Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: "Виявлення шкідливих програм IoT"

Виконав: студент
Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)
Гнатківський Любомир Васильович
підпис (прізвище та ініціали)

Керівник Стадник М. А.
підпис (прізвище та ініціали)

Нормоконтроль Тимощук Д. І.
підпис (прізвище та ініціали)

Завідувач кафедри Загородна Н.В.
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Гнатківському Любомиру Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Виявлення шкідливих програм IoT

Керівник роботи Стадник Марія Андріївна, к.т.н., доцент кафедри КБ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 15 » 04 2024 року № 4/7-350

2. Термін подання студентом завершеної роботи 21.06.2024

3. Вихідні дані до роботи Набір промаркованих даних IoT-23

4. Зміст роботи (перелік питань, які потрібно розробити)
Проаналізувати принцип роботи IoT системи. Проаналізувати атаки на IoT систему та можливі шляхи їх уникнення. Проаналізувати приклади застосування AI для захисту IoT системи. Розробити алгоритм та реалізувати методи AI для виявлення шкідливих програм методом аналізу мережевого трафіку IoT. Безпека життєдіяльності, основи охорони праці. Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1.Тема. 2. Мета. 3. Завдання. 4. Кількість підключень (у мільярдах) IoT у всьому світі з 2022 по 2023 рік, з прогнозами з 2024 по 2033 рік (позначено зірочкою). 5. Узагальнена архітектура IoT. 6. Результати порівняння протоколів, що використовуються в IoT. 7. Результати порівняння технологій IEEE 802.15.4 та Wi-Fi. 8. Можливі атаки на IoT систему. 9. Елементи дослідної системи IoT-23. 10. Типи атака у наборі IoT-23. 11. 23 характеристики даних у наборі IoT-23. 12. Послідовні етапи виявлення шкідливих програм в IoT системі. 13. Перших кілька значень даних у IoT-23. 14. Розподіл пустих значень одного сценарію в наборі даних IoT-23. 15. Кореляційна матриця характеристик в одному з наборів даних. 16. Результати попередньої обробки даних. 17. Кількість екземплярів кожної атаки (шкідливої програми) в наборі даних IoT-23. 18. Формування тренувального та тестового наборів даних. 19. Застосування моделі XGBClassifier та пошук її оптимальних параметрів. 20. Оптимальні параметри моделей XGBClassifier, GaussianNB, SVC. 21. Результати оцінки якості моделей-класифікаторів. 22. Висновки. 23. Дякую за увагу.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Мариненко С. Ю., к.т.н, доцент кафедри МТ		

7. Дата видачі завдання 29.01.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	29.01 – 01.02	Виконано
2.	Підбір джерел для аналізу IoT атака та відповідних шкідливих програм	02.02 – 09.02	Виконано
3.	Опрацювання джерел в галузі дослідження	12.02 – 16.02	Виконано
4.	Провести аналіз методів захисту від атак	17.02 – 11.03	Виконано
5.	Здійснення аналізу набору даних IoT-23	12.03-26.04	Виконано
6.	Пошук оптимальних параметрів класифікаторів для ідентифікації таки спричиненою шкідливою програмою в системі IoT	29.04– 10.05	Виконано
7.	Оформлення розділу «Сучасний стан інтернету речей»	10.05 – 15.05	Виконано
8.	Оформлення розділу «Типи атак на IoT систему та відповідні методи захисту»	16.05 – 23.05	Виконано
9.	Оформлення розділу «Виявлення шкідливих програм в мережі iot з використанням штучного інтелекту»	23.05-30.05	
10.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	31.05 – 04.06	Виконано
11.	Оформлення кваліфікаційної роботи	10.05 – 17.06	Виконано
12.	Нормоконтроль	17.06 – 18.06	Виконано
13.	Перевірка на плагіат	18.06-19.06	Виконано
14.	Попередній захист кваліфікаційної роботи	18.06 – 21.06	Виконано
15.	Захист кваліфікаційної роботи	25.06.2023	

Студент _____ Гнатківський Л. В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Стадник М. А.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Виявлення шкідливих програм IoT // Кваліфікаційна робота ОР «Бакалавр» // Гнатківський Любомир Васильович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБ-41 // Тернопіль, 2024 // С. 64, рис. – 12, табл. – 8 , кресл. – 23, додат. – 0.

КЛЮЧОВІ СЛОВА: IoT, XGBoost, SVM, AI, шкідлива програма, атака, мережевий трафік.

Кваліфікаційна робота присвячена дослідженню методів виявлення шкідливих програм в IoT системі. Шкідливі програми і відповідні атаки на всю систему IoT можуть завдати колосальних збитків навіть для систем персонального користування. У роботі розглянуто сучасний стан розвиту IoT, узагальнена трьох рівнева архітектура IoT, можливі атаки на кожному з рівнів та ймовірні вразливості. У роботі представлено роль штучного інтелекту для виявлення загроз щодо системи IoT.

Для виявлення шкідливих програм, що були інстальовані на пристрої системи IoT, і спричиняли аномалії у мережевому трафіку системи було використано алгоритми машинного навчання: XGBClassifier, SVC, GaussianNB. На основі порівняльного аналізу результатів класифікації найкращу якість класифікації (виявлення шкідливої програм чи атак) продемонстрував класифікатор XGBoost. Розроблений алгоритм машинного навчання може бути використаний в системах моніторингу мережевого трафіку критичних IoT систем.

Результати кваліфікаційної роботи можуть бути використані для лабораторних робіт в процесі навчання студентів, що проходять курс "Методи та системи штучного інтелекту".

ABSTRACT

IoT malware detection // Thesis of educational level "Bachelor"// Hnatkivskyi Lyubomyr Vasyliovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, group СБс-41 // Ternopil, 2024 // P. 64, fig. - 12, tab. - 8, draw. - 23, addition. – 0.

Keywords: IoT, XGBoost, SVM, AI, malware, attack, network traffic.

The qualification work is devoted to the research of methods of detecting malware in the IoT system. Malware and corresponding attacks on the entire IoT system can cause colossal damage even to personal use systems. The current state of IoT development, a generalized three-level IoT architecture, possible attacks at each of the levels, and probable vulnerabilities are considered in the thesis. The thesis presents the role of artificial intelligence in detecting threats to the IoT system.

Machine learning algorithms were used to detect malware that were installed on IoT system devices and caused anomalies in the network traffic of the system: XGBClassifier, SVC, GaussianNB. Based on the comparative analysis of the classification results, the best classification quality (malware or attack detection) was demonstrated by the XGBoost classifier. The developed machine learning algorithm can be used in network traffic monitoring systems of critical IoT systems.

The results of the qualification work can be used for laboratory work during the training of students taking the course “Methods and systems of artificial intelligence”.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AI	—	Artificial Intelligence
ANN	—	Artificial Neural Network
CoAP	—	Constrained Application Protocol
DoS	—	Denial of Service
FN	—	False Negative
FP	—	False Positive
IoT	—	Internet of Things
ML	—	Machine Learning
MQTT	—	MQ Telemetry Transport
NB	—	Naive Bayes
NFC	—	Near Field Communication
SSL	—	Secure Socket Layer
SVM	—	Support Vector Machine
TLS	—	Transport Layer Security
TN	—	True Negative
TP	—	True Positive
XMPP	—	Extensible Messaging and Presence Protocol

ВСТУП

Практично кожна людина зараз стикається із інтернетом речей (IoT): смарт годинники не лише показують відповідний час, але нагадують про важливу кількість кроків, яку потрібно пройти, про кількість сну, про рівень тиску чи серцебиття. Така своєрідна міні система для вимірювання життєво важливих показників. Надзвичайно зручно для кожної людини, саме тому кількість пристроїв з кожним роком зростає, і не лише для повсякденних задач людини, але для досягнення цілей критично важливої інфраструктури. Для прикладу це може бути система вимірювання тиску, датчики якої є розміщеними вздовж труби газопостачання. Невірна інформація з датчика може бути причиною того, що оператор мережі зменшить чи збільшить тиск і таким чином спровокує певну проблему у системі.

Специфіка систем IoT полягає в тому, що вони використовуються як у повсякденному житті людини, так і у важливих критичних сферах. Кількість IoT пристроїв, що підключаються у мережу зростає з кожним роком, тому завдання захисту системи IoT в цілому є актуальним. Про це завдання клопочуться не лише виробники пристроїв, виробники програмним додатків, але і кінцеві користувачі.

IoT система складається з рівня мережі речей, хмарних обчислень, програм та додатків. На кожному з цих рівнів виникають загрози і відповідно на кожному з них необхідні відповідні системи кіберзахисту.

Метою цієї роботи є виявлення шкідливих програм в системі IoT за допомогою аналізу мережевого трафіку між пристроями з використанням інструментів штучного інтелекту.

Основним завданням є розробка алгоритму ідентифікації шкідливих програм з використанням моделей класифікаторів машинного навчання та знаходження їх оптимальний параметрів для досягнення найвищих показників якості багатокласової класифікації.

Додатково в роботі буде проаналізовано сучасний стан IoT систем, її узагальнену архітектуру, типові атаки до відповідного рівня системи, можливі вразливості та шляхи їх усунення.

1 СУЧАСНИЙ СТАН ІНТЕРНЕТУ РЕЧЕЙ

1.1 Поняття інтернету речей (IoT)

Суспільство розвивається та комунікує, взаємодіючи з фізичними об’єктами та пристроями, що знаходяться навколо нас. Існує величезна кількість об’єктів (речей), якими ми щодня користуємося, наприклад, датчики різноманітних парметрів, кондиціонери, тостери, холодильники, смартфони, смарт-годинники, водопровідні крани, вентилятори, кондиціонери. Фундаментальна концепція інтернету речей (Internet of Things) полягає в тому, щоб об’єднати такого типу об’єкти (речі) за допомогою віртуального чи реального зв’язку та дозволити цим пристроям спілкуватися один з одним, щоб вони працювали розумно в загальному. Наприклад, термостат і кондиціонер – це два незалежні пристрої. Проте IoT дозволяє їм “спілкуватися” один з одним таким чином, що поточна температура в кімнаті чи середовищі, визначена термостатом, буде управлятись кондиціонером. Такий підхід забезпечить ефективне використання пристрою. Концепцію IoT можна коротко описати як повсюдну присутність різноманітних “речей” або “об’єктів”, такі як мітки радіочастотної ідентифікації (RFID), датчики, приводи, мобільні телефони, які за допомогою унікальних схем адресації здатні взаємодіяти один з одним і співпрацювати з сусідніми “розумними” компонентами для досягнення спільні цілі та результату [1].

Термін “Інтернет речей” (IoT) вперше ввів підприємець Кевін Ештон у 1999 році під час своїх досліджень у Auto-ID Lab [2]. За останні кілька років ця парадигма IoT стрімко розвивається і значно привертає увагу галузей у сфері бездротової телекомунікації.

Існує багато застосувань IoT, але в основному має справу з речами, присутніми в нашому навколишньому середовищі. Необхідно зазначити, що концептуально IoT можна застосувати в будь-якій сфері реального життя. Це свідчить про те, що у майбутньому IoT буде включено майже в усі сфери. На

рисунку 1.1 представлено загальні області застосування додатків і послуги, які IoT може надавати для цих відповідних доменів [1].

Рисунок 1.1 – Області застосування IoT

Значна кількість областей застосування IoT є причиною для популяризації інтернету речей та збільшенні кількості підключений пристроїв. Навіть вже сьогодні можна дистанційно запустити робот-порохотяг, нагріти воду у бойлері. Для цього лише достатньо мати підключення до мережі інтернет та встановлене відповідне програмне забезпечення. Згідно з даними Statista [3] у 2030 році очікується 32,1 мільярди пристроїв, які тим чи іншим чином будуть складовими відповідних систем IoT (див. рисунок 1.2).

Як стверджують співавтори з Atzori [4], IoT можна класифікувати за трьома підпарадигмами:

- інтернет-орієнтована підпарадигма (проміжне програмне забезпечення);
- орієнтована на речі підпарадигма (сенсори);
- семантична підпарадигма (знання).

Такий тип категоризації по підпарадигмах необхідний через міждисциплінарну природу самого IoT як такого. Необхідно зауважити, що ефективність і корисність IoT не пов’язані лише в областях застосування, де три

підпарадигми перетинаються одна з одною. Для прикладу, проміжне програмне забезпечення – це рівень програмного забезпечення, який діє як інтерфейс між програмою та операційною системою. В епоху IoT проміжне програмне забезпечення становить різні стеки мережевих протоколів разом із основними функціями пристроїв.

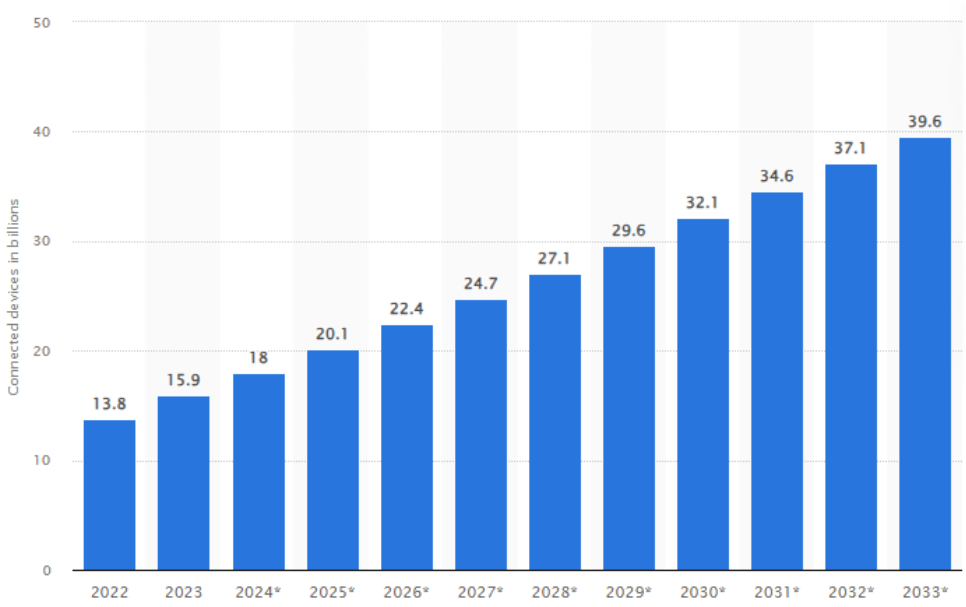

Рисунок 1.2 – Кількість підключень (у мільярдах) до Інтернету речей (IoT) у всьому світі з 2022 по 2023 рік, з прогнозами з 2024 по 2033 рік (позначено зірочкою)

Наступна орієнтована на речі підпарадигма IoT – це сенсорні вузли, по суті це електромеханічні пристрої, які відображають процеси навколишнього середовища. Згодом ці вузли з'єднуються один з одним за допомогою ефективних мережевих методів.

