Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«17» травня 2024 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи

галузь знань *12 Інформаційні технології*
(шифр і назва галузі знань)

спеціальність *125 Кібербезпека*
(код і назва спеціальності)

освітній ступень *магістр*

освітньо-наукова програма *Кібербезпека*
(назва освітньої програми)

на тему: «Метод захисту мереж IoT речей за допомогою штучного інтелекту»

Виконавець: студент ІІ курсу, групи КБм-21
_____ **Денис ЗИМБИЦЬКИЙ**
(підпис) (Ім'я, ПРІЗВИЩЕ)

	Ім'я, ПРІЗВИЩЕ	**Підпис**
Науковий керівник	Олександр ЛАПТЄВ	
Нормоконтроль	Юрій БАБЕНКО	

Київ 2024

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«17» листопада 2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності *125 Кібербезпека*
(код і назва спеціальності)

освітній ступень *магістр*

Здобувача(ки) КБм-21 (група) Зимбицького Дениса В'ячеславовича (прізвище ім'я по-батькові)

Тема кваліфікаційної роботи Метод захисту мереж IoT речей за допомогою штучного інтелекту

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 5 від 15.11.2023 р.

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень	Процес захисту мереж Інтернету речей (IoT) від потенційних загроз та атак.
Предмет досліджень	Метод захисту мереж IoT речей за допомогою штучного інтелекту
Мета	Вдосконалення методу захисту мереж IoT
Вихідні дані для проведення роботи	Метод захисту мереж IoT речей за допомогою штучного інтелекту.

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна удосконалено метод захисту електронного документообігу на підприємстві, який використовує аналіз потенційних загроз для мереж IoT на основі штучного інтелекту

Практична цінність покращення системи захисту даних мереж IoT

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Робота виконана у повному обсязі відповідно до теми.

5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Уточнення постановки задачі	17.11.2023 – 29.01.2024
Аналіз літературних джерел	30.01.2024 – 12.02.2024
Ознайомлення з сучасним станом досліджень в області інформаційної безпеки мереж інтернету речей	13.02.2024 – 21.02.2024
Розгляд методів забезпечення інформаційної безпеки мереж промислового Інтернету речей	22.02.2024 – 26.02.2024
Аналіз потенційних загроз для мереж Інтернету речей	27.02.2024 – 04.03.2024
Розгляд способів представлення даних	05.03.2024 – 10.03.2024
Дослідження методів детектування мережевих атак	11.03.2024 – 17.03.2024
Вибір способу представлення даних	18.03.2024 – 19.03.2024
Описати принцип роботи методу детектування мережевих атак	20.03.2024 – 17.04.2024
Оцінити точність методу	18.04.2024 – 25.04.2024
Оформлення пояснювальної записки згідно методичних рекомендацій	26.04.2024 – 12.05.2024
Подача пакету документів на розгляд ЕК	13.05.2024 – 18.05.2024

6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект Зниження збитків від атак на мережі IoT

Соціальний ефект Покращення технологій забезпечення захисту інформації даних мереж IoT

7. ДОДАТКОВІ ВИМОГИ

Завдання видав _____ Олександр ЛАПТЄВ
(підпис) (Ім'я, ПРІЗВИЩЕ)

Завдання прийняв до виконання _____ Денис ЗИМБИЦЬКИЙ
(підпис) (Ім'я, ПРІЗВИЩЕ)

Дата видачі завдання: 26.10.2023 р.
Термін подання кваліфікаційної роботи до ЕК 17.05.2024 р.

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Метод захисту мереж IoT речей за допомогою штучного інтелекту» складається зі вступу, основної частини, що містить 3 розділи, висновків і списку літератури та джерел. Загальний обсяг роботи – 79 сторінок. Робота містить 4 рисунки, 11 таблиці. Список використаних джерел включає 52 джерела.

Об'єкт дослідження – Процес захисту мереж Інтернету речей (IoT) від потенційних загроз та атак.

Мета роботи – розробити системи захисту електронного документообігу на підприємстві.

Предмет дослідження – метод захисту мереж IoT речей за допомогою штучного інтелекту

Наукова новизна - удосконалено метод захисту електронного документообігу на підприємстві, якій використовує аналіз потенційних загроз для мереж IoT на основі штучного інтелекту.

У розділі 1 дипломної роботи проводиться аналіз сучасного стану досліджень в галузі інформаційної безпеки мереж IoT з використанням технологій штучного інтелекту. Розглядаються основні проблеми, що виникають у забезпеченні безпеки мереж IoT, методи їхнього захисту та моніторингу, а також застосування штучних імунних систем для виявлення атак та аномалій в мережевому трафіку.

У розділі 2 розглядається розробка методів захисту мереж IoT з використанням штучного інтелекту. Аналізуються потенційні загрози для мереж IoT, розробляються методи їхнього виявлення та аналізу, розробляються та імплементуються алгоритми захисту на основі штучного інтелекту.

У розділі 3 представлено експериментальне дослідження та оцінка ефективності розроблених методів захисту мереж IoT. Проводиться постановка

експерименту, проведення експериментальних тестів на реальних пристроях IoT.

Ключові слова: Шифрування, аутентифікація, авторизація, виявлення вторгнень, фізичний захист, моніторинг безпеки

СПИСОК УМОВНИХ СКОРОЧЕНЬ

IoT - Інтернет речей (Internet of Things)

DDoS - Атака з відмовою в обслуговуванні (Distributed Denial of Service)

ІБ - Інформаційна безпека

AI - Штучний інтелект (Artificial Intelligence)

IDS - Система виявлення вторгнень (Intrusion Detection System)

IPS - Система запобігання вторгнень (Intrusion Prevention System)

CIM - Система управління подіями та інцидентами (Security Information and Event Management)

ML - Машинне навчання (Machine Learning)

SVM - Метод опорних векторів (Support Vector Machine)

LSTM - Довга краткочасна пам'ять (Long Short-Term Memory)

RF - Випадковий ліс (Random Forest)

ANN - Штучна нейронна мережа (Artificial Neural Network)

API - Інтерфейс програмування додатків (Application Programming Interface)

URL - Єдино адресований локатор (Uniform Resource Locator)

TCP/IP - Протокол керування передачею та Інтернет-протокол (Transmission Control Protocol/Internet Protocol)

HTTP - Протокол передачі гіпертексту (Hypertext Transfer Protocol)

HTTPS - Безпечний протокол передачі гіпертексту (Hypertext Transfer Protocol Secure)

SSL - Протокол безпеки рівня транспорту (Secure Sockets Layer)

TLS - Протокол безпеки транспортного рівня (Transport Layer Security)

DNS - Система доменних імен (Domain Name System)

MAC - Контроль доступу за адресою мережевої картки (Media Access Control)

IP - Інтернет-протокол (Internet Protocol)

DHCP - Протокол динамічної конфігурації хостів (Dynamic Host Configuration Protocol)

NAT - Мережева адресація з трансляцією мережевих адрес (Network Address Translation)

VPN - Віртуальна приватна мережа (Virtual Private Network)

IoT-пристрій - Пристрій Інтернету речей

AP - Точка доступу (Access Point)

Середовище IoT - Середовище Інтернету речей

БД - База даних

API - Інтерфейс програмування додатків

НСС - низькорівнева складова системи

ПЛК - програмований логічний контролер

ЗМІСТ

ВСТУП

У сучасному світі зростаюча кількість підключених пристроїв створює мережі Інтернету речей (IoT), які відіграють важливу роль у різних сферах життя, від побутових систем до промислових установок. Проте, разом із зростанням розповсюдженості IoT з'являються нові виклики і загрози, пов'язані з безпекою цих мереж. Безпека мереж Інтернету речей стає дедалі важливішою проблемою у зв'язку з можливими наслідками кібератак на підключені пристрої.

Основною метою даної дипломної роботи є розробка методів захисту мереж Інтернету речей з використанням штучного інтелекту. Штучний інтелект (ШІ) стає все більш потужним і дієвим інструментом для виявлення, аналізу та запобігання кіберзагрозам. Наше дослідження спрямоване на розробку і вдосконалення методів, які допоможуть підвищити безпеку мереж Інтернету речей та зменшити ризики їхнього використання для кібератак.

З ростом кількості підключених пристроїв IoT зростає ймовірність атак, спрямованих на порушення їхньої безпеки. До таких атак можна віднести DDoS-атаки, перехоплення конфіденційної інформації, фізичні вторгнення, викрадення ідентифікаційних даних та інші. Ці загрози можуть мати серйозні наслідки, включаючи втрату даних, порушення приватності користувачів, а також можливість витоку конфіденційної інформації.

Однією з ключових складових ефективної захисту мереж Інтернету речей є виявлення потенційних загроз та швидка реакція на них. Для цього використовуються різноманітні алгоритми та системи моніторингу, які базуються на штучному інтелекті. Саме це напрямок досліджень є ключовим у нашій роботі.

Крім того, важливим етапом нашої роботи є експериментальне тестування розроблених методів захисту на ізольованих тестових мережах. Це

дозволить нам оцінити ефективність та надійність наших рішень у реальних умовах, що є критично важливим для подальшої їхньої реалізації та впровадження в практику.

У цій роботі я планую розглянути наступні аспекти:

1. Дослідження та аналіз сучасного стану досліджень у галузі безпеки мереж Інтернету речей.

2. Дослідити найбільш поширені методи детектування мережевих атак, що використовують способи представлення даних. Виділити основні переваги, недоліки і області застосування кожного методу.

3. Вдосокналити метод детектування мережевих атак, заснований на використанні нейроеволюційних алгоритмів, з урахуванням проведеного раніше аналізу.

Ця робота спрямована на покращення безпеки мереж Інтернету речей та забезпечення їхньої надійності та стійкості до кіберзагроз. Ми віримо, що результати нашої роботи матимуть значний вплив на подальший розвиток цієї галузі та допоможуть створити більш безпечне середовище для використання підключених пристроїв IoT.

РОЗДІЛ 1
АНАЛІЗ СУЧАСНОГО СТАНУ ДОСЛІДЖЕНЬ В ОБЛАСТІ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ МЕРЕЖ ІНТЕРНЕТУ РЕЧЕЙ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ

1.1 Проблема забезпечення інформаційної безпеки мереж промислового Інтернету речей

Сучасний етап розвитку промисловості (Industry 4.0) в значній мірі пов'язаний з розвитком і впровадженням промислового Інтернету речей (Industrial Internet of Things, IIoT). За оцінками аналітичної компанії Research and Markets, щорічні темпи зростання світового ринку промислового Інтернету речей, починаючи з 2021 р., складають в середньому 21%, а обсяг світового ринку IIoT до 2026 р. повинен досягти 344,7 млрд. дол.

Промисловий Інтернет речей є розширенням Інтернету речей (Internet of Things, IoT), спрямованим на виконання промислових завдань. Інтернет речей (IoT) зазвичай відноситься до концепції обчислювальної мережі фізичних пристроїв («речей»), оснащених вбудованими технологіями для взаємодії один з одним або із зовнішнім середовищем. С точки зору технологій, Інтернет речей являє собою 4-ланкову систему: пристрої, що підключаються (сенсори, датчики, термінали); мережі, за якими вони взаємодіють; IoT-платформи і додатки для кінцевих користувачів.

Промисловий Інтернет речей (IIoT) – це Інтернет речей, машин, комп'ютерів і людей, що забезпечує інтелектуальні виробничі операції з використанням розширеної аналітики даних для якісно нових результатів бізнесу. У промисловому Інтернеті речей основними різновидами "речей", які треба підключати до мережі, є різні типи датчиків (сенсорів) і приводів. Ці пристрої, з одного боку, мають інтерфейс з комунікаційною мережею, а з іншого – інтерфейс, що забезпечує фізичну взаємодію з процесом, який

потрібно відстежувати. Комунікаційний інтерфейс є необхідною компонентою IIoT. Це може бути дротовий або бездротовий інтерфейс. Але, незалежно від того, яка технологія використовується на канальному і фізичному рівнях, пристрої повинні підтримувати протокол IP, щоб інтегруватися в інфраструктуру IIoT.

Поняття IIoT тісно пов'язане з поняттями кіберфізичної системи і АСУ ТП. Під АСУ ТП традиційно розуміється цілісне рішення, що забезпечує автоматизацію основних операцій ТП на виробництві в цілому або якійсь його ділянці, що випускає відносно завершене виріб, хоча останнім часом функції і цільове призначення АСУ ТП значно змістилися в бік ідеології IIoT. Схема типової архітектури IIoT і її застосування, представлені на рисунку 1.1.

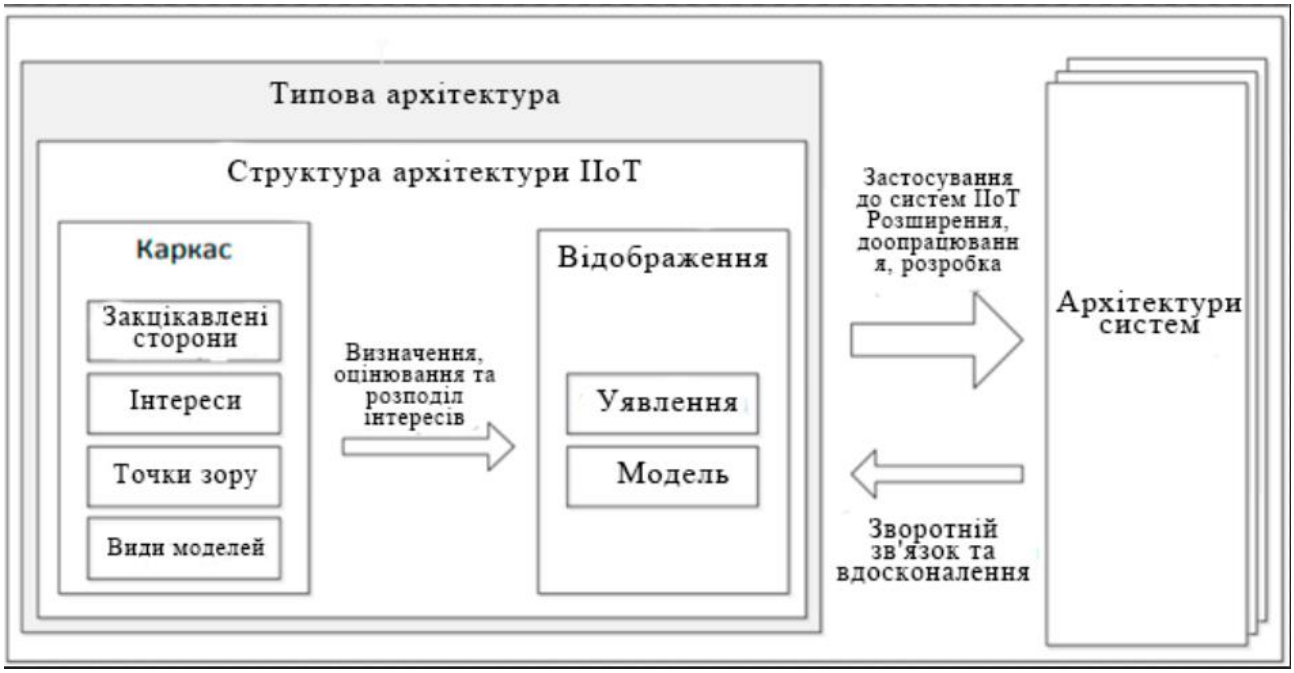

Рисунок 1.1 - Схема типової архітектури IIoT

Приклад трирівневої архітектури промислового Інтернету речей, представленої в стандарті з точки зору реалізації архітектурних патернів (шаблонів), наведено на малюнку 1.2. дана архітектура включає в себе наступні рівні:

- рівень підприємства, де реалізовані додатки, системи підтримки прийняття рішення, інтерфейси для кінцевих користувачів (тут здійснюється отримання потоків інформації з інших рівнів і видача керуючих команд);
- рівень платформи, де агрегується і обробляється інформація граничного рівня, перенаправляються команди управління з рівня підприємства на граничний рівень;
- граничний рівень, де здійснюється збір даних від граничних вузлів за допомогою мережі ближньої дії, а також реалізація керуючих команд.

