктивність ансамблю, роблячи його більш стійким до перенавчання.

При використанні методу випадкового лісу слід пам’ятати про його особливості, серед яких є те, що кількість дерев у випадковому лісі є важливим гіперпараметром, який впливає на точність та узагальненість моделі та те, що

випадковий ліс є стійким до перенавчання завдяки введенню випадковості на етапі навчання.

2.4.1 Математична модель випадкового лісу

Метод випадкового лісу базується на створенні великої кількості дерев прийняття рішень та класифікації об'єкта на основі передбачень всіх цих дерев. Потім знаходиться середнє значення передбачення різних дерев прийняття рішень, і в якості остаточного передбачення зазвичай використовується медіана всіх передбачень побудованих дерев.

Процес побудови набору дерев прийняття рішень на основі тренувального набору даних є складною задачею. Якщо кожного разу під час побудови нового дерева прийняття рішень використовувати однакові дані, то результатом буде набір ідентичних дерев. Дерева прийняття рішень з великою глибиною зазвичай можуть бути схильні до перенавчання, що означає, що їхні прогнози мають маленьке зміщення, але дуже високу варіабельність.

Основна мета методу випадкового лісу полягає у зниженні варіабельності передбачень окремих дерев прийняття рішень. З початкового тренувального набору обираються K підвибірок даних, і на кожній з цих підвибірок тренується дерево прийняття рішень. Це знижує варіабельність передбачень без шкоди для зміщення завдяки слабкій кореляції між деревами.

Ще один ключовий аспект полягає в тому, що під час побудови дерева прийняття рішень та вибору ознаки для поділу використовується випадкова підмножина ознак. Це дозволяє ще більше знизити кореляцію між деревами, оскільки зменшує кількість входжень дуже важливих ознак, які впливають на кінцеве передбачення.

Метод випадкового лісу відноситься до класу нелінійних методів класифікації, що ускладнює інтерпретацію результатів. Модель класифікації, побудована на основі випадкового лісу, часто розглядається як "чорний ящик", оскільки складається з великої кількості глибоких дерев прийняття рішень,

кожне з яких навчено на деякій підмножині вихідних об'єктів та ознак.

Один із способів отримання інформації про важливість ознак – це перебір ознак і порівняння роботи моделі з урахуванням та без урахування певної ознаки.

Інший спосіб – це підрахунок невизначеності та невпорядкованості піддерева, яке вилучає розбиття за цією ознакою. Мірою невизначеності може служити коефіцієнт Джині або ентропія.

У класичному визначенні дерева прийняття рішення функція передбачення визначається виразом

$$f(x) = \sum_{m=1}^{M} c_m I(x, R_m), \tag{2.3}$$

де x – вектор ознак;

M – кількість листків дерева;

R_m – регіон простору ознак, що відповідає листку m;

C_m– константа, що відповідає регіону;

I – індикаторна функція, яка повертає 1, якщо x належить R_m і 0 в іншому випадку.

Сумує внесок кожного регіону R_m, до якого належить вектор x. Якщо вектор x належить регіону R_m, індикаторна функція I дорівнює 1, і відповідне значення c_m включається в суму. Якщо вектор x не належить регіону R_m, індикаторна функція дорівнює 0, і значення c_m не включається в суму. Таким чином, *f(x)* обчислює передбачення моделі дерева прийняття рішень для даного вектора ознак x. У кінці *f(x)* поверне класову мітку або ймовірність для вектора ознак x.

Для випадкового лісу функція передбачення визначається як середнє значення передбачень окремих дерев

$$F(x) = \frac{1}{J}\sum_{j=1}^{J} f_j(x), \tag{2.4}$$

де J – кількість дерев у лісі.

Формула працює наступним чином: спочатку для кожного дерева j у випадковому лісі обчислюється передбачення в $f_j(x)$ для даного вектора ознак x. Це передбачення залежить від структури дерева і значень ознак x.

Далі передбачення всіх дерев усереднюються, щоб отримати остаточне передбачення $F(x)$.

Формула (2.2) означає, що сума всіх передбачень ділиться на кількість дерев, таким чином обчислюється середнє значення передбачень. Для задач класифікації кожне дерево повертає класову мітку або ймовірність.

Остаточне передбачення $F(x)$ може бути обчислено шляхом голосування більшості або шляхом усереднення ймовірностей, якщо кожне дерево повертає ймовірность.

Наприклад, якщо 3 дерева передбачають клас A, а 2 дерева – клас B, то остаточне передбачення буде клас A, що обрано за більшість голосів.

2.5 Метод опорних векторів

Метод опорних векторів є ефективним методом для виявлення аномалій у IoT. Він виявляє аномалії, аналізуючи характеристики трафіку та визначаючи, чи відрізняється зразок від нормальної поведінки. Метод опорних векторів містить наступні кроки для виявлення аномалій в IoT:

- збір даних;
- підготовка даних;
- маркування даних;
- вибір ядра;
- навчання моделі опорних векторів;
- оцінка моделі опорних;
- виявлення аномалій;
- аналіз результатів.

Спочатку відбувається збір даних з IoT пристроїв. Це включає збір різноманітних характеристик трафіку, таких як обсяг переданих даних, кількість підключень, час між підключеннями та інші показники, які можуть бути корисними для аналізу. Дані можуть збиратися за певний проміжок часу для отримання достатньої кількості інформації для аналізу.

Наступним етапом є підготовка даних. На цьому етапі дані очищаються від пропусків, а також масштабуються для нормалізації діапазонів значень ознак. Це робиться для того, щоб усі ознаки мали однакову вагу під час навчання моделі, і жодна з них не домінувала через свій діапазон значень.

