Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи виявлення аномального трафіку в IoT

(тема)

Виконав:

студент II курсу, групи КСМзм-21-1

Оборін О.О.
(прізвище, ініціали)

Спеціальність

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

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

Освітня програма

Комп'ютерні системи та мережі
(повна назва освітньої програми)

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

Допускається до захисту

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

2022 р.

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

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

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

Освітня програма Комп'ютерні системи та мережі
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Оборіну Олександру Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи виявлення аномального трафіку в IoT

затверджена наказом по університету від "24" жовтня 2022 р. № 178 Стз

2. Термін подання студентом роботи до екзаменаційної комісії 13 грудня 2022 р.

3. Вхідні дані до роботи

IoT

туманні обчислення

автоенкодер

4. Перелік питань, що потрібно опрацювати у роботі

Інтернет речей

Інформаційна безпека в IoT

Реалізація методу виявлення аномального трафіку в IoT

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 16 слайдів

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання. Аналіз літератури	25.10.2022–04.11.2022	
2	Огляд існуючих методів та алгоритмів.	05.11.2022–11.11.2022	
3	Аналіз існуючих пристроїв	12.11.2022–19.11.2022	
4	Розробка та реалізація методу	20.11.2022–29.11.2022	
5	Моделювання	30.11.2022–03.12.2022	
6	Отримання результатів	04.12.2022–06.12.2022	
7	Оформлення ПЗ	07.12.2022–12.12.2022	

Дата видачі завдання 24 жовтня 2022 р.

Студент _____
(підпис)

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

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 60 с., 12 рис., 1 дод., 20 джерел.

БЕЗПРОВІДНА СЕНСОРНА МЕРЕЖА, ІОТ, СИСТЕМА ВИЯВЛЕННЯ АТАК, АНОМАЛІЯ, ТРАФІК.

Метою кваліфікаційної роботи є аналіз методів виявлення аномального трафіку в IoT з урахуванням обмежень на споживчі ресурси.

У ході виконання кваліфікаційної роботи проведено аналіз методів виявлення аномального трафіку в IoT. Запропоновано метод детектування аномальної поведінки сенсорних пристроїв, що спирається на глибокі автоенкодери окремо для кожного пристрою, навченого на статистичних функціях, вилучених з незараженого трафіку

ABSTRACT

Master’s thesis: 60 pages, 12 figures, 1 appendices, 20 sources.

WIRELESS SENSOR MERIGE, IOT, ATTACK DETECTION SYSTEM, ANOMALITY, TRAFFIC.

The major goal of this thesis is to develop a method of data processing in a specialized computing device based on identifiers.

In order to overview of modern methods, algorithms and technical solutions of data processing based on their identifiers was carried out; analysis of data processing algorithms and classification of message groups of the target set based on identifiers. A method of creating data processing based on identifiers was also developed. A structural diagram of a specialized computing device for data processing based on identifiers was developed.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

IOT – Інтернет речей

БСМ – безпровідна сенсорна мережа

КВ – комутаційній вузол

СВА – система виявлення атак

СП – сенсорний пристрій

ВСТУП

Однією з найперспективніших технологій проектування систем автоматизації управління, моніторингу, контролю вважаються бездротовими сенсорні мережі (Wireless Sensor Networks, WSN) в силу таких їх переваг: мобільність, самоорганізація, швидке розгортання, створення тимчасової мережі біля, де прокладання звичайних кабелів ускладнена.

Безпровідна сенсорна мережа (БСМ) утворюється безліччю сенсорних пристроїв (СП) та комутаційних вузлів (КВ), з'єднаних за допомогою радіоканалу [9]. Джерелами даних у сенсорній мережі є датчики (сенсори), реєструючі дані (вимірювання) про довкілля – температурі, вологості, освітленні та ін. Взаємодіючи між собою та з КВ, наприклад маршрутизаторами, датчики створюють розподілену, самоорганізовану систему збору, обробки та передачі зареєстрованих даних.

Розподіл характеризується розосередженістю пристроїв БСМ в просторі. Самоорганізація означає, що пристрої БСМ здатні самостійно ідентифікувати один одного і у разі втрати будь-яких вузлів самостійно знаходити нові маршрути передачі.

Основна особливість БСМ полягає у відмові від безпосередньої участі людини в їхню роботу [15, 17]. Технології побудови вимірювальних систем на основі БСМ розвиваються вченими та інженерами не одне десятиліття. БСМ складаються з простих і компактні пристрої для обміну невеликими повідомленнями. До таких пристроїв належать, наприклад, пожежні датчики.

Черговим витком розвитку БСМ став інтернет речей (Internet of Things, IoT), що є готовими інтелектуальними системами – розумні системи (smart systems). «Розумні» функції інтернету речей багато в чому визначаються прикладними завданнями, але загалом IoT надає широкий набір послуг: екологічний контроль територій, охорона об'єктів, технологічний процес, моніторинг пацієнтів, «розумний дім» та багато іншого.

Метою кваліфікаційної роботи є аналіз методів виявлення аномального трафіку в IoT з урахуванням обмежень на споживчі ресурси.

Об’єкт дослідження: побудовані за ієрархічним принципом взаємодії вузлів мережі IoT.

Завдання:

- провести аналіз багаторівневої моделі архітектури IoT;
- провести аналіз методів виявлення аномального трафіку мереж IoT, що базуються на моделях глибокого навчання та системі метрик;
- розробити метод виявлення детектування аномальної поведінки сенсорних мереж;
- провести імітаційне моделювання функціонування мережі IoT з урахуванням розглянутих методів виявлення аномального трафіку.

1 ІНТЕРНЕТ РЕЧЕЙ (IOT)

Мережі інтернету речей характеризуються складною динамічною структурою, містять пристрої з різним програмним та апаратним забезпеченням. Поява інтелектуальних мобільних пристроїв, що використовують батареї в6 як джерело живлення, актуалізувало завдання своєчасного виявлення аномального трафіку, що витрачає енергію IoT. Під аномальним будемо розуміти мережевий трафік, який містить шкідливе програмне забезпечення, що реалізує атакуючий вплив на вузли IoT [2]. Своєчасне виявлення аномального трафіку сприяє збереженню терміну служби мережі інтернету речей і, відповідно, виконання IoT послуг.

Інтернет речей – це екосистема, що є програмно-апаратним рішенням, що забезпечує автономну реалізацію інформаційних процесів збору, обробки та передачі даних в інтересах, що надається користувача послуги. Екосистему IoT утворюють:

- сенсорні та виконавчі пристрої з функціями вимірювань та прийому-передачі даних,
- телекомунікаційна інфраструктура, що забезпечує транспортування даних,
- сервери, що виконують функції обробки даних,
- програмне забезпечення, що реалізує протоколи приймання-передачі, зберігання даних, методи аналізу отриманих даних та алгоритми прийняття рішень за наслідками аналізу.

Назва «екосистема» говорить про те, що елементи, що її утворюють. перебувають у закономірному взаємозв'язку друг з одним і створено умови їхнього спільного збалансованого та стійкого функціонування.

- системи інтернету речей складаються з вузлів наступного типу;
- сенсорні пристрої (СУ), що виконують функції вимірювання;

- виконавчі пристрої, що виконують функції впливу на контрольований об'єкт;

- вузли сполучення, що виконують роль агрегатора (концентратора), шлюзу або маршрутизатора в залежності від покладених на них функцій.

1.1 Сенсорні пристрої IoT

У типовій структурі сенсорного пристрою є один або кілька сенсорів, а також модулі для початкової обробки виміряних даних та взаємодія з іншими СП. На рисунку 1.1 представлена типова структура СП. Вимірник перетворює сигнали, що реєструються датчиком, з аналогового виду в цифровий (АЦП). Обчислювач відсіває непотрібні дані, корисні дані призводить до потрібного формату та записує їх у пам'ять СП. Приймач надає інтерфейс СП із зовнішнім середовищем і містить два буфери для прийому та передачі даних відповідно. У структуру СП також може бути інтегрована GPS-система. Робота СП забезпечується харчуванням від батареї. Час «життя» СП залежить від тривалості роботи елементів живлення [8]. Сенсорний пристрій утворює навколо себе зону чутливості з радіусом r, що визначає дальність дії датчика. Дальність дії датчика - це відстань, при якому вимірювач здатний реєструвати зміна вхідного сигналу.

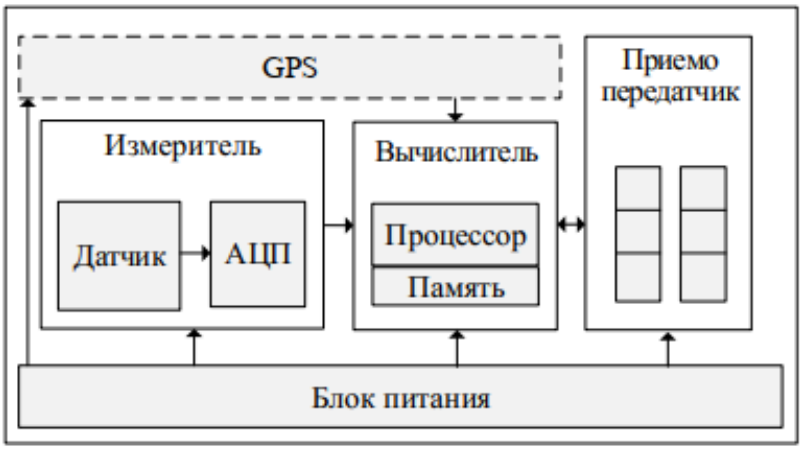

Рисунок 1.1 – Структурна схема СП

На базі СП будуються мережі моніторингу, наприклад, стану охоронюваного об'єкта, екологічної обстановки, території, що захищається і т.п. Мережі моніторингу зазвичай однорідні, тобто всі сенсорні вузли мають однаковими функціональними можливостями та енергоспоживанням. У медичних натільних сенсорних мережах (Wireless Body Area Network, WBAN) крім СУ, що вимірюють поточні медичні показники, використовуються виконавчі пристрої (рисунок 1.2), здатні впливати на об'єкт вимірів. Контролер прийняття рішення управляє виконавчим пристроєм, у структурі якого є активний елемент взаємодії з довкіллям. Як приклад можна привести пристрій до введення ліків пацієнту.

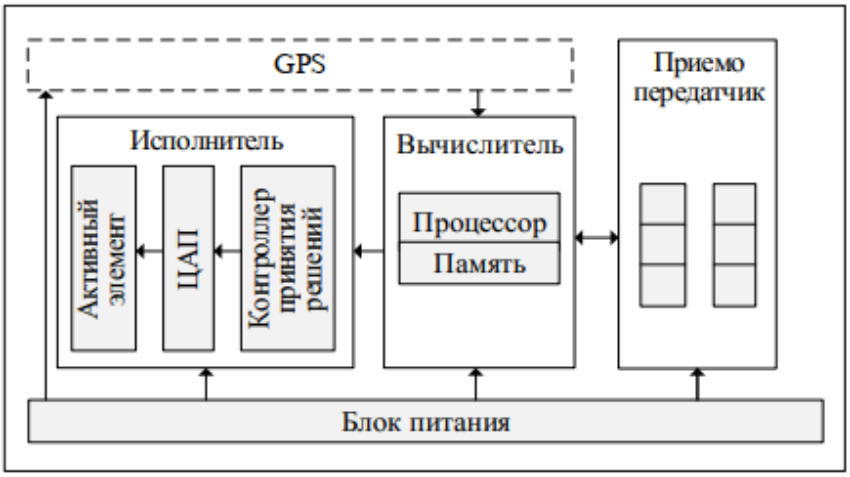

Рисунок 1.2 – Структурна схема виконавчого пристрою

У медичній натільній сенсорній мережі можуть бути різні пристрої типів, що виконують різноманітні функції, що володіють різним запасом енергії.