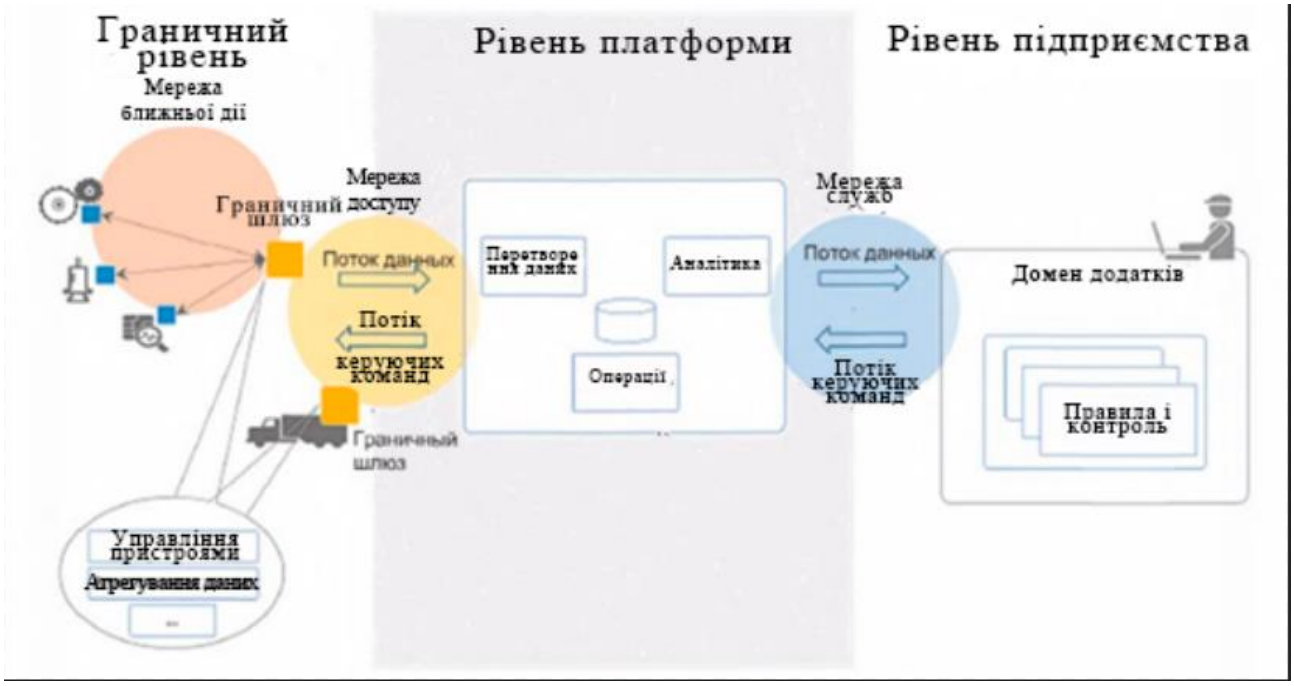

Рисунок 1.2 - Трирівнева архітектура реалізації IIoT

Однією з головних проблем в побудові та експлуатації промислового Інтернету речей є забезпечення ІБ його пристроїв і систем. Найбільш поширені причини слабкої захищеності IIoT:

- застаріле системне та прикладне ПЗ пристроїв IIoT, недостатня увага до програмних оновлень;
- передача даних без шифрування;
- стандартні заводські налаштування безпеки пристроїв;
- незахищені інтерфейси;
- уразливості в операційних системах (ОС) загального призначення;

- неможливість оснастити багато пристроїв вбудованими засобами безпеки.

Специфіка IIoT полягає в підключенні промислових систем до мережі Інтернет, можливості реалізації віддаленого управління ними, в тому числі з пристроїв, що знаходяться за межами підприємства; використання хмарних систем, в тому числі орендованих; обмеженості обчислювальних і енергетичних ресурсів автономних IIoT-пристроїв; їх слабкою захищеності; відсутності, найчастіше, засобів шифрування трафіку. У даній роботі не розглядаються питання шифрування і захисту хмарних систем.

Згідно зі звітом фірми Nokia "Threat Intelligence Report 2020" [1], в останні роки частка атак на пристрої Ios в загальному числі атак на мобільні пристрої збільшилася і досягла значення 32,7 %. За даними Check Point [2], 67% підприємств сьогодні вже зіткнулися з інцидентами безпеки, пов'язаними із застосуванням IIoT-пристроїв.У [3] також підкреслюється вразливість пристроїв промислового Інтернету речей, в якості слабких місць при цьому відзначається триваючий перехід на IPv6, слабкі аутентифікація і стандартні облікові записи, труднощі оновлення програмного забезпечення (ПЗ) і відсутність підтримки виробника, відкритий стан невикористовуваних портів, застосування текстових протоколів, незахищених мобільних технологій, хмарної інфраструктури, вразливого ПЗ і людський фактор.

Виробники, звичайно, намагаються вирішувати проблеми безпеки IIoT. Так, за даними [4], з метою забезпечення ІБ IoT в хмарній системі Microsoft Azure використовується методика моделювання загроз STRIDE, яка розглядає всі рівні і компоненти IoT з точки зору виникнення різних загроз, пропонуються заходи захисту. В [5] проводиться порівняння можливостей забезпечення ІБ таких IOT-фреймворків, як AWS IoT від компанії Amazon, ARM Bed від компанії ARM, Azure IoT від «Microsoft», HomeKit від «Apple», Brillo/Weave «Google», SmartThings від «Samsung», Calvin від «Ericsson» і Kura від «Eclipse». У [6] зазначається, що запобігти загрозі вигідніше, ніж компенсувати шкоду від наслідків її реалізації.

Необхідно відзначити, що в рамках IIoT часто використовуються бездротові сенсорні мережі (Wireless Sensor Networks, WAN), що складаються з великого числа автономних сенсорних вузлів, що збирають різні дані і обмінюються ними за допомогою бездротового з'єднання з більш потужним вузлом – базовою станцією. У зв'язку з їх розподіленою відкритою архітектурою і обмеженістю ресурсів сенсорних вузлів, такі мережі є дуже вразливими для атак [7]. Відповідно до [8, 9], зловмисник може скомпрометувати сенсорний вузол, Слухати, спотворювати та імітувати повідомлення, порушувати цілісність даних та збільшувати витрати ресурсів. Одним з найбільш поширених і небезпечних видів атак, що загрожують WSN, є атаки «відмова в обслуговуванні» (Denial of Service, DoS).

В даний час ведуться роботи з підвищення рівня захищеності WSN. [10] проаналізовано функціональні особливості WSN та найпоширеніші типи атак. В якості захисту від шкідливих або скомпрометованих вузлів пропонується використовувати адаптивне взаємодія елементів системи, засноване на аналізі поведінки сусідніх вузлів.

Серед міжнародних документів в області забезпечення ІБ IoT, Iot і промислових автоматизованих систем можна виділити звіт NISTIR [11], рекомендації та практики Міністерства внутрішньої безпеки США [12], Агентства Європейського Союзу з мереж та інформаційної безпеки (European Network and Information Security Agency, ENISA) [88-90], а також міжнародні стандарти [13].

1.2 Методи забезпечення інформаційної безпеки мереж промислового Інтернету речей

Вимоги до безпеки мереж промислового Інтернету речей визначаються вимогами до забезпечення безпеки систем, в рамках яких вони функціонують, і безпеки інформації, яку вони обробляють.

Відзначимо, що в даний час з метою підвищення рівня автоматизації процесів, пов'язаних з управлінням інцидентами ІБ, підвищення ефективності реагування на кіберзагрози, забезпечення комплексного захисту комп'ютерних мереж створюються і використовуються ситуаційні центри управління ІБ (Security Operation Center, SOC), що розглядаються в [14].

За даними [15], більшу частину часу оператор soc-центру працює з системами управління безпекою та подіями безпеки (Security Information and Event Management, SIEM). SIEM-системи здійснюють збір даних про події по всій мережі з різних джерел, зіставляють події між собою, виявляють підозрілі сукупності подій, які поза цих сукупностей можуть виглядати цілком легітимними. На основі кореляційного аналізу здійснюється більш глибока обробка даних про події і краще виявлення інцидентів ІБ.

Підсистеми кореляційного аналізу є невід'ємною складовою SIEM (Security Information and Event Management) – систем, що здійснюють управління інформацією про події та інциденти ІБ. Методи SIEM ефективно використовуються в тому числі для виявлення мережевих атак на IIoT [16].

Джерела інформації для SIEM-систем включають [17, 18]:

1 системи аутентифікації та контролю доступу, що надають інформацію про успішні або неуспішні спроби отримання доступу;

2 DLP-системи, що представляють відомості про спроби інсайдерських витоків, порушення прав доступу;

3 IDS / IPS-системи, що надають дані про мережеві атаки, зміни конфігурації і доступу до пристроїв;

4 міжмережеві екрани, що надають відомості про атаки, шкідливе ПЗ, спроби порушення правил доступу та ін.;

5 антивірусні програми, що генерують події про працездатність ПЗ, баз даних, зміну конфігурації і політик, шкідливий код;

6 журнали подій серверів і робочих станцій, що використовуються для контролю доступу, дотримання політик інформаційної безпеки;

7 мережеве активне обладнання, що використовується для контролю доступу, обліку мережевого трафіку;

8 сканери вразливостей, що надають дані про інвентаризацію активів, сервісів, ПЗ, вразливостей, топологічної структури;

9 системи інвентаризації та управління активами, що постачають дані для контролю існуючих активів в інфраструктурі та виявлення нових;

10 системи веб-фільтрації, що надають дані про відвідування співробітниками підозрілих або заборонених веб-сайтів.

Аналіз SIEM часто базується практично на «чистій» математиці та статистиці. Але відправною точкою служать задаються вручну правила. Наприклад, одноразове подія "login failed" не є істотним, в той час як п'ять і більше таких подій для одного облікового запису вже можуть свідчити про спроби підбору пароля. У найпростішому випадку, в SIEM системах правила представлені в форматі RAR (Rule-Based Reasoning) І містять набір умов, тригери, лічильники, сценарії дій. Кореляційний аналіз дозволяє виявляти неочевидні взаємозв'язки між подіями ІБ.

SIEM-системи знаходять застосування для вирішення завдань моніторингу та забезпечення ІБ в контексті IoT. Так, в [19] пропонується система виявлення розподілених DoS-атак (DDoS-атак) IOT-ботнетами на основі SIEM. Система виявляє і блокує трафік DDoS-атаки від скомпрометованого Ios пристрою за допомогою моніторингу специфічних типів пакетів, включаючи пакети TCP SYN, ICMP і DNS, що виходять від цих пристроїв. Показано, що підхід, заснований на SIEM, може бути успішно налаштований для точної ідентифікації і блокування шкідливого трафіку скомпрометованих Ios пристроїв. Технології SIEM застосовуються для виявлення інцидентів інформаційної безпеки IOT-систем також в [20]

У [21] пропонується інтеграція SIEM-системи OSSIM з бездротовою системою виявлення вторгнень для підвищення захищеності медичних систем IOT. Система виявлення вторгнень розгортається на Raspberry Pi, аналізується можливість використання такого пристрою в якості хоста для бездротової IDS.

Використовуються також можливості кореляції SIEM для повідомлень про виявлені аномалії і фільтрації помилкових спрацьовувань. Результати показали високу ефективність підходу, навіть у сильно завантаженому середовищі запропонований підхід дозволив успішно аналізувати трафік, завантаження процесора не перевищувало 5%, оперативної пам'яті-400 МБ за 8 годин.

Одним з недоліків SIEM-системи є її реактивна природа. За даними [22], проблема полягає в тому, що SIEM-система починає працювати тільки тоді, коли зловмисник вже проник в інфраструктуру. Тому для ефективної роботи SOC-центру класичні SIEM-системи необхідно доповнювати інтелектуальними системами, які дозволять виявити зловмисника ще на ранніх стадіях атаки або на етапі підготовки до злому

Таким чином, підхід на основі SIEM-систем, адаптований для роботи з великою кількістю джерел даних, добре підходить для інтеграції в IIoT-системи з метою аналізу та виявлення інцидентів ІБ, але обов'язково повинен бути доповнений засобами, що дозволяють виявляти зловмисні дії на ранніх етапах.

У [23] розглядається архітектура безпеки IOT-систем, яка ґрунтується на захисті систем зв'язку, пристроїв і взаємодії в мережі. Для захисту каналів застосовуються шифрування і перевірка справжності. Захист пристроїв розглядається як забезпечення безпеки і цілісності програмного коду. Тема безпеки коду не обговорюється, цілісність забезпечується підписанням коду та перевіркою підпису перед запуском. Є ймовірність зміни коду після запуску в момент завантаження, це компенсується хостовими засобами захисту, такими як харденінг (Hardening – посилення захищеності системи), розмежування доступу до системних файлів і ресурсів, контроль підключень і т.д. Контроль взаємодії в мережі грунтується на аналітиці, яка допомагає виявити підозрілі і зловмисні аномалії, загрози, які подолали наявні засоби і системи захисту.

У роботі [24] підкреслюється, що засоби моніторингу та аналітики можуть бути єдиним засобом вирішення завдань забезпечення безпеки в системах, де оновлення приладів в промислових системах управління неможливо без заміни всієї системи цілком (промислове виробництво,

нафтовидобуток та ін.). Підкреслюється, що в таких випадках системи виявлення аномалій особливо корисні, що багато мереж IoT характеризуються певними шаблонами поведінки, і відхилення легко ідентифікуються. Справа ускладнюється широким набором протоколів, але на допомогу приходить застосування засобів машинного навчання.

Зупинимося докладніше на проблемі виявлення атак і аномалій мережевого трафіку промислового Інтернету речей. У розглянутій трирівневої архітектурі IIoT (Рисунок 1.2) вони повинні виявлятися на граничному рівні, що включає пристрої IIoT, мережі ближньої дії та ін варто відзначити, що в роботі не розглядаються методи забезпечення безпеки хмарних обчислень, широкої використовуваних в IIoT. Питання забезпечення їх безпеки розглядаються в [25].

Розглянемо мережеві атаки детальніше. На сьогоднішній день можна виділити наступні види мережевих атак на Промислові мережі, представлені в таблиці 1.1.

Таблиця 1.1 - види мережевих атак на Промислові мережі IoT

Вид атаки	Опис атаки	Реалізація	Методи протидії
IP-спуфінг	видача зловмисником себе в якості легітимного Користувача за допомогою підміни IP адреси	введення неправдивої інформації або шкідливих команд у стандартний потік даних	-використання криптографічних засобів аутентифікації; - контроль доступу
Ін'єкція	міжсайтовий скриптинг (XSS атака), SQL ін'єкція, XPath ін'єкція.	модифікація запиту до бази даних, впровадження в веб-сторінку довільного коду	- кодування даних і керуючих символів; - правила побудови SQL-запитів; - регулярне

			оновлення.
Відмова в обслуговуванні (DoS)	порушення доступності сервісів і систем за допомогою великого числа запитів	Підтримка всіх з'єднань в зайнятому стані	- функції анти-DoS; - функції антиспуфинга; - застосування систем виявлення атак.
Фішинг-атаки	соціальна розробка або обман співробітників з метою розкрадання ідентифікаційних даних і подальшого несанкціонованого використання	розсилка листів через електронну пошту або повідомлень в месенджерах, як правило, містять посилання на фішинговий ресурс, застосування соціальної інженерії	-використання перевірених ресурсів; - застосування засобів антивірусного захисту (ЗАВЗ) в тому числі поштових, повідомлень своєчасним оновленням базисигнатур; - навчання і підготовка співробітників.
Мережева розвідка	збір інформації про мережу для планування атаки	Мережева розвідка за допомогою DNS або ICMP-запитів (Exo), сканування портів	- блокування ICMP відлуння відповідей прикордонних маршрутизаторів. - застосування систем виявлення атак.
Спеціалізовані програми	віруси, мережеві черв'яки, троянський кінь,	збір даних прихованим нелегітимним агентом	- ЗАВЗ з регулярно оновлюваними базами сигнатур;

	сніффер, руткіт	в системі, лавиноподібне поширення застосування:	– шифрування; - антисніфферів; - міжмережевих екранів; - антируткітів.