Після підготовки даних відбувається їх маркування. Це означає, що дані поділяються на дві категорії: нормальні та аномальні. Для навчання моделі методу опорних векторів зазвичай використовується лише нормальні дані, щоб модель могла виявляти відхилення від нормальної поведінки.

Далі відбувається вибір ядра для методу опорних векторів. Ядро визначає, як модель буде обробляти дані. Найпоширенішими є лінійне, поліноміальне та радіально базисно функціональне ядра. Вибір ядра залежить від характеру даних і завдання, яке потрібно вирішити.

Після вибору ядра відбувається навчання моделі. Навчальний набір даних використовується для тренування моделі, під час якого модель вивчає, як відрізняти нормальні зразки від потенційно аномальних.

Параметри моделі, такі як гіперпараметр C і параметри ядра, налаштовуються для досягнення найкращої продуктивності. Після навчання модель оцінюється на тестовому наборі даних.

Це дозволяє перевірити, наскільки добре модель здатна виявляти аномалії. Оцінка проводиться за допомогою метрик, таких як точність, відчутливість, специфічність та інших, щоб визначити ефективність моделі.

Навчена модель потім використовується для виявлення аномалій у нових зразках трафіку IoT. Модель класифікує зразки як нормальні або аномальні на основі своєї навченої здатності розрізняти ці два типи.

Останнім етапом є аналіз результатів. Виявлені аномалії аналізуються

для підтвердження їх відповідності реальним аномаліям у мережі.

Модель також може бути налаштована за потреби для покращення точності. Аналіз результатів дозволяє зрозуміти, які ознаки були найбільш важливими для виявлення аномалій і як можна поліпшити процес виявлення в майбутньому.

Основні концепти методу опорних векторів включають наступні аспекти:

- гіперплощина;
- опорні вектори;
- маржа;
- ядра;
- регуляризаційний параметр *C*;
- функція втрат.

Гіперплощина є лінійною розділювальною поверхнею, яка розділяє простір ознак на дві частини, кожна з яких належить до різних класів. Для задач класифікації методу опорних векторів потрібно знайти оптимальну гіперплощину, яка максимально віддалена від найближчих точок даних кожного класу.

Опорні вектори – це точки даних, які знаходяться найближче до гіперплощини. Вони визначають положення та орієнтацію гіперплощини. Ці точки мають найбільший вплив на побудову гіперплощини і є критичними для рішення задачі класифікації.

Маржа – це відстань між гіперплощиною і найближчими точками даних або ж опорними векторами з кожного класу. Методу опорних векторів прагне максимізувати цю маржу, щоб забезпечити найкраще розділення класів. Широка маржа допомагає зменшити ризик перенавчання і підвищує загальну узагальнюваність моделі.

Ядра дозволяють методу опорних векторів ефективно працювати з нелінійними даними. Найпоширенішими ядрами є:

- лінійне ядро;

- поліноміальне ядро;
- радіально-базисно функціональне ядро;
- сигмоїдне ядро.

Ядра перетворюють вхідні дані в простір вищих вимірів, де їх можна розділити лінійно.

Гіперпараметр C контролює баланс між максимізацією маржі та мінімізацією помилок класифікації. Високе значення C надає більшої ваги мінімізації помилок, що може призвести до перенавчання, тоді як низьке значення C сприяє максимізації маржі, але може допускати більше помилок.

Метод опорних векторів використовує функцію втрат для вимірювання помилок класифікації. Найпоширенішою є гінгевтратна функція, яка штрафує неправильні передбачення та передбачення, що знаходяться всередині маржі. На рисунку 2.4 зображені основні компоненти методу опорних векторів.

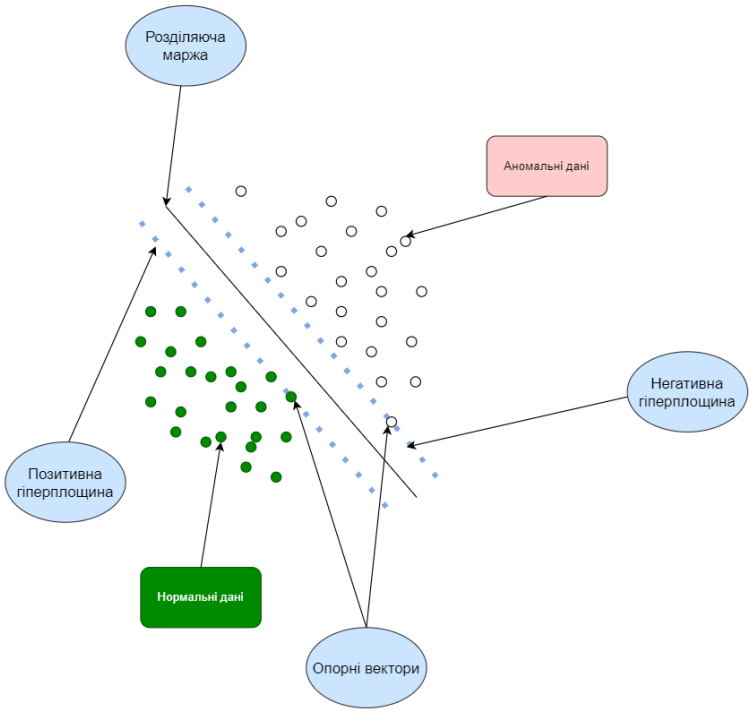

Рисунок 2.4 – Приклад компонентів методу опорних векторів

Метод опорних векторів є потужним інструментом для виявлення

аномалій в IoT мережах завдяки своїй здатності ефективно обробляти високовимірні дані та виявляти складні шаблони.