1.2 Телекомунікаційна структура IoT

Організація телекомунікаційної інфраструктури інтернету речей залежить від масштабу території і складності розв'язуваного завдання. На

відміну від мереж WiFi або WiMAX, час «життя» сенсорної мережі сильно залежить від споживання енергії сенсорними та виконавчими пристроями [16]. Споживання енергії відбувається за різних операцій: визначення маршруту доставки, передача та прийом даних, їх обробка тощо. В інтернеті речей системи передачі заощаджують енергію шляхом скорочення кількості операцій.

З метою економії енергії при передачі даних багатокрокове взаємодія є найпоширенішим варіантом в інтернеті речей та реалізується ієрархічною структурою (рисунок 1.3).

Рисунок 1.3 – Типи пристроїв і їх взаємозв'язок з фізичними речами

На нижньому рівні сенсорні пристрої об'єднані у кластери. СП відправляють зібрані дані головним вузлам. Головний вузол кластера агрегує дані від інших пристроїв свого кластера, формує пакети даних, надсилає їх найближчому маршрутизатору. Маршрутизатори утворюють топологію,

побудовану на принципі осередків-mesh, Топологія дозволяє вузлам ставати головним вузлом кластера і передавати пакети ролі комутатора. За кілька кроків (хопів) пакети дійдуть до шлюзу, що надає інтерфейс виходу до глобальної мережі (хмарні обчислення).

Виходячи з ієрархічної структури організації мереж IoT, можна виділити такі види взаємодії, що утворюють відкриті канали:

- «Device-to-Device» (D2D) – є взаємодія між СУ всередині сенсорної мережі;

- «Device-to-Server» (D2S) – являє собою модель взаємодії Клієнт/Сервер – на запит клієнта (клієнтської програми) сервер передає запитані дані від СП;

- «Server-to-Server», (S2S) – є серверна інфраструктура спільного використання даних із можливістю передачі їх назад пристроям, програм або користувачів.

Таким чином, на нижньому, фізичному рівні з сенсорних і виконавчих пристроїв утворюється бездротова сенсорна мережа, яка будується як сукупність кластерів. Відсутність обмеження на кількість кластерів, що дозволяє масштабувати розмір мережі під вимоги задач контролю та/або моніторингу територій. Визначення головного вузла у кластері організується з урахуванням балансу енергоспоживання. Існують різні алгоритми кластеризації, такі як LEACH, PEGASIS, TEEN та інші [22].

Вони об'єднані можливістю кожного вузла кластера стати головним вузлом Алгоритм LEACH (Low Energy Adaptive Cluster Hierarchy) передбачає стохастичний вибір головного вузла на основі енергетичних характеристик кожному новому часовому циклі (раунді). Головний вузол не може знову стати головним у наступному раунді. Вузли кластера обмінюються даними з головним вузлом як TDMA. Головний вузол створює розклад для зв'язку з вузлами кластеру. За алгоритмом PEGASIS (Power-Efficient Gathering Sensor Information) Systems) вузли організуються в ланцюжка. Дані від вузлів передаються по ланцюжку і тільки перші вузли ланцюжків передають

інформацію на вузол сполучення. Відповідно до алгоритму TEEN (Threshold-sensitive Energy Efficient Protocols) вузли передають дані головному вузлу, коли кількість накопичених даних досягло певного порога.

Головний вузол кластера протягом певного інтервалу часу виконує функції транзиту для СП – членів кластера, і тому витрачає більше енергії на прийом, обробку та передачу інформації. У зв'язку з цим, весь життєвий цикл сенсорної мережі ділять на раунди - інтервали часу, протягом 16 яких певний СП є головним вузлом кластера.

Для забезпечення баланс витрат енергії кожен раунд вибирається новий головний вузол. Фаза формування кластера та фази передачі даних становлять раунд. Головний вузол отримує дані від членів кластера та передає її іншим головним вузлам та/або шлюзу БСМ. У мережах зв'язку п'ятого покоління як такого шлюзу, наприклад, виступає базова станція. Після призначення головні вузли передають сенсорним пристроям сигнал RSS (Received Signal Strengt), формуючи їх у кластери. Потужність сигналу RSS дозволяє визначити кількість хопів від головного вузла сенсорного пристрою. Потім головні вузли широкомовною розсилкою передають на сенсорні пристрої інформацію про мережу – адресу, відстань тощо, і задають розклад передачі даних.

1.3 Протоколи IoT

Спеціальні протоколи IoT необхідні для забезпечення комунікації між речами та користувачами. Відомі такі протоколи згідно ділянкам взаємодії вузлів мережі IoT [22]:

- DDS: швидка шина для інтегрування інтелектуальних пристроїв (D2D);
- CoAP: протокол передачі інформації про стан вузла на сервер (D2S);
- MQTT: протокол для збору даних пристроїв та передачі їх серверам (D2S);

- XMPP: протокол для з'єднання пристроїв із користувачами, приватний випадок D2S-схеми, коли користувачі з'єднуються із серверами;

- STOMP: протокол обміну повідомленнями між пристроєм та сервером, реалізованими різними мовами та платформами (D2S);

- AMQP: система організація черг для з'єднання серверів між собою (S2S).

Перелічені протоколи є протоколами реального часу, існують десятки варіантів їх практичної реалізації. Усі ці протоколи володіють публікацією та підпискою, завдяки якій можуть об'єднувати тисячі пристроїв у мережу.

DDS (Data Distribution Service) – реалізує шаблон публікації-підписки для відправлення та прийому даних, зміни станів та команд, призначених кінцевим вузлам. Видавні вузли створюють інформацію за тематичним розділів – «topic», наприклад: температура, місце розташування, тиск тощо) і публікують ці шаблони як реляційної моделі даних.

Вузлам, підписаним на тематичні розділи протокол DDS, що реалізує пряму шинний зв'язок до бази даних, в якості транспортного протоколу використовується UDP. За допомогою DDS ефективно реалізується з'єднання численних пристроїв та користувачів. Користувачі та речі взаємодіють за методом "Запит-відповідь".

Анонімна модель взаємодії, за якої користувачеві немає потреби знати від якого СП отримано інформацію, а пристрої не знають своїх користувачів, що відрізняє DSS від інших протоколів. Така анонімність служить фундаментом масштабованості та самоорганізації БСМ. На цих властивостях побудований інтернет речей – при змінах у складі користувачів або пристроїв не треба переписувати сполучне програмне забезпечення.

CoAP (Constrained Application Protocol) призначений для взаємодії через Інтернет пристрої з обмеженими ресурсами. Основою для розробки CoAP послужив протокол HTTP, але на відміну текстового HTTP, CoAP є бінарним протоколом. Застосування бінарного протоколу знижує обсяг службових повідомлень, що зручно при малій потужності та низькому

споживання енергії IoT пристроями. Як транспортний протокол використовується UDP. CoAP організований у два шари: «Transactions» (транзакції) та "Request/Response" (Запит/Відповідь).

Шар "Transactions" обробляє обмін повідомлення між кінцевими точками. Шар «Request/Response» реалізує модель взаємодії Клієнт/Сервер, у якій СП зазвичай виконує функції сервера. На запит із клієнтської програми сенсорний пристрій починає передачу виміряних (зареєстрованих) даних. Протокол прикладного рівня CoAP може застосовуватись з будь-яким протоколом рівня: SMTP, FTP, HTTP, HTTPS.

Протокол MQTT (Message Queue Telemetry Transport) здійснює збір даних із пристроїв. Як транспорт використовується TCP. MQTT призначений для великих мереж з безліччю пристроїв, служить для забезпечення телеметрії та дистанційного моніторингу. До структури мережі, побудованої на MQTT, зазвичай входить сервер-видавець, сервер-брокер та один чи кілька клієнтів. Обмін повідомленнями здійснюється за шаблоном видавець-передплатник. Сервер-брокер управляє формуванням черг повідомлень та їх пріоритезацією. Таким чином, вся інформація, що передається, поділяється на різні канали, по одному каналу на кожного передплатника та видавця.

У протоколі MQTT передбачено вибір надійності обміну повідомленнями, який забезпечується трьома рівнями якості обслуговування (QoS, Quality of Service):

- QoS0 – повідомлення надсилається без підтвердження і лише один раз, найшвидший, але ненадійний;
- QoS1 – повідомлення передається до отримання підтвердження про доставку;
- QoS2 – повільний, але найнадійніший рівень якості. Реалізує стратегію одноразової доставки повідомлення. Застосовується чотириступінчаста процедура підтвердження доставки.

XMPP (Extensible Messaging and Presence Protocol) – протокол для обміну повідомленнями та інформацією про присутність. У XMPP

застосовується текстовий формат XML, протокол відкритий і добре розширюється. XMPP застосовується в невеликих мережах для адресації пристроїв, працює по TCP, не має високу швидкість, тому знайшов застосування в мережах IoT. На ділянці мережі між видавцем та брокером найчастіше застосовуються протоколи – CoAP, MQTT та XMPP.

Залежно від призначення та умов роботи мережі перевага надається конкретному протоколу. Протокол XMPP широко застосовується в системах управління освітленням та кліматом приміщеннях.

MQTT для моніторингу витоків та екологічного контролю на об'єктах небезпечного виробництва, контролю споживання енергії, управління світлом та інтелектуального садівництва, CoAP в системах розумного будинку. Для інтернету речей, що використовує обладнання різних платформ, рекомендується протокол STOMP.

STOMP (Simple Text Oriented Message Protocol) – простий протокол обміну текстові повідомлення. Протокол STOMP завдяки простоті та сумісності знайшов застосування різними мовами та платформами для зв'язку з брокером. Програмне забезпечення клієнта та сервера можуть бути написані на різних мовах програмування, протокол STOMP узгодить їхню взаємодію завдяки підтримці різних мов та бібліотек. Протокол STOMP забезпечує взаємодії сервера з брокером, на відміну від протоколу MQTT, що забезпечує зв'язок від брокера до СП та/або серверу.

Протокол AMQP розроблено для з'єднання серверів (S2S). Протокол AMQP (Advanced Message Queuing Protocol) – удосконалений протокол організації черги повідомлень. Протокол асинхронний, з надійною доставкою, обслуговує виключно черги. Працює поверх TCP. AMQP заснований на трьох поняттях:

- повідомлення – одиниця даних, що передаються, можуть забезпечуватися заголовками. Сервер не інтерпретує змісту повідомлення;
- брокер – приймач усіх повідомлень. Брокер розподіляє повідомлення в черги. Не зберігає повідомлення;

- черга – зберігає повідомлення до отримання передплатником.