Третьою підпарадигмою IoT є знання. Дані, що генеруються датчиком вузла (сенсором) повинні бути перетворені в інформацію, і з цієї інформації отримуються знання. Згодом ці знання використовуються для вирішення певних проблем та для досягнення цілей.

Завдяки сучасному дослідженню IoT авторами статті [5] було виявлено, що впровадження IoT у громадських місцях може заощадити велику кількість фізичних ресурсів, таких як електроенергія та вода. Таким чином можливо

провести модернізацію комунальних підприємств, запровадивши навіть найпростішу структуру IoT. Для прикладу міські комунальні підприємства, запровадивши IoT у будь яку свою сферу управління змогли би оптимізувати використання організаційних та фізичних ресурсів, таких як електроенергія, вода газ.

1.2 Архітектура IoT

Однією з перешкод при дослідженні архітектури IoT є те, що це достатньо широке поняття, що для нього досі не існує стандартної уніфікованої еталонної архітектури. Система IoT складається з різноманітних різнорідних датчиків, мереж, комунікаційних методологій і технологій обробки, але інтеграція цих різних типів технологій в одну злагоджену систему формує проблему взаємодії елементів. З метою вирішення проблеми сумісності необхідна стандартизована архітектура IoT. Автори статті [6] провели детальний аналіз та представили узагальнену IoT архітектуру, що зображено на рисунку 1.3.

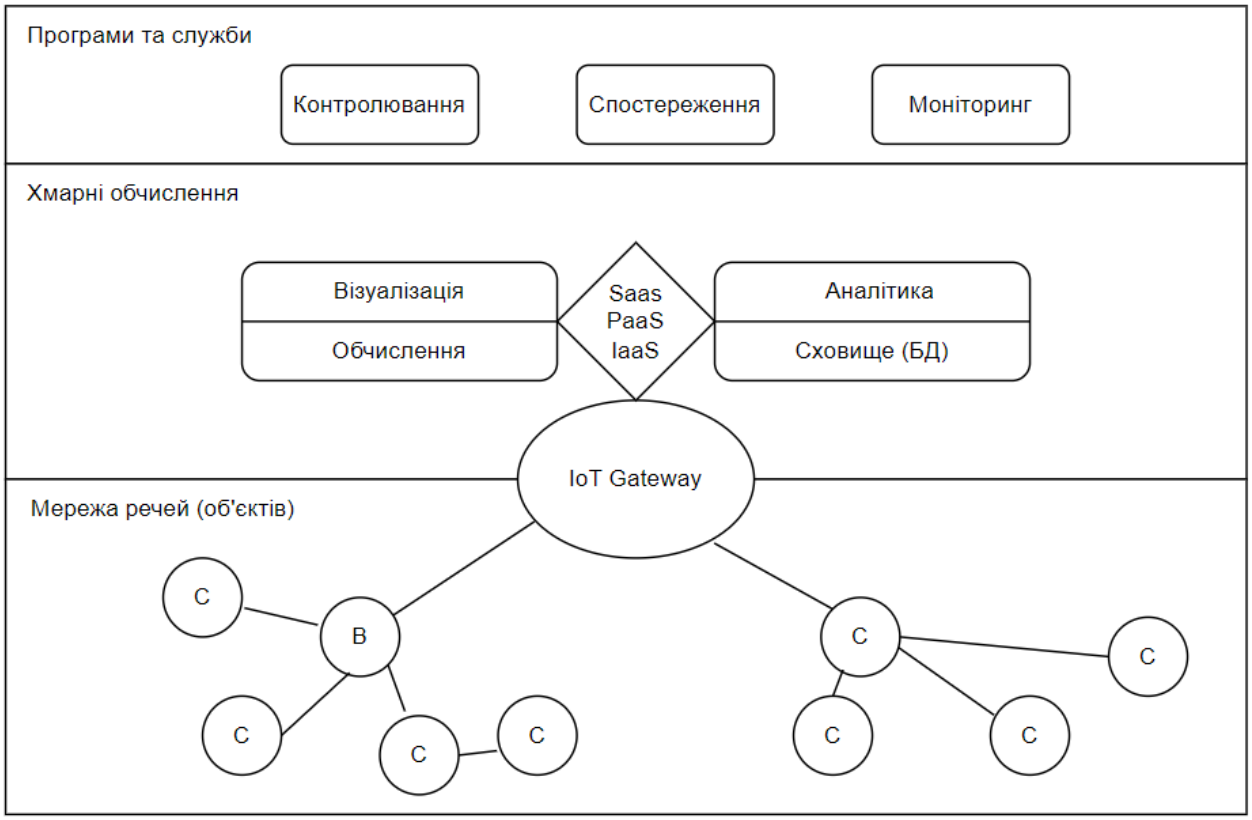

Рисунок 1. 3 – Узагальнена архітектура IoT

Таким чином, враховуючи що стандартної єдиної архітектури IoT не існує, проте архітектуру IoT можна розглядати як багатошарова система в загальному. На базовому рівні є різні типи вбудованих сенсорних вузлів і пристроїв або речей. Ці сенсорні вузли та пристрої з'єднані один з одним за допомогою дротової або бездротової мережі, такої як Wi-Fi, Bluetooth, NFC. IoT використовує магістраль мережі IP. Відповідно локальна мережа речей на базовому рівні може підключатися до глобального Інтернету через спеціалізований пристрій IoT, відомий як IoT Gateway (IoT шлюз).

Над базовим рівнем є рівень хмарної обробки. IoT Gateway діє як інтерфейс між рівнем хмарної обробки та локальною мережею речей. Концепція використання хмарних обчислень в IoT полягає в тому, що існує велика кількість вбудованих сенсорних пристроїв, локально пов'язаних між собою, що призводить до генерації величезної кількості даних у реальному часі. Щоб аналізувати та керувати такими даними, доцільним та оптимальним варіантом є хмарна платформа.

На вищому рівні та над рівнем хмарної обробки знаходиться рівень програм і служб. Цей рівень містить програми та служби, які управляють базовим рівнем на основі отриманої інформації із рівня хмарної обробки. Іноді програма може бути надзвичайно простою, як просто вимкнути електричний вимикач, а іноді може бути дуже складною, як керування критичною інфраструктурою.

1.3 Компоненти IoT системи.

На основі узагальненої архітектури, що представлена у попередньому підрозділі, представимо основні типові компоненти IoT.

Пристрої IoT (речі).

Речі в IoT можна визначити як будь-який вбудований пристрій, який може надсилати або отримувати дані через мережу та містить мікроконтролер або може бути процесором для певної програми. Оскільки ціна на 8- або 32-

розрядний блок мікроконтролера (MCU) є не захмарною, то його можна вибрати для використання в пристроях IoT. Проте існують обмеження MCU, що полягають в обмеженості ресурсів. Будь-який пристрій IoT повинен мати стек програмного забезпечення, який може містити програмний модуль, визначений для збору даних, керування входом, обробкою даних і стека протоколів дротового чи бездротового зв'язку. Для MCU це дуже складне завдання опрацьовувати величезні стеки програмного забезпечення IoT. Тому, багатофункціональний мікропроцесор для пристроїв IoT може бути кращим вибором, ніж MCU. Але знову ж таки це створює проблему економічності та управління енергією. Щоб вирішити цю ситуацію, найкращим вибором є використання як дешевого MCU, так і 8-розрядного процесора з низькою потужністю. Цей тип системи обробки називають гетерогенною системою обробки. Програмні модулі невеликого розміру, такі як збір даних і керування приводом, можуть бути вбудовані в MCU, а решта великих програмних модулів може оброблятися за допомогою мікропроцесора.

Операційна система для пристроїв IoT.

Оскільки пристрої IoT мають дуже обмежені ресурси, наприклад вбудована флеш-пам'ять, пам'ять тощо. Крім того, ці пристрої працюють від батарейок. Ці обмеження та особливості пристроїв IoT вимагають ефективної, надійної, портативної, гнучкої та легкої системи з дуже низьким відбитком RAM та ROM. Вбудовані Linux, Windows і т. д. багато хто з них беруть участь у розробці ОС IoT [7]. Операційну систему для пристроїв IoT можна охарактеризувати за такими параметрами:

- Реальний час. Оскільки пристрої IoT вбудовані в реальний світ і дані генеруються пристроями IoT у режимі реального часу, тому дані мають оброблятися в режимі реального часу. Щоб мати таку функціональність, операційна система для пристроїв IoT повинна бути операційною системою реального часу (RTOS).

- Модульність. Пристрій IoT потребуватиме модульної операційної системи, яка відокремлює основне ядро від проміжного ПЗ, протоколів і додатків [8]. Так як існує велика кількість категорій процесорів і мікроконтролерів, що

мають різні обсяги пам’яті, тому різна область застосування IoT вимагає іншого програмного пакета. Тому бажано, щоб операційна система IoT була модульною, щоб розробник IoT міг вибрати основні модулі відповідно до вимог оперативної пам’яті.

- Портативність. Операційна система IoT має бути легкою для переносу на різні апаратні платформи. Вона повинен підтримувати різноманітні класи пристроїв IoT. Мікроконтролери, які використовуються в Інтернеті речей, варіюються від 8 до 32 біт. ОС повинна мати можливість використовувати базову архітектуру. Крім того, IoT – це майбутнє в різноманітних сферах із широким спектром застосувань. ОС має бути адаптованою до конкретних потреб додатків і забезпечувати розумну абстракцію деталей.

- Мережа. Підтримка мережі є основною вимогою IoT. Концептуально IoT дозволяє пристроям спілкуватися один з одним за допомогою IP-мережі. Щоб задовольнити це обмеження, операційна система для пристроїв IoT повинна підтримувати широкий спектр стандартів зв’язку, наприклад стандарти IP і не IP, такі як Wi-Fi, Bluetooth і IEEE 802.15.4. Операційна система IoT має також підтримувати протоколи з ефективним використанням пропускної здатності, такі як 6LoWPAN. ОС IoT повинна бути гнучкою, щоб модернізувати існуючі пристрої новими параметрами підключення без переробки ядра вбудованого програмного забезпечення.

- Надійність. Багато систем IoT можна розгорнути в життєво важливих середовищах, таких як авіація, атомні електростанції, газові розподільчі вузли. У такому випадку пристрої IoT повинні бути бездоганно надійними. Щоб продемонструвати надійність пристроїв IoT, операційна система повинна бути сертифікована відповідним органом. Дотепер існують різні вбудовані операційні системи, такі як Contiki, RIOT, LiteOS, FreeRTOS, TinyOS і Embedded Linux. Ці операційні системи задовольняють одну або декілька бажаних характеристик операційної системи для IoT. Жодна операційна система на сьогоднішній день не сертифікована як операційна система IoT. Це відкриває двері для дослідників для роботи в області IoT ОС.

Стек протоколів локальної мережі та IoT.

Мережа є ключовим поняттям в архітектурі IoT. Пристрої IoT з'єднані між собою за допомогою технології зв'язку. Вибір цієї комунікаційної технології безпосередньо впливає на вартість проекту та вимоги до обладнання. Існує величезна область застосування IoT. IoT можна розгорнути вдома, будівлях, організаціях, фабриках, містах тощо. Нехай розглянемо завод як типовий випадок для системи IoT. Заводу знадобиться велика кількість підключених датчиків і приводів, розкиданих на великій території, і найкраще підійде бездротова технологія. Отже, бездротова сенсорна мережа (WSN) узагальнить технологію локальної мережі в IoT. Бездротова сенсорна мережа (WSN) складається з недорогих сенсорних вузлів, розгорнутих у густонаселених і віддалених районах для моніторингу навколишнього фізичного середовища [9].

Технологія WSN. Об'єкти в IoT можуть бути взаємопов'язані за допомогою технології WSN. Для формування WSN існують різні сполучні пристрої, які включають Wi-Fi та IEEE 802.15.4 технології.

Найпростішою мережевою технологією для пристрою IoT є Wi-Fi. Завдяки своїй повсюдній природі Wi-Fi може бути хорошим рішенням для багатьох доменів додатків IoT. Проте потреба в електроенергії Wi-Fi дуже висока. Є кілька пристроїв IoT, які не можуть дозволити собі такий рівень енергоспоживання: зазвичай такі пристрої IoT працюють від батареї, тому живити такі пристрої Wi-Fi стає недоцільно. У такому випадку система IoT потребує рішення з низьким енергоспоживанням для такого модуля приймача.

IEEE 802.15.4 є енергоефективним рівнем MAC для IoT. IEEE 802.15.4 є одним із радіостандартів малої потужності для пристроїв, що забезпечують доступ до інтернету речей. Дана технологія був реалізованою у 2003 році. Комерційні радіоприймачі з цим стандартом підтримують основу для систем малої потужності. Цей стандарт IEEE було розширено та вдосконалено у 2006 та 2011 роках за допомогою поправок 15.4e та 15.4g. Споживана потужність комерційних радіо частотних пристроїв скорочено вдвічі порівняно з кількома роками тому, і згідно тенденцією відбудеться ще на 50% зменшення споживної потужності у порівнянні з наступним поколінням пристроїв. У таблиці 1.1 представлено порівняння технологій IEEE 802.15.4 та Wi-Fi.

Таблиця 1.1 - Результати порівняння технологій IEEE 802.15.4 та Wi-Fi

Стандарт	IEEE 802.15.4	Wi-Fi
Частота	868/915 МГц, 2.4 ГГц	2.4, 5.8 ГГц
Швидкість передачі даних	250 Кб/с	11 до 105 Мб/с
Діапазон	10 до 300 м	10 до 100 м
Потужність	Дуже низька	Висока
Батарея	Лужна (від місяців до років)	Перезаряджання (години)

Крім цих двох радіотехнологій, існують також інші технології, як-от Bluetooth, Zigbee тощо. Але це не IP-технології, що і визначає їх не популярність у використання в інфраструктурі IoT.

Стек протоколів IP для IoT зазвичай містить:

- HTTP. Для клієнтсько-серверної моделі мережі через Інтернет основою є HTTP. В IoT безпека є головною проблемою. Щоб інтегрувати HTTP в IoT, зберігаючи проблеми з безпекою, реалізація HTTP може бути виконана так, щоб включити лише клієнта в наш пристрій IoT, а не сервер.

- WebSocket. Іншим простим протоколом, який зараз є в стеку протоколів IoT, є WebSocket. Він забезпечує повнодуплексний зв’язок через одне з’єднання TCP між клієнтом і сервером. Стандарт WebSocket спрощує більшу частину складності, пов’язаної з двонаправленим веб-зв’язком і керуванням з’єднаннями [14].

- XMPP (Extensible Messaging and Presence Protocol). В першу чергу він був розроблений для відкритого та контрольованого спілкування. XMPP бере свій початок у обміні миттєвими повідомленнями та інформації про присутність. Він розширився до сигналізації для VoIP, співпраці, полегшеного проміжного програмного забезпечення, синдикації вмісту та узагальненої маршрутизації даних XML. Це претендент на масове управління споживчими побутовими товарами, такими як пральні машини, сушильні машини, холодильники тощо.