Актуальною є проблема виявлення невідомих мережевих атак (Zero day), ймовірність реалізації яких імовірно можна знизити, використовуючи попереджувальні заходи. Інструментальні засоби оцінки вразливостей розглядаються в [26], методи оцінки ризиків реалізації невідомих мережевих атак – в [27], підвищення ефективності їх виявлення в [28]. Також можуть використовуватися методи проактивного моніторингу, наприклад, на основі аналізу часових рядів, а також системи виявлення аномалій, в тому числі на основі штучних імунних систем [29].

Під аномалією мережевого трафіку розуміється істотне відхилення трафіку мережевого пристрою від нормального профілю трафіку для даного пристрою або групи пристроїв [30].

Серед методів виявлення аномалій можна виділити статистичні методи, які будуються на зібраній статистиці параметрів нормального мережевого трафіку і порівняння з нею аналізованого трафіку [31]. Наприклад, може аналізуватися обсяг переданих даних, що мають загальні характеристики, або число з'єднань за 5 хвилин. Якщо значення параметрів різко відхиляються від очікуваних, значить, виникла аномалія. Іншим прикладом аналізованої характеристики може бути кількість пакетів, що надходять на певний порт [32]. Статистичні методи виявлення аномалій також розглядаються в [33], підкреслюється, що найчастіше в комерційних системах виявлення аномалій поєднується використання статистики з методами машинного навчання.

На підставі проведеного аналізу можна зробити наступні висновки. Проблема виявлення атак і аномалій мережевого трафіку є однією з ключових для IIoT-систем, її вирішенню має приділятися першорядна увага. Підхід до

виявлення інцидентів ІБ на основі SIEM-систем добре підходить для інтеграції в IIoT-системи, подібне об'єднання має підвищити ефективність вирішення завдань моніторингу ІБ промислового Інтернету речей.

1.3 Методи моніторингу ІБ мереж промислового Інтернету речей із застосуванням технологій інтелектуального аналізу даних

Моніторинг ІБ мереж IIoT здійснюється на основі даних про мережевий трафік. Завдання збору цих вхідних даних речей ускладнюється використанням IIoT-пристроями різних протоколів і типів підключень. У системах Інтернету речей застосовуються:

- бездротові локальні мережі (Wireless Local area Network, WLAN), бездротові персональні мережі (Wireless Personal Area Network, WPAN), включаючи мережі ближнього (малого і середнього) радіусу дії, такі протоколи, як: Wi-Fi, 6lowpan, ZigBee IP, Thread, Z-Wave, ZigBee, WirelessHart, BLE 4.2 (Bluetooth Mesh), MiWi.

- енергоефективні глобальні мережі (Low-Power Wide Area Network, LPWAN), технології для передачі невеликих даних на далекі відстані: Larawan, SIGFOX, CIoT, 4G LTE, 5g, NB-IoT та інші [34].

При використанні мобільних пристроїв з прямим доступом в Інтернет через SIM-карту, перехоплення трафіку можливий за допомогою способів, розглянутих в [35], включаючи установку агента на SIM-карту або на сам IOT-пристрій. У даній роботі не розглядаються питання збору даних мережевого трафіку IoT-пристроїв, що використовують мобільний Інтернет безпосередньо. Збір даних з відносно стаціонарних пристроїв, підключених до мережі Інтернет за допомогою певної внутрішньої мережевої інфраструктури, здійснюється за допомогою сніффінга або віддзеркалення трафіку з мережевого обладнання.

Необхідно враховувати розподілений характер об'єкта моніторингу ІБ, при цьому в якості джерел вхідних даних повинні виступати не тільки пристрої IIoT, але і мережеве обладнання: маршрутизатори, міжмережеві екрани (ме) та

ін., є можливість взаємодії з SIEM-системою. Також слід враховувати необхідність реалізації просторово-часової моделі збору вхідних даних для систем моніторингу. Вхідні дані повинні мати прив'язку до конкретних вузлів мережі IIoT і часу їх реєстрації, тільки є бути темпоральними.

Мережева взаємодія відноситься до граничного рівня архітектури IIoT, а системи управління, діагностики та моніторингу – до рівня платформи. З метою моніторингу ІБ мережі IIoT з граничного рівня можуть бути зібрані дані про мережеву взаємодію, стан IIoT-пристроїв, з рівня платформи – дані про поточний стан мереж і кінцевих точок IIoT, загальну кількість інцидентів, що надходять від зовнішніх систем моніторингу. До таких систем можуть ставитися системи SIEM і SCADA, а також, наприклад, система виявлення небезпечних станів промислових об'єктів, та ін. Передбачається збір даних про мережеву взаємодію з канального по транспортний рівнів мережевої моделі OSI.

Моніторинг ІБ мережі IIoT, в першу чергу, передбачає аналіз стану мережевого трафіку IIoT, інформація про який включає:

- часові ряди технологічних параметрів(ВРТП), тобто Параметри (Дані), що обробляються за допомогою мультисенсорних мереж;
- внутрішній мережевий трафік IIoT, тобто дані, що передаються по каналах зв'язку на кожному з рівнів управління IIoT і між рівнями управління
- зовнішній мережевий трафік IIoT, тобто дані, що надходять із зовнішнього середовища (Інтернет, передавачі, провайдери і т. д.) і передаються в зовнішнє середовище;
- дані, що надходять від взаємодіючої SIEM системи, про події (інциденти) ІБ.

Таким чином, система моніторингу ІБ мережі IIoT повинна бути розподіленою, враховувати характер зібраних вхідних даних, гетерогенність відповідних джерел, що відноситься і до ІІС як нижньому рівню системи моніторингу ІБ мережі. Необхідно враховувати різнорідність вхідних даних також на етапі їх нормалізації (приведення до єдиного формату подання).

Відзначимо також, що визначення конкретного складу зібраних і аналізованих даних залежить від використовуваних протоколів і технологій конкретного об'єкта. У даній роботі не пропонується будь-якої певний склад параметрів, найбільш універсальний для всіх мереж IIoT або, навпаки, найбільш підходящий для певної вузької області. Для навчання і роботи системи виявлення атак може бути використаний будь-який набір параметрів, достатній для визначення на його основі безпеки тієї чи іншої мережевої взаємодії, обраний експертами відповідно до використовуваних на конкретному об'єкті мережевими технологіями, протоколами, що забезпечує можливість ефективної класифікації. Разом з тим, загальний підхід до нормалізації даних повинен включати в себе кодування їх якісних, текстових або лінгвістичних значень числовими параметрами, перетворення вихідного діапазону кількісних значень до використовуваного системою діапазону, докладніше питання нормалізації розглядаються в [36].

На етапі навчання і тестування СОА стосовно мережі IIoT будемо відштовхуватися від параметрів, що використовуються в різних, найбільш часто використовуваних наборах даних про мережеві з'єднання, що містять параметри трафіку як для нормальних мережевих з'єднань, так і для різного роду атак – датасетах (ДС). Дані по деяким з них наведені в таблиці 1.2.

Таблиця 1.2 - ДС, використовувані для навчання СОА на IoT / IIoT

Найменування ДС	Кількість параметрів/ атрибутів	Кількість видів/класів атак	Специфіка	Атака
KDD-99	41	22	трафік інформаційно телекомунікаційних мереж (ІТКС)	мережеві атаки на ІТКС
NSL-KDD	41	22		
UNSW-NB15	48	9		
LITNET-2020	85	12		

TON_IoT	44	9	Логи ОС Windows, Ubuntu, IoT/ IIoT, мережевий трафік ІТКС	
AWID 2	154	16	WiFi-трафік	мережеві атаки в WiFi-мережах
AWID3	253	16		
VARIoT	83	0	трафік IoT-пристроїв розумного будинку	Ні
IoTID20	83	4		атаки на IoT
IoT-23	21	8		ботнет трафік
N-BaIoT	115	2		
BoTNeTIoT-L01	23	2		
Bot-IoT	46 або 10	4	трафік IoT-пристроїв	
NF-BoT-IoT	43 або 12	4	NetFlow-версія Bot-IoT	
DS2OS	12	7	дані прикладного рівня	
WSN-DS	23	4	трафік WAN по протоколу LEACH	атаки на WSN

У таблиці 1.2 для деяких датасет зустрічаються два значення кількості параметрів, наприклад, для Bot-IoT – «46 або 10». Мається на увазі, що існує дві версії ДС з більшою і меншою кількістю параметрів. Крім того, для датасета Variety вказано кількість містяться атак, рівне нулю; це пов'язано з тим, що Variety містить дані тільки про нормальну мережеву взаємодію.

Також відзначимо, NF-BoT-IoT є NetFlow-версією датасета Bot-IoT, представленої університетом Квінсленда (Австралія). На відповідній сторінці [37] сайту даного університету представлені NetFlow-версії та інших мережевих наборів даних, докладно описані в [38], такі як NFUNSW-ТИ 15, NF-ToN-It, NF-BoT-IoT, NF-CSE-CIC-IDS2018, NF-UQ-nids.

BoTNeTIoT-L01 є доопрацьованою версією N-BaIoT зі зменшеною надмірністю, з вибором тільки параметрів 10-секундного часового вікна. NSL-KDD-допрацьована версія KDD-99, що не містить надлишкових і повторюваних записів.

Більшість з цих ДС містять або ботнет-трафік, тобто трафік пристроїв, які вже заражені, або трафік, характерний для класичних інформаційно-телекомунікаційних систем (ІТКС), що не містять дані, характерні для IIoT-пристроїв, або вузькоспеціалізовані дані, такі як атрибути CAN і дані прикладного рівня.

Як правило, класичними для побудови, навчання і тестування систем виявлення вторгнень вважаються Набори навчальних даних (датасети) KDD-99 і його вдосконалена версія – NSL-KDD. Розглянемо задачу нормалізації параметрів і зменшення розмірності простору цих параметрів на прикладі NSL-KDD.

NSL–KDD-містить набір векторів-рядків, кожна з яких складається з 41 параметра з'єднання, представлені в таблиці 1.3. Кожен рядок позначений, чи відповідає він якомусь виду атаки чи нормальному стану системи. Всього представлено 22 класу атак, об'єднаних в 4 групи: User to Root (U2 r), Remote to Local (R2 l), Probe, denial of Service (DoS). Всі дані, представлені в наборі, можуть бути згруповані в три категорії: характеристики окремих з'єднань, особливості з'єднання, параметри з'єднання за проміжок часу.

Таблица 1.3 - Параметри датасета NSL-KDD

Найменування параметрів		
duration;	su_attempted;	same_srv_rate;
protocol_type;	num_root;	diff_srv_rate;
service;	num_file_creations;	srv_diff_host_rate;
flag;	num_shells;	dst_host_count;
src_bytes;	num_access_files;	dst_host_srv_count;
land;	is_host_login;	dst_host_same_srv_rate;
wrong_fragment;	is_guest_login;	dst_host_diff_srv_rate;
urgent;	count;	dst_host_srv_diff_host_rate;
hot;	srv_count;	dst_host_serror_rate;
num_failed_logins;	serror_rate;	dst_host_srv_serror_rate;
logged_in;	srv_serror_rate;	dst_host_rerror_rate;
num_compromised;	rerror_rate;	st_host_srv_rerror_rate.
root_shell;	srv_rerror_rate;	

Класи мережевих атак, представлені в NSL-KDD, наведені в таблиці 1.4.

Таблица 1.4 - Мережеві атаки, що містяться в NSL-KDD

Вид атаки	Класс атаки
R2L	ftp_write
	guess_passwd
	imap
	multihop
	phf
	spy
	warezclient
	warezmaster
probe	ipsweep

	nmap
	portsweep
	satan
DoS	Back
	land
	neptune
	pod
	smurf
	teardrop
U2R	buffer_overflow
	loadmodule
	perl
	rootkit

В [39] на прикладі датасета KDD-99, який за складом використовуваних параметрів (ознак) мережевого трафіку аналогічний розглянутому NSL-KDD, пропонуються різні підходи до зменшення розмірності (стиснення) простору ознак, тобто до визначення найбільш інформативних з них. У [40] для стиснення простору параметрів застосовується сингулярне розкладання матриць. У [41] вирішується задача стиснення простору параметрів NSL-KDD з використанням методики розрахунку впливу атрибутів через механізм аналізу відповідностей.

Однак в [42] пропонується для кожної атаки використовувати свій окремий список параметрів. Тобто для атаки DoS пропонується один набір параметрів, для U2R-інший і т.д. при використанні параметрів, в обчислювальних експериментах виявити загрози взагалі не вдалося. Стиснення з використанням сингулярного розкладання матриць дозволяє проводити класифікацію одноразово стислих даних, проте не дозволяє аналізувати все нові і нові рядки мережевої активності. Тому в даній роботі вирішувалася також

завдання визначення інформативних (значущих) параметрів використовуваних датасетов.

Після збору і нормалізації дані про мережевий трафік піддаються аналізу. Для цього використовуються різні методи та технології видобутку даних, Інтелектуальні системи сьогодні в першу чергу асоціюються з штучними нейронними мережами (ІНС), зарекомендували себе в якості ефективного методу класифікації, що застосовується для вирішення величезної кількості різних завдань, включаючи виявлення атак і мережевих аномалій, в тому числі з використанням технологій глибокого навчання. Методи машинного навчання також включають в себе алгоритми дерев рішень (ДР), випадкового лісу (ВЛ), мурашиної колонії, нечіткої логіки, k-найближчих сусідів (k nearest neighbors, KNN), машини опорних векторів (SVM), наївний байєсівський Класифікатор (НБК), генетичні алгоритми (ГА) і ін ці методи широко використовуються для вирішення завдань інтелектуального аналізу даних, в тому числі для виявлення і класифікації атак на основі аналізу даних про мережевий трафік.

Для вирішення завдань виявлення атак також знайшли досить широке застосування штучні імунні системи (ІВС). Вони характеризуються здатністю виявляти невідомі атаки, можливістю постійного фонового Самонавчання. За даними, ІВС можуть в рази перевершувати своїх головних конкурентів (ІНС і ГА) за швидкодією, а також характеризуються вдвічі меншою кількістю помилок. Іншою перевагою ІІС є їх застосовність в розподілених системах, в тому числі в рамках реалізації багатоагентного підходу. Підвищення ефективності виявлення мережевих атак і аномалій можливо також за рахунок спільного використання декількох методів штучного інтелекту (ШІ) в рамках гібридної інтелектуальної системи (ГІС), що об'єднує в своєму складі дві або більше різні технології ШІ з метою отримання синергетичного ефекту, нівелювання недоліків однієї технології перевагами іншої. Так, системи нечіткої логіки зрозумілі і прозорі для Користувача, але у них відсутня здатність до навчання. ІНС, навпаки, здатні до навчання, але непрозорі для користувача. Їх спільне використання в складі нечіткої нейронної мережі (ННС)

дозволяє отримати адаптивну систему, здатну до навчання і одночасно в значній мірі прозору для користувача.