Завдяки максимізації маржі та використанню різних версій ядер для обробки нелінійних даних, метод опорних векторів може забезпечити високу точність і надійність виявлення аномалій.

2.5.1 Математична модель опорних векторів

У математичній моделі опорних векторів використовується набір пар векторів

$$(\overrightarrow{x_n}, y_n), \tag{2.5}$$

де $\overrightarrow{x_n}$ – вектор ознак для n-го зразка;

y_n – значення цільової змінної для n-го зразка.

У класичній постановці передбачається, що об'єкти двох класів у навчальній вибірці лінійно роздільні, тобто існує гіперплощина у просторі, відносно якої об'єкти двох класів розташовані по різні боки.

Модель опорних векторів розділяє гіперплощиною простір векторів на два класи, але крім цього він будує гіперплощину таким чином, щоб мінімальна відстань між побудованою гіперплощиною та вектором $\overrightarrow{x_n}$ з вибірки була максимальна.

Гіперплощина визначається рівнянням

$$\vec{w} \times \vec{x} - b = 0, \tag{2.6}$$

де $\vec{w}$ – вектор нормалі до гіперплощини;

$\vec{x}$ – вектор ознак;

b – скалярний зсув.

Тоді для лінійно роздільної вибірки можна вибрати дві паралельні

гіперплощини, які розділяють вибірку на два класи таким чином, що відстань між цими гіперплощинами максимальна.

Регіон, обмежений цими гіперплощинами, називається відступом. Вираз двох гіперплощин має вигляд

$$\vec{w} \times \vec{x} - b = 1; \tag{2.7}$$

$$\vec{w} \times \vec{x} - b = -1. \tag{2.8}$$

Відстань між цими гіперплощинами дорівнює

$$\frac{2}{\overrightarrow{||w||}}. \tag{2.9}$$

Оскільки завдання полягає в максимізації цієї відстані, потрібно мінімізувати вектор нормалі до гіперплощини.

Виходячи з того, що гіперплощини знаходяться на максимальній відстані, випливає, що на кожній з цих двох гіперплощин знаходиться принаймні один вектор $\vec{x}$ з вибірки. Такі вектори називаються опорними векторами.

Якщо результат передбачення збігся з правильним розташуванням об'єкта відносно відступу, функція втрат дорівнює нулю, інакше її значення пропорційне відстані від роздільної гіперплощини відповідного класу до об'єкта.

Завершальний функціонал, який підлягає мінімізації, має вигляд

$$\mathrm{C}\left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_1(\vec{w} \times \vec{x}_i - b)\right)\right] + \overrightarrow{||w||}^2. \tag{2.10}$$

Мінімізація норми $\vec{w}$ зазначена в методі опорних векторів з самого початку. Це можна розглядати як аналог регуляризації в логістичній регресії.

Коефіцієнт C відповідає за компроміс між збільшенням норми вектора $\vec{w}$ та забезпеченням того, що вектори $\vec{x}_i$ з навчальної вибірки знаходяться по правильну сторону від відступу.

Отже, метод опорних векторів будує гіперплощину, яка не лише розділяє класи, але й максимізує відстань до найближчих точок кожного класу, забезпечуючи оптимальну класифікацію з високою узагальнюючою здатністю.

3 РЕАЛІЗАЦІЯ І ДОСЛІДЖЕННЯ МЕТОДІВ ВИЯВЛЕННЯ АНОМАЛЬНОГО ТРАФІКУ В ІОТ

3.1 Опис загальної характеристики дослідження

У рамках кваліфікаційної роботи були реалізовані 3 моделі машинного навчання: модель випадкового лісу, модель опорних векторних та моделі дерева прийняття рішень. Тестування та навчання проводилося на наборі даних IoT23, який складається з 23 записів мережевого трафіку IoT, що містить як нормальний, так і аномальний мережевий трафік.

Результати дослідження показують, що ці моделі машинного навчання ефективні для виявлення аномального трафіку у пристроях IoT.

3.2 Проведення попередньої обробки даних

Для навчання моделей був обраний датасет IoT-23, оскільки він пропонує добре задокументований набір даних, придатний для навчання моделей машинного навчання.

На рисунку 3.1 зображені пристрої IoT, які використовуються у датасеті IoT-23. Серед них є персональний домашній асистестент Amazon Echo, розумний дверний замок Somfy та розумна світлодіодна лампа Philips HUE.

Рисунок 3.1 – Пристрої IoT з датасету IoT-23

Метою датасету IoT-23 є запропонувати великий набір даних помічених нормальних та аномальних даних з IoT. На рисунку 3.2 продемонстрований короткий опис шкідливих сценаріїв з датасету.