- CoAP (Constrained Application Protocol). Протокол обмежених додатків (CoAP) був розроблений IETF для використання в мережах з низьким енергоспоживанням і обмеженими мережами. CoAP використовує UDP, який

стає надійним для пристроїв IoT для миттєвої передачі даних. CoAP є хорошим вибором для пристроїв IoT через його підхід до енергозбереження. Комбіноване використання CoAP і 6LoWPAN дозволяє здійснювати двонаправлений Інтернет-зв'язок між сенсорними пристроями та Інтернет-серверами [10].

- MQTT. MQ Telemetry Transport – це протокол з відкритим кодом для обмежених пристроїв і мереж з низькою пропускною здатністю та високою затримкою. Це надзвичайно легкий транспортний засіб для публікації/підписки, який ідеально підходить для підключення невеликих пристроїв до обмежених мереж. Протокол MQTT у середовищі Інтернету речей може гарантувати негайне відновлення після несправності.

Результати порівняльного аналізу протоколів, що використовуються в інтернеті речей представлено у вигляді таблиці 1.2.

Таблиця 1.2 - Результати порівняння протоколів, що застосовуються у системі IoT

Протокол	CoAP	XMPP	RESTful HTTP	MQTT
Транспортний	UDP	TCP	TCP	TCP
Повідомлення	Запит/ Відповідь	Опублікувати/ Підписатися Запит/ Відповідь	Запит/ Відповідь	Опублікувати/ Підписатися Запит/Відповідь
2G, 3G (1000 вузлів)	Відмінна	Відмінна	Відмінна	Відмінна
LLN (1000 вузлів)	Відмінна	Низька	Низька	Низька
Обчислювальні ресурси	10Ks RAM/ Flash	10Ks RAM/ Flash	10Ks RAM/ Flash	10Ks RAM/ Flash
Приклади застосування	Комунальні польові мережі	Дистанційне управління побутовою технікою	Інтелектуальний енергетичний профіль 2 (управління енергією приміщення, домашні послуги)	Розширення корпоративного обміну повідомленнями в програмах IoT

Хмарне середовище.

Основою IoT є взаємозв’язок пристроїв IoT. У типовій системі IoT є величезна кількість пристроїв. У результаті системи IoT генеруватимуть величезну кількість даних. Не завжди всі пристрої IoT стабільні в деяких областях застосування пристрої IoT можуть бути мобільним. Отже, для таких систем IoT потрібна обчислювальна система з швидшою обробкою, розподіленою природою та здатна обробляти великі дані. Тому хмарні обчислення є одним із варіантів для більшості систем Інтернету речей. Існує кілька постачальників послуг хмарних обчислень. Це включає платформу Microsoft Azure, хмарну службу Google, Amazon тощо. Для системи IoT вибір служби хмарних обчислень може ґрунтуватися на таких параметрах, як:

Сервісна модель: Існує кілька моделей обчислювальних послуг, наприклад: інфраструктура як послуга (IAAS), платформа як послуга (PAAS) і програмне забезпечення як послуга (SAAS). Яка модель сервісу хмарних обчислень найкраще підходить для IoT, залежить від області її застосування IoT.

Підтримка програмного забезпечення: Постачальник хмарних послуг повинен підтримувати стек протоколів IoT, як-от MQTT, Restful, HTTP і CoAP.

Вартість послуги: Очевидно, що це основний параметр для вибору сервісу хмарних обчислень. За бюджетом IoT-проекту та домену IoT-додатку можна вибрати постачальника хмарних послуг.

Безпека відіграє життєво важливу роль, коли ці пристрої знаходяться поблизу нас і надсилають свої дані через мережу. Пристрої IoT також широко використовуються в промисловості. Тому важливо враховувати ризики кіберуразливості та атак у середовищі IoT і впроваджувати рекомендації, щоб певною мірою захистити середовище IoT.

2 ТИПИ АТАК НА IOT СИСТЕМУ ТА ВІДПОВІДНІ МЕТОДИ ЗАХИСТУ

2.1 Цілі безпеки IoT

Системи IoT проникли у багато важливих сфер, таких як медицина, освіта та фінанси. Незважаючи на це, IoT широко використовується вдома, в офісах, на вулицях міста тощо. Усі вони настільки ж значні. Крім того, IoT полегшує підключення до інтелектуальних речей, хмарних сервісів і різних програм. Крім того, світ швидко почав покладатися на штучний інтелект, що є зростаючою тенденцією в цифровому світі. У результаті виробники та розробники Інтернету речей відчувають тиск, щоб захистити цю технологію, щоб вони могли задовольнити майбутні потреби з високими вимогами.

Довіра до пристрою IoT починається з підтвердження його безпеки. Це важливий крок, особливо якщо такий пристрій підключено до Інтернету. Це з'єднання робить його вразливим до численних ризиків безпеки, таких як конфіденційність програмного забезпечення, шкідливі напади, кібератаки та атаки зловмисного програмного забезпечення. Однак постійна потреба у вдосконаленні та оновленні в цій області не може покладатися на існуючі методи забезпечення безпеки. Оскільки регулярно з'являються нові загрози, необхідно постійно оновлювати існуючі фреймворки та пропонувати нові рішення, одночасно оновлюючи поля IoT.

Щоб зробити середовище IoT безпечним [11], усі компоненти IoT повинні намагатися досягти наведених нижче цілей безпеки:

- Конфіденційність - це збереження конфіденційності даних, щоб лише авторизовані користувачі (як люди, так і пристрої) мали доступ до цих даних. Криптографія є ключовою технологією для досягнення конфіденційності.
- Цілісність - це процес, у якому зберігається повнота та точність даних при передачі інформації між пристроями в мережі речей та на вищих рівнях.
- Безвідмовність - це процес, у якому система IoT може перевірити інцидент або не інцидент події та діяти згідно встановлених політик, не зупиняючи систему в цілому.

- Доступність - це здатність системи IoT забезпечувати доступ до своїх послуг, якщо цього вимагають авторизовані об’єкти або користувачі.
- Конфіденційність - це процес, у якому система IoT дотримується правил або політики конфіденційності та дозволяє користувачам контролювати свої конфіденційні дані.
- Моніторинг забезпечує здатність системи IoT здійснювати надійний моніторинг своїх дій.
- Підзвітність - це процес, у якому система IoT зобов’язує користувачів брати на себе відповідальність за свої дії.
- Надійність - це забезпечення здатності системи Інтернету речей підтверджувати особу та підтверджувати довіру до третьої сторони.

Якщо одна із цілей не забезпечується системою, то автоматично це стає вразливістю і певним слабким місцем. Лише досягнувши та забезпечивши усі цілі безпеки, IoT система є захищеною і може функціонувати для подальшого її використання.

2.2 Атаки на IoT систему

Згідно з оновленим звітом SonicWall Cyber Threat Report за середину року, за перші шість місяців 2023 року кількість зловмисного програмного забезпечення IoT у всьому світі зросла на 37%, що призвело до 77,9 мільйонів атак у порівнянні з 57 мільйонами атак за перші шість місяці 2022 року [12].

Цього разу кількість атак зменшилася в Північній Америці на 3%, але в Азії та Латинській Америці спостерігалося тризначне зростання – 170% і 164% відповідно. Що стосується країн з найбільшим зростанням і зниженням, то кількість атак IoT в Індії зросла на приголомшливі 311%, але в Німеччині знизилася на 30%.

Якщо говорити про галузеві цифри, то хороша новина полягає в тому, що уряд і освіта зафіксували падіння на 73%, фінанси та охорона здоров’я – на 60%, і лише в секторі роздрібної торгівлі кількість атак зросла на 13%. .

Характеристики Інтернету речей, включаючи численні підключення до системи та обробку величезних обсягів даних, підвищують ймовірність того, що хакери будуть націлюватися на неї. Крім того, постачальники не єдина сторона, яка хвилюються щодо кібербезпеки IoT, але споживачі потребують технологій, яким вони можуть довіряти. Найкращі рішення безпеки для цієї технології, що розвивається, є надзвичайно необхідними. Тим не менш, діапазон запропонованих методів і пропонованих рішень, які були надані останніми дослідженнями кібербезпеки IoT, спонукав до наступних запитів щодо того які методи виявилися найбільш ефективними для виявлення вразливості IoT чи які типи атак можуть бути спрямовані на системи IoT.

Програми Інтернету речей використовуються багатьма користувачами, але водночас вони можуть наражати користувачів на безпрецедентні загрози безпеці та проблеми. Більшість пристроїв IoT напряму підключаються до Інтернету та обмінюються своїми даними з певним рівнем довіри без виконання будь-яких тестів безпеки. Тож більшість атак, які існують у кіберпросторі, також можливі в IoT. IoT використовує бездротову сенсорну мережу як основу, тому атаки WSN також присутні в середовищі IoT. Нижче наведено кілька атак (таблиця 2.1), можливих на різних рівнях архітектури IoT [13].

Нижче описано деякі з вищезазначених атак, щоб зрозуміти природу атак, які завдають шкоди на різних рівнях середовища IoT.

- Апаратний троян. Однією з основних проблем безпеки для мікросхем є апаратні трояни. Вони зловмисно модифікують IC, щоб дозволити зловмисникам використовувати їх функціональні можливості та отримати доступ до програмного забезпечення, що з ними працює.

- Реплікація вузла. Головною метою такої атаки є зловмисне додавання об’єкта шляхом дублювання ідентифікаційного номера одного об’єкта до поточного набору об’єктів. Через цю атаку може статися значне зниження продуктивності мережі. Крім того, після надходження пакетів до репліки це може не тільки пошкодити пакети, але й скерувати їх неправильно, завдаючи серйозної шкоди системам IoT, дозволяючи зловмиснику отримати доступ до параметрів безпеки (наприклад, спільних ключів). Він також здатний відкликати

авторизовані вузли, оскільки він може виконувати протокол відкликання об’єктів (див. таблицю 2.1).

Таблиця 2.1 - Можливі атаки на IoT систему структуровані по архітектурних рівнях

Рівень	Можлива атака
Мережа речей	Апаратний троян, реплікація вузла, DoS-атаки (позбавлення сну, розрядження батареї, атака збою), фізична атака, зловмисний вузол, атака на бічний канал, підслуховування, атаки на перехоплення, шум у даних, атака повтору. Атака на канал зв’язку, атака на зіткнення, атака на фрагментацію, атаки на маршрутизацію (Hello packet flood, Hole, Sybil Attack, Worm Hole, Selective Forwarding, Black Hole), прослуховування, введення шкідливих пакетів, несанкціонована розмова, DoS-атаки, десинхронізована атака.
Хмарні обчислення	Атака на веб-браузер, атака з використанням підпису, ін’єкція зловмисного програмного забезпечення в хмару, атака на хмару/сервер, ін’єкція SQL.
Програми та додатки	Впровадження коду, переповнення буфера, фішинг-атака, автентифікація та авторизація, викрадення особистих даних, втручання в програму на основі вузла, діра в безпеці програми, віддалене налаштування.

- Атаки на відмову в обслуговуванні (DoS): атаки DoS на обчислювальних вузлах можна класифікувати за трьома категоріями: атаки з депривацією сну, збоями та розрядкою акумулятора на крайньому рівні. У режимі депривації сну вузол, що працює від батареї, може отримати величезну

кількість запитів, які виглядають як законні, надіслані зловмисником. Деякі пристрої IoT працюють від акумулятора. Атака, пов’язана з розрядженням батареї, є надзвичайно потужною, що призводить до шкідливих наслідків, таких як відключення електроенергії. Атаки збоїв відбуваються, коли об’єкт IoT перестає виконувати свої основні функції. Це могло статися через небажану помилку на етапі виробництва, депривацію сну та впровадження коду.

- Фізична атака. У деяких об’єктах додатків IoT, розгорнутих у агресивному середовищі, такі об’єкти вразливі до фізичного доступу, що може призвести до атак на апаратне/програмне забезпечення. Маючи фізичний доступ до об’єкта, зловмисник може отримати дорогоцінну криптографічну інформацію, змінити операційну систему та знищити схему, що може призвести до тривалого знищення.

- Зловмисний вузол. У середовищі Інтернету речей деякий вузол отримує неавторизований доступ до мережі Інтернету речей та інших об’єктів і порушує функції та безпеку середовища.

- Атаки каналу. Це сильна атака на методи шифрування, яка може вплинути на їх безпеку та надійність. Під час атаки каналу на рівні крайового вузла об’єкти виконують свої звичайні операції, існує ймовірність того, що такі об’єкти можуть розкрити критичну інформацію, атаки каналу на рівні зв’язку не є інвазивними, оскільки вони виявляють лише навмисно витік інформації.

- Атаки зіткнення. Цей тип атак може бути запущений на рівні зв’язку. Одним із способів є додавання шуму в канал зв'язку, що призводить до повторної передачі пакетів і виснаження обмежених енергоресурсів.

- Атаки фрагментації. Хоча 6LoWPAN не має жодного механізму захисту, його безпека забезпечується базовими рівнями (наприклад, IEEE 802.15.4). IEEE 802.15.4 має максимальну одиницю передачі (MTU) 127 байт, тоді як IPv6 має мінімальний MTU 1280 байт. Завдяки технології фрагментації 6loWPAN забезпечує передачу пакетів IPv6 через IEEE 802.15.4. У цьому випадку зловмисник може вставити шкідливий пакет серед інших фрагментів, оскільки 6loWPAN розробив без методів автентифікації.

- Атаки маршрутизації. Для передачі даних у середовищі IoT багато протоколів маршрутизації використовуються в мережі. Зловмисний вузол модифікував пакет, генерував підроблені пакети, змінював маршрут. Згідно з дослідженням літератури, у ньому можливі атаки Sybil, Gray Hole, Wormhole, Hello flood, а також можливі типи атак із вибірковим перенаправленням.

- Неавторизована розмова. Для обміну даними та доступу до них кожен об'єкт IoT потребує зв'язку з іншими об'єктами. При цьому кожен об'єкт повинен взаємодіяти лише з набором об'єктів, яким потрібні його дані. Такий вид обмеженої взаємодії запобігатиме неавторизованому доступу до об'єктів IoT, що є фундаментальною вимогою безпеки IoT. Наприклад, термостат у розумному будинку значною мірою залежить від даних детектора диму, щоб вимкнути систему опалення у разі небезпеки. Тим не менш, незахищений обмін даними з іншими об'єктами детектором диму може поставити під загрозу весь розумний дім.

- Flood атака в хмарі. Це одна з форм атак на відмову в обслуговуванні в хмарі. Тут зловмисники постійно надсилають запити до служби в хмарі, що виснажує ресурси в хмарі, тим самим впливаючи на якість обслуговування. Коли хмарна система виявляє, що поточний екземпляр служби не відповідає вимогам; це перенесе уражену службу на інші сервери. Це призведе до збільшення навантаження на інші сервери.

- Ін'єкція зловмисного програмного забезпечення в хмарі: зловмисник може змінити дані, отримати контроль і запустити шкідливий код, впровадивши екземпляр шкідливої служби або віртуальну машину в хмару.