Загальна ідея побудови гібридних інтелектуальних систем виявлення атак (Гіса) обговорюється в ряді робіт. Як правило,в основі побудови ГІСОА використовується об'єднання ІНС, алгоритмів кластерного аналізу, ін, SVM з іншими різними за своєю ідеологією методами ШІ. Окрему перспективну групу СОА займають ГІСОА на базі ІІС в доповненні з іншими технологіями.

Стосовно до промислових мереж і мереж Інтернету речей використовуються детально розглянуті в] підходи, що включають в себе:

1. нейронні мережі глибокого навчання-для виявлення мережевих атак і аномалій в IoT / IIoT мережевих атак на кіберфізичні системи;

2. методи машинного навчання-для виявлення атак на IoT , IIoT, у тому числі в порівнянні різних алгоритмів з ІНС, де кращу точність демонструють ІНС і СЛ;

3. штучні імунні системи-для ідентифікації вторгнень в мережі IoT.;

4. гібридні інтелектуальні системи виявлення атак на IoT, IoMT, АСУ ТП і IIoT, у тому числі використовують алгоритми ІІС

Таким чином,для вирішення завдання забезпечення мережевої безпеки промислового Інтернету речей сьогодні запропоновані різні методи ШІ, включаючи методи машинного навчання, ІНС, ІІС, а також гібридні інтелектуальні системи, що дозволяють використовувати переваги різних підходів і нівелювати їх недоліки. Особливий інтерес в даному випадку представляє застосування ІІС, в тому числі в складі гібридних інтелектуальних систем, завдяки їх здатності виявляти невідомі атаки, постійно самонавчатися, високій швидкодії, низькому рівню помилок, застосовності для реалізації в класі розподілених багатоагентних систем

У даній роботі пропонується відповідна концепція побудови багаторівневої гібридної розподіленої інтелектуальної системи виявлення атак і аномалій мережевого трафіку IIoT на основі багатоагентної платформи з використанням механізмів штучних імунних систем, методів машинного

навчання, взаємодії з підсистемою кореляційного аналізу подій ІБ або SIEM системою.

Висновки до розділу 1

1. Системи промислового Інтернету речей в даний час є в достатній мірі вразливими. Система стандартизації в даній новій області поки тільки розробляється, виробники все ще ставлять в пріоритет вигоду виробництва і функціональність пристроїв на шкоду їх безпеки. Разом з тим, вже зараз починають активно розроблятися методи і засоби захисту IIoT.

2. З метою забезпечення мережевої безпеки промислового Інтернету речей знаходять застосування різні методи штучного інтелекту, включаючи методи машинного навчання, ІНС, а також гібридні інтелектуальні системи, що дозволяють використовувати переваги різних підходів і нівелювати їх недоліки. Особливий інтерес представляє використання ІІС, в тому числі в складі гібридних інтелектуальних систем, завдяки їх здатності виявляти невідомі атаки, самонавчатися, високій швидкодії, низькому числу помилок, застосовності для реалізації в класі розподілених багатоагентних систем.

3. ІІС є розподіленою адаптивною системою виявлення подій ІБ, нехарактерних для нормального стану системи, або, в залежності від реалізації, небезпечних подій, в тому числі невідомих.

Існують приклади застосування ІІС для захисту мереж промислового Інтернету речей, проте відкритими залишаються питання побудови перспективної дворівневої ІІС, заснованої на комплексуванні алгоритмів негативної селекції, клонального відбору, дендритних клітин, оновлення і пам'яті детекторів, що реалізує взаємне навчання і самонавчання агентів нижнього рівня, аналіз небезпеки на верхньому рівні в складі гібридної розподіленої системи моніторингу ІБ мереж промислового Інтернету речей.

РОЗДІЛ 2
МЕТОДИ ЗАХИСТУ МЕРЕЖ ІНТЕРНЕТУ РЕЧЕЙ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

2.1 Аналіз потенційних загроз для мереж Інтернету речей

Мережі Інтернету речей (IoT) стають дедалі більш важливими в різних сферах життя, включаючи побутові, комерційні та промислові сектори. Однак разом зі зростанням їхнього використання також зростає кількість потенційних загроз для цих мереж. У цьому підрозділі проводиться аналіз найбільш типових атак на мережах IoT з метою визначення основних загроз та вразливостей.

2.1.1 Ідентифікація типових атак на мережі Інтернету речей

У контексті безпеки мереж Інтернету речей (IoT) ідентифікація типових атак є першочерговим завданням для розробки ефективних заходів захисту. Розгляд цього аспекту допомагає виокремити загрози, які можуть бути спрямовані на системи IoT, і розробити відповідні стратегії протидії.

1. **DDoS-атаки (атаки з відмовою в обслуговуванні)**:

DDoS-атаки, або атаки з відмовою в обслуговуванні, є однією з найпоширеніших та найагресивніших форм кібератак на мережі Інтернету речей (IoT). У цих атаках зловмисники намагаються перевантажити мережу, підключену до Інтернету, через величезний об'єм запитів, що надходять одночасно. Це може призвести до тимчасової або повної недоступності системи для легітимних користувачів.

DDoS-атаки можуть бути розподілені (Distributed Denial of Service - DDoS), коли атакуючий складається з багатьох різних джерел, що робить їх ускладненими для виявлення та блокування. Ці атаки можуть бути здійснені за

допомогою зламаних пристроїв IoT, які стають частиною ботнету (мережа компрометованих пристроїв, керованих зловмисниками).

Наслідки DDoS-атак можуть бути серйозними, зокрема втрата бізнесу, втрата клієнтів, збитки у репутації та фінансові втрати. Час, необхідний для відновлення роботи після атаки, може бути значним, що призводить до збоїв у нормальному функціонуванні бізнесу та негативно впливає на відносини з клієнтами.

Для захисту від DDoS-атак необхідно використовувати комплексний підхід, який включає в себе застосування спеціалізованих засобів моніторингу мережі, виявлення та фільтрації незвичайного трафіку, а також посилення архітектури мережі для оптимізації пропускної здатності та реактивне виявлення атак та реагування на них.

2. **Перехоплення та витік конфіденційної інформації**:

Ця загроза виникає, коли зловмисники здійснюють спроби отримати доступ до конфіденційної інформації, яка передається через мережу Інтернету речей (IoT). Це може статися через перехоплення передаваних даних або витік інформації з самого пристрою або мережі.

Мережі IoT часто включають в себе різноманітні типи пристроїв, які можуть передавати чутливу інформацію, таку як особисті дані користувачів, конфіденційні дані підприємств або медичні дані. Зловмисники можуть намагатися перехопити ці дані з метою використання їх у шкідливих цілях, таких як крадіжка особистих даних, шахрайство або шантаж.

Методи перехоплення та витоку конфіденційної інформації можуть включати в себе перехоплення мережевого трафіку, використання програмних вразливостей пристроїв IoT для витоку даних, або навіть фізичний доступ до пристроїв.

Для захисту від цієї загрози необхідно використовувати шифрування даних, захист мережевого трафіку за допомогою брандмауерів та інших засобів безпеки мережі, а також ретельне контролювання доступу до пристроїв та застосунків. Також важливо регулярно оновлювати програмне забезпечення

пристроїв IoT та виявляти та виправляти вразливості, що можуть бути використані зловмисниками для витоку інформації.

3. **Фізичні атаки**:

Фізичні атаки є серйозною загрозою для мереж Інтернету речей (IoT), оскільки зловмисники намагаються отримати фізичний доступ до пристроїв IoT з метою їхнього пошкодження або недозволеного втручання у їхню роботу.

Ці атаки можуть бути різноманітними, включаючи фізичні пошкодження, такі як руйнування або відірвання пристроїв, або намагання отримати несанкціонований фізичний доступ до внутрішніх компонентів або роз'ємів пристроїв для встановлення шкідливого програмного забезпечення або зміни конфігурації.

Особливо вразливі до фізичних атак можуть бути пристрої IoT, розташовані у відкритому доступі або у вразливих місцях, де їх можна легко підірвати або фізично пошкодити.

Для захисту від фізичних атак необхідно вживати ряд заходів, таких як фізичне забезпечення пристроїв та інфраструктури, використання механізмів аутентифікації та авторизації для обмеження доступу до пристроїв, а також застосування шифрування даних для запобігання несанкціонованому доступу до конфіденційної інформації у випадку втрати або крадіжки пристроїв. Також важливо встановити механізми виявлення та реагування на випадки фізичних атак для швидкого виявлення і реагування на подібні загрози.

4. **Викрадення ідентифікаційних даних**:

Ця загроза виникає, коли зловмисники намагаються отримати несанкціонований доступ до ідентифікаційних даних, що використовуються для аутентифікації та авторизації в мережі Інтернету речей (IoT). Ідентифікаційні дані можуть включати логіни, паролі, сертифікати або інші форми ідентифікації, які дозволяють користувачам або пристроям отримати доступ до системи або мережі.

Зловмисники можуть намагатися використовувати різні методи для викрадення ідентифікаційних даних, включаючи атаки на системи

аутентифікації, перехоплення даних під час їх передачі через мережу, фішингові атаки або використання вразливостей програмного забезпечення для отримання доступу до сховищ ідентифікаційних даних.

Наслідки викрадення ідентифікаційних даних можуть бути серйозними, оскільки зловмисники можуть використовувати отримані дані для несанкціонованого доступу до систем або мережі, викрадення конфіденційної інформації або вчинення фінансового шахрайства.

Для захисту від цієї загрози необхідно використовувати захищені методи аутентифікації, такі як двофакторна аутентифікація або біометричні методи, шифрування даних під час їх передачі через мережу, регулярне оновлення програмного забезпечення для усунення вразливостей та використання моніторингових систем для виявлення незвичайної активності або спроб несанкціонованого доступу до системи. Також важливо навчати користувачів профілактичним заходам та практикувати безпечне зберігання та обробку ідентифікаційних даних.

5. **Маніпулювання даними**:

Маніпулювання даними - це атака, в результаті якої зловмисники змінюють, перехреснують або навіть знищують дані, що передаються через мережу Інтернету речей (IoT). Ця загроза може мати серйозні наслідки, оскільки вона може спричинити помилкові висновки, порушити нормальну роботу системи або навіть спричинити небезпеку для користувачів або оточуючого середовища.

Маніпулювання даними може включати в себе такі дії, як зміна значень параметрів, що передаються пристроями IoT, внесення помилкових записів в бази даних, перехресне впливання на дані в різних системах або навіть знищення чи блокування доступу до даних.

Зловмисники можуть використовувати маніпулювання даними для різних цілей, включаючи злам систем безпеки, вчинення шахрайства, руйнування репутації або навіть навмисне завдання шкоди фізичному середовищу.

Для захисту від цієї загрози необхідно використовувати різноманітні технічні та організаційні заходи, такі як шифрування даних, цифрові підписи, механізми контролю цілісності даних, моніторингові системи для виявлення незвичайних патернів в даних та вчасні заходи для відновлення даних у випадку їхнього пошкодження або втрати. Також важливо навчати користувачів профілактичним заходам та практикувати безпечне зберігання та обробку даних для запобігання маніпулюванню даними.

Ретельне дослідження та розуміння цих типових атак є важливим етапом у розробці ефективних стратегій захисту для мереж Інтернету речей, що забезпечує безпеку та надійність їхньої роботи.

2.2 Способи представлення даних

За описовою природою способи подання даних принципово можна розділити на використовують:

- багатовимірні часові ряди без перетворення з подальшим аналізом;
- стислі, агреговані або оброблені іншими способами багатовимірні часові ряди;
- фрактальне представлення топології системи;
- графові структури різних видів.

Тут і далі перераховуються основні способи опису даних і постановки завдань, а також причини використання цих способів з виділенням їх переваг, недоліків і основних областей застосування.

2.2.1 Багатовимірні часові ряди без перетворення з подальшим аналізом

Пропонований підхід розглядає дані з НСС (низькорівнева складова системи) АСУ: з актуаторів, ПЛК (програмованих логічних контролерів), сенсорів. Надходять дані перетворюються в багатовимірні тимчасові ряди. Використання багатовимірних часових рядів обгрунтовано наступним

принципом: даний метод зберігає велику інформативність для подальшого аналізу за рахунок збереження зв'язків між пристроями.

Для детектування відхилень у процесах функціонування КФС засобами навченої моделі прогнозування багатовимірного часового ряду виконується передбачення майбутнього стану системи. На вхід моделі надходять показання поточного стану, а на виході виходить передбачений результат. Далі обчислюється помилка-різниця між реальним значенням стану КФС і передбаченим за допомогою навченої моделі. Якщо величина помилки знаходиться вище порогового значення, система фіксує детектування аномального стану.

Багатовимірний часовий ряд являє собою наступну сукупність: $X = \{X(1), X(2), \ldots, X(m)\}$, де кожне значення в момент часу $t\ i$ представлено вектором: $X = \{x1^{(i)}, x2^{(i)}, \ldots, xn^{(i)}\}$. Спочатку дані, отримані від компонентів системи, нормуються: приводяться до єдиного виду і єдиного масштабу: $xi = \frac{xi - xmin}{xmax - xmin}$

Передбачення наступного значення часового ряду являє собою побудову моделі: $\hat{y} = model(x_{t-1}, \ldots, x_{t-n})$.

Для обчислення помилки виконується наступний набір дій:

- обчислюється різниця між передбаченим $\hat{y}$ і спостережуваним у значенням для кожної ознаки $e_t = |y - \hat{y}|$;

- далі можна виявляти аномалії на підставі умови $_{max}e_t > T$, де T – деякий поріг.

Використовуючи даний підхід, необхідно відзначити, що всі атаки, що проводяться на системи, є тривалими в часі. Для фіксування даного положення і підвищення точності детектування враховуються всі зафіксовані максимальні помилки (сплески) в деякому фіксованому за розміром вікні часу: $Erri = \sum max\ e_t > T$

Переваги способу:

- висока точність короткострокового прогнозування і, як наслідок, гарне виявлення атак, що укладаються в одне або кілька вікон при відбудові на конкретних, заздалегідь відомих тимчасових параметрах;
- варіативність застосовуваних аналізаторів даних.

Недоліки способу:

- необхідність ручного підбору гіперпараметрів провісника, наприклад на основі підходу пошуку по решітці;
- варіативна ширина вікна вибірки даних і, як результат, висока ймовірність помилок для атак з сильно помітною тривалістю: пр і вузькому вікні висока ймовірність пропустити тривалі атаки, і навпаки, при досить широкому вікні зростає ймовірність пропустити короткочасні атаки;
- у загальному випадку даний підхід показав не найвищу точність виявлення, проте один з найвищих показників універсальності.

Область застосування: системи, що мають достатні обчислювальні ресурси для тимчасового зберігання і обробки багатовимірних часових рядів без перетворення і вимагають глибокого аналізу взаємозв'язків з допустимим нехтуванням топології.