#	Name of Dataset	Duration (hrs)	#Packets	#ZeekFlows	Pcap Size	Name
1	CTU-IoT-Malware-Capture-34-1	24	233,000	23,146	121 MB	Mirai
2	CTU-IoT-Malware-Capture-43-1	1	82,000,000	67,321,810	6 GB	Mirai
3	CTU-IoT-Malware-Capture-44-1	2	1,309,000	238	1.7 GB	Mirai
4	CTU-IoT-Malware-Capture-49-1	8	18,000,000	5,410,562	1.3 GB	Mirai
5	CTU-IoT-Malware-Capture-52-1	24	64,000,000	19,781,379	4.6 GB	Mirai
6	CTU-IoT-Malware-Capture-20-1	24	50,000	3,210	3.9 MB	Torii
7	CTU-IoT-Malware-Capture-21-1	24	50,000	3,287	3.9 MB	Torii
8	CTU-IoT-Malware-Capture-42-1	8	24,000	4,427	2.8 MB	Trojan
9	CTU-IoT-Malware-Capture-60-1	24	271,000,000	3,581,029	21 GB	Gagfyt
10	CTU-IoT-Malware-Capture-17-1	24	109,000,000	54,659,864	7.8 GB	Kenjiro
11	CTU-IoT-Malware-Capture-36-1	24	13,000,000	13,645,107	992 MB	Okiru
12	CTU-IoT-Malware-Capture-33-1	24	54,000,000	54,454,592	3.9 GB	Kenjiro
13	CTU-IoT-Malware-Capture-8-1	24	23,000	10,404	2.1 MB	Hakai
14	CTU-IoT-Malware-Capture-35-1	24	46,000,000	10,447,796	3.6G	Mirai
15	CTU-IoT-Malware-Capture-48-1	24	13,000,000	3,394,347	1.2G	Mirai
16	CTU-IoT-Malware-Capture-39-1	7	73,000,000	73,568,982	5.3GB	IRCBot
17	CTU-IoT-Malware-Capture-7-1	24	11,000,000	11,454,723	897 MB	Linux,Mirai
18	CTU-IoT-Malware-Capture-9-1	24	6,437,000	6,378,294	472 MB	Linux.Hajime
19	CTU-IoT-Malware-Capture-3-1	36	496,000	156,104	56 MB	Muhstik
20	CTU-IoT-Malware-Capture-1-1	112	1,686,000	1,008,749	140 MB	Hide and Seek

Рисунок 3.2 – Короткий опис шкідливих сценаріїв IoT

Нижче наведено приклади значеннь з датасету:

- tf – мітка часу захоплення;
- uid – ідентифікатор захоплення;
- id_orig.h – IP-адреса джерела атаки;
- id_orig.p – порт джерела атаки;
- id_resp.h – IP пристрою IoT;
- id_resp.p – порт пристрою IoT;
- proto – протокол транспортного рівня з'єднання;
- duration – кількість часу обміну даними між пристроєм IoT і зловмисником;
- orig_bytes – кількість даних, надісланих на пристрій IoT;
- resp_bytes – кількість даних, надісланих пристроєм IoT;
- conn_state – стан підключення;

- missed_bytes – кількість пропущених байтів у повідомленні;
- orig_pkts – кількість пакетів, які надсилаються на пристрій IoT;
- orig_ip_bytes – кількість байтів, які надсилаються на пристрій IoT;
- resp_pkts – кількість пакетів, що надсилаються з пристрою IoT;
- resp_ip_bytes – кількість байтів, які надсилаються з пристрою IoT;
- label – тип захоплення, доброякісне чи шкідливе.

Варто зазначити, що у цьому датасеті описано кожен сценарій файлом з розширенням PCAP.

Також у дослідженні використовується система аналізу трафіку і виявлення мережевих вторгнень. система аналізу трафіку і виявлення мережевих вторгнень Zeek. Вона дозволяє виявляти зловмисне програмне забезпечення, вторгнення та іншу шкідливу діяльність шляхом аналізу мережі в режимі реального часу.

Для обробки даних була використана бібліотека Zat, для перетворення файлів журналу Zeek у структуру фрейму даних була обрана бібліотека Pandas. Також була використана бібліотека Dask для паралельних обчислень.

До початку навчання були видаленні стовпці з нульвими значеннями та у кінці набір даних мав лише 9000 зразків на клас. Загалом 63000 зразків.

Тому, набір даних достатній для завантаження в оперативну пам’ять за допомогою бібліотеки Pandas.

3.3 Вибір апаратних та програмних засобів для проведення дослідження

Для дослідження було використано ноутбук HP Omen з 4 гігабайтами оперативної пам'яті, мову програмування Python та веб-додаток Jupyter Notebook, який дозволяє створювати і ділитися документами, які містять живий код, рівняння, візуалізації та пояснювальний текст. На рисунку 3.3 представлений інтерфейс веб-додатку Jupyter Notebook, де наведена програмна реалізація.

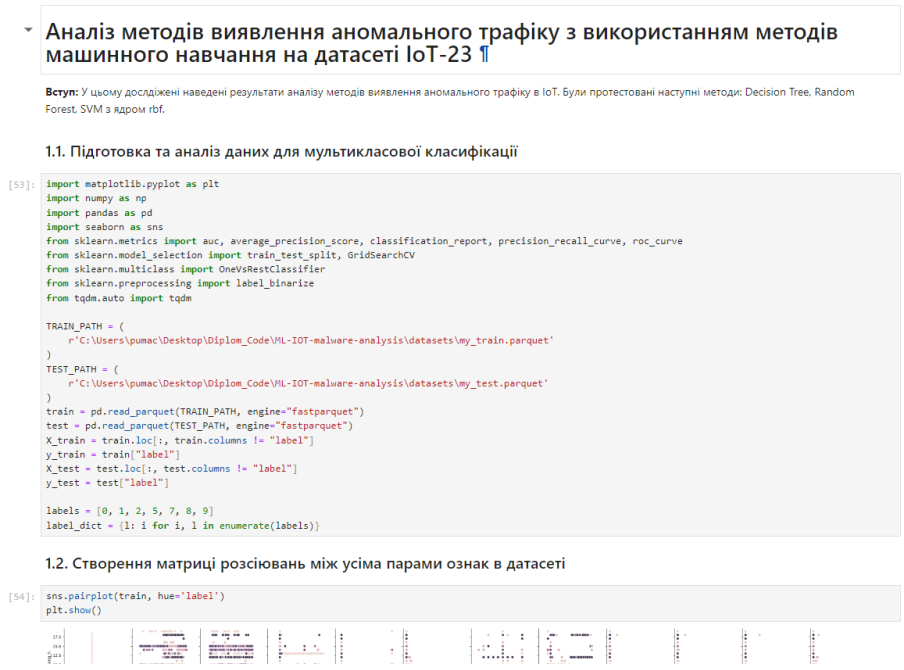

Рисунок 3.3 – Інтерфейс веб-додатку Jupyter Notebook

Також було використано бібліотеку scikit-learn, адже вона надає інструменти для попередньої обробки, кластеризація, класифікації та зменшення розмірності. Також вона містить метрики та колекції наборів даних, які допомагають порівнювати та тестувати моделі.