- Атака загортання підпису. Хмарна система використовує XML-підпис для забезпечення цілісності служби. Зловмисник змінює підслухані повідомлення, не роблячи підпис недійсним. Деякі хмари використовують SOAP. Зловмисники використовують уразливості в SOAP, щоб змінити прослухані повідомлення.

- Атака SQL-ін'єкції. Зловмисники використовують веб-інтерфейс або інтерфейс мобільного додатка, щоб запускати оператори SQL для операцій читання, запису та видалення. Така атака може не тільки отримати особисті дані

користувача, але й загрожувати всій системі баз даних. Коли веб-програми піддаються атаці за допомогою ін’єкції SQL, поточна сторінка показує інші результати порівняно з правдивою інформацією.

- Атаки на прикладному рівні в основному спрямовані на (несанкціонований) доступ до конфіденційних даних користувача. Зловмисники зазвичай використовують уразливі місця програм і додатків (наприклад, впровадження коду, переповнення буфера) або несанкціонований доступ для атаки. Одним із підходів для отримання неавторизованим агентом такого ж дозволу, як і законні користувачі, є підробка особи. Окрім цих атак, віруси, хробаки та трояни також загрожують прикладному рівню. Крім того, інші шкідливі програми (Rootkit, шпигунське програмне забезпечення, рекламне програмне забезпечення тощо) також підривають конфіденційність користувачів.

Більшість із зазначених вище атак можливі через неправильну конфігурацію та недотримання певних стандартів у середовищі IoT. Багато організацій працюють над оцінкою безпеки та наданням рекомендацій щодо безпечного налаштування середовища IoT.

OWASP (Open Web Application Security Project, проект безпеки відкритих веб-додатків) працює над деякими проблемами безпеки та постачається для IoT, і є розробленим, щоб допомогти виробникам, розробникам і споживачам краще зрозуміти проблеми безпеки, пов’язані з Інтернетом речей, і надати користувачам будь-які можливості контекст для прийняття кращих рішень щодо безпеки під час створення, розгортання або оцінки технологій IoT. OWASP запропонував деякі поширені проблеми в додатках IoT і кроки протидії для їх захисту.

2.3 Методи захисту IoT від можливих атак

Розглянемо класичні методи захисту системи IoT від можливих атак та представимо рекомендації щодо запобіганням такого роду атакам [14].

Однією із поширених вразливостей є погана фізична безпека, що характеризується тим, що слабкі сторони присутні, коли зловмисник може розібрати пристрій, щоб легко отримати доступ до носія інформації та будь-яких даних, що зберігаються на цьому носії. Слабкі сторони також присутні, коли порти USB або інші зовнішні порти можна використовувати для доступу до пристрою за допомогою функцій призначених для налаштування або обслуговування. Це може призвести до легкого несанкціонованого доступу до пристрою або даних.

Основними причинами виникнення такої вразливості є доступ до програмного забезпечення через порти USB або фізичне видалення носія даних.

Для того, щоб запобігти вразливості поганої фізичної безпеки користувач повинен бути впевненим, що:

- носій даних не можна легко видалити;
- збережені дані зашифровані в стані спокою;
- пристрій нелегко розібрати;
- не можна використовувати порти USB або інші зовнішні порти для отримання зловмисного доступу до пристрою;
- потрібні лише зовнішні порти для функціонування виробу;
- продукт має можливість обмеження адмініструвальні можливості.

Наступною поширеною вразливістю є небезпечне програмне забезпечення чи прошивка. Пристрої повинні мати можливість оновлюватися, коли виявляються вразливості, а оновлення програмного забезпечення чи прошивки можуть бути небезпечними, якщо самі оновлені файли та мережеве з’єднання, через яке вони доставляються, не захищені. Програмне забезпечення також може бути небезпечним, якщо воно містить жорстко закодовані конфіденційні дані, наприклад облікові дані. Неможливість оновлення програмного забезпечення та мікропрограми означає, що пристрої залишаються вразливими до проблеми безпеки, яку оновлення мало би вирішити. Крім того, якщо пристрої мають жорстко закодовані конфіденційні облікові дані, і якщо ці облікові дані розкриті, вони залишаються такими протягом невизначеного періоду.

Основними причинами виникнення такого типу вразливості є шифрування, що не використовується для отримання оновлень; оновлення не перевірено перед завантаженням; файл оновлення не зашифровано; мікропрограма містить конфіденційну інформацію.

Для того, щоб запобігти вразливості небезпечного програмного *забезпечення чи прошивки* користувач повинен бути впевненим, що:

- пристрій має можливість оновлення;
- файл оновлення зашифровано з використанням прийнятних методів шифрування;
- файл оновлення передається через зашифроване з'єднання;
- файл оновлення не розкриває конфіденційні дані;
- оновлення підписується та перевіряється, перш ніж дозволити його завантажувати та застосовувати;
- сервер оновлення безпечний.

Незахищені мережеві служби – також є однією з вразливостей IoT системи. Це стосується вразливостей у мережевих службах, які використовуються для доступу до пристрою IoT, які можуть дозволити зловмиснику отримати несанкціонований доступ до пристрою або пов’язаних з ним даних.

Причинами такої вразливості є вразливі служби, переповнення буфера, відкриті порти через UPnP, Xploitable UDP Services, відмова в обслуговуванні, DoS через мережевий пристрій, фузінг.

Для того, щоб запобігти вразливості *незахищених мережевих служб* користувач повинен бути впевненим, що:

- служби не вразливі до атак переповнення буфера та фузінгу;
- відкриті та доступні лише необхідні порти;
- служби не вразливі до атак DoS, які можуть вплинути на сам пристрій або інші пристрої та/або користувачів у локальній мережі чи інших мережах;
- мережеві порти або служби не піддаються доступу до Інтернету, наприклад, через UPnP.

Відсутність транспортного шифрування також є вразливістю. Тут йдеться про обмін даними з пристроєм IoT у незашифрованому форматі. Це може легко

призвести до того, що зловмисник отримає дані та або захопить ці дані для подальшого використання, або скомпрометує сам пристрій. Причинами можуть бути нешифровані послуги через Інтернет, незашифровані послуги через локальну мережу, погано реалізований SSL/TLS, неправильно налаштований SSL/TLS.

Для того щоб запобігти вразливості відсутності транспортного шифрування потрібно, щоб:

- дані шифрувались за допомогою таких протоколів, як SSL і TLS, під час проходження по мережах;
- інші промислові стандартні методи шифрування повинні використовуватись для захисту даних під час транспортування, якщо SSL або TLS недоступні;
- використовувались лише прийняті стандарти шифрування та необхідно уникати використання власних протоколів шифрування.

Недостатній рівень автентифікація/авторизація може виникнути через неефективність механізмів для автентифікації користувача IoT через інтерфейс та погана авторизація, за допомогою яких користувач може отримати вищі дозволені рівні доступу.

Для того щоб запобігти вразливості недостатнього рівня автентифікація/авторизація потрібно забезпечити наступне:

- високу надійність паролів;
- облікові дані повинні бути захищеними належним чином;
- застосувати двофакторну автентифікацію, де це можливо;
- забезпечити надійні механізми відновлення пароля;
- для конфіденційних функцій потрібна повторна автентифікація.

Однією з вразливістю є незахищений хмарний інтерфейс. Зазвичай це означає поганий контроль автентифікації або дані, що переміщуються в незашифрованому форматі надання зловмиснику доступу до пристрою або основні дані. Щоб уникнути вразливості незахищеного хмарного сервісу необхідно:

- під час першого налаштування потрібно змінити стандартні імена користувачів і пароль;
- механізми скидання пароля не повинні бути вразливими;
- має бути певний механізм блокування облікового запису після кількох невдалих спроб неавторизованого доступу;
- хмарний веб-інтерфейс повинен бути не сприйнятливий до XSS, SQLi або CSRF.

Більшість вразливостей є породженими не вірним конфігуруванням сервісів, пристроїв, неуважністю адміністраторів та недотриманням протоколів.

2.4 Роль машинного навчання (AI) в системах IoT

Щоб динамічно захищати системи від кіберзагроз, багато експертів з кібербезпеки звертаються до штучного інтелекту (AI). AI найчастіше використовується для виявлення вторгнень у сфері кібербезпеки шляхом аналізу шаблонів трафіку та перегляду діяльності, характерної для нападу.

Існує два основних види машинного навчання: контрольоване та неконтрольоване. Контрольоване навчання (з вчителем) – це коли люди вручну позначають навчальні дані як зловмисні або законні, а потім вводять ці дані в алгоритм, щоб створити модель із “класами” даних, які порівнюють з трафіком, який він аналізує. Неконтрольоване навчання (без вчителя) відмовляється від тренувальних даних і ручного маркування, а замість цього алгоритм групує подібні фрагменти даних у класи, а потім класифікує їх відповідно до узгодженості даних в одному класі та модульності даних між класами.

Одним із популярних алгоритмів машинного навчання для кібербезпеки є наївний класифікатор Байєса, який прагне класифікувати дані на основі теореми Байєса, згідно з якою припускається, що всі аномальні дії походять від незалежних подій, а не від однієї атаки. Класифікатор наївний Байєса – це контрольований алгоритм навчання, і після того, як він навчений і створив свої класи, аналізуватиме кожну дію, щоб визначити ймовірність того, що вона є

аномальною. Алгоритми машинного навчання також можна використовувати для створення інших моделей.

Дерево рішень – це тип штучного інтелекту, який створює набір правил на основі своїх навчальних зразків даних. Він використовує ітераційний поділ, щоб знайти опис (часто просто “атака” або “норма”), який найкраще класифікує трафік, який він аналізує. Прикладом такого підходу в кібербезпеці є виявлення атак DoS шляхом аналізу швидкості потоку [15], розміру та тривалості трафіку. Наприклад, якщо швидкість потоку низька, але тривалість трафіку велика, ймовірно, це буде атака, і, отже, буде класифікуватися як така. Дерево рішень також можна використовувати для виявлення атак із впровадженням команд у роботизованих транспортних засобах шляхом класифікації значень із споживання ЦП, потоку мережі та обсягу записаних даних, як показано на рисунку 2.1.

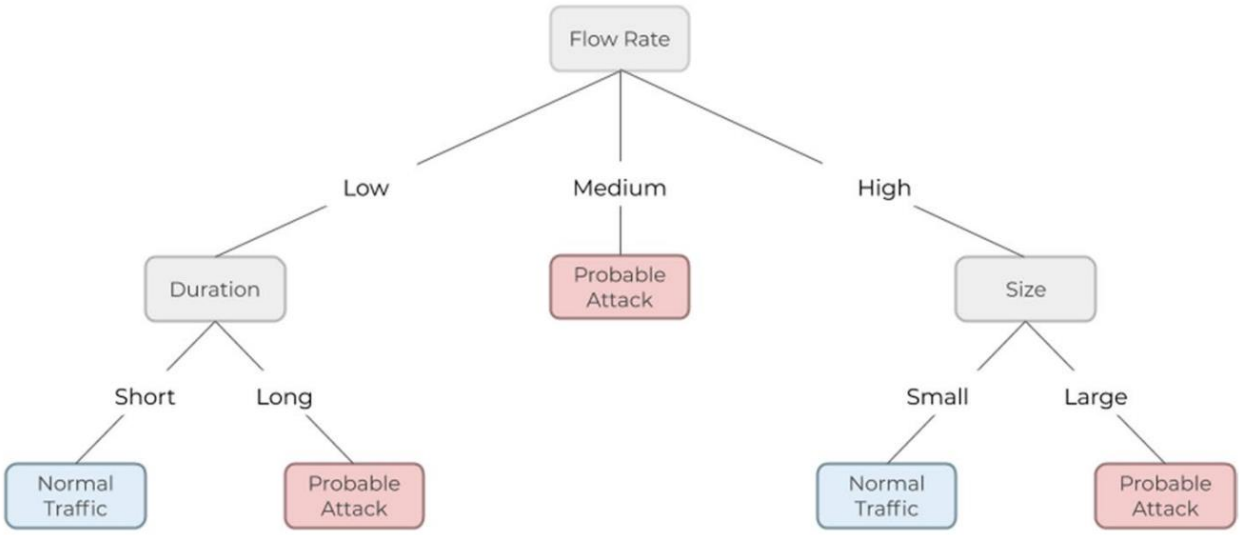

Рисунок 2.1 – Приклад використання дерева рішень для класифікації мережевого трафіку

Цей метод популярний, оскільки він інтуїтивно зрозумілий у тому, що штучний інтелект розглядає і не розглядає аномальний трафік, відомий розробнику. Крім того, як тільки буде знайдено ефективну серію правил, штучний інтелект може аналізувати трафік у режимі реального часу, надаючи майже миттєве сповіщення, якщо виявлена незвичайна активність.

Іншим підходом до дерев прийняття рішень є техніка навчання правилам, яка шукає набір характеристик атаки на кожній ітерації, одночасно максимізуючи певну оцінку, яка вказує на якість класифікації (тобто кількість неправильно класифікованих зразків даних). Основна відмінність між традиційними деревами рішень і методами навчання правилам полягає в тому, що традиційні дерева рішень шукають характеристики, які призведуть до класифікації, тоді як техніка навчання правилам знаходить повний набір правил, які можуть описати клас. Це може бути перевагою, оскільки воно може врахувати поради людини під час генерації правил, що створює оптимізований набір правил.

Для ідентифікації атак у мережевому трафіку автори використовували також модель XGBoost [16] або eXtreme Gradient Boosting – це алгоритм машинного навчання в рамках ансамблевого навчання. XGBoost створює прогностичну модель, об'єднуючи прогнози кількох окремих моделей, часто дерев рішень, ітераційним способом.

Алгоритм працює шляхом послідовного додавання слабких учнів до ансамблю, при цьому кожен новий учень зосереджується на виправленні помилок, допущених наявними. Він використовує техніку оптимізації градієнтного спуску, щоб мінімізувати попередньо визначену функцію втрат під час навчання.

Ключові особливості алгоритму XGBoost включають його здатність обробляти складні зв'язки в даних, методи регуляризації для запобігання надмірному оснащенню та включення паралельної обробки для ефективних обчислень.

Іноді може бути недостатньо покладатися на результати лише однієї моделі машинного навчання. Ансамблеве навчання пропонує систематичне рішення для поєднання передбачуваної здатності кількох учнів. Результатом є єдина модель, яка дає сукупний результат кількох моделей.

Моделі, які утворюють ансамбль, також відомі як базові учні, можуть бути або з одного алгоритму навчання, або з різних алгоритмів навчання. Багінг і бустинг – це два широко використовувані інструменти для вивчення ансамблю.

Хоча ці дві методики можна використовувати з декількома статистичними моделями, найбільше використання було з деревами рішень.

Баггінг. Хоча дерева рішень є однією з моделей, які найлегше інтерпретувати, вони демонструють дуже різноманітну поведінку. Розглянемо єдиний навчальний набір даних, який ми випадковим чином розділили на дві частини. Тепер давайте використаємо кожну частину для навчання дерева рішень, щоб отримати дві моделі.