2.2.2 Адаптивний алгоритм фільтра Калмана

Показання компонентів системи представляються у вигляді хаотичної траєкторії руху деякого тіла зі змінною швидкістю в одномірному просторі з використанням класичних фізичних рівнянь шляху, швидкості і прискорення матеріальної точки. Алгоритм пр іменування фільтра Калмана для обчислення майбутніх значень представлений на рисунку 2.1

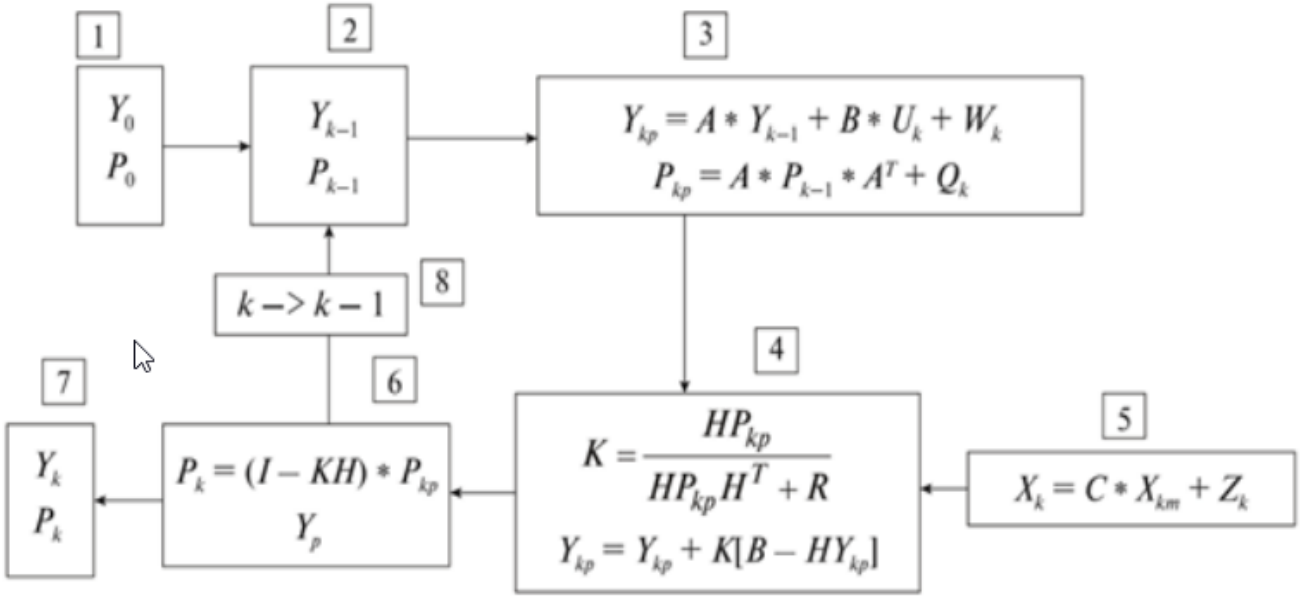

Рисунок 2.1 - Алгоритм застосування фільтра Калмана

У блоці 1 описуються початкові оціночні значення: $Y0$ – коор дината тіла і $P0$ – коваріаційна матриця помилок; в блоці 2 представлені значення тих же параметрів на K-1 кроці. У блоці 3 Показані рівняння оцінки поточного стану тіла на основі стану з попереднього кроку. Матриця Q-шум, що приймається за нульову матрицю, виходячи з фізики задачі. Матриця коваріації $P\ k\ p$ оновлюється на основі попередньої з використанням матриці еволюції процесу A. У блоці 4 описується обчислення Калманова посилення: під змінною h розуміється допоміжна матриця, яка використовується в приведенні посилення Калмана до матриці потрібної кінцевої розмірності; R – помилки вимірювань. У блоці 5 використовуються дані від компонентів системи з обчисленими значеннями швидкостей, ці дані представлені матрицею C і шумовий матрицею Z. блоці 6 описує кроки отримання фінального результату фільтра. Блок 7 містить значення $Y\ k$ і$P\ k$, необхідні при розрахунках. У блоці 8 описується ітерація фільтра необхідну кількість разів виходячи з того, що обчислені для кроку k значення стануть новими значеннями для кроку k-1.

Показання датчиків системи представляються у вигляді часових рядів, фільтр Калмана застосовувався до кожного з рядів окремо, отримані показання

представляються у вигляді єдиного багатовимірного ряду, який надалі використовується для аналізу даних і подальшого передбачення стану на наступному кроці.

2.2.3 Дискретне вейвлет-перетворення

Вейвлет-перетворення, в загальному випадку, є узагальненням спектрального аналізу, і базується на розкладанні одержуваних даних на величини і кінцеві базиси, більш зручні для кінцевого аналізу і обробки. На відміну від Фур'є-перетворення, вейвлет-перетворення має можливість аналізу даних в частотно-часовій області.

Ознайомившись з математичним апаратом даного перетворення, стає зрозуміло, що будь-яку послідовність можна розкласти на вейвлет-функції і базисні масштабуючі функції. Безпосередньо саме перетворення виконується для розкладання досліджуваних послідовностей на дві послідовності коефіцієнтів: $coeffA$ і $coeffD$. Коефіцієнти наближення пов'язані з функцією масштабування, а коефіцієнти деталізації $coeffD$, згідно з роботою [22], пов'язані з базовою вейвлет-функцією. У випадках, коли рівень р азложенія приймає величину більше одиниці, необхідно у всіх наступних кроках виконати вейвлет-перетворення отриманих на кожному рівні коефіцієнтів апроксимації $coeffA$.

Після всіх проведених операцій, на виході алгоритму на рівні розкладання j будемо мати набір коефіцієнтів [$coeffAj$, $coeffDj$, $coeffDj$-1 ,$coeffDj$-2 ,..., $coeffD2$, $coeffD1$], який надалі використовується для аналізу даних і подальшого передбачення стану системи на наступному кроці.

Переваги обох способів:

- агрегація, стиснення або обробка трафіку з приведенням до однорозмірних величин спрощують і стандартизують аналіз;
- підвищення продуктивності аналізатора;

- відсутність витрат на зберігання даних через використання тільки поточного стану системи; можливість використовувати різні способи вирішення, не обов'язково зав'язані на статистичному аналізі;

- абстрагування від топології системи шляхом заміщення мережевої структури багатовимірними часовими рядами з подальшою обробкою;

Недоліки обох способів:

- у загальному випадку статистичні інструменти обробки даних не враховують топологію системи і її фізичні властивості, що може дуже сильно позначитися на точності аналізу за рахунок зниження інформативності;

- складність уніфікації настройки аналізатора одночасно на короткочасні і довготривалі атаки за рахунок фізичної неможливості статистичного апарату.

Область застосування: знижені вимоги до ресурсоємності і уніфікації даних додатково стимулюють використання даного підходу в гомоморфних системах з малим ступенем мінливості, наприклад в самоподібних і/або лінеаризованих структурах, не сильно зав'язаних або не зав'язаних зовсім на власній топології. Найбільшу ефективність даний підхід повинен показати в системах, сильно схильних до статистично передбачуваним флуктуацій трафіку, наприклад в мережевих магістралях

2.3 Графові структури різних видів

Через велику кількість існуючих графових структур тут і далі будуть розглядатися найбільш популярні та ефективні топології в області описових способів представлення даних в кіберфізичних системах. Зокрема, будуть розглянуті:

- класичні графи;
- динамічні графи;
- подієві графи;
- сигнальні графи.

2.3.1 Класичні графи

Даний тип графових структур найбільш загальний, простий і різнобічний. Як наслідок, дані графи є базисом для більшості наступних графів або допоміжним пристроєм в інших описових методах. Іноді, втім, використовуються якісь модифікації стандартних графових структур, які складно віднести до нового виду за рахунок малої кількості змін. Дані структури, зважаючи на свою простоту, добре себе зарекомендували в багатьох сферах застосування, в тому числі і в розглянутій.

Методом дослідження є представлення мережевої інфраструктури досліджуваного об'єкта у вигляді орієнтованого графа, опис цільової функції як безлічі маршрутів на графі і систематизація кібератак у вигляді унарних операцій над графом.

Повнота, чіткість і коректність розробленої моделі підтверджується сформульованою і доведеною теоремою

Переваги способу:

- простота і наочність зі збереженням принциповості системи;
- мала вимогливість до ресурсів;
- велика існуюча алгоритмічна база;

Недоліки способу:

- недостатня варіативність-або доводиться додавати нові операції/функції/способи відображення, або видозмінювати і доповнювати використовуваний математичний апарат;
- поява складних залежностей у разі гетероморфності (різнорідності) структури-наприклад, TG-графи, в яких дуже велика важливість напрямку обміну даними, тип і кількість існуючих реберних зв'язків

Область застосування: повсюдне застосування елементів даної топології в структурах, не сильно зав'язаних на складних зв'язках і/або відображеннях, стимулює і далі використовувати підхід, описаний в даному методі, проте сам по собі спосіб опису не є вичерпною і часто вимагає доповнень і/або видозмін.

2.3.2 Динамічні графи

Динамічний граф є набір або впорядковане в часі безліч класичних графів, в якому операція переходу від і-го графа до і+1-го графу визначається функцією часу і в найпростішому випадку є простим лінійним відображенням масиву подій в часі на масив графів

Дана графова структура поки не знайшла широкого застосування в забезпеченні безпеки і детектуванні аномалій, однак при належному підході і наявності в системі достатніх обчислювальних р есурсів повинна себе добре зарекомендувати як найпростіший спосіб, що дозволяє реалізувати механізм циклічності обробки даних аналізатором, який описується далі

Додатково варто сказати, що хоч подібний підхід і може вимагати додаткові ресурси в системі на збереження і обробку «зліпків» стану системи в графовому відображенні, на практиці подібне рішення зводиться до зберігання списку змін графової топології для мінімізації витрачається простору на зберігачах інформації.

Переваги способу:

- додатковий ступінь свободи за рахунок використання осі часу;
- можливість більш детального аналізу та навчання аналізатора на основі часових параметрів;
- можливість буферизації даних для циклічного аналізу системи.

Недоліки способу:

- підвищена вимогливість до обчислювальних ресурсів системи;
- необхідність навчати аналізатор, так як інші рішення малозастосовні для подібного підходу;
- необхідність зберігати великі обсяги даних (частково вирішується збереженням не «зліпків» системи, а списків змін у графі).

Область застосування: системи, що вимагають підвищеної надійності і/або мають сильну прив'язку до часу роботи і володіють достатніми

обчислювальними ресурсами, в належній мірі підходять для реалізації подібної описової системи. Окремо варто відзначити, що подібний підхід досить докладно і просто дозволяє описати циклічність роботи аналізуючого пристрою, вимагаючи при цьому не найвищі показники продуктивності системи. Є так званим проміжним варіантом між класичними і сигнальними графовими структурами, внаслідок чого удобопріменім в досить широких областях.

2.3.3 Подієві графи

Такий підхід дозволяє виконати аналіз поведінки програм на основі подій, що генеруються в процесі функціонування системи [ ]. Також представлена архітектура системи, перерахований список подій, відстеження яких виконується на відповідних їм рівнях. Додатково проаналізовано метрики, що дозволяють оцінювати подобу отриманого графа і структури графів заданих додатків. Наводяться результати експериментів, що ілюструють ефективність і точність розробленого підходу.

Варто згадати, що в контексті інформаційної безпеки під подією розуміється будь-яка зміна стану інформаційної системи, що відбивається на стані її безпеки.

Хоча в загальному випадку рішення, використовувані в зв'язці з даним способом опису даних в КФС, не завжди зводяться до оцінок самоподібності, даний підхід застосовується, грунтуючись на трьох критеріях самоподібності системи з алгоритмічним базисом реалізації в структурах з підвищеною складністю топології:

- знаходження максимального загального підграфа двох графів;
- знаходження максимального загального підграфа і мінімального загального надграфа двох графів;
- функція обчислення відстані.

Всі три оцінки розглядаються далі.

Переваги способу:

- обробка різнорівневих подій і цілісність картини дій, що відбуваються в системі;

- фрагментація системи на різні рівні відповідальності;

- відсутність учня аналізатора за рахунок введення критерію самоподібності (варіативно).

Недоліки способу:

- необхідність складання списку подій і, як наслідок, відкритість до вразливостей, що використовують нові тригери / підходи;

- необхідність розробки агентів окремо під кожну зону відповідальності і додаткові перевірки покриття системи всіма зонами відповідальності, що тягне зайві витрати на узгодження роботи агентів зокрема і описової моделі в цілому;

- можлива непрямість аналізу: у наш час задача визначення ізоморфізму двох графів є не вирішеною. Зазвичай аналіз виконується безпосередньо за непрямий ознаками.

Область застосування: складноструктуровані багаторівневі системи, що мають явно нелінійну топологію, наприклад системи з гібридним виконанням мережевої інфраструктури (такі як операційні системи, фізико-біологічні системи зв'язку і взаємодії, вбудовані пристрої, імплантати і так далі), є ідеальними кандидатами для застосування даного способу опису циркулюючих даних через складність стикування різних логічних і/або виконавчих рівнів за допомогою використання інших методів зокрема і своєї фізичної природи в цілому.

2.3.4 Сигнальні графи

Варто згадати сигнальні графи, так як останні мають особливе значення для фізичних систем і їх моделювання. Під сигнальними графами розуміють Зважені орієнтовані графи. Вершини таких графів логічно відображають деякі змінні, які, в свою чергу, описують стану систем і підсистем. Вага кожної з

таких вершин задає функцію часу і / або деякі величини, що відображають відповідну змінну (зазвичай стан підсистеми КФС). У свою чергу, дуги сигнальних графів характеризують зв'язку між змінними: вага кожної такої дуги являє собою відношення (чисельне або функціональне), що описує передачу сигналу від однієї вершини до іншої.

Варто також відзначити, що використання сигнальних графів є досить поширеною практикою в теорії ланцюгів і механізмів, а також в оцінці ризиків, розрахунку можливої кількості відмов системи в одиницю часу і так далі. У слідстві вищесказаного можливість застосування сигнальних графів в області аналізу безпеки і виявлення аномалій вельми перспективна.

Окремо варто загострити увагу на максимальній близькості способів опису фізичних процесів в сигнальних графах до їх реальних витоків і, як результат, зручність використання даного способу в сферах безпеки КФС за рахунок чіткого поділу високорівневої (логічної) і низькорівневої (фізичної) складових системи (ВСС і НСС відповідно) і операцій, що проводяться в системі, з подальшою обробкою і вирішенням завдань різних рівнів зв'язності, як буде показано далі.

Переваги способу:

- нативне відображення фізичних процесів на безліч залежностей в графовій структурі, зручність застосування за рахунок готового математичного апарату, донині успішно застосовується у фізичних і технічних областях зв'язку;
- можливість логічного розбиття і / або паралельної обробки / відображення двох різних типів даних, наприклад ВСС і НСС;
- відсутність обов'язкової прив'язки до часу і, як результат виконання даної умови, знижені вимог до обчислювальних ресурсів системи.

Недоліки способу:

- початкова складність опису системи за рахунок близькості до фізичних процесів;

- підвищена ресурсоємність (за умови додаткового використання циклічності епох роботи заданих фізичних пристроїв та / або аналізатора у випадках необхідності підвищених вимог до безпеки системи, як зазначається далі);

- як і у випадку з усіма іншими графовими структурами, можливість легкого розширення, масштабування і доповнення структури під потреби кінцевого користувача.

Область застосування: складні системи, безпека яких в тому числі повинна грунтуватися на багатоплановому/багаторівневому аналізі двох і більше параметрів зі спільною обробкою даних, що акцентують увагу на фізиці процесів, повинні бути ідеальними кандидатами для застосування даної методології. Як приклади можна розглянути системи АСУ ТП, IIoT, SCADA і так далі.

2.4 Ступінь зв'язності рішення задачі з фізичної точки зору

Наступним кроком логічно розглянути ступінь зв'язності модулів аналізатора або самих аналізаторів. Як прийнято в моделюванні фізичних процесів, завдання обробки даних, що акцентують увагу на зв'язності, зазвичай діляться на 3 типи:

- Segregated Approach (SA) або незв'язане/роздільне рішення;
- Iteratively Coupled Approach (ICA) або часткова/ітеративна зв'язність рішення;
- Fully Coupled Approach (FCA) або повна зв'язаність рішення.