Залежно від версії протоколу можуть бути різні механізми роботи брокера:

- fanout – повідомлення передається у всі причеплені до брокера черги;
- direct – повідомлення передається в чергу з ім'ям, що збігається з ID маршруту;
- topic – повідомлення передається у чергу на тему передплати.

Клієнт може отримувати інформацію з кількох черг, у яких зберігаються повідомлення та очікують доставки клієнту. За допомогою протоколу AMQP реалізуються аналітичні функції на серверній стороні протокол використовується при обміні діловими повідомленнями.

1.4 Архітектура IoT

Єдине уявлення про еталонну архітектуру інтернету речей відсутній. Це пов'язано з різноманітністю технологій, що застосовуються в організації IoT: хмари (cloud computing), тумани (Fog computing), граничні обчислення (Edge Computing), бездротові та провідні мережі, локальні та Світові інфокомунікації.

Проте існують кілька ініціатив стандартизації, таких як OpenFog Consortium, Edge Computing Consortium, mF2C H2020 EU проект. Вони пропонують свої референсні моделі архітектур IoT. Референсна архітектура є шаблоном для розробки різних систем інтернету речей, залежно від вимог замовника.

Аналіз літературних джерел показує, що класична (що з'явилася першою) є трирівнева архітектура IoT, що включає фізичний, мережевий та прикладний рівні. Пристрої інтернету речей збирають дані про події, що відбуваються в фізичному середовищі, що передаються на найближчий маршрутизатор або шлюз, який передає зібрані дані для подальшої їх

обробки та зберігання "хмара". Шлюзи та глобальна мережа, що представляє транспорт для зібраних даних у «хмару», реалізують функції мережевого рівня архітектури IoT. Прикладний рівень включає послуги обслуговування клієнтів, такі як «розумний дім», «розумний транспорт», «розумна енергетика» та багато інших. Функції прикладного рівня реалізуються технологіями хмарних обчислень.

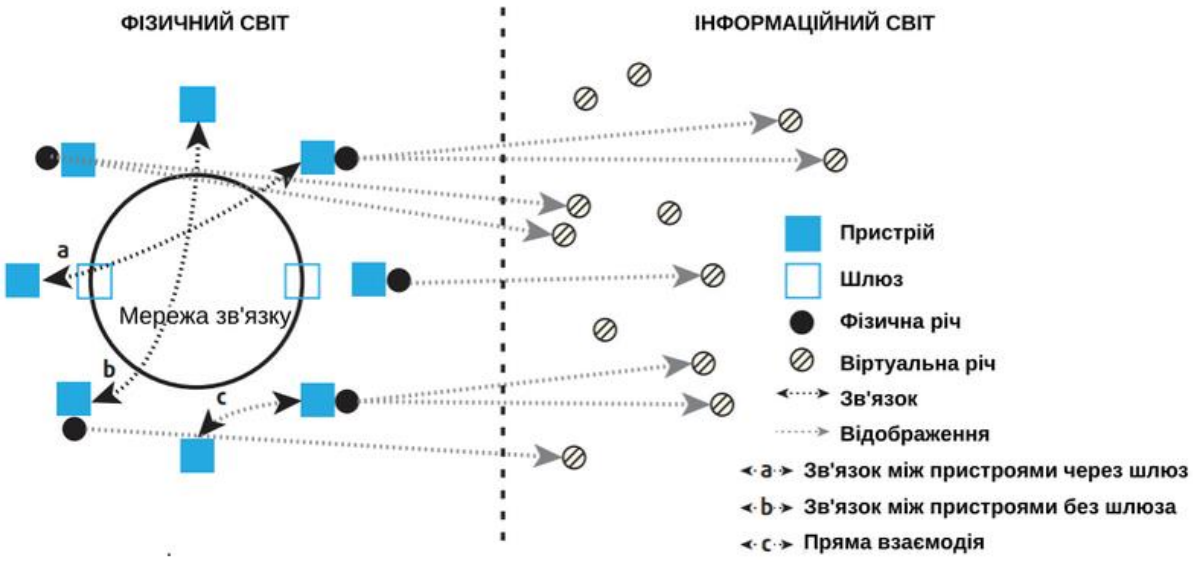

Рисунок 1.4 – Технічний огляд IoT

Трирівнева архітектура є варіантом реалізації «наскрізного» взаємодії пристроїв із додатками. Використання проміжних рівнів або підрівнів з функціями перетворення, передобробки чи інтерпретації даних у такій архітектурі не передбачено. З одного боку, це дозволяє формувати єдину інфраструктуру інтернету. речей, але з іншого – передача в хмару величезних масивів даних від підключених до IoT медичних установ, виробництв, звичайних користувачів і т.п., обробка цих даних, формування керуючих впливів та їх доставка за розумний час вимагають високої продуктивності від хмарних ресурсів та відповідної смуги пропускання мережної інфраструктури. Зі зростанням обсягів цифрових даних подібна архітектура

не дозволяє вирішувати завдання ефективного витрачання фізичних ресурсів інтернету речей – енергії, часу доставки, обсягів пам'яті.

У зв'язку з цим популярними стали архітектури з проміжними рівнями чи підрівнями, куди покладено функції, покликані вирішувати завдання ефективного витрачання фізичних ресурсів Інтернету речей.

До таких архітектурам належать:

- проміжні платформи;
- туманні обчислення;
- граничні обчислення.

В інтернеті речей хмарні технології поділяються на хмарні обчислення (Cloud Computing) та туманні обчислення (Fog Computing), їх концепції схожі, але різниця є. Потреба поділу продиктована зростаючою кількістю «розумних» речей, їх користувачів та даних, які вони генерують і які потрібно зберігати та обробляти в інтернеті речей.

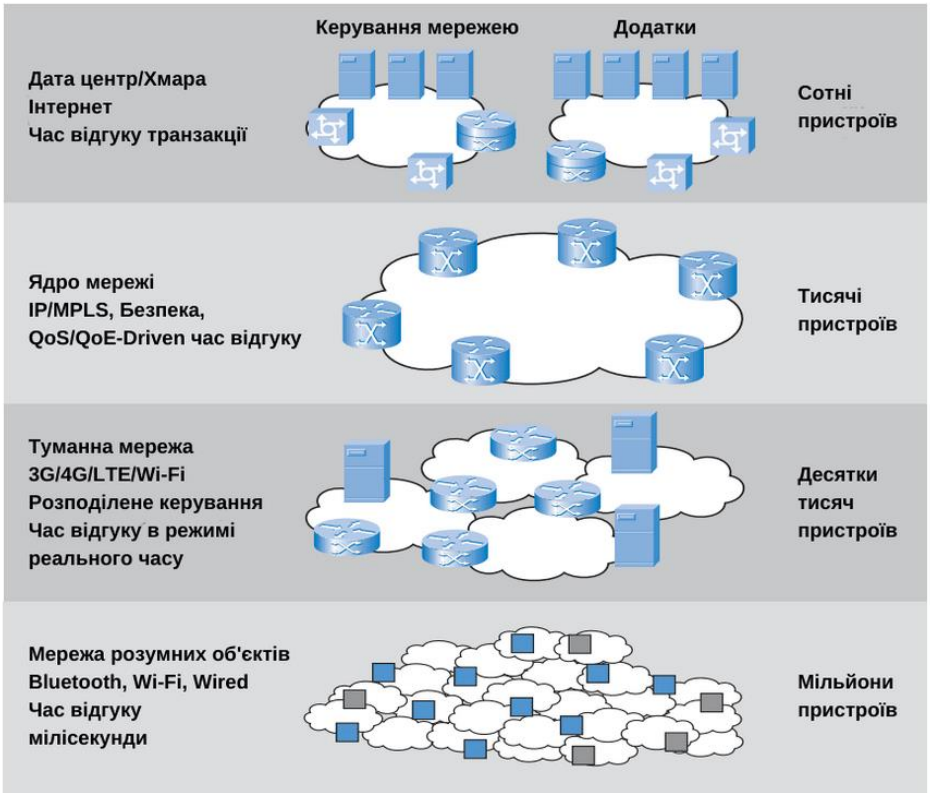

Рисунок 1.5 – Туманні обчислення

У хмарних обчисленнях обробка відбувається у віддалених центрах обробці даних (ЦОД), тому функціональні можливості та час затримки вищі, а безпека нижче. Хмарна архітектура централізована, масштабована. Зазвичай хмара складається з кількох великих серверів. У туманних обчисленнях обробка та зберігання даних відбувається ближче до користувачеві, наприклад, на сервері локальної мережі.

Туманна архітектура (рисунок 1.5) децентралізована, може включати мільйони дрібних вузлів. Обчислення на граничному сегменті призводять до миттєвого відгуку та мінімальним затримкам. Обробка поруч із джерелом даних тому час затримки нижче, відгук миттєвий. Основна відмінність між туманними та хмарними обчисленнями полягає в тому, що хмара є централізованою системою, а туман є розподіленою децентралізованою інфраструктурою.

1.5 Архітектура IoT з використанням проміжних платформ

Проміжна платформа в архітектурі IoT (рисунок 1.7) є своєрідним програмним інтегратором безлічі технологій, що забезпечують функціонування IoT-пристроїв в єдиній екосистемі, є інтернет речей.

У функції платформи, як правило, входять:

- підключення та нормалізація: забезпечення зв'язку користувачів з IoTпристроями, підтримка протоколів інформаційного обміну, безпечний обмін даними, очищення даних від шумів та ідентифікація пристроїв. Реалізується програмними агентами та бібліотеками;

- управління пристроями: управління станом підключених IoT пристроїв, передача на них команд управління, оновлення програмного забезпечення забезпечення IoT-пристроїв тощо. Реалізується програмним забезпеченням IoT пристроїв та інструментами для ручного керування IoT-пристроями;

- обробка та реакція: інструменти для налаштування сценаріїв

поведінки екосистеми Інтернету речей. Наприклад, відправляти sms-повідомлення користувачеві при настанні певних подій віддавати команди виконавчих пристроїв для взаємодії з навколишнім середовищем. Реалізується обробниками правил;

- візуалізація: інструментарій налаштування графічного відображення одержуваних даних для зручності сприйняття та аналізу даних користувачем (графіки, діаграми та інші графічні елементи). Реалізується спеціальним програмне забезпечення графічного відображення даних від IoT-пристроїв;

- аналітика: інструменти для створення алгоритмів, завдяки яким можна створювати прогнози, наприклад, передиктивний ремонт обладнання промисловий інтернет речей. Реалізується алгоритмами машинного навчання;

- додаткові інструменти: інструменти для розробки та управління додатками, п також самі додатки.

- зовнішній інтерфейс: механізм взаємодії із зовнішніми системами, наприклад, для інтеграції з провайдерами послуг, які отримують дані з платформи, та на їх основі надають послуги.

Таким чином, проміжна платформа інтернету речей є своєрідною інфраструктурою програм, що реалізують екосистему IoT.