Коли ми підберемо обидві ці моделі, вони дадуть різні результати. Кажуть, що дерева рішень пов'язані з високою дисперсією через таку поведінку. Об'єднання або посилення агрегації допомагає зменшити розбіжності у будь-якого учня. Кілька дерев рішень, які генеруються паралельно, формують базу тих, хто вивчає техніку упаковки. Дані, відібрані із заміною, передаються цим учням для навчання. Остаточним прогнозом є усереднений результат усіх учнів.

Бустинг. Під час посилення дерева будуються послідовно таким чином, що кожне наступне дерево спрямоване на зменшення помилок попереднього дерева. Кожне дерево навчається у своїх попередників і оновлює залишкові помилки. Отже, дерево, яке росте наступним у послідовності, вивчатиме оновлену версію залишків.

Основні учні, які навчаються у бустингу, це слабкі учні, у яких упередження високі, а передбачувана здатність трохи краща, ніж випадкове вгадування. Кожен із цих слабких учнів надає деяку життєво важливу інформацію для прогнозування, що дає змогу техніці посилення виробляти сильних учнів шляхом ефективного поєднання цих слабких учнів. Остаточний сильний учень знижує як упередженість, так і дисперсію.

На відміну від таких методів пакетування, як Random Forest, у яких дерева вирощуються максимально, підсилення використовує дерева з меншою кількістю розколів. Такі маленькі дерева, які не дуже глибокі, добре інтерпретуються. Такі параметри, як кількість дерев або ітерацій, швидкість, з якою посилення градієнта вивчається, і глибина дерева, можна оптимально вибрати за допомогою методів перевірки, таких як k-кратна перехресна

перевірка. Наявність великої кількості дерев може призвести до переобладнання. Тому необхідно ретельно підбирати критерії зупинки для форсування.

Техніка k-найближчого сусіда (k-NN) вивчає вибірки даних для створення класів, аналізуючи евклідову відстань між новим фрагментом даних і вже класифікованими фрагментами даних, щоб вирішити, до якого класу слід віднести новий фрагмент, Наприклад, нова частина даних, коли k, кількість найближчих сусідів, дорівнює трьом (3), буде класифікована до класу два (2), але коли k дорівнює дев'яти (9), нова частина буде класифікована в клас 1 як показано на рисунку 2.2.

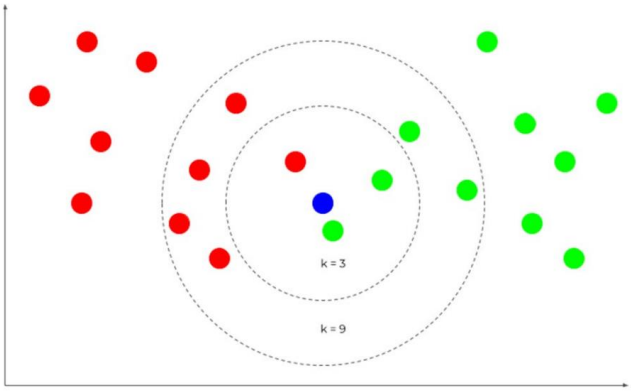

Рисунок 2.2 – Схематичне представлення методу k-NN (метод може по-різному класифікувати точку даних за різних значень k)

Техніка k-NN є привабливою для систем виявлення вторгнень, оскільки вона може швидко навчатися на нових шаблонах трафіку, щоб помітити раніше невидимі атаки, навіть атаки нульового дня. Експерти з кібербезпеки також досліджують застосування k-NN для виявлення кібератак у реальному часі. Цей метод використовувався для виявлення атак, таких як хибні атаки введення даних, і добре працює, коли дані можна представити через модель, яка дозволяє вимірювати їх відстань до інших даних, тобто через розподіл Гауса або вектор.

Метод опорних векторів (SVM) є розширенням моделей лінійної регресії, які розміщують площину, яка розділяє дані на два класи (див. рисунок 2.3).

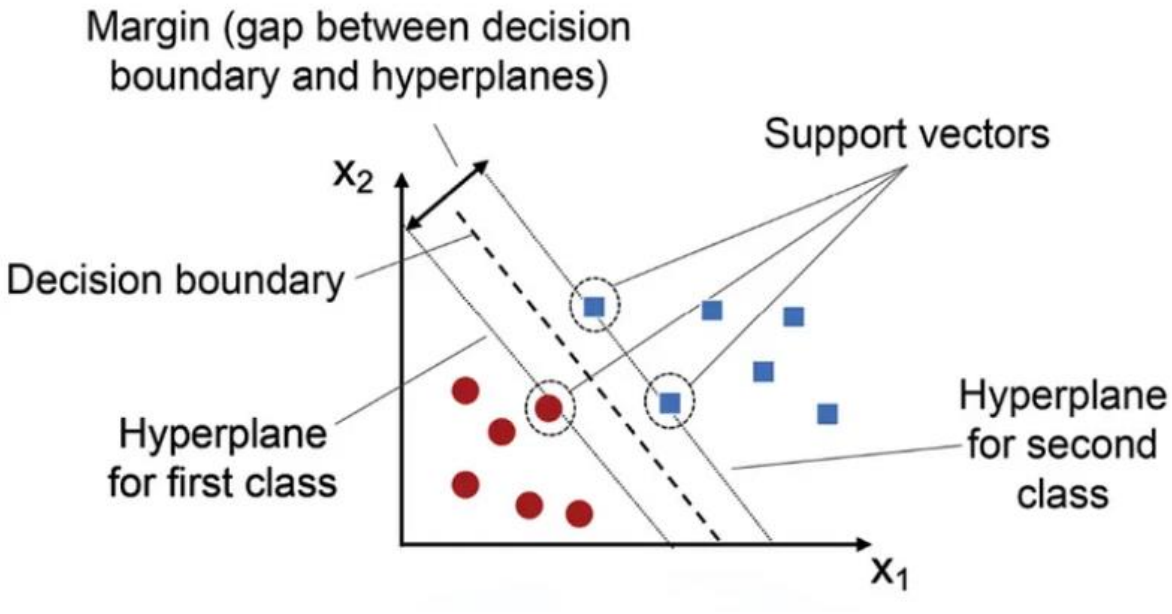

Рисунок 2.3 – Схематичне зображення методу опорних векторів

Ця площина може бути лінійною, нелінійною, поліноміальною, гаусовою, сигмоподібною тощо, залежно від функції, яка використовується в алгоритмі. SVM також можуть розділяти дані на більше ніж два класи, використовуючи більше ніж одну площину. У кібербезпеці ця техніка використовується для аналізу моделей інтернет-трафіку та розділення їх на класи компонентів, такі як HTTP, FTP, SMTP. Оскільки SVM є технікою керованого машинного навчання, її часто використовують у додатках, де можна симулювати атаки, наприклад, використовувати мережевий трафік, згенерований тестуванням на проникнення, як навчальні дані, такі як HTTP, FTP, SMTP.

Штучні нейронні мережі (ANN) – це техніка, заснована на тому, як нейрони взаємодіють один з одним у мозку, щоб передавати та інтерпретувати інформацію. У ANN нейрон – це математичне рівняння, яке зчитує дані та виводить цільове значення, яке потім передається наступному нейрону на основі його значення. Потім алгоритм ANN виконує ітерації, поки вихідне значення не стане прийнятно близьким до цільового значення, що дозволяє нейронам вивчати та виправляти свої ваги шляхом вимірювання похибки між очікуваним значенням і попереднім вихідним значенням. Після завершення навчання методом зворотного поширення помилки алгоритм представляє математичне

рівняння, яке виводить значення, яке можна використовувати для класифікації даних.

Великою перевагою ANN є те, що вони можуть коригувати свої математичні моделі, коли їм надають нову інформацію, тоді як інші математичні моделі можуть застаріти, оскільки нові типи трафіку та атак стають поширеними. Це також означає, що ANN вміють відловлювати раніше невидимі атаки та атаки нульового дня, оскільки вони враховують нову інформацію більш ретельно, ніж статичні математичні моделі. Через це ANN забезпечують надійне виявлення вторгнень систем і добре показали такі атаки, як DoS [17].

В даний час використання ШІ в кібербезпеці є невеликою, але є сферою, що швидко розвивається. Це також дорого та ресурсомістко, тому використання ШІ для захисту невеликої системи може бути неможливим. Однак компанії, які мають великі мережі, можуть отримати вигоду від цих рішень, особливо якщо вони розглядають або вже впровадили пристрої IoT у свою мережу. Кібербезпека штучного інтелекту також буде корисною для масивних систем, які можна знайти в розумному місті, і штучний інтелект зможе забезпечувати дуже швидкий час відгуку, що важливо в таких системах, як керування трафіком. У майбутньому кібербезпека штучного інтелекту також може бути інтегрована в менші системи, такі як безпілотні автомобілі чи розумні будинки. Крім того, багато заходів кібербезпеки штучного інтелекту виявляють або перешкоджають поточним атакам, а не запобігають атакам, а це означає, що інші запобіжні заходи безпеки також повинні бути введені.

Для оцінки алгоритмів машинного навчання, описаних у попередньому розділі, необхідно використовувати метрики, що відображатимуть якість класифікаціх. Є чотири сутності, які необхідно представити перед детальним пояснення суті метрик:

TP – кількість фактичних позитивних результатів, які були правильно визначено;

TN – кількість фактичних атак, які були правильно визначено;

FP – кількість фактичних позитивних результатів, які були визначені як атаки чи шкідлива програма;

FN – кількість фактичних атак, які були визначені не як загроза, а як норма.

Матриця помилок – це таблиця, яка дозволяє візуалізувати ефективність моделі, показуючи, які значення, на думку моделі, належать до яких класів. Вона має розмір N×N, де n – кількість класів, причому стовпці представляють фактичні класи, а рядки прогнозовані класи.

В роботі буде використано наступні метрики оцінки якості моделі машинного навчання: Accuracy (A), Precision (P), Recall (R).

Precision (P) – це показник, який оцінює модель шляхом обчислення частки правильно ідентифікованих позитивних результатів. Для її обчислення використовується формула (2.1).

$$P = \frac{TP}{TP+FP} \tag{2.1}$$

Accuracy (A) – це показник, який оцінює модель шляхом обчислення частки правильних прогнозів від загальної кількості прогнозів. Для обчислення цієї метрики використовується формула (2.2).

$$A = \frac{TP+TN}{TP+FP+FP+FN} \tag{2.2}$$

Recall (R) – це показник, який оцінює модель шляхом обчислення частки фактичних позитивних результатів, які були правильно визначені. Формула для цього показника якості (2.3).

$$R = \frac{TP}{TP+FN} \tag{2.3}$$

На основі проаналізованих алгоритмів машинного навчання та їх застосування для виявлення атаки у мережевому трафіку чи шкідливої програми у системі IoT було обрано для реалізації практичної частини методи XGBoost, SVM, NB.

3 ВИЯВЛЕННЯ ШКІДЛИВИХ ПРОГРАМ В МЕРЕЖІ IoT З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

3.1 Огляд набору даних IoT-23

В роботі було використано IoT-23 набір даних мережевого трафіку від пристроїв Інтернету речей (IoT). Цей мережевий трафік IoT був зафіксований у Stratosphere Laboratory, AIC group, FEL, CTU University, Чеська Республіка [18]. Мета створення такого набору – запропонувати дослідникам великий набір даних реальних і помічених шкідливих програм IoT і безпечного трафіку IoT для розробки алгоритмів машинного навчання, проведення дослідження в сфері захисту IoT. Цей набір даних і його дослідження фінансує компанія Avast Software, Прага.

Набір даних IoT-23 складається з двадцяти трьох сценаріїв різного мережевого трафіку IoT. Набір даних IoT-23 містить 20 сценаріїв зловмисного програмного забезпечення, виконаних на пристроях IoT, і 3 захоплення для безпечного трафіку пристроїв IoT. Він був вперше опублікований у січні 2020 року. Усі сценарії поділяються на двадцять записів мережі із заражених пристроїв IoT (які матимуть назву зразка зловмисного програмного забезпечення, виконаного в кожному сценарії) і три захоплення мережі реального мережевого трафіку пристроїв IoT (з назвами пристроїв, де трафік було зафіксовано).

Для кожного зловмисного сценарію було виконано певний зразок шкідливого програмного забезпечення в Raspberry Pi, який використовував кілька протоколів і виконував різні дії. Мережевий трафік, зафіксований для безпечних сценаріїв, був отриманий шляхом запису мережевого трафіку трьох різних пристроїв IoT: інтелектуальної світлодіодної лампи Philips HUE, інтелектуального домашнього персонального помічника Amazon Echo та розумного дверного замка Somfy, що зображено на рисунку 3.1. Важливо зазначити, що ці три пристрої IoT є реальним обладнанням. Це дозволило дослідниками фіксувати та аналізувати реальну поведінку мережі. І зловмисні, і

доброякісні сценарії працюють у контрольованому мережевому середовищі з необмеженим підключенням до Інтернету, як і будь-який інший справжній пристрій IoT.

Рисунок 3.1 – Елементи (речі) дослідної системи IoT-23

Набір даних у повному вигляді містить: файли .pcap, які є вихідними файлами запису мережі, файли conn.log.labeled, які створюються за допомогою аналізатора мережі під назвою Zeek, різні деталі та інформацію про кожен із записів завдяки тому факту, що легше працювати виключно з файлами conn.log.labeled, тільки вони були використані в цьому проекті. Файли .pcap створюються мережею завдяки програмі захоплення Wireshark, і їх можна відкрити лише за допомогою неї. Проте робота з цим файлами є трудомісткою і виявилася зайвою складністю для кваліфікаційної роботи бакалавра.

Типи атак, що є представленими в наборі даних представлені у таблиці 3.1. Для розуміння набору даних необхідно зрозуміти, яка інформація представляє відповідний стовбець файлу. Кожен із файлів conn.log.labeld містить 23 стовпці даних, типи яких представлено в таблиці 3.2. Стовпець conn-state є змінною, та специфічною лише для Zeek, і відображає стан з’єднання між двома пристроями. Наприклад, S0 означає, що пристрій намагається підключитися, але сторона-отримувач не відповідає. У цьому наборі даних усі значення, яких не було в будь-

якому із записів, було позначено тире “-”, лише IP-адреса була позначена двома крапками “::”.