3.4 Вибір метрик оцінки ефективності моделей

У даній кваліфакаційній роботі було використано наступні метрики оцінки ефективності моделей, як от:

- точність (accuracy) – частка правильних передбачень серед усіх передбачень;
- F1-міра (F1-score) – гармонійне середнє між точністю передбачень і відгуком;
- точність передбачень (precision) – частка правильних позитивних передбачень серед усіх позитивних передбачень;
- відгук (recall) – частка правильно передбачених позитивних випадків

серед усіх справжніх позитивних випадків;

- ROC-крива (ROC – receiver operating characteristic) – крива, що показує співвідношення між true positive rate (TPR) та false positive rate (FPR);
- AUC (AUC – area under curve) – площа під ROC-кривою, яка узагальнює ефективність моделі по всьому діапазону порогів та має значення від 0 до 1;
- макросередня точність (macro-averaged precision) середня точність для кожного класу, що показує загальну ефективність моделі;
- макросередня повнота (macro-averaged recall) – це середня повнота для кожного класу, що показує загальну здатність моделі правильно ідентифікувати позитивні випадки;
- макросередня F1-міра (macro-averaged F1-score) – середнє гармонійне значення між макроусередненою точністю та макроусередненою повнотою для кожного класу, що показує загальну збалансовану ефективність моделі;
- конфузійна матриця (confusion matrix) – інструмент для візуалізації продуктивності моделі класифікації, дозволяючи детально оцінити, як модель виконує свої передбачення для кожного класу, і виявити, де вона робить помилки.

3.5 Навчання моделей для виявлення аномального трафіку

Навчання моделей відбувалося за наступними кроками:

- спочатку були завантажені необхідні дані з датасету;
- потім дані були очищені від шумів;
- далі дані були розділені на тренувальні та тестові набори;
- потім кожна модель була навчена на тренувальних даних та протестована на тестових даних;
- наприкінці була оцінена ефективність кожної з моделей за допомогою метрик оцінки.

У дослідженні були навчені наступні моделі:

- модель випадкового лісу;
- модель опорних векторів з радіально-базисно функціональним ядром;
- модель дерева рішень.

Покращення точності моделей відбувалася за рахунок використання методу перехресного пошуку у сітці. Цей метод дозволяє навчати моделі з діапазоном різних значень гіперпараметрів, з оцінкою продуктивності моделі за допомогою перехресної перевірки та з вибором набору гіперпараметрів, які є найоптимальнішими.

3.6 Аналіз результатів дослідження

Модель випадкового лісу показала найкращі результати за всіма критеріями, включаючи макросередню точність, макросередній відгук та макросередню F1-міру.

Це робить її найкращим варіантом для даного набору даних з найвищою загальною точністю 99.98%. Найкращі обрані гіперпараметри для моделі випадкового лісу:

- criterion – entropy;
- max_features – 0.5;
- n_estimators – 200.

Модель опорних векторів показала найгірші результати серед трьох моделей, з точністю 97.49% та найнижчими середніми значеннями. Незважаючи на те, що модель опорних векторів часто ефективна для різних задач класифікації, та у цьому конкретному випадку вона відстала від інших моделей. Найкращі обрані гіперпараметри для цієї моделі опорних векторів:

- C – 1000;
- gamma – 0.01.

Модель дерева прийняття рішень демонструє результати, дуже близькі до випадкового лісу, але з невеликим зниженням точності до 99.97%.

Це підтверджує, що для деяких наборів даних простіші моделі можуть

наблизитись до або навіть перевершити складніші ансамблі моделей за певних обставин. Найкращі обрані гіперпараметри для моделі дерева прийняття рішень:

- criterion – gini;
- max_depth – 30;
- min_samples_leaf – 1;
- min_samples_split – 2.

У таблиці 3.1 наведена порівняльна характеристика результатів навчання моделей.

Таблиця 3.1 – Макросередні показники для кожної моделі

	Макросередня точність	Макросередній відгук	Макросередня F1-міра
Модель випадкового лісу	99.98%	99.99%	99.98%
Модель опорних векторів	97.50%	97.48%	97.47%
Модель дерева рішення	99.97%	99.97%	99.97%

На рисунку 3.4. крива точності та відгуку для моделі випадкового лісу.

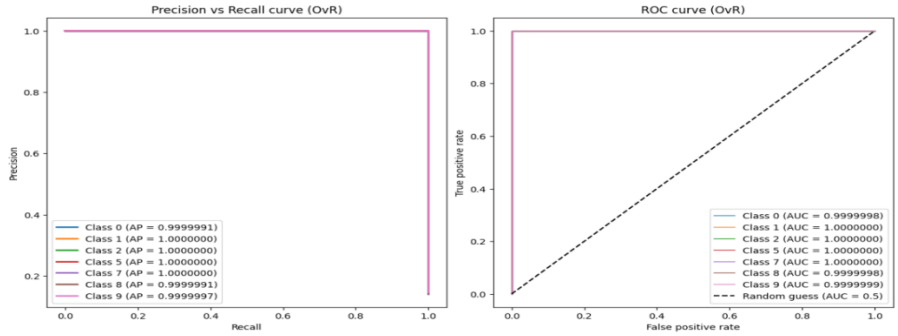

Рисунок 3.4 – Криві точності та відгуку для моделі випадкового лісу

Крива точності та відгуку для моделі випадкового лісу показує співвідношення між точністю і відгуком для різних класів з датасету. Для всіх класів моделі випадкового лісу криві показують високу точність та відгук, що

свідчить про хорошу здатність моделі правильно передбачати позитивні випадки та мінімізувати хибні випадки. Значення середньої точності для всіх класів майже дорівнює 1, що вказує на високий рівень точності і повноти для кожного класу.