1.6 Архітектура IoT з використанням туманних обчислень

Архітектура систем туманних обчислень передбачає децентралізовану обробку даних на граничному сегменті мережі, тим самим розвантажуючи хмару від рутинної роботи. При цьому зберігаються усі технологічні переваги такі як віртуалізація, контейнеризація, керованість тощо, але обробка даних здійснюється близько до джерела інформації, що дозволяє створювати високопродуктивні системи IoT.

Таким чином, центральну хмару утворюють тисячі хмарних ЦОД, що надають ресурси для виконання складних програм IoT. на На наступному

рівні знаходяться десятки тисяч розподілених керуючих ЦОД, в яких міститься "інтелект" Fog Computing ("Обробка туману"), а на На останньому рівні розташовані мільйони розумних пристроїв.

Для обробки в «тумані» застосовуються ті самі методи аналізу даних, що і для централізованої хмари, але спеціалізовані під конкретні бізнес-завдання. Можна сказати, що туманні обчислення необхідні для спеціалізованої обробки даних у реальному масштабі часу. Основні функції рівня туманних обчислень – це отримання запитів користувачів, збір та аналіз даних, необхідних для виконання цих запитів, та відповідно, надання відповідей. При цьому має забезпечуватись необхідний рівень безпеки при зборі та передачі даних.

Туманні обчислення відносять до найважливіших технологічних тенденцій останніх років. У 2015 р. створено міжнародний консорціум OpenFog Consortium, об'єднує компанії та академічні інститути у сфері високих технологій, який займається стандартизацією та просуванням туманних обчислень. У 2017 р. консорціумом запропоновано еталонну архітектуру туманних обчислень – OpenFog Reference Architecture (RA), прийнятої у 2018 р. як офіційний стандарт. Стандарт регулює використання еталонної архітектури як багатоцільової технологічної платформи.

Архітектура дозволяє обробляти величезні масиви даних для додатків інтернету речей, штучного інтелекту та інших сучасних технологій. Інфраструктура туманних обчислень є децентралізованою архітектурою, ресурси якої розподілені від "речей" до "туманів". Розподіл ресурсів слугує меті прискорення всіх процесів, за рахунок наближення обчислень до користувачів.

Архітектура OpenFog RA базується на таких основних технологічних принципах: безпека, масштабованість, відкритість, автономність, програмованість, надійність, адаптивність, ієрархічність. Вимоги до безпеки Fog-обчислень формуються з урахуванням конкретних застосувань, типів та місця розміщення вузлів IoT. При проектування рішень із застосуванням

еталонної архітектури важливо виконати вимоги щодо захисту інфраструктури. Це конфіденційність, анонімність, цілісність, довіра, атестація, перевірка та вимірювання [9].

Виконання вимог гарантує безпечне середовище обчислень. Після включення живлення ланцюжок довіри поширюється від компонентів довіреного апаратного забезпечення до інших апаратних та програмних компонентів. Вузли, розміщені на межі туману, керують доступом та здійснюють шифрування даних. Вони забезпечують контекстуальну цілісність та ізоляцію даних [12]. Масштабованість передбачає нарощування продуктивності в мережі у разі збільшення кількості користувачів, програм або пристроїв. Це дозволяє змінювати розміри мережі, зменшувати чи збільшувати кількість вузлів на рівнях, додавати засоби зберігання або програми, що призводить до розширення функціональності.

Відкритість сприяє збільшенню кількості постачальників, прискоренню розвитку систем, розроблення та застосування різних платформ, протоколів та додатків. Цей принцип забезпечує розміщення вузлів на будь-якому ієрархічному рівні туманних мереж, додавання різноманітних вузлів та додатків, формування програмно-конфігурованих вузлів.

Автономність визначається здатністю вузлів туманних мереж, у разі збоїв, відмов чи відсутності зовнішніх сервісів, продовжувати виконання заданих функцій. Цей принцип в архітектурі OpenFog RA поширюється на всі елементи Fog-обчислень.

Програмованість дозволяє оптимізувати використання ресурсів, в тому числі використовувати контейнеризацію, формувати інфраструктуру, налаштовуватися під конкретні вимоги розгортання додатків. Програмування дозволяє логічно ізолювати виконавчі середовища різних користувачів. Вона автоматизує оновлення засобів безпеки і дозволяє швидше реагувати на загрози, що виникають. Надійність є невід'ємним атрибутом архітектури IoT, сприяючи зменшенню часу простою, безперервності бізнес-процесів спрощення сервісного обслуговування.

Особливо це важливо для туманних інфраструктур на промисловому чи небезпечному виробництві, на важкодоступні об'єкти або в несприятливому середовищі. Адаптивність в архітектурі OpenFog RA означає обробку даних поблизу джерела їх отримання для прискорення прийняття оперативного рішення.

Ієрархічність передбачає, що це компоненти туманних обчислень можуть бути представлені у вигляді логічної ієрархічної інфраструктури подібно до Інтернету речей: фізичний рівень (сенсори, виконавчі пристрої, відеокамери, смартфони, планшети), сенсорні мережі (Wi-Fi, ZeegBee та інші), мережевий рівень (шлюзи та глобальні мережі), прикладний рівень (додатки та послуги) [13, 15].

1.7 Архітектура систем граничних обчислень

Граничні обчислення (edge computing) – це різновид розподілених обчислень, у якій обробка інформації відбувається поруч із джерелом даних. Основна відмінність хмарних обчислень від граничних полягає в обробці та зберіганні даних у віддалених ЦОД.

Граничні обчислення відрізняються від локальних тим, що є частиною системи, що включає в себе інструменти для збору та аналізу статистики, централізованого управління та оновлення програм на пристроях. У ролі пристрою граничних обчислень можуть бути мікрокомп'ютери, наприклад Raspberry Pi, мобільні пристрої, персональні ноутбуки, розумні камери та інші пристрої, здатні виконувати додаток для обробки даних.

Граничні, як і туманні обчислення – це обчислення, які розташовані поруч із джерелом одержуваних даних. Відрізняються вони тим, що при туманних обчисленнях обробка здійснюється на пристроях, постійно підключених до мережі.

У граничних обчисленнях обробка здійснюється як на сенсорах, розумних пристроях – без передачі на шлюз, так і на рівні шлюзу та на

кластерах, що утворюють туманні обчислення. Наприклад, робота московських камер "Стрілка" повністю відповідає технології граничних обчислень. Первинна обробка даних відбувається «на стовпі», а проміжна обробка в відокремлених обчислювальних кластерах у різних відомств.

До переваг впровадження граничних обчислень в архітектуру класичного інтернету речей відносяться:

- зменшення обсягу переданого трафіку за рахунок попередньої обробки даних на самому пристрої та передачі тільки результуючої інформації. Обробка даних близько до їхнього джерела особливо ефективна при обробці потокового відео, стиснутого звуку або великої кількості зображень;

- зменшення часу відгуку та затримок, прискорення прийняття оперативне рішення;

- персональні дані не виходять із певного контуру;

- пристрої певний час можуть працювати незалежно – без доступу до центральних серверів, що підвищує стійкість до відмови системи. Від втрати даних при відмові граничних пристроїв захищає централізований збір результуючих даних.

Серед недоліків граничних обчислень виділяють такі:

- забезпечення гарантованої відмовостійкості для всіх edgeпристроїв;

- устрою гетерогенні, тобто. мають відмінні платформи та версії операційних систем, що потребує створення кількох версій сервісів;

- для керування великою кількістю пристроїв необхідна платформа, вирішальна технічні завдання граничних обчислень. Декілька великих компаній просувають на ринок інтернету речей свої проекти з відкритим вихідним кодом, завдяки чому зрілість технології граничних обчислень стрімко зростає.

Зокрема, консорціум Linux Foundation Edge (LF EDGE), компанія IBM, яка застосовує певні принципи при проектуванні рішень граничних обчислень та є одним з лідерів у галузі хмарних технологій.

Референсну архітектуру граничних обчислень складають пристрої, сервера або шлюзи, хмара та гібридна хмара в ЦОД. При використанні референсної архітектури на пристрої IoT, залежно від розв'язуваних завдань, можуть бути такі сервіси/дані: модель обробки даних, сервіс аналітики, інтерфейс користувача, база даних та будь-які інші сервіси.

Оскільки моделі машинного навчання запускаються на десятках або тисячах пристроїв, то у хмарі граничних обчислень реалізується навчання моделей, обробка статистичних даних, графічне відображення зведеної інформації. На рівні шлюзів обробка виконується на мікрокластери мікро-ЦОД за підтримки кластерних технологій.

Для синхронізації великої кількості пристроїв, програм, управління різноманітними сервісами у різних кластерах використовується фреймворк, який реалізує функції керування різнорідними розподіленими обчисленнями.

Застосування такого фреймворку демонструє переваги граничних обчислень перед туманними обчисленнями. У фреймворку, крім центральної частини з управління моделями та сервісами, кожному з рівнів присутні агенти. Агенти необхідні для контролю та управління сервісами на окремих пристроях та кластерах.

Граничні обчислення знайшли широке застосування у багатьох областях, незважаючи на те, що спрямовані на певне коло завдань.

Нині дедалі більше сценаріїв застосування граничних обчислень можна побачити в секторі фінансів, торгівлі, охороні здоров'я, страхуванні та виробництві.

В інтернеті речей склалася тенденція об'єднання розрізнених мереж для поєднання даних, процесів, взаємодій.

Експерти прогнозують швидкий перехід до Всеосяжного Інтернету (Internet of Everything, IoE). Він послужить меті об'єднанню «всього» для надання різноманітних сервісів та послуг.

1.8 Характеристики IoT

До характеристик мереж інтернету речей, що відображають їх основні властивості належать такі. Зона відповідальності. Під зоною відповідальності мережі інтернету речей слід розуміти територію або область простору, що покривається зонами чутливість сенсорів (датчиків). Сенсорним полем називають простір, покривається сенсорною мережею [5, 7].

Гетерогенність. Ця характеристика відбиває спосіб побудови мережі інтернету речей з позиції набору інфокомунікаційних, що використовуються технологій. Загалом, при побудові IoT допускається використання кількох технологій бездротового зв'язку, різні за функціональністю вузли комутації, які мають забезпечувати можливість спільного використання у межах однієї мережі. У гетерогенній мережі всі або частина вузлів можуть бути рухомими, мати різні швидкісні характеристики, стандарти зв'язку та протоколи обміну даними.

Зв'язність. Це потенційна можливість реалізації доставки даних від будь-якого активного сенсорного пристрою в «хмару», що може характеризуватись ймовірністю зв'язності. Доставка даних здійснюється по логічному каналу, що зв'язує сенсорний пристрій зі шлюзом адресата для подальшої передачі щодо нього даних. Таких логічних каналів, зв'язуючих джерело та адресат, може бути декілька.

Оскільки маршрут може проходити через кілька вузлів, які виконують функції транзиту, то ймовірність наявності маршруту між двома вузлами буде залежати від довжини маршруту (числа транзитів) та ймовірностей зв'язності цих вузлів. Таким чином, зв'язність є мірою взаємозв'язку між будь-яким сенсорним вузлом та центром обробки та зберігання даних. Цей захід може бути описано ймовірністю існування маршруту для будь-якого вузла мережі.