Таблиця 3.1 - Типи атак, що є представлені у наборі даних IoT-23

Тип атаки	Опис атаки
Attack	загальна позначка, яка приписується аномаліям, які неможливо ідентифікувати
Benign	вказує на відсутність підозрілих або зловмисних дій у з’єднаннях.
C&C	вказує на те, що заражений пристрій було підключено до сервера CC (контролю та командування).
FileDownload	вказує на те, що файл завантажується на наш заражений пристрій. Це виявляється за допомогою фільтрації з’єднань із байтами відповіді понад 3 КБ або 5 КБ.
HeartBeat	вказує на те, що пакети, надіслані через це з’єднання, використовуються для відстеження зараженого хоста сервером C&C. Це було виявлено за допомогою фільтрації з’єднань із байтами відповіді, нижчими за 1 Б, і з періодичними подібними з’єднаннями.
Okiru	Ця мітка додається, коли потоки мають схожі шаблони з найпоширенішими відомими атаками ботнету Mirai.
PartOfAHorizontal PortScan	вказує на те, що підключення використовуються для горизонтального сканування портів для збору інформації для подальших атак.
Torii	ця позначка вказує на те, що підключення мають характеристики ботнету Torii. Це рішення про маркування було прийнято з тими ж параметрами, що й для Mirai, але з тією різницею, що ця сімейство ботнетів менш поширена.

Таблиця 3.2 - Типи інформації в наборі даних IoT-32

Назва колонки	Опис	Тип
ts	час, коли було зроблено захоплення, виражене в Unix Time	int
uid	ідентифікатор запису	str
id_orig.h	IP-адреса, де сталася атака, IPv4 або IPv6	str
id_orig.p	порт, який використовує відповідач	int
id_resp.h	IP-адреса пристрою, на якому відбувся запис	str
id_resp.p	порт, який використовується для відповіді від пристрою, де відбувся запис	int
proto	мережевий протокол, який використовується для пакету даних	str
service	протокол застосування	str
duration	кількість часу обміну даними між пристроєм і зловмисником	float
orig_bytes	кількість даних, надісланих на пристрій	int
resp_bytes	кількість даних, надісланих пристроєм	int
conn_state	стан з'єднання	str
local_orig	чи з’єднання виникло локально	bool
local_resp	чи була відповідь локальна	bool
missed_bytes	кількість пропущених байтів у повідомленні	int
history	історія стану зв'язку	str
orig_pkts	кількість пакетів, які надсилаються на пристрій	int
orig_ip_bytes	кількість байтів, які надсилаються на пристрій	int
resp_pkts	кількість пакетів, які надсилаються з пристрою	int
resp_ip_bytes	кількість байтів, які надсилаються з пристрою	int
tunnel_parents	ідентифікатор підключення, якщо воно тунельоване	str
label	тип захоплення, доброякісне чи зловмисне	str
detailed_label	якщо захоплення зловмисне, тип захоплення, як описано в таблиці 3.1	str

3.2 Алгоритм виявлення шкідливих програм в мережі IoT з використанням методів AI

В попередньому розділі було досліджено які саме методи та моделі штучного інтелекту використовувались науковцями для ідентифікації атаки у мережевому трафіку на основі проаналізованої літератури. До списку цих алгоритмів входять Random Forest, ADA Boost, ANN, MLP, XGBoost, NB, SVM. В цій кваліфікаційній роботі було використано не усі з попереднього переліку, а лише ті, які показали значні результати в аналогічних роботах щодо аналізу атак в звичайному мережевому трафіку, а саме: XGBoost, NB, SVM. Ці моделі машинного навчання є класичними і не є представниками глибокого навчання, тобто штучних нейронних мереж [19].

Задача виявлення шкідливих програм, що спричиняють аномалії у мережевому трафіку системи IoT, є задачею класифікації на основі промаркованих вручну даних.

Для виявлення шкідливих програм інстальованих на пристрої IoT необхідно розробити алгоритм, щоб відображав відповідні дії для досягнення конкретної цілі – виявлення аномалії у трафіку, відповідно шкідливого програмного забезпечення на пристрої.

На рисунку 3.2 зображено схематичне зображення етапів виявлення шкідливих програм в IoT з використанням моделей-класифікаторів.

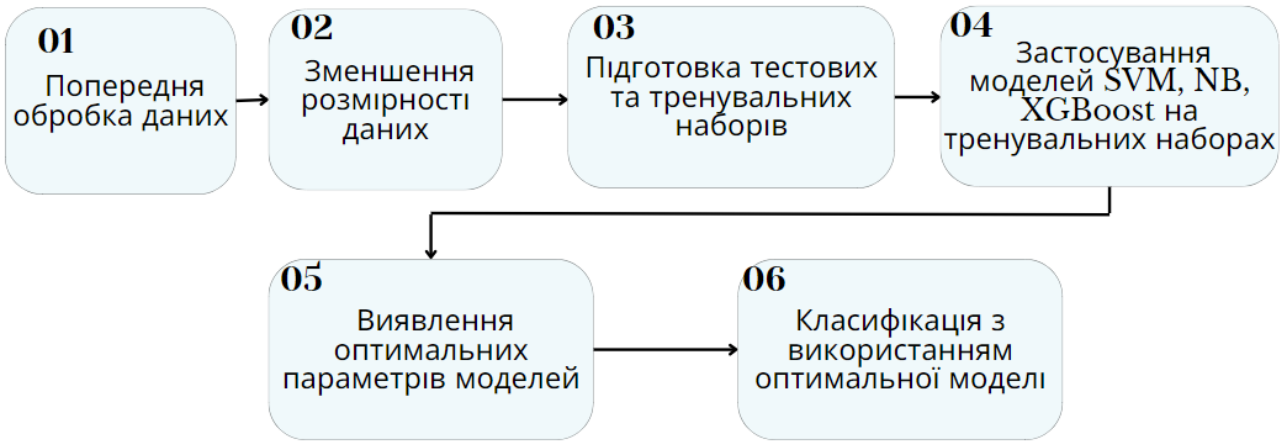

Рисунок 3.2 – Схематичне зображення алгоритму виявлення шкідливих програм в IoT з використанням моделей SVM, NB, XGBoost

Попередня обробка даних є важливим етапом при застосуванні будь якого алгоритму машинного навчання. Оскільки дані попадають на вхід моделі AI, необхідно щоб дані не були пустими (NULL), мали дозволені символи, відображали реальний стан сутності, за яку вони відповідають, тобто щоб їх значення було із дозволеного діапазону. У випадку не виконання попередньої обробки даних будуть отримані значні класифікаційні помилки, що в подальшому унеможливіть досягнення поставленої мети кваліфікаційної роботи.

Другим етапом є зменшення розмірності даних. Досліджуваний набір даних може містити значну кількість колонок, що відображають певну характеристику об’єкту дослідження, проте яка може бути не значною для моделі класифікатора і відповідно не нестиме інформації і не впливатиме на результат класифікації. Такі характеристики необхідно попередньо видалити із набору даних і це також зменшить розмірність і водночас швидкість тренування моделі.

Для реалізації процесу навчання з вчителем необхідно розділити дані на тренувальну та тестову вибірки, що генеруються випадковим за розподілом 80-20 (або за вказаними значеннями). Дуже важливо, щоб перед розподілом даних набір містив збалансовані значення кількостей усіх промаркованих представників. Інакше кажучи, якщо набір має лише два позначення (label) – true та false, то необхідно, щоб вони були представлені у наборі даних приблизно рівномірно. Лише в такому випадку у тестовому та тренувальному наборі будуть рівномірно представлені представники з кожного класу [20].

Наступним логічним етапом є застосування моделей-класифікаторів і отримання первинних значень точності моделей. На основі більшого значення точності можна обрати одну чи кілька моделей для подальшого її конфігурування та знаходження оптимальних параметрів.

Для знаходження оптимальних параметрів моделі зазвичай використовують функцію GridSearchCV, що дозволяє автоматично зробити комбінації параметрів моделі із вказаного діапазону та отримати відповідні значення точності. На основі показника точності буде обрано оптимальний набір параметрів моделі.

Останнім етапом є класифікації з використання моделі із її оптимальними параметрами для виявлення шкідливого ПЗ, що спричиняє аномалії у трафіку мережі IoT.

3.3 Попередня обробка та зменшення розмірності даних

Представимо перших кілька значень та характеристик із будь-якого сценарію на рисунку 3.3.

	id.resp_h	id.resp_p	proto	service	duration	orig_bytes	...	\
344375	120.10.164.48	23.0	tcp	NaN	<NA>	<NA>	...	
883270	61.182.158.196	9527.0	tcp	NaN	<NA>	<NA>	...	
94977	107.159.11.133	9527.0	tcp	NaN	0.998941	0	...	
552763	172.180.210.202	41534.0	udp	NaN	<NA>	<NA>	...	
796999	134.153.182.254	2323.0	tcp	NaN	<NA>	<NA>	...	
...	...	...	...	...	...	...	...	
295364	95.27.190.142	7243.0	udp	NaN	<NA>	<NA>	...	
955070	34.119.243.60	8080.0	tcp	NaN	2.998561	0	...	
589521	239.200.66.127	22646.0	udp	NaN	<NA>	<NA>	...	
69445	66.230.166.194	8080.0	tcp	NaN	2.998805	0	...	
1005051	75.10.105.178	23.0	tcp	NaN	<NA>	<NA>	...	

	local_resp	missed_bytes	history	orig_pkts	orig_ip_bytes	resp_pkts	\
344375	NaN	0.0	S	1.0	60.0	0.0	
883270	NaN	0.0	S	1.0	60.0	0.0	
94977	NaN	0.0	S	2.0	120.0	0.0	
552763	NaN	0.0	D	1.0	40.0	0.0	
796999	NaN	0.0	S	1.0	60.0	0.0	
...	...	...	...	...	...	...	
295364	NaN	0.0	D	1.0	40.0	0.0	
955070	NaN	0.0	S	3.0	180.0	0.0	
589521	NaN	0.0	D	1.0	40.0	0.0	
69445	NaN	0.0	S	3.0	180.0	0.0	
1005051	NaN	0.0	S	1.0	60.0	0.0	

Рисунок 3.3 – Перші кілька значень відповідних характеристик одного сценарію із набору даних IoT-23

Кілька характеристик є у стрічковому форматі і відображають IP-адресу, проте такий формат не підходить для інтерпретації класифікатором, тому потрібно провести перекодування цих даних у відповідний форму, яка буде зрозумілою для класифікатора. Для прикладу на рисунку 3.4 наведено інтерпретацію даних характеристики “id_resp_h” у зручний формат з використанням функції “label_encoder.fit_transform”. Такі дії були виконаними до усіх характеристик, що містять значення IP-адрес.

```
df['id.resp_h']= label_encoder.fit_transform(df['id.resp_h'])
df['id.resp_h'].value_counts()
```

id.resp_h		id.resp_h	
147.231.100.5	887	39585	887
89.221.214.130	304	163650	304
213.239.154.12	293	94905	293
37.187.104.44	278	121527	278
192.168.100.103	62	76879	62
	...		...
226.170.84.29	1	102917	1
31.31.8.190	1	117661	1
158.78.202.145	1	48583	1
114.19.233.206	1	12662	1
75.10.105.178	1	151656	1

Рисунок 3.4 - Перекодування характеристики “id_resp_h” у зручний формат з використанням функції “label_encoder.fit_transform”

Автори набору даних попередньо попередили, що усі значення, які є пустими, то будуть позначені як тире “-”, лише IP-адреса є позначеною двома крапками “::”. Для того, що краще працювати із пустими значеннями з використання вбудованих інструментів замінимо на звичайне NaN у Python і виведемо розподіл даних, у яких є пусті значення (див. лістинг 3.1) (див.рисунок 3.5).

Лістинг 3.1 - Ідентифікація пустих значень в наборі даних IoT-23 та обчислення їх кількості

```
df.replace('-', pd.NA, inplace=True)
null_values = df.isnull().sum()
plt.figure(figsize=(10, 6))
sns.barplot(x=null_values.index, y=null_values)
```

```
plt.xticks(rotation=45, ha='right')
plt.xlabel('Column')
plt.ylabel('Number of Null Values')
plt.title('Null Values in Malware_Detect_Data.csv')
plt.tight_layout()
plt.show()
null_values = df.isnull().sum()
null_percentage = (null_values / len(df)) * 100
columns_with_null = null_percentage[null_percentage > 0]
```

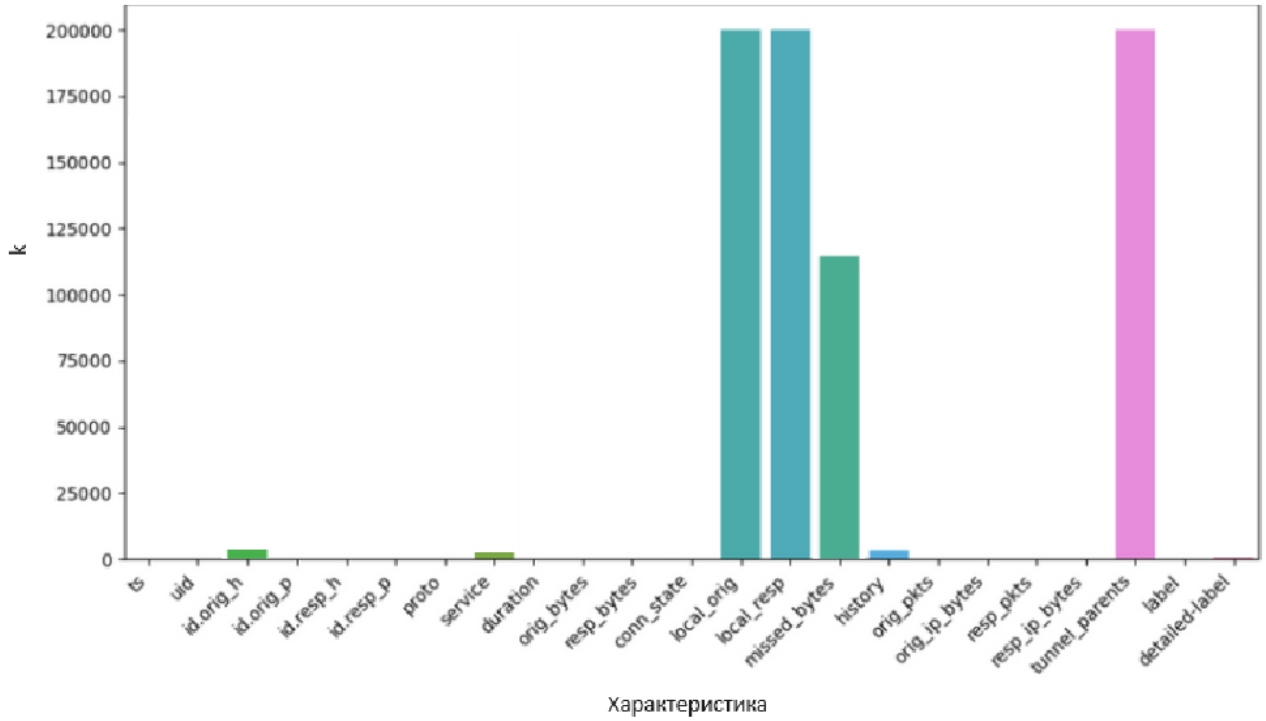

Рисунок 3.5 – Розподіл пустих значень одного сценарію в наборі даних IoT-23

Необхідно зауважити, що наявність пустих значень було перевірено на усіх 20 записах-сценаріях, що містять промарковані дані з вказанням виду атаки на систему IoT.