ROC-крива для моделі випадкового лісу показує співвідношення між TPR і FPR для кожного класу. Всі криві майже дотичні до верхнього лівого кута графіка, що означає високе значення TPR і низьке значення FPR.

Значення AUC для всіх класів майже дорівнює 1, що вказує на високу здатність моделі правильно розрізняти між позитивними і негативними випадками для кожного класу. Чорна пунктирна лінія показує випадкове вгадування, що порівнюється з результатами моделі.

На рисинку 3.5 представлена крива точності та відгуку для моделі опорних векторів.

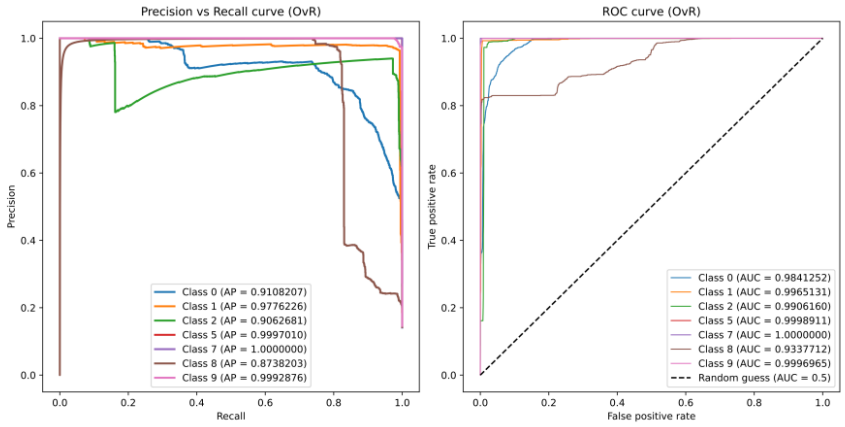

Рисунок 3.5 – Криві точності та відгуку для моделі опорних векторів

На кривій точності та відгуку для моделі опорних векторів клас з номером 0 має середню точність 0.910820 відсоткової частки , а клас 3 має 0.969655 відсоткової частки. Для інших класів, наприклад, класу з номером 5 з середньою точнстю 0.736111 відсоткової частки крива показує гіршу ефективність, що свідчить про наявність хибних передбачень і недостатньо високу точність або відгук.

Більшість кривих для моделі опорних векторів мають значення AUC понад 0.9 відсоткової частки, що вказує на високу здатність моделі опорних векторів правильно розрізняти між позитивними і негативними випадками для цих класів. Наприклад, клас з номером 3 має 0.998139 відсоткової частки, що є дуже хорошим результатом. Однак, є класи з нижчими значеннями AUC, наприклад, клас з номером 5 має 0.849547 відсоткової частки, що вказує на менш ефективну здатність моделі розрізняти між позитивними і негативними випадками для цього класу.

На основі графіків можна зробити висновок, що модель опорних векторів працює добре для більшості класів, проте для деяких класів є місця для покращення. ROC-крива підтверджує, що хоча модель має високу ефективність для більшості класів, деякі класи все ж мають нижчу продуктивність.

На рисинку 3.6 представлена крива точності та відгуку для моделі опорних векторів.

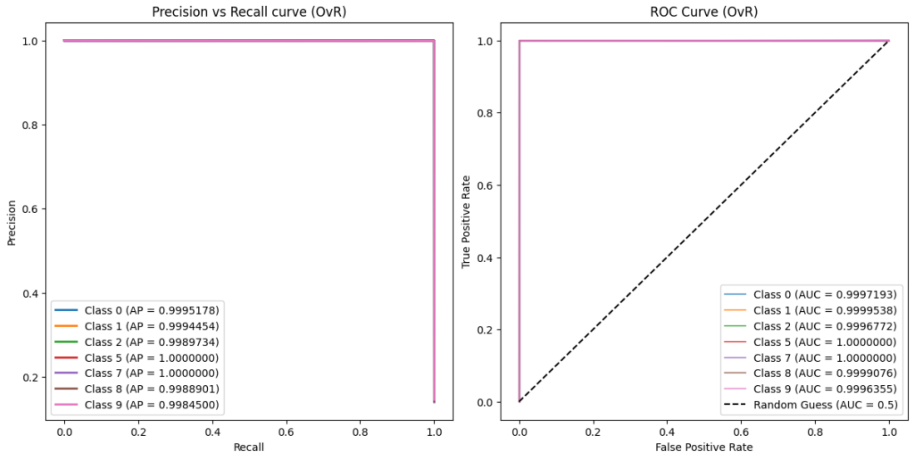

Рисунок 3.6 – Криві точності та відгуку для моделі дерева рішення

Значення середньої точності для кожного класу моделі дерева рішення є дуже високі та варіюються від 0.999178 до 0.999520 відсоткової частки. Криві ROC майже вертикальні, що вказує на те, що модель добре розрізняє класи з мінімальним компромісом між точністю та відгуком.