Кластеризація. Це кількісна характеристика ступеня зв'язності вузлів, які є найближчими сусідами вузла, що розглядається. Для аналізованого

вузла коефіцієнт кластеризації виражається через ймовірність, що два вузли, які є найближчими сусідами вузли самі є один одному найближчими сусідами.

Трафік інтернет речей. Під трафіком інтернету речей розуміють характеристики потоку даних, що генеруються сенсорними вузлами мережі. Трафік, як правило, описується такими параметрами як: розмір пакета даних, інтенсивність, функція розподілу інтервалів часу між надходженнями пакетів даних [15].

В основі протоколів із встановленням логічного з'єднання лежить пакетне передавання даних. Операційні системи фрагментують пакети з потоку трафіку, що передається, піклуються про правильну послідовність їх прийому та знову об'єднують отримані пакети в потік трафіку. Розмір пакета та його форма визначаються стандартом мережі, пов'язані з апаратним забезпеченням мережі, що використовується топологією та типом середовища передачі даних. Загалом, размер38 пакета, прийнятий в інтернеті речей менше, ніж пакет локальної або глобальної мережі. Наприклад, IEEE Std 802.15.4-2011 визначає розмір пакета 133 байт.

У системі інтернету речей передаються два типи пакетів: керуючі (запити, квитанції, підтвердження, спеціальні команди) та інформаційні (дані з СП). Формат пакету включає два типи даних: службову інформацію та дані користувача, звані також корисним навантаженням. Службова інформація необхідна для доставки даних: адреси відправника та одержувача, коди виявлення помилок та інформація про черговість. Як правило, службова інформація міститься в заголовку та хвості пакета, а між ними розміщується поле даних. Керуючі пакети не містять поля даних.

Інтенсивність – це частота генерації пакетів даних сенсорними пристроями, пакетів/с, позначимо λ. Функція розподілу характеризує розподіл інтервалів часу між надходженнями пакетів даних Дослідження останніх півтора десятка років доводять, що мережевий трафік за своєю природою є самоподібним (self-similar) або фрактальним (Fractal).

Самоподібні процеси описуються властивостями, не характерними для класичних потоків теорії телетрафіку. До цих властивостей відносять:

- довготривалу залежність, яка визначена пульсуючою структурою процесу;

- повільно спадаючу дисперсію, яка свідчить про наявність сплесків, які неможливо згладити простим усередненням та дозволяє формалізувати поняття самоподібності в математичному значенні у вигляді розподілу з важкими хвостами.

2 ІНФОРМАЦІЙНА БЕЗПЕКА В ІОТ

Суть інтернету речей полягає в можливості взаємодії із реальним фізичним світом. Атака на сенсори, маніпуляція з даними, що надходять у програму, підміна команд управління, вплив безпосередньо на виконавчі механізми – успішні атаки на будь-які ланки цього ланцюжка можуть призвести до захоплення контролю за системою інтернету речей [1]. Під атакою на вузол IoT розумітимемо аномальний трафік, що містить шкідливий код та спрямований на виснаження енергії вузла.

Завдання забезпечення інформаційної безпеки в IoT-мережах можна умовно поділити на першочергові та другорядні. Першочергові завдання загальновідомі і включають забезпечення конфіденційності, цілісності, автентифікації та доступності даних. Технології для їх вирішення можуть відрізнятися від тих, що доступні для комп'ютерних мереж особливостей (характеристик) інтернету речей, наведених вище.

Другорядні цілі забезпечення безпеки включають такі поняття як свіжість даних, самоорганізація, тимчасова синхронізація, захищена локалізація.

Конфіденційність в IoT-мережах має на увазі необхідність захисту даних, що передаються з метою закрити до них доступ для потенційних зловмисників за допомогою різних механізмів, таких як контроль доступу, шифрування тощо [2].

Аутентифікація даних необхідна для підтвердження справжності їх походження або першоджерела, що дозволяє перевірити легітимність відправника та одержувача даних. Аутентифікація даних вирішується засобами одноключових у БСС та двоключових у «хмарі» систем забезпечення безпеки. В IoT-мережах забезпечення аутентифікації даних без постійної участі людини стає досить складним завданням.

Цілісність даних в IoT-мережах має на увазі використання механізмів забезпечення захисту даних від зміни під час їх транспортування або зберігання.

Доступність даних в IoT-мережах означає можливість роботи мережі та виконання покладених на неї функцій. Доступність даних сенсорної мережі може бути легко скомпрометована за допомогою атаки та виведення з ладу базової станції чи головного вузла кластера мережі [3]. Свіжість даних в IoT-мережах дозволяє визначити, що дані, отримані датчиком, відправлені через мережу вперше, а не є копією повторно надісланих повідомлень. Для визначення свіжості даних у пакет може бути впроваджена мітка часу, яка служить сигналом видалення пакета у разі перевищення time-out.

Самоорганізація IoT-мережі є фундаментальною характеристикою, що відрізняє їх від інфраструктурних мереж, і означає самовідновлення топології кластерних структур залежно від різноманітних ситуацій. В разі відсутності самоорганізації результат атакуючого впливу на неї може бути руйнівним.

Тимчасова синхронізація в IoT-мережах необхідна визначення загальної шкали часу для всіх вузлів мережі та їх локальних вбудованих тимчасових механізмів. Захищена локалізація не дозволяє зловмиснику маніпулювати інформацією про місцезнаходження, наприклад, за допомогою хибних повідомлень про інтенсивності сигналу або відтворення сигналів. Таким чином, при проектуванні систем інтернету речей особливе увага приділяється забезпеченню безпеки даних, в основному переданих бездротовими каналами.

2.1 Класифікація атак на IoT-мережі

Атакуючі впливи в IoT-мережах схожі з атаками на інфраструктурні мережі. Однак, БСМ використовують відкрите фізичне середовище, а також широкомовний зв'язок за допомогою радіоканалу, тому їх важче захистити [11]. На рисунку 2.1 наведено класифікацію атак на IoT-мережі.

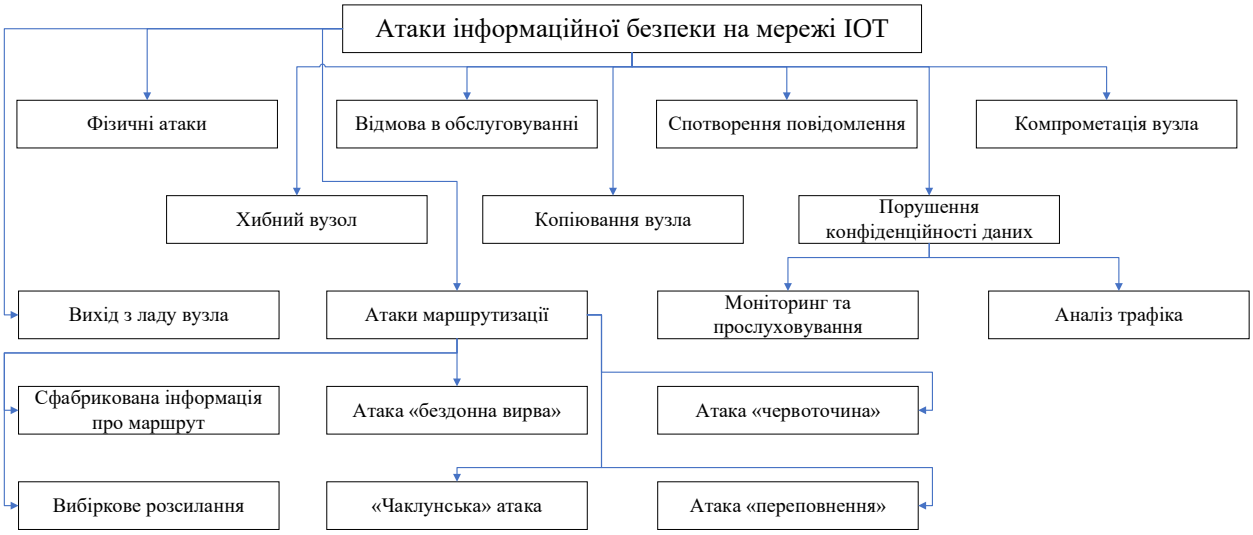

Рисунок 2.1 – Класифікація атак на мережі IoT

Фізичні атаки припускають зовнішній фізичний вплив на сенсорні вузли, які функціонують у природних умовах без постійного контролю (нагляду) із боку людини. Це єдиний вид атак, вплив яких позначається на вузли безповоротно.

Відмова в обслуговуванні пов'язана з проявом атак типу Distributed Denial of Service (DDoS – розподілена відмова в обслуговуванні), які націлені на виведення з ладу сервісів, що надаються сенсорною мережею. Аномальний трафік, що несе DDoS-атаку, по суті, є лавиною «порожніх» запитів, що надсилаються атакованому вузлу, в результаті якого відбувається заповнення пам'яті – зростає черга обслуговування. Ця атака багатоаспектна: від різкого зниження якості надання сервісів та ресурсів мережі до виведення її з ладу.

У реалізації DDoS-атаки роль атакуючого виконує так звана бот-мережа. Бот-мережа може стати джерелом лавиноподібного зростання «порожніх» запитів. Джерелами запитів є вузли, заражені шкідливими програмами. Програми безперервно посилають запити на вузол, що атакується, заважаючи його нормальній роботі.

Спотворення повідомлення є проявом атаки, що призводить до порушення цілісності даних, що передаються через мережу.

Компрометація вузла призводить до того, що атакований вузол здатний розсилати неправдиві дані. Якщо скомпрометовано вузол комутації – головний вузол кластера, маршрутизатор, шлюз, це призводить до порушення цілісності всієї сенсорної мережі.

Хибний вузол є наслідком впровадження в мережу вузла, що відправляє своїм сусідам шкідливий код у вигляді звичайних даних. Наявність помилкового вузла, що особливо виконує роль вузла комутації може призвести до виходу з ладу всієї сенсорної мережі.

Захоплення вузла є проявом атаки, орієнтованої на розкриття конфіденційної інформації, наприклад, ключів шифрування, що у свою черга може призвести до дискредитації мережі загалом.

Копіювання вузла мережі є атакою, при якій зловмисник виконує клонування захопленого вузла комутації. Дана атака реалізується після атаки «захоплення вузла»: ідентифікаційні дані, отримані в результаті захоплення вузла, використовуються для інтегрування в діючу мережу підготовлених вузлів-клонів. Таким чином, зловмисник отримує керування тими сегментами мережі, куди ці вузли-клони впроваджено.

Порушення конфіденційності даних включає атаки двох видів:

- моніторинг та прослуховування, внаслідок чого зловмисник перехоплює дані, що передаються. Реалізація цього виду атаки особливо загрожує при передачі команд управління;

- аналіз трафіку є атакою, націленою на знімання статистики щодо активності сенсорних вузлів. Різнопланова статистика утворює комунікаційний патерн, аналіз якого може дати зловмиснику інформацію, з урахуванням якої може плануватися атака. Зазначимо, що наявність Шифрування не рятує від аналізу трафіку.