2.4.1 Segregated Approach

Sa системи зазвичай застосовуються при дослідженні та/або моделюванні ряду завдань, що акцентують увагу на будь-яких конкретних явищах, чий зв'язністю з побічними явищами можна знехтувати або розрахувати окремо, не

враховуючи додаткові залежності з побічними явищами. У нашому випадку подібне рішення не має місця, так як потрібно враховувати в належній мірі обидві складові, які практично завжди є пов'язаними як мінімум одностороннім відображенням типу HCC → BCC

2.4.2 Iteratively Coupled Approach

Підхід, що описує ICA рішення, зазвичай заснований на послідовному ітеративному вирішенні ряду завдань. На I-му кроці виконується рішення задачі X. отримані дані надходять в задачу Y, з Y в Z. після закінчення послідовних розрахунків на i-му кроці дані з задачі Z надходить знову в задачу X, а крок i стає i+1.

У разі ICA рішення виникає проблема, схожа з описаною в частині sa підходу. Так, наприклад, складність прийняття рішень зв'язковими модулями може бути нівельована створенням проміжного блоку, що приймає сигнали від обох модулів, аналізує дані і реалізує вибір за обома типами вихідних даних, однак, знову ж таки, в своєму роді даний аналізатор представляє 3 модуля: BCC, HCC і вирішувач. Видозміна будь-якого з модулів просто і легко робиться за рахунок модульності розробки, проте все одно доведеться переписувати всі модулі, так як, наприклад, модуль BCC може мати нові дані, які вважаються аномалією тільки при виникненні схожих флуктуацій в модулі HCC, а HCC модуль буде вважати ці флуктуації нормальними, так як не була оновлена його прошивка.

До того ж ітеративний підхід сильно позначається на продуктивності за рахунок неможливості розпаралелювання рекурентних обчислень.

2.4.3 Fully Coupled Approach

Рішення FCA моделі-найбільш просте, зручне для доопрацювання і використання, що демонструє максимальну точність. По суті, воно являє собою

єдиний аналізатор, який, безсумнівно, так само складно переписувати, як і ІСА, але його простіше сертифікувати, оновлювати залежні бази даних (якщо вони є) і використовувати на практиці, так як немає необхідності в узгодженні модулів. Також такий підхід менше навантажує систему , що часто є критичним критерієм при виборі підходу до вирішення в реальних завданнях промислового масштабу, проте за це кінцевому користувачеві доведеться розплатитися малою енергоефективністю аналізатора.

2.5 Підсумок по вибору способів подання даних

Виходячи з усього вищесказаного, при розгляді КФС, до яких пред'являються підвищені вимоги в сфері інформаційної та фізикотехнічної безпеки, рекомендується акцентувати увагу на FCA зв'язності завдання за рахунок необхідності обліку обох рівнів зв'язку.

Як способів, найбільш відповідних принципу FCA, можна виділити подієві графи, сигнальні графи і їх поєднання з багатовимірними часовими рядами. Перші рекомендується використовувати в разі підвищеної складності топології та/або можливості проблем обробки стикувальних зон системи за рахунок гетероморфності структур другімі методами; інакше ж, у випадках розгляду класичних КФС типу АСУ ТП, IIoT і SCADA, рекомендується застосовувати сигнальні графи через їх більшої пристосованості до реалізації циклічної обробки даних, описуваної в наступній статті. Для повноти аналізу, збереження прінціпа глибини зв'язків оброблюваних даних і поширення зворотних зв'язків рекомендується використовувати перераховані графові структури з багатовимірними часовими рядами.

Для більшої наочності основні переваги і недоліки способів подання, додаткові примітки та інші матеріали введені в таблиці 2.1 і таблиці 2.2

Таблиця 2.1 - Основні риси способів подання даних

Завдання\спосіб представлення	Часові ряди	Алгоритм Калмана	ДВП	Фрактали	Графи
Варіативність	+	+	+	-	+
Короткострокові атаки	+	+	+	+	+
Довгострокові атаки	+/-	+	-	+	+
Ручна настройка	+	-	-	+	+/-
Агрегація даних	+/-	+	+	+	+/-
Продуктивність	+/-	+	+	+	+/-
Облік нелінійних процесів	+/-	+	-	-	+/-
Облік топології системи	+/-	-	-	+	+
Уніфікація завдання	+	-	-	-	+/-

Таблиця 2.2 – Основні риси графових структур

Завдання\вид графів	Класичні	Динамічні	Подієві	Сигнальні
Варіативність	+	+	+	+
Короткострокові атаки	+	+	+	+
Довгострокові атаки	+	+	+	+
Ручна настройка	створення нового способу	-	+	+
Агрегація даних	можлива	можлива	+	можлива
Продуктивність	+	-	+	-
Облік нелінійних процесів	-	+	-	+
Облік топології системи	+	+	+	+
Уніфікація завдання	+	+	+/-	+

2.6 Методи детектування мережевих атак

Аналізуючи механізми, засоби і математичний апарат, використовувані в методах детектування мережевих атак, спрямованих на КФС, принципово можна виділити наступні підходи:

- оцінка критеріїв самоподібності системи;
- прогнозування стану системи на основі статистичних інструментів;
- прогнозування стану системи на основі машинного навчання.

Тут і далі наводяться основні методи детектування мережевих атак, засновані на досліджених в минулому розділі способах подання даних, з виділенням їх переваг, недоліків і областей застосування.

2.6.1 Оцінка критеріїв самоподібності системи.

Велика частина описаних раніше способів легко і зручно застосовується у випадках КФС зі слабо вираженою гетероморфностью структури, згідно з описаними раніше переваг, логічно використовувати механізм оцінки критеріїв самоподібності структури, так як, як зазначалося авторами раніше, при належному дотриманні досить жорстких умов, виграш від використання подібних рішень часто переважує строгість і жорсткість структури системи.

Так, наприклад, в роботі [32] авторами в якості критерію самоподібності використовується показник Херста: $\frac{R}{S} = (\frac{N}{2})^H$, де H-показник Херста, R-розмах перших n значень ряду, S – стандартне відхилення.

З [33, 34] відомо, що процес вважається самоподібним, якщо виконується наступна нерівність: $0.5 \leq H \leq 1$.

2.6.2 Прогнозування стану системи на основі статистичних інструментів

Унаслідок існування відмінностей між перевагами і недоліками серед механізмів передбачення стану системи на основі статистичних інструментів, дані підходи будуть окремо розглянуті в наступних розділах. Особливу увагу в даному аналізі буде приділено складності математичного апарату і

застосовності теорії ймовірностей укупі з одержуваної результативність від застосування даних підходів.

Аналіз буде проводитися для пошуку точок розкладання на основі байєсівського онлайн алгоритму і використання коефіцієнтів множинної кореляції. Причини вибору-найбільша популярність і найвища ефективність серед вивченого безлічі інструментів статистичного аналізу і теорії ймовірностей в сфері забезпечення безпеки

Зважаючи на схожість підходів, їх переваг, недоліків і областей застосування, останні будуть перераховані разом, узагальнюючи даний розділ.

2.6.3 Пошук точок розкладання на основі байєсівського онлайн алгоритму

Серед методів виявлення розкладання (розкладання - розбіжності очікуваної випадкової величини з отриманою на основі прогнозування більш ніж на якусь величину N) зазвичай прийнято використовувати алгоритми і методи, засновані на формулі Байеса. Так, наприклад, у дослідженні [35] авторами розглядається можливість використання адаптованого байєсівського онлайн-алгоритму для виявлення точок розкладання. Даний метод цілком логічно використовувати і для виявлення мережевих атак та інших аномалій в мережах в дослідженні [36]. Авторами відзначається досить високі показники ефективності і невисокою ресурсоємності при використанні в якості заміщення методу більш «дорогих» підходів, що аналізують трафік магістральних мереж Інтернет.

Байєсівський онлайн-алгоритм будується на обчисленні розподілу довжин прогону щодо вхідних даних, згідно [37]. Основа алгоритму полягає у використанні формули Байєса:

$$\mathrm{P}(r_t \mid x_{1:t}) = \frac{P(xt \mid rt-1, x1{:}t-1) \sum P(rt \mid rt-1) P(rt-1 \mid x1{:}t-1)}{\mathrm{P(x1{:}t)}},$$

де кожна з ймовірностей, що використовуються в даній формулі, визначається наступним чином:

1. P($xt|rt$-1, $x1:t$-1) – імовірність того, що нові надходження даних задовольняють параметрам розподілу в поточній довжині прогону;

2. $P(rt|rt$-1) – імовірність того, що довжина пробігу або зростає з приходом нових даних, або виникає точка розладу;

3. $P(rt$-1$|x1:t$-1) – значення, обчислені на попередньому кроці.

2.7 Прогнозування стану системи на основі машинного навчання

У сфері безпеки КФС з машинного навчання, згідно з емпіричним досвідом авторів і сформованим практикам даної області, перевага зазвичай віддається нейронним мережам різних конфігурацій (NN, RU, SM, AE, NTM та інші) і еволюційним алгоритмам (переважно генетичним, але також всречаются, наприклад, PSO, ABC, ACO та інші). До слова, хоч нейронні мережі і зайняли домінуюче становище в цій області, а генетичні алгоритми знайшли меншу популярність за рахунок можливих проблем з подоланням локальних екстремумів, однак, наприклад, в роботі [47] генетичні алгоритми показали свою ефективність.

Зважаючи на велику кількість видів і методів машинного навчання, варто відзначити базові підходи прогнозування.

Принципово обрана нейронна мережа або генетичний алгоритм позначаються лише на точності, швидкості і вимогливості до ресурсів роботи аналізатора за рахунок своїх внутрішніх пристроїв і прийнятих рішень. Акцент же, в свою чергу, віддається нейронних мереж за рахунок максимальної гнучкості настройки аналізатора для кожної конкретної теоретикоопісательной і фізичної моделі. Так, наприклад, в роботі [16] докладно описуються причини

виборів конфігурації нейронної мережі, шари, ступінь просіювання та інші необхідні деталі.

Загальний підхід вирішення сформульованих раніше завдань полягає в отриманні даних від системи, зіставленні/перетворенні/відображенні їх відповідно до прийнятої моделі і подальшому передбаченні майбутнього стану системи. Отриманий результат порівнюється з нинішнім станом. У разі різниці результатів, що перевершує якесь порогове значення, поведінка системи вважається аномальним.

Переваги рішення:

- висока варіативність застосовуваних конструкцій і, як результат, широкий вибір між швидкістю, якістю і вимогам до ресурсоємності системи;
- можливість найбільш глибокого і надійного детектування аномалій в НСС за допомогою застосування циклічності аналізатора, описаного далі;
- можливість реалізації найбільш глибокого аналізу і підвищеного рівня безпеки системи. Недоліки рішення:
- первісна складність настройки аналізатора;
- необхідність навчання системи;
- апріорі підвищені вимоги до ресурсів системи в порівнянні з усіма іншими рішеннями;
- неможливість перенесення навченої моделі на нову топологію (на відміну від більшості інших підходів), необхідність перенавчання.

Область застосування: дане рішення застосовується з усіма названих раніше способами подання даних КФС, що допускають будь-яку ступінь гомоморфності (однорідності) структури системи, проте до останньої пред'являються підвищені вимоги в області ресурсоємності. У разі необхідності підвищеної чутливості до аналізу НСС дане рішення легко доповнюється механізмом циклічності. Найбільш зручні сфери застосування ті, що не мають на увазі частого зміни топології мережі з точки зору зміни проектів і/або їх реалізації. Наприклад, функціонуючі АСУ ТП, SCADA і IIoT

2.7.1 Реалізація циклічності аналізатора на основі нейронних мереж

Також варто відзначити, що існують конфігурації нейронних мереж, здатні передбачати не тільки одиничне майбутнє значення системи, але і, за рахунок буферизації тимчасових змінних, періоди. Даний підхід дозволяє вирішити задачу циклічності аналізатора, а саме навчати нейронну мережу не послідовним набором даних, дискретизованим, наприклад, за часом, а набором даних, відповідним певному циклу роботи безлічі пристроїв.

Очевидно, що цикл навчання, тобто. ширину вибірки тимчасового інтервалу варто задавати по найбільшому часу $t\ k$ одного циклу з усіх пристроїв, що розглядаються в системі, якщо даний цикл пристрою є $GCD(t1\ ,t2\ ,\ ...,tn\ ) = tk$ (НОД($t1\ ,t2\ ,\ ...,tn\ ) = tk$, найбільшим загальним дільником), в іншому випадку тривалість циклу задається добутком тривалостей циклів N пристроїв таким чином, щоб до кінця циклу всі пристрої повернулися в свій початковий фізичний стан, інакше відбувається накладення багатовимірних кривих циклів пристроїв і викликається помилкове детектування аномалії.

Такий підхід дозволить знаходити фізичні аномалії НСС за відсутності прояву аномальної поведінки у високорівневій складовій системи (ВСС) навіть у тих випадках, коли аномалія на рівні НСС не була виявлена механізмами статистичного аналізу або критеріями самоподібності. Наприклад, можна розглянути процес загартування в металургійному цеху. Так температура, покладемо, індукційної печі за час циклу становить складну криву через певного технічного процесу, яка ні в якому разі не може бути порушена з причин усадки металу або інших фізичних явищ, що виникають в разі відхилення роботи системи від заданого алгоритму. У разі порушення даних ВСС продовжує вважатися легітимним за рахунок компенсованого зміни величин (покладемо, Рівного циклічного відхилення від середньої величини багатовимірної кривої, зміненої якимось шумом із середнім значенням, які

прагнуть до нуля, доданим шкідливим програмним забезпеченням), проте фізичний процес порушується. Так, наприклад, зберігається умовна лінія тренда або усереднені за певний період значення, однак, в разі використання циклічності аналізатор здатний виявити відхилення конкретних фізичних пристроїв від заданої багатовимірної кривої і виявити аномалію.

Висновки до розділу 2

Виходячи з усього вищесказаного, при розгляді КФС, до яких пред'являються підвищені вимоги в сфері інформаційної та кіберфізичної безпеки, при наявності достатньої обчислювальної потужності, авторами рекомендується акцентувати увагу на рішеннях, заснованих на застосуванні машинного навчання зважаючи підвищеної варіативності і можливості застосування механізму циклічності аналізатора для додаткового глибокого аналізу НСС з метою максимізації безпеки системи.

В інших випадках, наприклад, при неможливості виконання критеріїв достатньої обчислювальної ресурсоємності системи і/або можливості допущення або тільки короткочасних, або тільки довготривалих атак, допустимі застосування як статистичних інструментів вирішення поставлених завдань, так і використання критеріїв самоподібності. Останні, в свою чергу, рекомендуються саме у випадках малої гетероморфності системи для більшої ефективності і надійності, або у випадках мультифрактальності, коли можна окремо застосувати критерії для кожної підсистеми, або при необхідності детектувати різні за тривалістю аномалії, але неможливості використовувати машинне навчання.

В окремих окремих випадках, наприклад, в разі розгортання системи з периферійні обчислення, створення DTN мереж, мереж військовооперативного призначення та інших особливих випадків, рекомендується використовувати модифікації графових структур через легкість перетворення останніх. Таке рішення дозволить забезпечити максимальну гнучкість і прив'язку до вельми

вузьконаправленим завданням в системі з чр езмерно високою гетероморфностью. У разі малої обчислювальної здатності або існування великої затримки рекомендується в таких мережах використовувати статистичні інструменти аналізу станів проміжних пристроїв і логічних вузлів.

РОЗДІЛ 3
ВДОСКОНАЛЕННЯ МЕТОДУ ЗАХИСТУ МЕРЕЖ ІНТЕРНЕТУ РЕЧЕЙ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

Провівши ретельний аналіз існуючих способів подання даних, їх переваг, недоліків, областей застосування, а також існуючих методів детектування мережевих атак, можна перейти до безпосереднього вдосконалення методу детектування мережевих атак.