Значення AUC для кожного класу також дуже високі, всі приблизно 0.999 відсоткової частки, що свідчить про відмінну продуктивність. Криві ROC майже всі дорівнюють 1, залишаються близько до верхнього лівого кута, що вказує на дуже високу чутливість моделі при підтриманні дуже низького рівня хибних результатів.

Загалом, модель дерева рішень показує чудову продуктивність як за метриками точності та відгуку, так і за метриками крової ROC.

Високі значення середньої точності та AUC для всіх класів вказують на те, що модель дуже ефективно класифікує різні класи з мінімальними помилками.

На рисунку 3.7 наведені конфузійні матриці для моделі випадкового лісу, моделі опорних векторів та моделі дерева рішень.

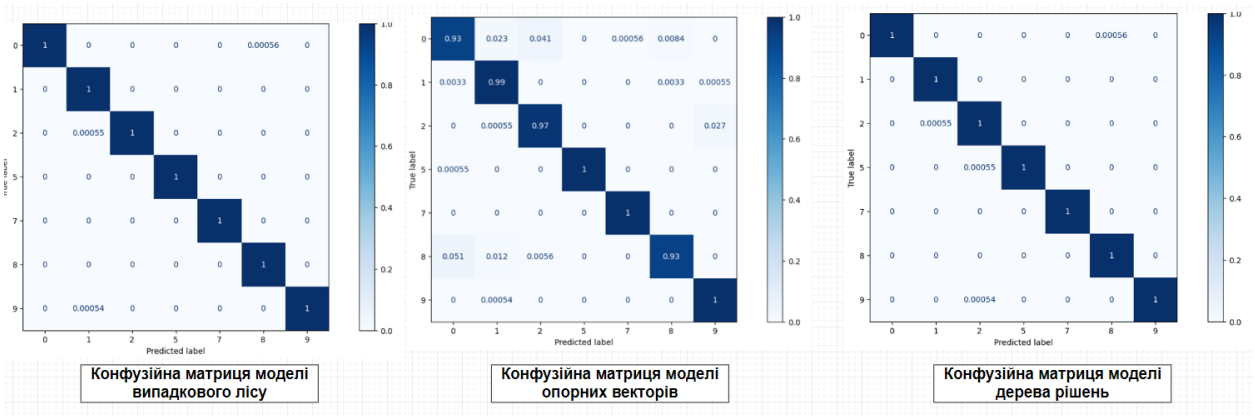

Рисунок 3.7 – Конфузійні матриці для моделей

Конфузійна матриця моделі випадкового лісу показує майже ідеальні передбачення з дуже низькими значеннями помилок. Значення на діагоналі дорівнюють 1, що означає 100% правильних передбачень для кожного класу, за винятком дуже незначної помилки для класу 0, що дорівнює 0.00056 відсоткової частки.

Конфузійна матриця моделі опорних векторів має трохи більше помилок у порівнянні з випадковим лісом і деревом рішень. Діагональні значення трохи

нижчі за 1 для класів 0, 2, 3, 5 та 7, що вказує на кілька неправильних передбачень. Наприклад, для класу 0 точність становить 0.93 відсоткової частки, а для класу 7 – 0.93 відсоткової частки. Є також помилки, такі як 0.027 відсоткової частки для класу 2.

Конфузійна матриця моделі дерева рішень показує дуже високі результати, подібно до моделі випадкового лісу. Значення на діагоналі дорівнюють 1, що свідчить про 100% правильних передбачень для кожного класу, за винятком дуже незначної помилки для класу 0, що дорівнює 0.00056 відсоткової частки.

Загалом, конфузійні матриці демонструють, що моделі випадкового лісу та дерева рішень мають дуже високу точність, тоді як модель опорних векторів показує трохи нижчі результати з деякими помилками у передбаченнях для кількох класів.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи були досліджені методи виявлення аномального трафіку IoT, такі як: метод випадкового лісу, метод опорних векторів та метод дерева прийняття рішень. У результаті проведеного дослідження було здійснено підбір оптимальних гіперпараметрів для реалізованих моделей для їх кращої продуктивність при виявленні аномального трафіку в IoT.

Для вибору оптимальних гіперпараметрів був використаний метод перехресної валідації, що дозволяє підібрати різні комбінації гіперпараметрів та вибирати ті, які забезпечують найкращу продуктивність моделі.

Для методу випадкового лісу було підібрано оптимальну кількість дерев та їх максимальну глибину, що дозволило збільшити точність виявлення аномалій за рахунок зменшення ймовірності перенавчання.

Для методу опорних векторів були підібрані значення параметра регуляризації *C* та тип ядра з урахуванням необхідності балансування між похибками першого та другого роду.

Для методу дерева прийняття рішень були підібрані глибина дерева та мінімальна кількість зразків для розщеплення з метою забезпечення стабільності моделі при обробці великої кількості даних IoT.

Метод випадкового лісу показав найкращі результати з точністю 99.98%. Цей метод забезпечує високу стійкість до перенавчання та може ефективно працювати з даними із шумами.

Метод опорних векторів показав нижчі результати порівняно з іншими моделями, з точністю 97.49%. Він був менш ефективний через складність налаштування гіперпараметрів та високу обчислювальну складність.

Метод дерева прийняття рішень продемонстрував хороші результати з точністю 99.97%, що підтверджує його ефективність для деяких наборів даних та можливість наблизитися до точності більш складних моделей.

Результати дослідження показують, що метод випадкового лісу є найбільш перспективним методом для виявлення аномального трафіку в IoT.