Атаки маршрутизації, до них належать такі:

- змінена маршрутна інформація призводить до появи закільцьованих

маршрутів, маршрутів, некоректних з погляду ефективності передачі даних, маршрутів, побудованих через клоновані вузли, у результаті дані втрачаються чи збільшується час їх доставки;

- вибіркове розсилання спрямоване на зниження якості обслуговування мережі: скомпрометовані вузли запрограмовані на вибіркове видалення команд управління чи даних користувача;

- атака «бездонна вирва» спрямована на збір усього трафіку у кластері бездротовий сенсорної мережі докладно вирву.

- атака «червоточина» передбачає побудову спеціального маршруту між зараженими вузлами Si та Sj з метою транспортування перехоплених пакетів. Пакети захоплюються на вузлі Si і послідовно передаються від одного транзитного вузла інший до кінця маршруту – вузла Sj. Транзитні вузли, через які пройшли пакети, прийняті «червоточиною» вузла Sj, будуть хибно сприймати вузол Si як що знаходиться на відстані одного хопа. Атака цього типу вважаються важко детектованими;

- атака «переповнення» є широкомовною атакою, при реалізації якої в мережу вкидається маса необов'язкових повідомлень, обробка та передача яких сприяє скороченню терміну служби сенсорної мережі.

- атака «збірний пункт». Заражений вузол розсилає неправдиві відомості про себе щоб переконати інші вузли кластера, що він головний вузол;

- атаки викиду пакетів. Дані атаки особливо ефективні якщо скомпрометований головний вузол. Бувають двох видів – «чорна діра» та «сіра дірка». При атаці «чорна діра», заражений вузол стирає всі пакети, що приходять. При атаці «сіра діра» дані видаляються вибірково, що ускладнює виявлення атаки;

- атака Сивіли пов'язана з множинним уявленням скомпрометованого вузла під різними хибними ідентифікаторами в простір мережі. Атака насамперед націлена на порушення роботи механізмів маршрутизації, агрегації даних, розподіленого зберігання;

- атака зациклювання. Заражені вузли можуть знижувати пропускну здатність, втрачати дані та збільшувати енергоспоживання шляхом створення логічних кілець та зациклювання;

- Rush-атака. Скомпрометовані вузли надсилають велику кількість службових повідомлень маршрутизації, знижуючи пропускну спроможність, збільшуючи енергоспоживання та скорочуючи термін служби мережі інтернету речей.

2.2 Механізми забезпечення безпеки мережі IoT

Механізми забезпечення безпеки інфокомунікаційних мереж, та IoT-мереж, зокрема, сприяють детектуванню та запобіганню аномального трафіку. Умовно їх можна класифікувати як механізми високого та низького рівня (рисунок 2.2).

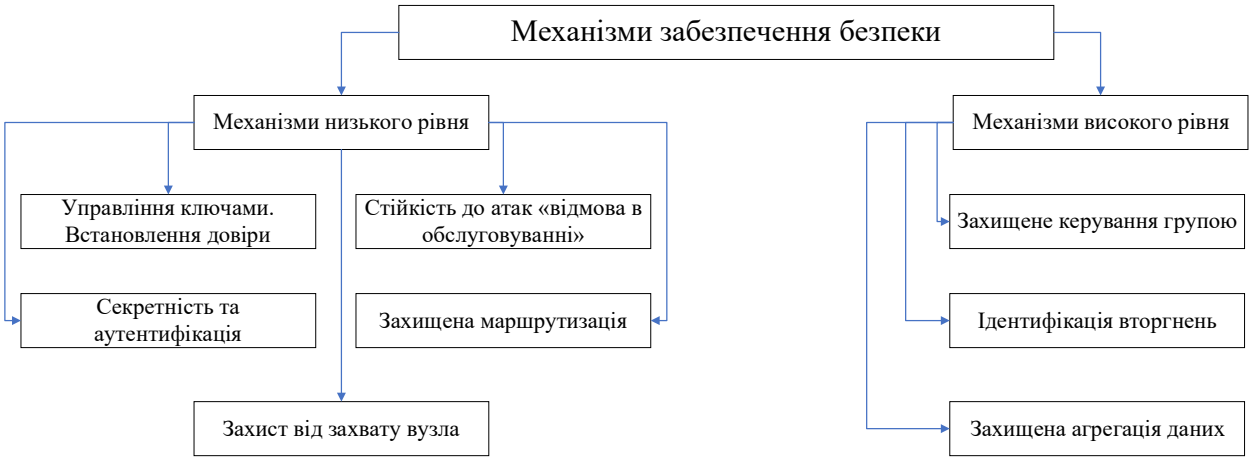

Рисунок 2.2 – Механізми забезпечення безпеки мережі IoT

Управління ключами та встановлення довіри. Обмежені ресурси бездротовий сенсорної мережі, особливо енергетичні, визначають вибір схеми керування ключами на користь симетричної схеми шифрування проти асиметричної схеми, яка потребує більше енергетичних ресурсів. Симетричний ключ встановлюється між кожною парою взаємодіючих вузлів,

включаючи головні вузли кластеру. Очевидним недоліком застосування симетричною схемою шифрування є можливість відновлення всього пулу ключів при компрометації значної кількості мережевих з наступним дешифруванням даних.

Секретність та автентифікація. Бездротові сенсорні мережі також як та будь-які комп'ютерні мережі потребують організації захисту від розмноження, прослуховування та модифікації пакетів даних. Стандартним захистом для комп'ютерними мережами є криптографічні методи, однак їх застосування у бездротових сенсорних мережах ускладнене специфікою пасивної технології передачі даних – технології при якій вузол, який прослуховує свого сусіда може не передавати дані, якщо сусідній вузол передає точно такі ж дані.

Типовим прикладом роботи такої технології є алгоритм LEACH, згідно з яким сенсорні вузли передають дані на головний вузол кластеру. Застосування криптографічних методів на канальному рівні з одного боку полегшує процедуру обміну ключами, але, з іншого сторони, транзитним вузлам простіше здійснювати перехоплення та модифікацію даних.

Стійкість до відмов у обслуговуванні. Відмова в обслуговуванні може виникнути внаслідок нестачі обчислювальних ресурсів, несправностей у програмному та апаратному забезпеченні, спеціально створених впливів зі сторони довкілля чи комплекс всіх перелічених вище причин. Припустимо, що зловмисник намагається зробити сенсорну мережу непрацездатною шляхом подачі потужного сигналу, здатного повністю заглушити комунікаційні функції сенсорних вузлів. Захист від подібної атаки може вирішуватися застосуванням технології розширеного спектра. Технологія заснована на реалізації методів, що сприяють розширенню діапазону частот сигналу, має намір більшого, ніж потрібно для організації передачі даних. Такий сигнал нагадує шум, який протистоїть навмисній інтерференції сигналу, відправляється зловмисником.

Захищена маршрутизація. Атаки, розглянуті вище, більшою ступеня спрямовані на протоколи маршрутизації. Використання хибної інформації про маршрути в IoT-мережі створює проблеми при організації доставки даних адресату. Захист мережі від атак компрометації маршрутної інформації вирішується інтегруванням нових схем аутентифікації та захищених протоколів маршрутизації у роботу IoT-мережі [14].

Захист від захвату вузла. Захоплення вузла зловмисником реалізується з метою заволодіння інформацією, що зберігається на ньому, перепрограмування вузла або клонування вузла. Поширеними технологіями протидії захоплення вузла є застосування антивандальних упаковок, програмне забезпечення для шифрування та приховування даних, хешування. Механізми забезпечення безпеки високого рівня включають такі процедури: Захищене керування групою.

Завдяки властивості самоорганізації та мобільності БСМ, кластери IoT-мережі не є стаціонарними – число сенсорів у кластерах з часом змінюється. Тому для керування кластерами сенсорної мережі необхідні спеціальні механізми організації процедури автентифікації для вузлів, що знову прибули в кластер, захищеного доступу сенсорних пристроїв до головного вузла кластера та захищених комунікацій усередині кластерів.

Ідентифікація вторгнень. Завдання ідентифікації вторгнень вирішується організацією постійного моніторингу IoT-мережі з веденням журналу обліку вторгнень та розпізнавання спроб нелегітимного проникнення в мережу з повідомленням користувачів про ці спроби. Захищена агрегація даних. Агрегація даних виконується на головних вузлах кластерів, які також зазнають атак з боку зловмисника та вимагають надійного захисту. Атаковані вузли агрегації використовуються як джерело поширення хибних даних. Захищена агрегація передбачає застосування процедури аутентифікації та захищених протоколів маршрутизації.

2.3 Методи виявлення атак в мережах IoT

Базовим засобом детектування аномального трафіку Інфокомунікаційною мережею є системи виявлення атак (СВА). СВА спочатку збирає дані про роботу мережі, потім аналізує їх і робить висновок наявності чи відсутності атаки. Аналізу піддаються відомості про передані по мережі пакетах даних, навантаженні, яке вони створюють при обробці на вузлах60 мережі, необхідних ресурсів пам'яті та продуктивності вузлів обробки, швидкість роботи додатків, статистика доступу до файлів і т.і. Функціонування СОА засноване на методах детектування атак, як правило, сигнатурних та/або поведінкових. Детектування атаки включає чотири стадії (рисунок 2.4), першою є – збір метрик, які беруть участь у встановленні факту атаки на вузли мережі.

Рисунок 2.4 – Схема процесу виявлення атаки

Періодичний знімання метрик, що характеризують роботу мережі, реалізують спеціалізовані програмні агенти СВА – мережні чи вузлові, які інтегруються у програмне забезпечення управління роботою IoT-мережі. У функції мережевих агентів входить збір метрик про пакети даних, що передаються в контрольовані ділянки мережі. У функції вузлових агентів входить збір метрик про події, що відбуваються в контрольованих вузлах мережі, наприклад, статистика про пакетів, що відправляються, приймаються і оброблюються. Допускається контроль вузла відразу з боку кількох вузлових агентів, кожен із яких може збирати свою групу метрик. Метрики, зібрані агентами, аналізуються засобами СВА. Детектування атаки реалізується із застосуванням сигнатурного або поведінкового підходу залежно від цього, якому рівні ієрархії IoT мережі функціонує СВА. Сигнатурний підхід до виявлення атак ґрунтується на створенні бази даних шаблонів (сигнатур) відомих атак. Роль шаблону може грати:

- ділянка контексту;
- семантичний вираз спеціальною мовою;
- формальна математична модель.

У зібраних агентами даних шукається шаблон, максимально наближений до зареєстрованих у базі даних сигнатур атак. При визначенні збігу встановлюється факт наявності атаки. Сигнатурний метод має низький кількістю хибних детектувань, проте він не здатний виявити нову атаку, яка відсутня в сигнатурних базах. В основі поведінкового підходу лежить модель штатного процесу функціонування мережі, яку можна назвати патерном мережі. Суть поведінкового підходу полягає в оцінці відхилення реального режиму роботи мережі від патерну штатної роботи мережі. Якщо відхилення перевищують рівень допустимого відхилення, то поведінка мережі сприймається як аномалія. До переваг поведінкового підходу належить можливість детектування нових чи невідомих атак. Як недоліки даних методів можна відзначити хибні спрацьовування при непередбачуваній мережній активності та складність створення точної моделі .