Переведемо пусті значення в колонці “duration” в числовий формат, який позначатиме нульову тривалість і буде розпізнано, як те, що запис не відбувся, або це початок запису (див. лістинг 3.2).

Лістинг 3.2 - Переведення NaN значення характеристики “duration” в числове значення “0”

```
label_encoder = preprocessing.LabelEncoder()
#label encode label
df['label']= label_encoder.fit_transform(df['label'])
df.head()
df['duration'] = pd.to_numeric(df['duration'])
```

Наступним кроком було виконано кореляційний аналіз даних з метою ідентифікації тих характеристик набору даних, які мають значний взаємозв’язок із міткою, тобто вплив. Результати кореляційного аналізу для одного з 20 сценаріїв наведені на рисунку 3.6.

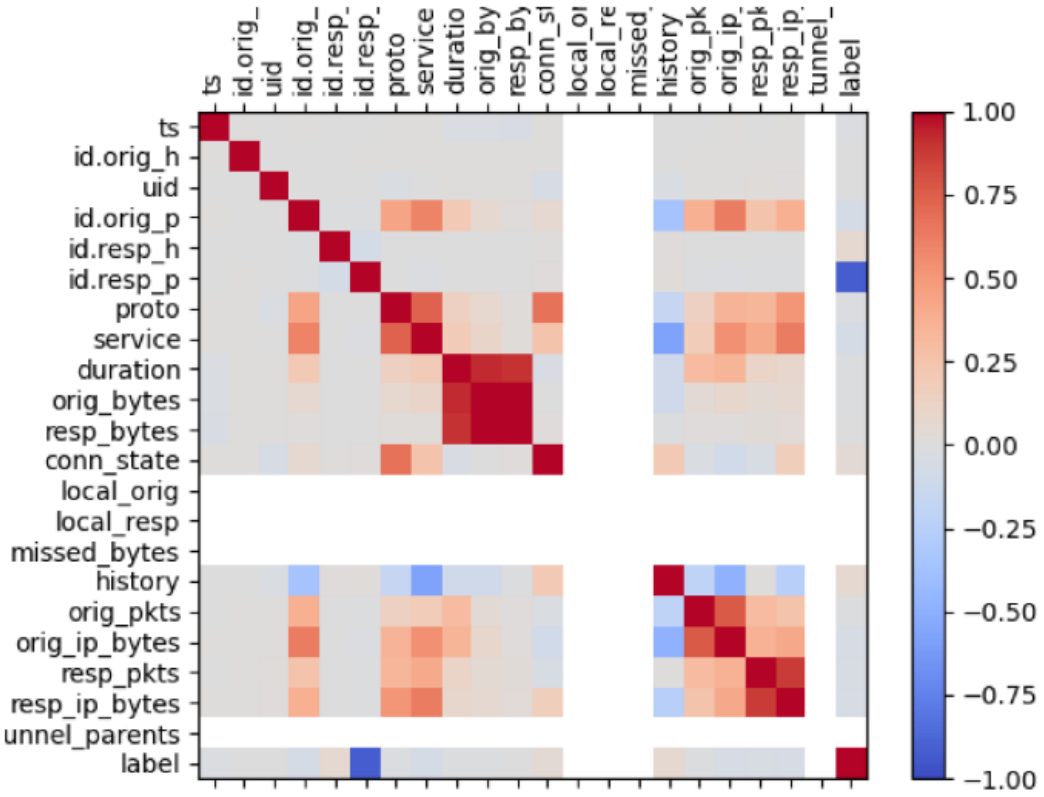

Рисунок 3.6 – Кореляційна матриця в одному з наборів даних IoT-32

Кореляційна матриця є розмірністю 23×23, що відображає взаємозв’язок кожної характеристики одна з одною. Кореляційна матриця представлена у шкалі від синього (негативний взаємозв’язок) до оранжевого (позитивний взаємозв’язок). Сірий колір означає слабкий зв'язок або його відсутність. Такий кореляційний аналіз був виконаний для усіх двадцяти сценаріїв і в результаті чого було зменшено розмірність даних.

Характеристики “ts”, “uid”, “id_orig.h”, “local_orig”, “local_resp”, “missed_bytes”, “tunnel_parents”. Перші три характеристики були видалені із набору даних, оскільки вони не мали впливу на значення мітки, а останні чотири характеристики були видалені через те, що більшість їх значень були пустими,

відповідно вони не несуть ніякої інформації. В результаті набір даних міститиме на 23 характеристики, а лише 16.

В результаті попередньої обробки даних було виконано:

- перетворення характеристик, що місять IP-адресу, у зручний формат значення завдяки функції “label_encoder.fit_transform” бібліотеки Scikit-learn;
- замінено пусті значення в характеристиці “duration” на числове значення “0”;
- проведено аналіз на пусті значення характеристик і в результаті було видалено 4 характеристики: “local_orig”, “local_resp”, “missed_bytes”, “tunnel_parents”;
- проведено кореляційний аналіз і в результаті було видалено “ts”, “uid”, “id_orig.h”, які не мають взаємозв’язку із міткою, тобто не будуть впливати на результати класифікації.

Після попередньої обробки наступним етапом є підготовка тренувальних і тестових наборів та застосування моделей.

3.4 Застосування моделей SVM, NB, XGBoost

Оскільки набір даних IoT-23 містить 23 сценарії, то після попередньої обробки кожного сценарію їх було об’єднано в один файл .csv з метою забезпечення опрацювання даних з одного джерела і для застосування над один файлом даних усіх трьох алгоритмів. Звичайно, для більш детального дослідження можна було би досліджувати алгоритми на кожному сценарії, потім порівнювати результати і шукати оптимальну модель. В кваліфікаційній роботі бакалавра було обрано спрощений варіант, щоб показати результати застосування моделей штучного інтелекту для виявлення шкідливих програм.

Перед тим як підготовляти тестові та тренувальні набори необхідно дослідити розподіл кожного класу атаки на мережевий трафік у наборі даних, щоб уникнути втрати точності та появи помилок класифікації.

Оскільки первинний набір даних не є збалансованими (див. таблицю 3.3), то перед розділенням на тренувальну та тестові вибірки було використано метод

передескретизації даних SMOTE, який штучно генерує дані з метою зменшення такого значного розриву між загальною кількістю представників кожного типу атаки. Для прикладу для C&C буде згенеровано оригінальні дані на основі представників у наборі, а для Okiru відбудеться вилучення зайвих, по суті укрупнення.

Для формування тренувальної та тестової вибірки було застосовану функцію train_test_split (див. лістинг 3.3), що розподіляє дані згідно вказаного відсоткового співвідношення. У роботу було використано 80% для тренувального набору, та 20% для тестового. Після того їх було нормалізовано.

Таблиця 3.3 - Кількість екземплярів кожної атаки (шкідливої програми) в наборі даних IoT-32

Атака/ шкідлива програма	Кількість записів
Attack	9398
Benign	30858735
C&C	22048
Heart Beat	34518
Mirai	2
Torii	30
DDoS	19538713
File Download	18
Okiru	60990711
Part of Horizontal Port Scan	213853817
Всього	325307990

Лістинг 3.3 - Формування тестувальної та тренувальної вибірок

```
from sklearn.model_selection import train_test_split
X = df.drop('label', axis=1)
y = df['label']
# Розподіл на X та Y
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
y_train.head()
y_train.value_counts()
```

```
from sklearn.preprocessing import Normalizer
scaler = Normalizer()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

У кваліфікаційній роботі було використано три моделі SVM, NB, XGboost. Розпочнемо із представлення моделі XGBoost та пошуку її оптимальних параметрів. Для її ініціалізації було використано бібліотеку XGBoost та модель XGBClassifier, що представлено у лістингу 3.4. Для пошуку оптимальних параметрів застосовано функцію RandomizedSearchCV.

Лістинг 3.4 - Застосування моделі XGBClassifier та пошук її оптимальних парметрів

```
from xgboost import XGBClassifier
from sklearn.model_selection import RandomizedSearchCV
import numpy as np

# Визначення класифікатора та масивів його можливих параметрів
xgb_model = XGBClassifier()
param_space = {
    'n_estimators': [100],
    'max_depth': [3, 4, 5, 6],
    'learning_rate': [0.01, 0.1, 0.2, 0.3],
    'subsample': [0.8, 0.9, 1.0],
    'colsample_bytree': [0.8, 0.9, 1.0],
    'gamma': [0, 0.1, 0.2, 0.3],
}

# Створення об'єкту RandomizedSearchCV
random_search = RandomizedSearchCV(
    xgb_model,
    param_space,
    n_iter=5,
    scoring='accuracy',  # Use the appropriate scoring metric
    n_jobs=-1,
    cv=5,  # Number of cross-validation folds
    random_state=42,  # Set a random seed for reproducibility
    verbose=3
)