По-перше, метод випадкового лісу показав найвищу точність серед досліджених методів – 99.98%. Це зумовлено використанням ансамблевого навчання, яке об'єднує прогноз кількох дерев рішень, що допомагає зменшити варіацію та підвищити точність моделі

По-друге, метод випадкового лісу демонструє високу стійкість до перенавчання. Це досягається за рахунок використання великої кількості дерев рішень, кожне з яких навчено на різних підмножинах даних. Такий підхід дозволяє моделі краще узагальнювати на нових даних та уникати проблеми перенавчання, яка часто виникає при використанні одиничних моделей машинного навчання.

По-третє, метод випадкового лісу має високу гнучкість у роботі з різними типами даних та ознак. Це дозволяє ефективно використовувати його для аналізу складних та багатовимірних даних IoT, що містять як числові, так і категорійні ознаки. Завдяки цьому, модель може враховувати різноманітні аспекти мережевого трафіку та виявляти аномалії, що виникають у різних контекстах.

Використання машинного навчання для виявлення аномалій дозволяє автоматизувати процес аналізу даних та підвищити ефективність виявлення аномальних патернів у реальному часі.

Однією з головних проблем є потреба у великих обсягах помічених та оброблених даних, що може бути дорогим та часозатратним процесом. Існують також складності з адаптацією моделей до нових умов та типів даних, що може вимагати додаткових ресурсів для навчання моделей.

Рекомендації для подальших досліджень включають покращення методів та моделей виявлення аномалій на рівні пристроїв IoT для підвищення їхньої ефективності та зниження кількості хибнопозитивних результатів, розробку більш адаптивних моделей, які можуть ефективно працювати з різноманітними типами даних та умовами в середовищі IoT, а також пошук

ефективних способів обробки енергетичних та обчислювальних обмежень, що є важливими для забезпечення стабільної роботи пристроїв IoT.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Altangerel, G., Tejfel, M., & Tsogbaatar, E. (2023). IoT Anomaly Detection with 1D CNN Using P4 Capabilities. CTA Electrotechnica et Informatica, Vol. 23, No. 2, pp. 3–12. DOI: 10.2478/aei-2023-0006.

2. Bhatti, I. K., Belsare, P., Watpade, V., Bhargave, A., & Gharate, D. (2022). Sensors in IoT. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 2(2). ISSN (Online) 2581-9429. DOI: 10.48175/IJARSCT-2776.

3. Chatterjee, A., & Ahmed, B. S. (2022). IoT Anomaly Detection Methods and Applications: A Survey. Internet of Things, 100568. Elsevier BV.

4. Cook, A. A., Mısırlı, G., & Fan, Z. (2020). Anomaly Detection for IoT Time-Series Data: A Survey. IEEE Internet of Things Journal, 7(7), 6481–6494.

5. Fahim, M., & Sillitti, A. (2019). Anomaly detection, analysis and prediction techniques in IoT environment: A systematic literature review, IEEE Access 7, 81664–81681.

6. Hasan, M., Islam, M. M., Islam Zarif, M. I., & Hashem, M. M. A. (2019). Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet of Things, 7, 100059. DOI: 10.1016/j.iot.2019.100059.

7. Kovalenko, A., Kuchuk, H., Kuchuk, N., & Kostolny, J. (2021). Horizontal scaling method for a hyperconverged network. 2021 International Conference on Information and Digital Technologies (IDT), Zilina, Slovakia. DOI: https://doi.org/10.1109/IDT52577.2021.9497534.

8. Li, H., & Boulanger, P. (2020). A Survey of Heart Anomaly Detection Using Ambulatory Electrocardiogram (ECG). Sensors, 20(5), 1461. DOI: https://doi.org/10.3390/s20051461.

9. Parimala, V. K. (Ed.). (2024). Anomaly Detection - Recent Advances, AI and ML Perspectives and Applications. IntechOpen. DOI: https://doi.org/10.5772/intechopen.110988. ISBN: 978-1-83769-027-5.

10. Ramesh, P., Sri Venkat Rami Reddy, M., & Bhaskara Reddy, P. (2021). Architecture, Protocols, Layers and Elements of IoT. International Journal of Creative Research Thoughts (IJCRT), 9(9). ISSN: 2320-2882.

11. Rokach, L., & Maimon, O. (2005). Decision Trees. In The Data Mining and Knowledge Discovery Handbook (pp. Chapter 9). DOI: 10.1007/0-387-25465-X_9.

12. Ruban, I. V., Martovytskyi, V. O., Kovalenko, A. A., & Lukova-Chuiko, N. V. (2019). Identification in Informative Systems on the Basis of Users' Behaviour. Proceedings of the International Conference on Advanced Optoelectronics and Lasers, CAOL2019-September, 9019446, 574-577. DOI: https://doi.org/10.1109/CAOL46282.2019.9019446.

13. Коваленко А. А., Кучук Г. А. (2018). Методи синтезу інформаційної та технічної структури системи управління об’єктом критичного застосування. Сучасні інформаційні системи, 2(1), 22–27. DOI: https://doi.org/10.20998/2522-9052.2018.1.04.

14. Кучук, Н., Коваленко, А., Рубан, І., Шишацький, А., Заковоротний, О., Шевяков, І. (2023). Traffic Modeling for the Industrial Internet of NanoThings. 2023 IEEE 4th KhPI Week on Advanced Technology, KhPI Week 2023 - Conference Proceedings, 194480. DOI: http://dx.doi.org/10.1109/KhPIWeek61412.2023.1031285.

15. Марченко, Р., Коваленко, А., Знайдюк, В. (2024). Аналіз методів виявлення аномального трафіку в мережах IoT. Системи управління, навігації та зв’язку. Збірник наукових праць, 1(75), 133-136. DOI: https://doi.org/10.26906/SUNZ.2024.1.133.