3 РЕАЛІЗАЦІЯ МЕТОДУ ВИЯВЛЕННЯ АНОМАЛЬНОГО ТРАФІКА МЕРЕЖІ IOT

3.1 Виявлення атак на рівні IoT-пристрою

У IoT-пристроях складно чи практично неможливо реалізувати рішення безпеки, наприклад, такі як шифрування, що вимагають значних обчислювальних ресурсів. У зв'язку з цим усі енерговитратні процедури вирішуються засобами систем виявлення атак, що переносяться на кордон середовища IoT (прикордонні маршрутизатори та шлюзи) та/або сервери.

Пропоноване рішення є першою лінією захисту всіх пристроїв IoT, що реалізується межі середовища IoT. У процесі функціонування для кожного пристрою Інтернету речей виникає послідовність подій, що характеризуються значеннями наборів метрик. У дискретні моменти часу реєструються значення метрик IoT-пристроїв, які можна розглядати як часові ряди.

У ролі патерну поведінки будь-якого вузла сенсорної мережі можуть виступати як навантажувальні характеристики, так і статистичні характеристики часового ряду, що відображає функціонування пристрою. Очевидно, що статистичні характеристики густини розподілу ймовірностей тимчасових затримок можуть бути паттерном поведінки сенсорний пристрій. По-перше, дана характеристика показує реальну навантаження сенсорного пристрою в часі, і по-друге, сам розподіл вже є патерном поведінки.

Дана пропозиція щодо побудови патерну виникла в результаті аналізу трафіку розумних пристроїв Наприклад, для розумної розетки Xiaomi ряд експериментів з передачі даних з розетки на смартфон (стан s_1) та зі смартфона на розетку (стан s_2) через канал Bluetooth дозволили апроксимувати функції розподілу часових інтервалів. Стану s_1 та s_2 відповідають активному режиму роботи розетки, за якого розетка приймає

або відправляє дані та команди управління у випадкові моменти часу, після чого перетворюється на фоновий режим – стан «сну». У фоновому режимі розетка не отримують запитів.

3.2 Виявлення атак на рівні мережних сегментів

Для виявлення атак (аномалій) пропонується мережний підхід, який використовує глибоке навчання. Суть підходу полягає в тому, що детектор аномалій будується на глибоких автоенкодерах – для кожного пристрою IoT свій.

Автоенкодер – це нейронна мережа, здатна стискати вхідні дані і на виході відновлювати їх. На рисунку 3.1 наведено архітектуру мережі. Функція стиснення гарантує, що нейронна мережа знаходить відповідні відносини від вхідних функцій, з яких відновлює вихідні дані на виході з прийнятною помилкою ε.

За архітектурою автоенкодер схожий на персептрон – нейронну мережу прямого поширення. Позначимо:

- безліч вхідних даних як X,
- приховане безліч (стислий простір) як H,
- безліч вихідних даних як Y. Тоді архітектура автоенкодера може бути записана так

$$\mathbf{X} \in \mathbf{R}^d = \mathbf{XH} \in \mathbf{R}^p = \mathbf{Y},\ d<p. \tag{3.1}$$

Мета автоенкодера – отримати на вихідному шарі відгук ε, найбільш близький до вхідного. Таким чином, автоенкодери являють собою моделі навчання без вчителя, тобто не потрібна навчальна вибірка.

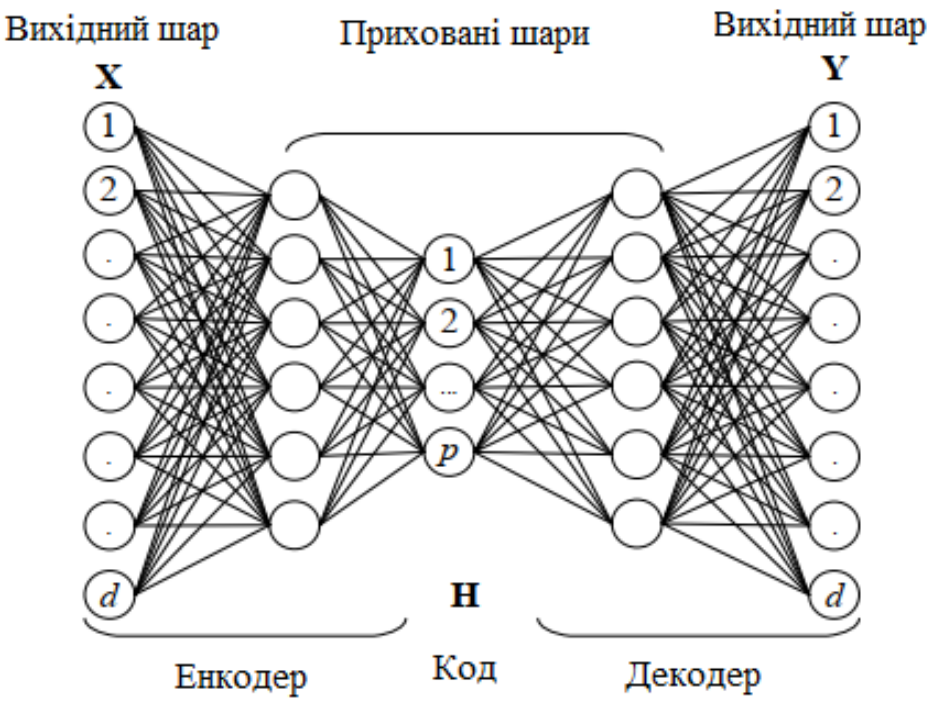

Рисунок 3.1 – Архітектура автоенкодера

Значення ваги і зсувів, як правило, встановлюються випадково та таким чином, після цих значень на кожній ітерації процесу навчання змінюються, наближаючись до мінімальної різниці між одержуваним і очікуваним результатом.

Загальною метрикою для оцінки якості стиснення функцій f, g у методі головних компонентом є пояснена дисперсія. Пояснювальна дисперсія показує частку у відсотках від величини вихідної дисперсії, з якою стиснуте подання дозволяє відновити вихідні дані із прийнятним рівнем помилки.

Теоретично можна побудувати будь-яку архітектуру автоенкодера, задаючи розмір коду та ємність енкодера та декодера на основі складності модельованого розподілу.

Переваги використання глибокого автоенкодера до виявлення аномальної поведінки пристроїв інтернету речей очевидні:

- завдяки функції стиснення, що реалізується методом головних компонентів, навчанні беруть участь лише основні статистичні характеристики розподілу;

- детектування аномальної поведінки сенсорних пристроїв відбувається в режимі онлайн – методу не потрібні значні обчислювальні ресурси та ресурси пам'яті на пристроях IoT, оскільки всі обчислення відбуваються в граничних обчисленнях.

Суть пропонованого підходу до виявлення нападів ботнету IoT, спирається на побудову автоенкодера для кожного пристрою, навченого на статистичних функціях, витягнутих із незараженого трафіку. При застосування до нових (можливо заражених) даних пристрою інтернету речей, виявлені аномалії, можуть вказувати на зараження пристрою.

Процес детектування включає такі основні етапи:

- збір даних;
- вилучення ознак;
- навчання детектора аномалій (автоенкодера);
- безперервний моніторинг.

На етапі збору даних необхідно зібрати необроблені дані мережевого трафіку (у форматі pcap), використовуючи дзеркало на маршрутизаторі, через який відбувається трафік. Щоб гарантувати, що навчальні дані є «чистими», звичайний трафік необхідно збирати відразу після підключення IoT пристрою до мережі. Макет для збору даних наведено на рисунку 3.2.

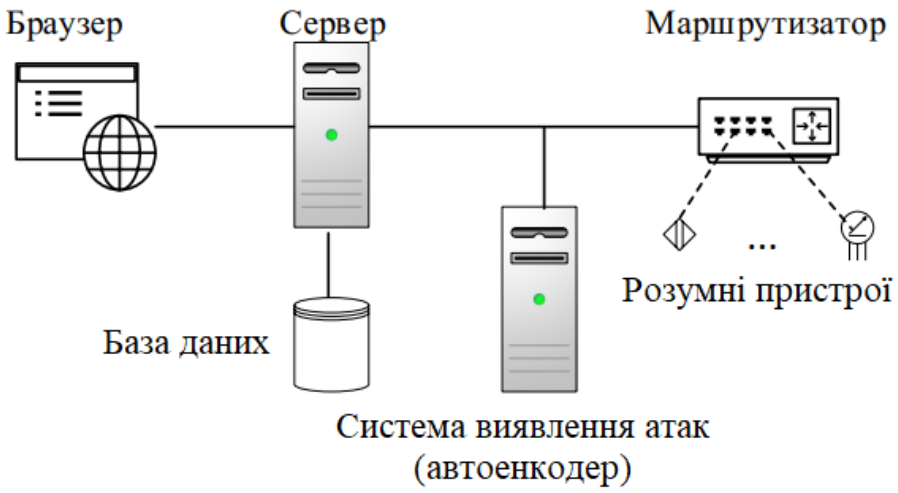

Рисунок 3.2 – Макет для збору даних про мережний трафік

Оптимізація параметрів та гіперпараметрів для кожного автоенкодера відбувається таким чином, що при вилученні значень статистичних функцій, які не відповідають еталонній (навченій) поведінці модель максимізує true positive rate (TPR) – успішне виявлення атаки та мінімізує false positive rate (FPR) – помилкові маркування незаражених даних. Для навчання та оптимізації використано два окремих набори даних, які містять статистичні величини та функції незараженого трафіку, у тому числі модель витягує шаблони нормальної поведінки. У постановочному експерименті брали участь дані від наступних пристроїв IoT:

веб-камери Samsung SNH 1011 N;

камери відеоспостереження SimpleHome XCS7-1003-WHT;

бездротовий радіоняні Philips B120N/10;

розумний метеостанції Xiaomi MiJia Miaomiaoce E-Ink.

датасети S_t і S_o були взяті з репозиторію UCI (UCI Machine Learning Repository) – найбільшого репозиторію реальних і модельних завдань машинного навчання.