Описуваний метод буде грунтуватися на обробці отриманих часових рядів з актуаторів і сенсорів, використанні модифікованого алгоритму NEAT-гіперкуба для передбачення подальшого стану системи, і обчисленні помилки між передбаченим і реальним значеннями.

Безпосередньо сам алгоритм роботи NEAT-гіперкуба заснований на симбіозі двох інших механізмів: нейронних мереж і генетичних алгоритмів, що виконують конфігурацію нейронної мережі. Основні моменти реалізації будуть описані в наступних розділах.

Тестування створеного і реалізованого методу детектування мережевих атак на КФС буде виконуватися на наборі даних TON_IOT DATASETS.

3.1. Первинна обробка даних

Початкові Дані, отримані з ton_iot DATASETS вибірки, були представлені в csv файлах.

Виконана обробка та агрегація даних включала наступні кроки:

1. Уніфікація часу подання даних за часом з кроком в 1 секунду (виконана нормування за часом: усереднення даних, отриманих за проміжок в 1 секунду, якщо такі існують, або дублювання даних в разі відсутності таких за проміжок в 1 секунду).

2. Присвоєння ідентифікаційних номерів кожному з 7 пристроїв (id, нумерація виконувалася довільно, проте зі збереженням логічного зв'язку «відправник-одержувач»).

3. Зміна показників стану для тих пристроїв, які не могли вимірювати ступінь своєї «завантаженості», але підтримують можливість вимірювати ступінь розряду живильного пристрою (в даному випадку враховувалася швидкість розряду за 1 годину; для зручності даний показник був зведений до зміни процентного стану заряду в секунду).

3.2 Вибір способу представлення даних

Зважаючи розглянутих раніше переваг і недоліків існуючих методів подання даних, було вирішено зупинитися на використанні багатовимірних часових рядів. Основні причини даного вибору: варіативність застосовуваних аналізаторів даних, можливість ручного налаштування гіперпараметрів вирішувача, а також високий ступінь варіативності методу – можливість застосовувати в гетерогенних системах різного типу. У разі використання багатовимірних часових рядів зазвичай проводиться навчання нейронної мережі на валідних даних для передбачення майбутнього стану системи і обчислення різниці (помилки) між передбаченим і реальним станом. Шляхом аналізу помилки проводиться детектування аномальних станів в системі.

До складу розглянутої IoT системи увійшло 7 пристроїв, причому кожен мав 4 базисні складові, тобто розмірність багатовимірного часового ряду склала 28.

3.3 Причини вибору та модифікації алгоритму сімейства NEAT

Виходячи з проведених раніше досліджень стає очевидним, що багатовимірні часові ряди зазвичай використовуються з нейронними мережами з огляду на те, що останні показали досить високу точність детектування

мережевих атак укупі з використанням цього способу опису даних в КФС. Щоб уникнути перераховані раніше недоліки використання нейронних мереж, а саме первісну складність настройки аналізатора і складність складання топології нейронної мережі, було вирішено використовувати нейроеволюціонний алгоритм NEAT-гіперкуб.

NEAT (NeuroEvolution of Augmenting Topology) – це генетичний алгоритм для створення нейронних мереж, що розвиваються. Даний метод був розроблений в Остіні, в Техаському університеті. Принцип роботи алгоритму зводився до зміни ваг і двомірної структури нейронної мережі – пошуку найбільш оптимального значення методами генетичних алгоритмів.

Окремо варто відзначити можливість модульного виконання neat алгоритмів. Так як реалізація методу зводиться не тільки до виставлення заданих гіперпараметрів, виконавець має можливість конфігурувати як дані, використовувані для обробки нейронною мережею, саму нейронну мережу, так і модифікувати генетичну складову алгоритму під свої потреби, щоб створювати топологію саме тієї спрямованості, яку вимагає завдання

NEAT-гіперкуб (Hypercube-based NEAT) – це генеративне кодування, яке розвиває штучні нейронні мережі з інципами широко використовуваного алгоритму NEAT. Це новий метод розвитку великомасштабних нейронних мереж з використанням геометричних закономірностей предметної області. Він використовує мережі створення композиційних шаблонів (CPPN, Compositional pattern-producing networks).

Мережі створення композиційних шаблонів (CPPN) - це різновид штучних нейронних мереж, архітектура яких визначається генетичними алгоритмами.

У той час як нейронні мережі часто містять саме сигмовидні і гауссовские функції активації, CPPN зазвичай базуються на більш складних функціях, так як перші не здатні повною мірою вирішити задачу оптимізації. Вибір функцій для канонічного набору може бути зміщений в бік певних типів шаблонів і закономірностей. Наприклад, періодичні функції, такі як синус, створюють

сегментовані шаблони з повтореннями, тоді як симетричні функції, такі як гауссова, створюють симетричні візерунки. Лінійні функції можуть бути використані для створення лінійних або фрактальних візерунків. Таким чином, архітектор системи генетичного мистецтва на основі CPPN може змінювати типи генеруються нею патернів, вибираючи набір канонічних функцій, які необхідно включити.

Крім того, на відміну від звичайних нейронний мереж, мережі композиційних шаблонів зазвичай можуть бути застосовні до всіх можливих вхідних даних, так що вони можуть являти собою повну структуру. Оскільки вони є композиціями функцій, CPPN фактично кодують структури з нескінченною роздільною здатністю і можуть бути дискретизовані для конкретного рішення з будь-якою оптимальною роздільною здатністю.

У загальному випадку алгоритм NEAT-гіперкуб працює з вхідною, вихідною сітками і проміжними шарами, сконфігурованими Користувачем, однак такий підхід не дозволяє повною мірою автоматично конфігурувати топологію кінцевої нейронної мережі.

3.4 Принцип роботи методу детектування мережевих атак

Описуваний метод грунтується на обробці отриманих багатовимірних часових рядів, складених з даних передбаченні майбутнього стану системи засобами модифікованого нейроеволюційного алгоритму NEAT-гіперкурб і аналізі виникаючих помилок – розбіжності між реальними значеннями стану системи і передбаченими.

Методів включає в себе 2 етапи – підготовчий і робочий. Підготовчий етап націлений на автоматичне конфігурування оптимальної топології нейронної мережі і має на увазі під собою наступні кроки:

1. Підготовка тестових даних-нормалізація і складання багатовимірних часових рядів.

2. Передача отриманих багатовимірних рядів на вхід нейронної мережі, первинно налаштованої Користувачем.

3. Навчання нейронної мережі на переданих даних і її реконфігурація генетичної складової нейроеволюційного алгоритму до тих пір, поки не буде отримана задана точність на тестових даних.

Робочий етап має на увазі під собою безпосереднє детектування мережевих атак, спрямованих на КФС, і включає в себе наступні кроки:

1. Підготовка реальних даних функціонуючої КФС-нормалізація і складання багатовимірних часових рядів.

2. Передача отриманих багатовимірних рядів на вхід нейронної мережі, оптимально сконфігурованої генетичної складової нейроеволюційного алгоритму.

3. Передбачення майбутнього стану системи нейронною мережею на основі отриманих багатовимірних часових рядів.

4. Обчислення помилки між передбаченим станом системи і реальним

5. Фіксування наявності або відсутності атак на КФС на основі отриманої помилки.

Схема роботи методу представлена на рисунку 3.1

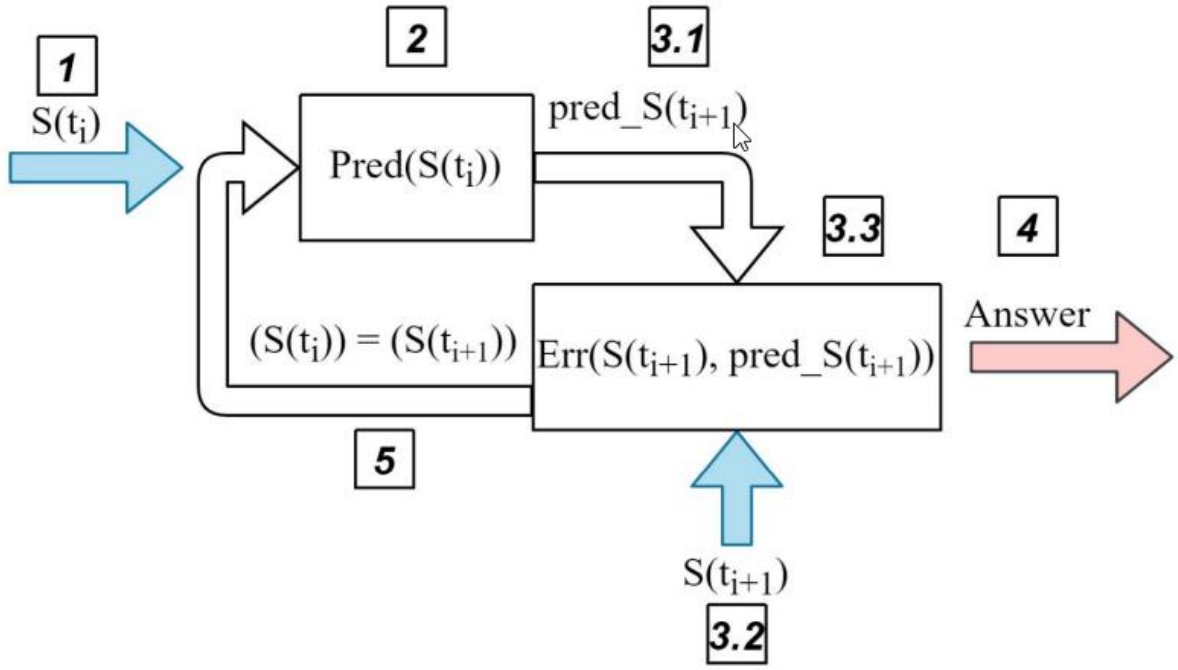

Рисунок 3.1 - Принципова схема роботи методу детектування мережевих атак

На етапі 1 сформований багатовимірний часовий ряд S(ti) від часу ti подається на вхід нейронної мережі. На етапі 2 виконується операція передбачення майбутнього ряду pred_S(ti+1) на основі ряду s(ti), де Pred() – функція передбачення, що виконується нейронною мережею. На етапах 3.1 і 3.2 прогнозований багатовимірний ряд pred_S(ti+1) і багатовимірний ряд, отриманий з реальних показників системи S(ti+1), надходять в блок порівняння. На етапі 3.3 обчислюється різниця між показниками і відбувається накопичення помилки. На етапі 4 на основі порівнюваних даних в блоці 3.3 ми отримуємо відповідь про наявність або відсутність атак. На етапі 5 Значення багатовимірного часового ряду від часу ti замінюється значеннями від часу ti+1, після чого алгоритм повторюється.

3.4.1 Прогнозування стану системи

Як вже раніше зазначалося, дані, що пройшли процедуру нормалізації, повинні бути піддані попередній обробці: для кожної точки часового ряду визначається спрогнозоване значення.

Для передбачення подальшого значення стану системи через часовий ряд необхідно виконати операцію: $\gamma pred = pred(xt\text{-}1\ ,\ldots,\ xt{-}n)$

Операція передбачення виконується засобами нейронної мережі, сконфігурованої алгоритмом гіперкуба.

3.4.2 Врахування помилок передбачення стану системи і фіксування наявності атак на систему

Стани системи, передбачені нейронною мережею, можуть в деякому ступені відрізнятися від реальних значень, тому необхідно враховувати помилку – можливу різницю між показниками. Для розрахунку помилки між передбаченим станом системи і реальним виконується наступний ряд дій:

обчислення різниці між передбаченим $\gamma\ pred$ і реальним $\gamma\ real$ значенням: $errt = |\gamma pred - \gamma real|$;

- фіксування наявності або відсутності атаки на основі умови перевищення значення помилки реального станів і передбаченого більш ніж на фіксовану величину: $MAX(errt\ ) > T$, де T - порогове значення прояву аномальної поведінки в системі.

Однак виникає ймовірність помилкових спрацьовувань за рахунок короткочасних "викидів" великих помилок передбачення в малі проміжки часу, тому необхідно врахувати усереднену помилку за деякий проміжок часу: $ERRi = \langle \sum MAX(errt\ )\ i\ t{=}i{-}k \rangle > T$.

3.5 Реалізація розробленого методу

Програмна реалізація була виконана засобами мови Python. Первинна обробка даних виконувалася стандартною бібліотекою, що дозволяє працювати з файлами формату «.csv». Створення багатовимірних часових рядів виконувалося за допомогою математичної бібліотеки Pandas.

Нейромережа будувалася і навчалася за алгоритмом модифікованого гіперкуба з використанням бібліотеки NEAT-Python мови Python. Бібліотека NEAT-Python використовує набір гіперпараметрів, які впливають на виконання та точність алгоритму NEAT.

У роботі використовувалися нижчеописані гіперпараметри (наведені найбільш важливі):

1. Функція активації всіх вузлів мережі є сигмоїдальною, а входи вузлів агрегуються функцією суми: activation_default = sigmoid, aggregation_default = sum. Даний вибір обумовлений прагненням виділити слабкі сигнали і постаратися уникнути насичення і перенасичення від сильних сигналів.

2. Тип закодованої мережі-повнозв'язна мережа зворотного поширення: feed_forward = True, initial_connection = full_direct. В основі даного

рішення лежить бажання оптимізувати швидкість сходження нейрогенетичного алгоритму топології нейронної мережі. Небажання використовувати топологію мережі зі зворотним розподілом в початковій субстраті обгрунтовано відсутністю необхідності відновлення форми і частоти первинних даних. Однак в подальшому субстрат АТО р азрешено еволюціонувати до мережі зі зворотними зв'язками, що і спостерігається на практиці.

3. Під час еволюції нові мережеві вузли та зв'язки додаються та/або видаляються з певною ймовірністю. Очевидно, ймовірність додавання і видалення вузлів була виставлена з більш низьким значенням, чому у ймовірності появи і видалення зв'язків. Дане рішення обгрунтовано желаніемоптімізіровать топологію мережі саме засобами поширення взаємопов'язаних даних, появою і наявністю зворотних зв'язків, мінімізацією створення «Мертвих» вузлів.

4. Імовірність додавання / видалення вузла-node_add_prob = 0.05, node_delete_prob = 0.05.

5. Імовірність додавання / видалення зв'язку-conn_add_prob = 0.3, conn_delete_prob = 0.3.

Усі зв'язки ввімкнені за замовчуванням з дуже низькою ймовірністю відключення через мутацію: enabled_mutate_rate = 0.01. Хоча топологія нейронної мережі і генерується відносно довільним чином, не варто відмовлятися від операцій просіювання і «drop-шарів». Введення мутації відключення довільних вузлів і / або зв'язків дозволяє позбутися від паразитних непрямих залежностей, які не завжди здатні позначитися на прогнозі стану системи позитивно. Знову ж таки, в разі вдалого виникнення зворотних зв'язків не паразитичного характеру, фітнес-функція не дозволить загинути популяції з настільки вдалою мутацією

Щоб стимулювати різноманітність видів, задамо сильний вплив надлишкових/непересічних частин батьківських геномів на відстань між геномами: параметри відстані між геномами – compatibility_disjoint_coefficient = 1.0. Дане рішення дозволяє спочатку створити максимально можливу

псевдовипадкову популяцію особин для подальшого кросовера. В іншому випадку, при створенні ідентичних особин, доводиться чекати додатковий час виникнення стартових мутацій, необхідний для успішних операцій кросовера

3.6 Оцінка точності методу

Тестування програмної реалізації створеного методу виконувалося на вибірці даних TON_IOT DATASETS.