# Оптимізація гіперпараметрів
random_search.fit(train_data, train_labels)

# Отримання оптимальних параметрів моделі
best_xgb_hps = random_search.best_params_
best_xgb_model = random_search.best_estimator_
```

Аналогічно було виконано для двох інших моделей GaussianNB із бібліотеки “sklearn.naive_bayes” та SVC із бібліотеки “sklearn.naive_svm” . В таблиці 3.4 представлено оптимальні параметри для кожної моделі.

Для оцінки якості моделей-класифікаторів було використано перехресну перевірку K-fold CV. Суть її полягає в тому, що набір даних ділиться на k рівних частин, для яких здійснюється оцінювання k разів із кожним із отриманих k підмножин, призначених для тестового набору. У роботі було використано значення k=5, тому кожен разу тестовий набір відповідав 20% усіх даних. Ця методика перевірки узагальнює результати ефективності на основі статистичного аналізу. Точність моделі в тестовому наборі порівнюється з точністю моделі в навчальному наборі. Для оцінки якості класифікації було обрано такі метрики Accuracy(A), Precision (P) і Recall (R) були обрані як показники. Отримані результати представлено в таблиці 3.5 у відповідності до запропонованої моделі з оптимальними параметрами. Завдяки застосуванню k-кратної перехресної перевірки отримані показники є фактично середніми для кожного критерію.

Таблиця 3.4 - Оптимальні параметри моделей XGBClassifier, GaussianNB, SVC

Модель	Оптимальні параметри моделі
XGBClassifier	'subsample': 0.8, 'n_estimators': 100, 'max_depth': 5, 'learning_rate': 0.01, 'gamma': 0.2, 'colsample_bytree': 0.8
GaussianNB	'priors':None, 'var_smoothing':1e-07
SVC	'C': 0.01, 'gamma': 0.001, 'kernel': 'rbf'

Таблиця 3.5 - Результати оцінки якості моделей-класифікаторів

Модель	Метрики якості		
	A, %	P, %	R, %
XGBClassifier	0,984	0,994	0,992
GaussianNB	0, 975	0,987	0, 973
SVC	0,863	0,923	0,781

Найкращі результати показує модель класифікатора XGBClassifier, яку і було використано для подальшої ідентифікації шкідливих програм як причини певних аномалій у мережі системи IoT.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Долікарська допомога при харчових отруєннях

Харчове отруєння – це захворювання, викликане вживанням зараженої чи зіпусутої їжі. Інфекційні організми, включаючи бактерії, віруси та паразити, або їх токсини є найпоширенішими причинами харчових отруєнь. Дедалі більше людей зіткнулися з цією проблемою, адже через постійне відключення світла дуже складно зберегти продукти у свіжому вигляді. А зіпсовані продукти – це основна причина отруєння. У цій статті ми докладніше розповімо про те, що таке харчове отруєння та як його уникнути.

Будь-яка їжа може стати причиною розладів органів шлунково-кишкового тракту. Але найчастіше, отруєння викликають:

- риба (особливо сира чи слабосолена);
- продукція холодного копчення;
- сирі яйця;
- молоко та молочні продукти;
- кондитерські вироби з кремом (торти, тістечка);
- домашні консерви.

Симптоми харчового отруєння залежить від джерела забруднення. Більшість видів харчового отруєння викликають одну або кілька з наступних ознак та симптомів:

- нудота;
- блювання;
- водяниста чи кривава діарея;
- біль у животі та судоми;
- висока температура, жар.

Такий симптом отруєння, як блювання, може відбуватися понад 15 разів на добу, а при діареї виділення мають водянистий характер і часто містять кров та слиз.

Ознаки та симптоми можуть виявитися через кілька годин після вживання зараженої їжі або через кілька днів або навіть тижнів. Хвороба, викликана харчовим отруєнням, зазвичай триває від кількох годин за кілька днів.

Перше, що потрібно зробити при отруєнні, це оцінити тяжкість симптомів та звернутися за лікарською допомогою. Якщо ви відчуваєте будь-які з наступних ознак або симптомів, негайно зверніться за медичною допомогою:

- нудота;
- блювання;
- часті епізоди блювання та нездатність утримувати рідини;
- кров'янисте блювання або діарея;
- діарея понад три дні;
- сильний біль чи спазми у животі;
- температура вище 38 °C;
- ознаки або симптоми зневоднення - надмірна спрага, сухість у роті, мале сечовипускання або його відсутність, сильна слабкість, запаморочення або непритомність;
- неврологічні симптоми, такі як нечіткість зору, м'язова слабкість та поколювання в руках.

Основні принципи першої допомоги за будь-яких видів отруєнь полягають у тому, щоб позбутися харчового отруєння, потрібно промити шлунок: варто випити 1-2 літри кип'яченої води (теплої), додавши до неї соди (1 ст.л. на 1 л води). Повторити процедуру потрібно 2-3 рази, доки у блювоті не з'являться чисті промивні води. Маленьким дітям шлунок можна промивати виключно в лікарні через зонд, тому в такому разі потрібно відразу звертатися за медичною допомогою.

Перша допомога при отруєнні: для боротьби зі зневодненням хворому потрібно кожні 10 хвилин пити трохи води. Можна використовувати зелений або чорний чай (неміцний) та негазовану мінеральну воду.

Також можна прийняти ентеросорбенти. Цей вид препаратів слід прийняти після промивання шлунка, що зменшить концентрацію токсинів у кишківнику.

Перед вживанням препарату бажано проконсультуватися з лікарем та прочитати інструкцію до цих ліків.

Звичайно легше попередити харчове отруєння чим потім ліквідовувати їх наслідки. Для цього потрібно дотримуватись наступних правил:

- ретельно та часто мити руки. Перед приготуванням та вживанням їжі, після повернення додому з вулиці, до та після туалету. Батьки повинні стежити за виконанням цих правил дітьми, тому дії дорослого - найкращий приклад для дитини.
- мити овочі та фрукти перед вживанням. Обов'язково промивайте їх у чистій проточній воді. Не дозволяйте їсти немиті продукти, навіть якщо збираєте з власного городу.
- правильно зберігати продукти та загалом стежити за гігієною вживання їжі. Харчуйтеся в перевірених місцях, обирайте безпечні методи приготування, дбайте про правильний температурний режим зберігання їжі.
- термічно обробляйте м’ясо та рибу. Вживайте тільки гарантовано свіжу продукцію. Навіть натяк на неприємний запах — привід викинути м’ясо/рибу у смітник. Це правило працює не лише з м'ясними та рибними продуктами, а й з будь-якими іншими.
- вживайте лише термічно оброблені яйця. Уникайте вживання продуктів, що приготовлені з сирих яєць (майонез, креми, соус тартар), особливо гусячих.
- безпечно розморожуйте продукти. Заборонено розморожувати їжу за кімнатної температури. Найкращий спосіб — перекласти продукт з морозилки в холодильник.
- зберігайте сирі продукти окремо від готових, це допоможе запобігти перехресному забрудненню.

4.2 Проведення інструктажів з охорони праці

Одним із обов’язків роботодавця є забезпечення проведення інструктажів з охорони праці на підприємстві. Згідно Закону України “Про охорону праці” від

14.10.1992 року з внесеними змінами від 21.11.2002 року працівники під час прийняття на роботу та протягом роботи мають проходити інструктаж з питань охорони праці. Тих, хто не пройшов інструктаж, не допускають до роботи. Оскільки працівники в сфері кібербезпеки працюють із пристроями, комп'ютерами, то вони аналогічно до будь-яких інших працівників повинні пройти інструктаж з охорони праці.

Працівники під час прийняття на роботу та періодично повинні проходити на підприємстві інструктажі з питань охорони праці, надання першої медичної допомоги потерпілим від нещасних випадків, а також з правил поведінки та дій при виникненні аварійних ситуацій, пожеж і стихійних лих.

Порядок проведення інструктажів з питань охорони праці на підприємстві визначає глава 6 Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці від 26.01.2005 р. № 15 (далі — Типове положення). Інструктажі залежно від характеру та часу проведення поділяються на види:

- вступний;
- первинний;
- повторний;
- позаплановий;
- цільовий.

Вступний інструктаж проводиться:

- з усіма працівниками, які приймаються на постійну або тимчасову роботу, незалежно від їх освіти, стажу роботи та посади;
- з працівниками інших організацій, які прибули на підприємство і беруть безпосередню участь у виробничому процесі або виконують інші роботи для підприємства;
- з учнями та студентами, які прибули на підприємство для проходження трудового або професійного навчання;
- з екскурсантами у разі екскурсії на підприємство.

Вступний інструктаж проводиться спеціалістом служби охорони праці або іншим фахівцем відповідно до наказу (розпорядження) по підприємству, який в

установленому Типовим положенням порядку пройшов навчання і перевірку знань з питань охорони праці. Вступний інструктаж проводиться в кабінеті охорони праці або в приміщенні, що спеціально для цього обладнано, з використанням сучасних технічних засобів навчання, навчальних та наочних посібників за програмою, розробленою службою охорони праці з урахуванням особливостей виробництва. Програма та тривалість інструктажу затверджуються керівником підприємства.

Запис про проведення вступного інструктажу робиться в журналі реєстрації вступного інструктажу з питань охорони праці (додаток 5 Типового положення), який зберігається службою охорони праці або працівником, що відповідає за проведення вступного інструктажу, а також у наказі про прийняття працівника на роботу.

Первинний інструктаж проводиться до початку роботи безпосередньо на робочому місці з працівником:

- новоприйнятим (постійно чи тимчасово) на підприємство або до фізичної особи, яка використовує найману працю;
- який переводиться з одного структурного підрозділу підприємства до іншого;
- який виконуватиме нову для нього роботу;
- відрядженим працівником іншого підприємства, який бере безпосередню участь у виробничому процесі на підприємстві.

Первинний інструктаж проводиться з учнями, курсантами, слухачами та студентами навчальних закладів:

- до початку трудового або професійного навчання;
- перед виконанням кожного навчального завдання, пов’язаного з використанням різних механізмів, інструментів, матеріалів тощо.

Первинний інструктаж на робочому місці проводиться індивідуально або з групою осіб одного фаху за діючими на підприємстві інструкціями з охорони праці відповідно до виконуваних робіт.

Повторний інструктаж проводиться на робочому місці індивідуально з окремим працівником або групою працівників, які виконують однотипні роботи,

за обсягом і змістом переліку питань первинного інструктажу. Повторний інструктаж проводиться в терміни, визначені нормативно-правовими актами з охорони праці, які діють у галузі, або роботодавцем (фізичною особою, яка використовує найману працю) з урахуванням конкретних умов праці, але не рідше:

- на роботах з підвищеною небезпекою — 1 раз на 3 місяці;
- для решти робіт — 1 раз на 6 місяців.

Позаплановий інструктаж проводиться з працівниками на робочому місці або в кабінеті охорони праці:

- при введенні в дію нових або переглянутих нормативно-правових актів з охорони праці, а також при внесенні змін та доповнень до них;\
- при зміні технологічного процесу, або модернізації устаткування, приладів та інструментів, вихідної сировини, матеріалів та інших факторів, що впливають на стан охорони праці;
- при порушеннях працівниками вимог нормативно-правових актів з охорони праці, що призвели до травм, аварій, пожеж тощо;
- при перерві в роботі виконавця робіт більш ніж на 30 календарних днів – для робіт з підвищеною небезпекою, а для решти робіт – понад 60 днів.

Позаплановий інструктаж з учнями, студентами, курсантами, слухачами проводиться під час проведення трудового і професійного навчання при порушеннях ними вимог нормативно – правових актів з охорони праці, що можуть призвести або призвели до травм, аварій, пожеж тощо.

Позаплановий інструктаж може проводитись індивідуально з окремим працівником або з групою працівників одного фаху. Обсяг і зміст позапланового інструктажу визначаються в кожному окремому випадку залежно від причин і обставин, що спричинили потребу його проведення.

Цільовий інструктаж проводиться з працівниками:

- при ліквідації аварії або стихійного лиха;
- при проведенні робіт, на які відповідно до законодавства оформлюються наряд-допуск, наказ або розпорядження.

Цільовий інструктаж проводиться індивідуально з окремим працівником або з групою працівників. Обсяг і зміст цільового інструктажу визначаються залежно від виду робіт, що виконуватимуться.

Первинний, повторний, позаплановий і цільовий інструктажі проводить безпосередній керівник робіт (начальник структурного підрозділу, майстер) або фізична особа, яка використовує найману працю.

Ці інструктажі завершуються перевіркою знань у вигляді усного опитування або за допомогою технічних засобів, а також перевіркою набутих навичок безпечних методів праці, особою, яка проводила інструктаж.

При незадовільних результатах перевірки знань, умінь і навичок щодо безпечного виконання робіт після первинного, повторного чи позапланового інструктажів протягом 10 днів додатково проводяться інструктаж і повторна перевірка знань. При незадовільних результатах перевірки знань після цільового інструктажу допуск до виконання робіт не надається. Повторна перевірка знань при цьому не дозволяється.

Про проведення первинного, повторного, позапланового та цільового інструктажів та їх допуск до роботи, особа, яка проводила інструктаж, уносить запис до журналу реєстрації інструктажів з питань охорони праці на робочому місці. Сторінки журналу реєстрації інструктажів повинні бути пронумеровані, прошнуровані і скріплені печаткою.

У разі виконання робіт, що потребують оформлення наряду-допуску, цільовий інструктаж реєструється в цьому наряді-допуску, а в журналі реєстрації інструктажів не обов’язково. Перелік професій та посад працівників, які звільняються від повторного інструктажу, затверджується роботодавцем. До цього переліку можуть бути зараховані працівники, участь у виробничому процесі яких не пов’язана з безпосереднім обслуговуванням об’єктів, машин, механізмів, устаткування; застосуванням приладів та інструментів, збереженням або переробкою сировини, матеріалів тощо.

ВИСНОВКИ

Аналіз сучасного стану кількості IoT систем та кількості пристроїв підключених до глобальної мережі Інтернет свідчать про те, що з кожним роком ці показники зростатимуть і відповідно зростатиме потреба в їх захисті.

IoT система є багаторівневою і на основі аналізу проведеному у другому розділі кожен рівень є вразливим до певного рівня атак та впливу шкідливих програм. На кожному рівні IoT виникають своєрідні вразливості, що потребуються специфічних саме для того рівня методів їх усунення.

Завдяки методам та інструментам штучному інтелекту можна з високою точністю виявляти атаки, що спричиненні шкідливим програмних забезпеченням, встановленим на пристрої IoT, та які спричиняють аномальні у мережевому трафіку системи. До таких моделей-класифікаторів, що ідентифікують атаки на основі мережевого трафіку належать випадковий ліс, нейронні мережі, SVM, класифікатор наївного Байєса, метод найближчого сусіда.

Для проектування моделі ідентифікації шкідливих програм в IoT було використано попередньо промаркований набір даних IoT-23 і виконано його попередню обробку з метою досягнення вищих показників якості класифікації.

В результаті попередньої обробки було перетворенно характеристик, що місять IP-адресу, у зручний формат значення завдяки функції “label_encoder.fit_transform” бібліотеки Scikit-learn; замінено пусті значення в характеристиці “duration” на числове значення “0”; проведено аналіз на пусті значення характеристик і в результаті було видалено 4 характеристики: “local_orig”, “local_resp”, “missed_bytes”, “tunnel_parents”; проведено кореляційний аналіз і в результаті було видалено “ts”, “uid”, “id_orig.h”, які не мають взаємозв’язку із міткою, тобто не будуть впливати на результати класифікації.

Для визначення моделі, яка є найбільш чутливою до набору даних IoT-23 було виконано пошук оптимальних параметрів для XGBClassifier, SVC, GaussianNB моделей. Такий перелік класифікаторів було обрано на основі

аналізу наявних досліджень і показників точності кожної з моделі у досліджених наукових статтях.

Найкращі показники якості класифікації продемонструвала модель XGBClassifier, а саме: A – 0,984, P – 0,994, R – 0,992.

Отримана модель і відповідний алгоритм виявлення шкідливих програм може бути використаний для аналізу мережевого трафіку в IoT системах, запобігати атакам в реальному часі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. B. Mazon-Olivo, “Internet of Things: State-of-the-art, Computing Paradigms and Reference Architectures”, IEEE Latin America Transactions 20(1):49-63
2. P. Fremantle and P. Scott, “A security survey of middleware for the Internet of Things PrePrints,” 2015.
3. IoT connections worldwide 2022-2033 | Statista. Statista. URL: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (дата звернення: 17.06.2024).
4. Бекер, І., Тимощук, В., Маслянка, Т., & Тимощук, Д. (2023). МЕТОДИКА ЗАХИСТУ ВІД ПОВІЛЬНИХ ТА ШВИДКИХ BRUTE-FORCE АТАК НА IMAP СЕРВЕР. Матеріали конференцій МНЛ, (17 листопада 2023 р., м. Львів), 275-276.
5. Abdul-Ghani, H. A., & Konstantas, “A Comprehensive Study of Security and Privacy Guidelines, Threats, and Countermeasures: An IoT Perspective”. Journal of Sensor and Actuator Networks, 8(2), p.22.
6. Ванца, В., Тимощук, В., Стебельський, М., & Тимощук, Д. (2023). МЕТОДИ МІНІМІЗАЦІЇ ВПЛИВУ SLOWLORIS АТАК НА ВЕБСЕРВЕР. Матеріали конференцій МЦНД, (03.11. 2023; Суми, Україна), 119-120.
7. B. Aziz, “A formal model and analysis of an IoT protocol,” Ad Hoc Networks, pp. 1–9, 2015
8. Іваночко, Н., Тимощук, В., Букатка, С., & Тимощук, Д. (2023). РОЗРОБКА ТА ВПРОВАДЖЕННЯ ЗАХОДІВ ЗАХИСТУ ВІД UDP FLOOD АТАК НА DNS СЕРВЕР. Матеріали конференцій МНЛ, (3 листопада 2023 р., м. Вінниця), 177-178.
9. S. A. Alvi, B. Afzal, G. A. Shah, L. Atzori, and W. Mahmood, “Ad Hoc Networks Internet of multimedia things: Vision and challenges,” Ad Hoc Networks, vol. 33, pp. 87–111, 2015
10. P. Persson and O. Angelsmark, “Calvin – Merging Cloud and IoT,” Procedia - Procedia Comput. Sci., vol. 52, pp. 210–217, 2015.

11. Тимощук, В., Долінський, А., & Тимощук, Д. (2024). СИСТЕМА ЗМЕНШЕННЯ ВПЛИВУ DOS-АТАК НА ОСНОВІ MIKROTIK. Матеріали конференцій МЦНД, (17.05. 2024; Ужгород, Україна), 198-200. https://doi.org/10.62731/mcnd-17.05.2024.008
12. SonicWall Cyber Threat report | SonicWall. SonicWall. URL: https://www.sonicwall.com/medialibrary/en/white-paper/2024-cyber-threat-report.pdf (дата звернення: 17.06.2024).
13. Тимощук , В., Долінський , А., & Тимощук , Д. (2024). ВИКОРИСТАННЯ ТЕХНІКИ ДИНАМІЧНОГО ВІДКРИВАННЯ МЕРЕЖЕВИХ ПОРТІВ ДЛЯ ПІДВИЩЕННЯ БЕЗПЕКИ СЕРВЕРІВ. Collection of Scientific Papers «ΛΌГОΣ», (May 24, 2024; Zurich, Switzerland), 233–234. https://doi.org/10.36074/logos-24.05.2024.051
14. The Role of Artificial Intelligence in IoT and OT Security. *CSO Online*. URL: https://www.csoonline.com/article/566503/the-role-of-artificial-intelligence-in-iot-and-ot-security.html (дата звернення: 17.06.2024).
15. M. Kuzlu, C. Fair, O. Guler, “Role of artificial Intelligence in the internet of things”, Discover internet of things, 1:7, 2021, pp. 3-12.
16. N. Zagorodna, M. Stadnyk, B. Lypa, M.Gavrylov, R. Kozak, “Network Attack Detection Using Machine Learning Methods”, Proceeding of 3rd International Conference CNDGS’2022, 2022. PP. 55-61.
17. S. Zeadally, E. Adi, Z. Baig, I. Khan, “Harnessing artifcial intelligence capabilities to improve cybersecurity”, IEEE Access. 2020;8:23817–37.
18. Sebastian Garcia, Agustin Parmisano, & Maria Jose Erquiaga.. IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]. Zenodo.
19 Skorenkyy, Y., Kozak, R., Zagorodna, N., Kramar, O., & Baran, I. (2021, March). Use of augmented reality-enabled prototyping of cyber-physical systems for improving cyber-security education. In Journal of Physics: Conference Series (Vol. 1840, No. 1, p. 012026). IOP Publishing.

20 Zagorodna N., Skorenkyy Y., Kunanets N., Baran I., Stadnyk M (2022), Augmented Reality Enhanced Learning Tools Development for Cybersecurity Major, CEUR Workshop Proceedings, 3309 , pp. 25-32.