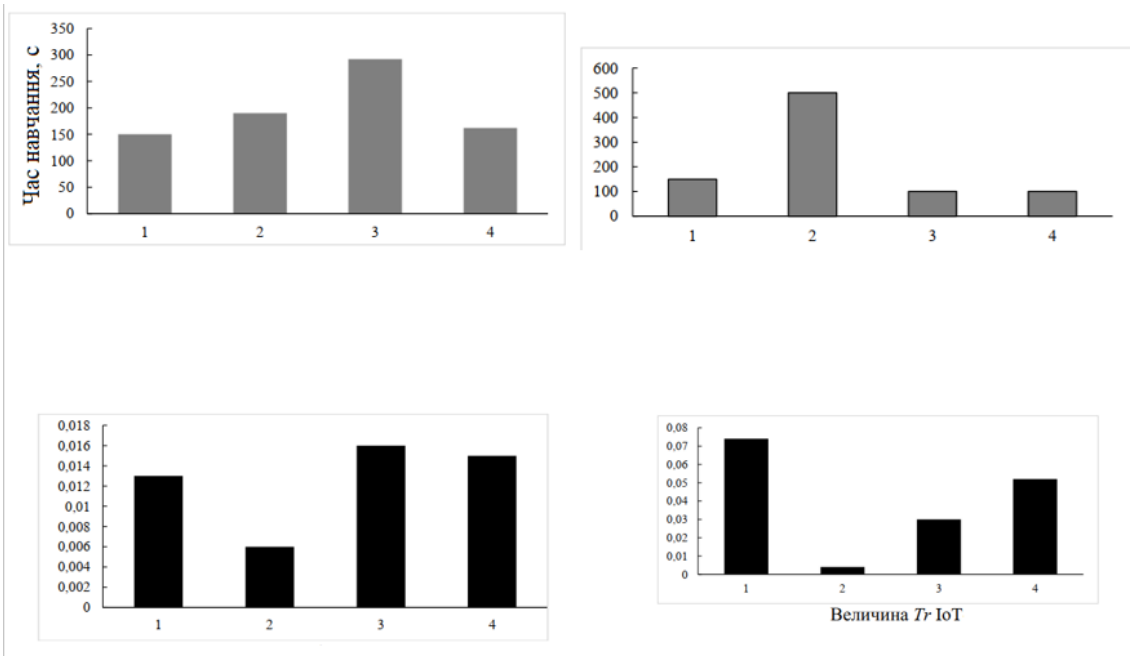

Рисунок 3.3 – Навчання автоенкодеру та аналіз результатів

Метод з використанням автоенкодерів в середньому на виявлення вимагав 0,17 с. Припускаючи, що СВА може автоматично виконати відключення зараженого пристрою від мережі, то запущені атаки можуть бути зупинено менше, ніж за секунду. Це суттєво нижче, ніж триває в середньому DDoS атака – 20-90 секунд.

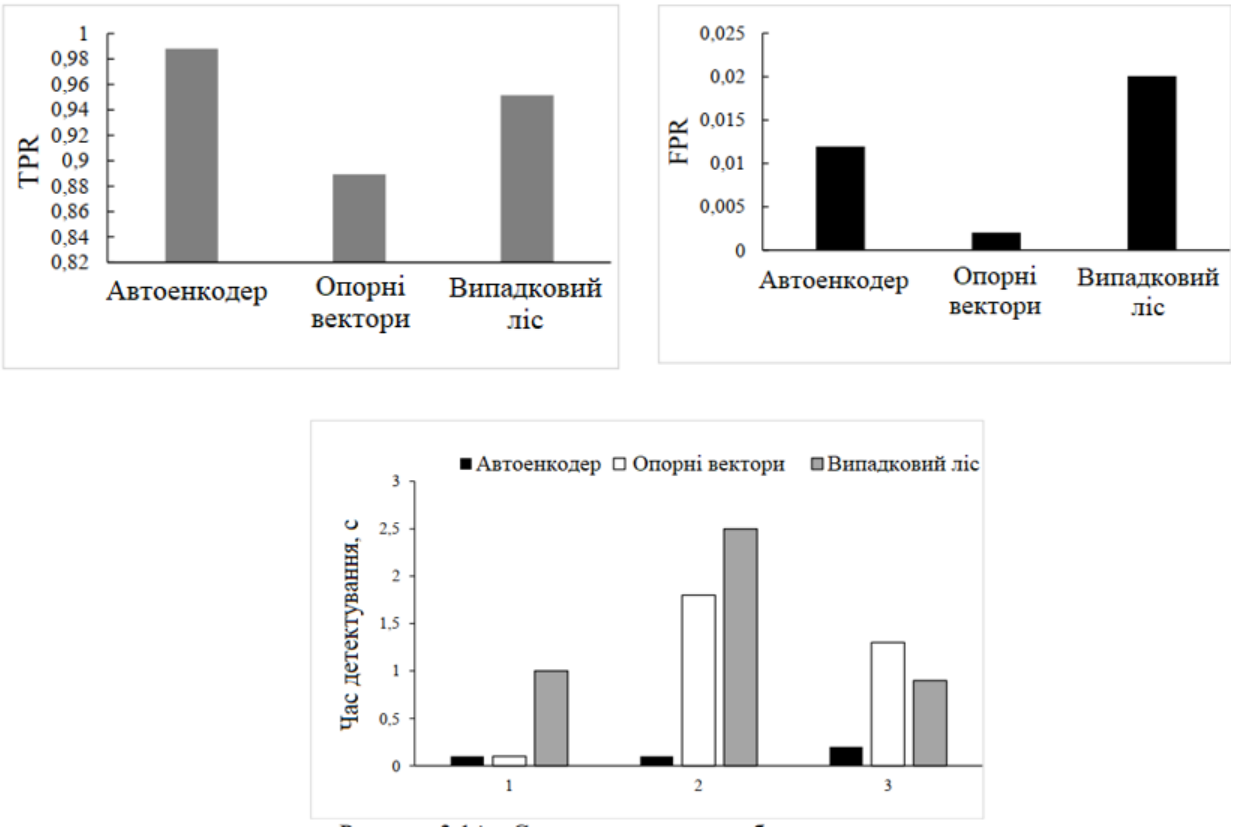

Рисунок 3.4 – Результати

Після етапу навчання автокодера та оптимізації, також були розглянуті два інших алгоритми (рисунок 3.4), які найчастіше використовуються для виявлення аномалій:

- машина опорних векторів (SVM – support vector machine);
- випадковий ліс (Random Forest).

Гіперпараметри цих алгоритмів також були оптимізовані. З точки зору TPR і FPR та часу виявлення атак, автоенкодери показали свою перевагу над методом опорних векторів та випадкового лісу.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи проведено аналіз методів виявлення аномального трафіку в IoT.

Запропоновано метод детектування аномальної поведінки сенсорних пристроїв, що спирається на глибокі автоенкодери окремо для кожного пристрою, навченого на статистичних функціях, вилучених з незараженого трафіку

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Белянинова Е.Г. Векторы вирусных атак завтрашнего дня / Е.Г. Белянинова,Е.В. Рыбанин, П.Ю. Богданов // Информационные технологии и системы: управление, экономика, транспорт, право. – 2019. – № 2 (34). – С. 219-222.

2. Белянинова Е.Г. Вирусные атаки в современном мире умных устройств / Е.Г. Белянинова, П.Ю. Богданов // Современные проблемы гидрометеорологии и мониторинга окружающей среды на пространстве СНГ: Сборник тезисов Международной научно-практической конференции, посвященной 90-летию РГГМУ. – 2020. – С. 578-579.

3. Белянинова Е.Г. Применение технологии NFC в системах контроля и управления доступом / Е.Г. Белянинова, П.Ю. Богданов // Информационные технологии в образовании. Сборник статей научно-практической конференции студентов, аспирантов и молодых ученых. Российский государственный гидрометеорологический университет. Санкт-Петербург – 2021. – С. 41-43.

4. Белянинова Е.Г. Исследование информации, собираемой о пользователях фитнес-браслетов / Е.Г. Белянинова, Е.В. Рыбанин, П.Ю. Богданов // Земля и Человек. Актуальные вопросы современного состояния окружающей среды. Сборник статей Межвузовской научно-практической конференции студентов, аспирантов и молодых ученых, посвященной празднованию 90-летия РГГМУ. – 2020. – С. 247-249.

5. Бескид П.П. Результаты исследований в области дистанционных методов обнаружения нефтяных загрязнений на водной поверхности, проводимых в РГГМУ / П.П. Бескид, П.Ю. Богданов, В.А. Миклуш, Т.М. Татарникова, Е.А. Чернецова, А.Д. Шишкин // Гидрометеорология и экология. – 2020. – № 60. – С. 371-391

6. Heikki Halttula, Harri Haapasalo, Risto Silvola. Managing data flows в

infrastructure projects - the lifecycle process model // Journal of Information Technology in Construction March 2020 (25) pp. 193-211 DOI: 10.36680/j.itcon.2020.012.

7. Богатырев В.А. Оценка надежности выполнения кластерами запросов реального времени / В.А. Богатырев, А.В. Богатырев, С.В. Богатырев // Известия вузов. Приборостроение. – 2014. – Т. 57. – № 4. – С. 46–48

8. Браницкий А.А. Анализ и классификация методов обнаружения сетевых атак / А.А. Браницкий, И.В. Котенко // Труды СПИИРАН. – 2016. – № 2(45). – С. 207- 244.

9. Варгаузин В.А. Радиосети для сбора данных от сенсоров, мониторинга и управления на основе стандарта IEEE 802.15.4 / В.А. Варгузин //ТелеМультиМедиа. – 2005. – №6. – С. 23-28.

10. Варгаузин В.А. Сетевая технология ZigBee / В.А. Варгаузин // ТелеМультиМедиа. – 2005. – С. 29-32.

11. Викулов, А.С. Анализ трафика в сети беспроводного доступа стандарта IEEE 802.11 / А.С. Викулов, А.И. Парамонов // Труды учебных заведений связи. – 2017. – Т. 3. – № 3. – С. 21-27.

12. Вишневский В.В. Энциклопедия Wi-Max. Путь к 4G / В.В. Вишневский, С.Л. Портной, И.В. Шахнович. – М.: Техносфера, 2009. – 471 с

13. Вишневский В. Mesh-сети стандарта IEEE 802.11s – технологи и реализация / В. Вишневский, Д. Лаконцев, А. Сафонов, С. Шпилев // Первая миля. – 2008. – № 2-3. – С. 26-31.

14. Гребнєв, В. В. 16-розрядні мікроконтролери з Flash-пам'яттю та функцією DSP фірми Infineon (сімейство XC166) / В.В. Гребнів. - М: РадіоСофт, 2008. – 528 с.

15. Восков Л.С. Web вещей – новый этап развития интернета вещей / Л.С. Восков, Н.А. Пилипенко // Качество. Инновации. Образование. – 2013. – № 2. – С. 44- 49.

16. Вітіска, Н. І. Мікропрограмний принцип відображення алгоритмів

вирішення складних фізичних завдань на машинах з реконфігурованою мультимікроконвеєрною обчислювальною структурою / Н. І. Вітіска, Д. В. Задорожній, В. І. Шмойлов // Вісник Воронезького113 державного технічного університету - 2009. - Т. 5. - № 5. - С. 196- 200.

17. Мальцев Г.М. Кодування повідомлень у системах радіоуправління без зворотного інформаційного каналу// Мальцев Г.М., Чернявський Є.В. Інформаційно-керуючі системи. 2011. № 4 (53). З. 60-65.

18. Гузєєв А.В. Формування розподілу ймовірностей появи окремих повідомлень джерела за статистичного кодування // T-Comm: Телекомунікації та транспорт. 2010. Т. 4. №6. С. 12-16.

19. Шоломов Л.А. Елементи теорії недовизначеної інформації // Прикладна дискретна математика. Додаток. 2009. № 2. С. 18-42

20. Коваленко А.А., Оборін О.О., Покора К.В. Метод виявлення аномального трафіку в IoT// Проблеми інформатизації : десята міжнародна науково-технічна конференція. Черкаси – Баку – Бельсько-Бяла – Харків, 2022, т.2, с.4.