Порогова величина помилки визначається емпірично, і в даному випадку (на даному досліджуваному датасеті) величина T була встановлена в значення 0,398. При даному пороговому значенні були розраховані наступні величини:

1. Accuracy (наскільки близький результат вимірювання до справжнього значення) = $(TP + TN) / (P + N)$.

2. Precision (наскільки близькі вимірювання одного і того ж об'єкта др уг до одного) = $TP / (TP + FP)$.

3. True Positive Rate = $TP / (TP + FN)$.

4. True Negative Rate = $TN / (TN + FP)$.

5. False Positive Rate = $FP / (FP + TN)$.

6. False Negative Rate = $FN / (FN + TP)$.

7. Positive Predictive Value = $1 - FP / (FP + TP)$.

8. Negative Predictive Value = $TN / (TN + FN)$.

9. F1 Score = $2TP / (2TP + FP + FN)$.

В даному випадку: 1. TP-кількість вірних детектувань нормального стану системи (True Positive). 2. TN-кількість вірних детектувань атак на систему (True Negative). 3. FP-кількість нерозпізнаних атак (False Positive). 4. FN-кількість нормальних станів системи, розпізнаних як атаки (False Negative). 5. P-загальна кількість нормальних станів КФС (Positive). 6. N-Загальна кількість станів КФС, що включають в себе атаки (Negative)

Далі наводяться значення для всіх розглянутих часових проміжківв вигляді таблиць. Для зручності, значення розбиті за типами атак і по розглянутих тимчасових інтервалах. Після кожного часового проміжку, а також після всіх розрахунків і викладок, слідують короткі висновки про точність розробленого методу.

Таблица 3.1-Отримані дані на відрізку «DoS атаки». 48 годин

All	1209600
Positive	394496
Negative	815104
True Positive	363748
True Negative	760838
False Positive	54266
False Negative	30748

Таблица 3.2-Точність методу на відрізку «DoS атаки». 48 годин

Accuracy	0,9297
Precision	0,8702
True Positive Rate	0,9221
True Negative Rate	0,9334
False Positive Rate	0,0666
False Negative Rate	0,0779
Positive Predictive Value	0,8702
Negative Predictive Value	0,9612
F1 Score	0,8954
Matthews Correlation Coefficient	0,8433

Аналізуючи проміжок DoS, можна сказати, що тут метод показав себе позитивно. Дані слова підтверджую як висока близькість рішень (Accuracy), так і висока загальна точність класифікації (Precision). Окремо варто відзначити наступні величини: False Positive Rate і False Negative rate – їх значення склали менше 0,1 і знаходяться дуже близько один одному. Дані показники свідчать про те, що частота помилкових детектувань становить малу частку від загальної, а перенесення значень в сторону FP або Fn відсутня.

Таблица 3.3-Отримані дані на відрізку «DDoS атаки».48 годин

All	1209600
Positive	486456

Negative	723144
True Positive	451619
True Negative	667022
False Positive	56122
False Negative	34837

Таблица 3.4-Точність методу на відрізку «DDoS атаки». 48 годин

Accuracy	0,9248
Precision	0,8895
True Positive Rate	0,9284
True Negative Rate	0,9224
False Positive Rate	0,0776
False Negative Rate	0,0716
Positive Predictive Value	0,8895
Negative Predictive Value	0,9504
F1 Score	0,9085
Matthews Correlation Coefficient	0,8453

Як і у випадку з проміжком, що включає в себе DoS атаки, метод також добре відпрацював на проміжку ddosатак. Значення загальної точності (Precision) трохи збільшилася, а в іншому можна зробити висновки, аналогічні випадку з DoS – на заданому проміжку метод прекрасно впорався зі своїм завданням.

Таблица 3.5-Отримані дані на відрізку «всі атаки». 144 години

All	3628800
Positive	1662561
Negative	1966239
True Positive	1474275
True Negative	1729175
False Positive	237064
False Negative	188286

Таблица 3.6 -Точність методу на відрізку «всі атаки». 144 години

Accuracy	0,8828
Precision	0,8615
True Positive Rate	0,8867
True Negative Rate	0,8794
False Positive Rate	0,1206
False Negative Rate	0,1133

Positive Predictive Value	0,8615
Negative Predictive Value	0,9018
F1 Score	0,8739
Matthews Correlation Coefficient	0,7647

Окремо, крім часових проміжків, що включають в себе дискретні атаки, варто розглянути точність методу на всіх проміжках, що включають в себе атаках.

Не складно помітити, що загальна точність (Precision) і близькість PPP (Accuracy) зменшилися відносно тих же показників, але у випадках DOS і DDoS атак. Можливі причини даної поведінки - досить рідкісне дублювання відправляються пакетів під час атаки, яке легко загубити на тлі реальної втрати пакетів і легітимного дублювання.

Для отримання більш чіткого уявлення про точність методу був виконаний ROC аналіз. На рисунку 3.2 наводиться апроксимована ROC крива на часовому проміжку «за весь час» за 192 години.

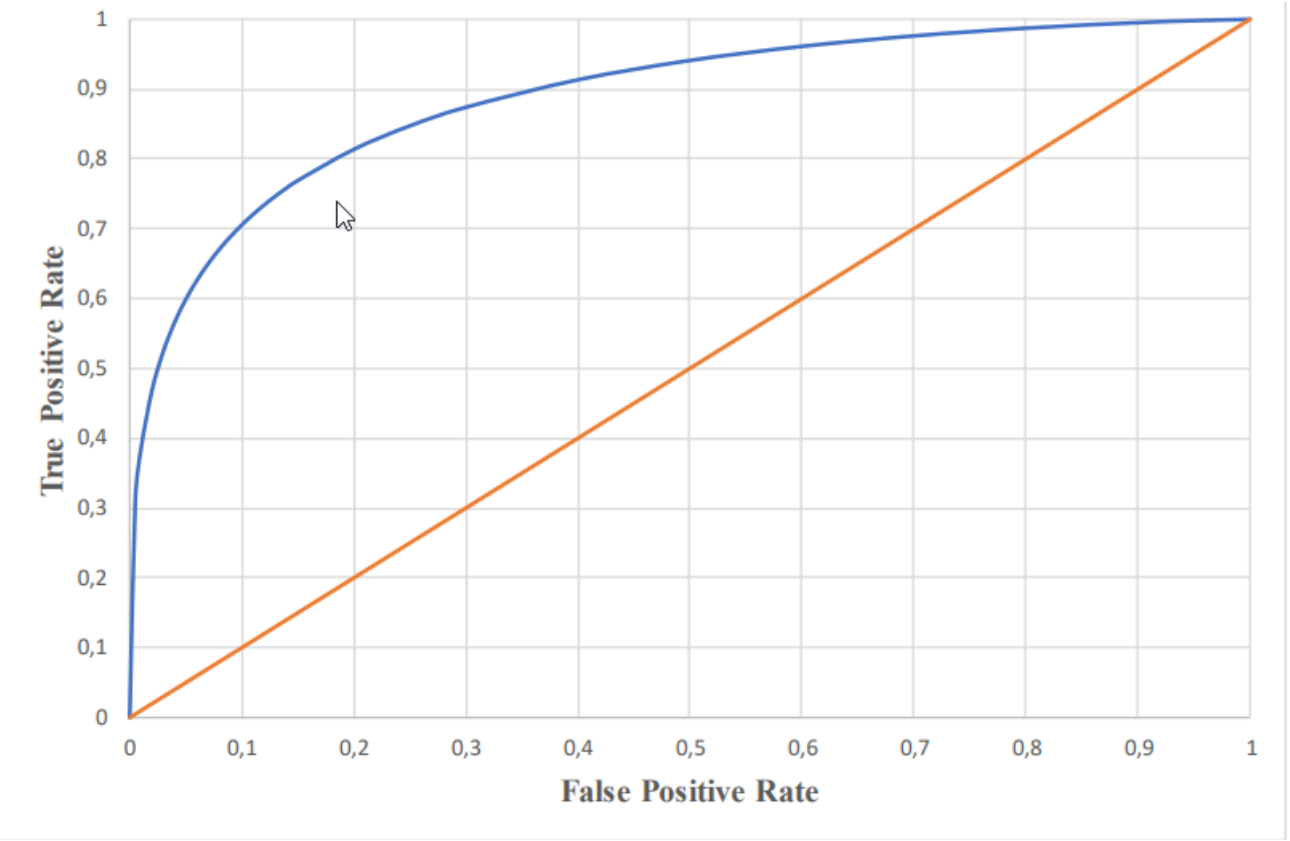

Рисунок 3.2- ROC крива для часового проміжку «за весь час»

Показник AUROC (площа під графіком кривої) можна трактувати як еквівалентність ймовірності того, що бінарний Класифікатор при виконанні оцінки присвоїть більшу вагу випадково обраної позитивній характеристиці або показнику, ніж випадково обраному негативному. В ідеальних умовах даний показник прагне до 1, а в разі рівноймовірнісного «вгадування» на безлічі – до 0,5 (дана пряма представлена на рисунку 3.2 помаранчевим кольором). Для даного випадку показник auroc склав 0,89, що підкреслює достатню точність методу і вдало обрану порогову величину помилки.

ВИСНОВКИ

У наш час IoT є невід'ємною частиною нашого повсякденного життя, використовуючи безліч підключених пристроїв для полегшення різних аспектів нашого життя. Проте зі зростанням кількості та різноманітності цих пристроїв зростає й загроза їхньої безпеки. Ризики, пов'язані з безпекою мереж Інтернету речей, включають в себе можливість втрати конфіденційності даних, порушення приватності, знищення або втручання в роботу систем, а також можливість використання пристроїв для здійснення кібератак.

У роботі я досліджував та вдосконавлював метод захисту мереж Інтернету речей з використанням штучного інтелекту. Метод передбачає використання нейроеволюційного алгоритму сімейства NEAT: модифікований NEAT-гіперкуб.

Виявлення мережевих атак, здійснювалося в кілька етапів:

1. Первинна обробка даних і подання їх у вигляді багатовимірних часових рядів.

2. Конфігурування нейронної мережі генетичної складової NEAT-гіперкуба.

3. Навчання налаштованої нейронної мережі на тестовому наборі.

4. Передбачення майбутнього стану системи на основі поточних даних.

5. Розрахунок помилки між передбаченим і реальним станами системи.

6. Порівняння отриманої помилки з мінімальним пороговим значенням T.

Тестування проводилося на наборі даних TON_IOTDATASETS. Отримані загальна точність (Precision; 0,9152) і близькість рішень (Accuracy; 0,8861), а також величини False Positive Rate (0,1206) і False Negative rate (0,1094) свідчать про відсутність перенавчання моделі і високої надійності даного методу.

Загалом, моя дипломна робота розкрила важливі аспекти безпеки мереж Інтернету речей та розробила ефективні методи їх захисту, що можуть бути

використані для забезпечення безпеки та надійності цих систем у майбутньому. Мої дослідження можуть бути використані як основа для подальших досліджень та реалізації в реальних системах IoT.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бураковський Л.В., Хоменко І.В. Інформаційні технології в системах безпеки життєдіяльності. Київ: Національний технічний університет України «Київський політехнічний інститут», 2016. 201 с.
2. Гаврилюк В.В. Основи кібербезпеки. Київ: Видавництво “Центр учбової літератури”, 2020. С. 127-128
3. Дейнека О.М. та ін. Захист інформації: навч. посіб. Київ: Видавництво НПУ ім. М.П.Драгоманова, 2018. 220 с.
4. Кліменко О.В. та ін. Інформаційна безпека та криптографія: навч. посіб. Київ: Видавничий дім «Київський університет», 2017. С. 60-62
5. Козачук Р.М. Безпека життєдіяльності: навч. посіб. Київ: Видавничий дім “ПРОМІНЬ”, 2019. 207 с.
6. Коломієць С.М. та ін. Комп’ютерна безпека: навч. посіб. Київ: Видавничий дім «Центр учбової літератури», 2018. 152 с.
7. Короп Н.В. та ін. Безпека інформаційних систем: навч. посіб. Київ: Видавництво “Нова книга”, 2017. 195 с.
8. Кононов О.Ю., Кононова О.В. Безпека інформаційних технологій: навч. посіб. Київ: Видавничий дім “Центр учбової літератури”, 2018. 125 с.
9. Луняк О.В. та ін. Інформаційна безпека: навч. посіб. Київ: Видавничий дім “ПРОМІНЬ”, 2020. С. 83-95
10. Морозов О.М. та ін. Безпека життєдіяльності: навч. посіб. Київ: Видавництво “Нова книга”, 2018. 46 с.
11. Науменко В.О. Безпека інформаційних систем: навч. посіб. Київ: Видавничий дім “Центр учбової літератури”, 2017. 113 с.
12. Подвирна О.О., Гладун О.В. Основи кібербезпеки: навч. посіб. Київ: Видавництво “Київський університет”, 2019. 140 с.

13.Пономаренко О.І. Інформаційна безпека в інформаційно-комунікаційних системах: навч. посіб. Київ: Видавничий дім “Центр учбової літератури”, 2018. С. 198-200

14.Решетнік О.В. Комп’ютерна безпека: навч. посіб. Київ: Видавничий дім “Київський університет”, 2017. 249 с.

15.Скоробагатько І.В. Безпека інформаційних технологій: навч. посіб. Київ: Видавництво “Нова книга”, 2019. 112 с.

16.Стасюк В.В. Інформаційна безпека: навч. посіб. Київ: Видавництво “Нова книга”, 2020. 163 с.

17.Трофимов В.Г. та ін. Комп’ютерна безпека: навч. посіб. Київ: Видавництво “Центр учбової літератури”, 2018. 159 с.

18.Федоров Д.В., Корженевський Д.В. Безпека інформаційних систем: навч. посіб. Київ: Видавництво “Нова книга”, 2018. С. 237-239

19.Харламов В.О., Луцька Т.М. Безпека інформаційних систем: навч. посіб. Київ: Видавництво “Нова книга”, 2019. 213 с.

20.Шевченко Ю.М. Основи кібербезпеки: навч. посіб. Київ: Видавництво “Нова книга”, 2017. 107 с.

21.Шульгіна І.І. Інформаційна безпека: навч. посіб. – Київ: Видавництво “Нова книга”, 2018. 141 с.

22.Іванов І.І., Іванова І.І. Комп’ютерна безпека: навч. посіб. – Київ: Видавництво “Центр учбової літератури”, 2019. 209 с.

23.Ігнатова О.В. Інформаційна безпека в інформаційно-комунікаційних системах: навч. посіб. Київ: Видавництво “Нова книга”, 2020. С. 152-155

24.Ільченко О.М. Основи кібербезпеки: навч. посіб. Київ: Видавництво “Нова книга”, 2019. 241 с.

25.Кравченко О.В. Безпека інформаційних технологій: навч. посіб. Київ: Видавництво “Нова книга”, 2020. 56 с.

26.Кузнецов О.М. Інформаційна безпека: навч. посіб. Київ: Видавництво “Нова книга”, 2019. 112 с.

27.Мартиненко В.Г., Петров В.І. Безпека інформаційних систем: навч. посіб. Київ: Видавництво “Центр учбової літератури”, 2018. 216 с.

28.Міненко С.М. Безпека інформаційних систем: навч. посіб. Київ: Видавництво “Нова книга”, 2018. 170 с.

29.Онопрієнко Л.В. Інформаційна безпека в інформаційно-комунікаційних системах: навч. посіб. Київ: Видавництво “Центр учбової літератури”, 2020. 109 с.

30.Поляков В.І. Безпека інформаційних технологій: навч. посіб. Київ: Видавництво “Центр учбової літератури”, 2019. 38 с.

31.Wavelet methods for the detection of anomalies and their application to network traffic analysis Quality and Reliability Engineering International. 2006. №. 8. P. 953-969.

32.Adams R. P., MacKay D. J. C. Bayesian online changepoint detection. arXiv preprint arXiv:0710.3742. 2007. P. 2-12

33.Anderson K. C. A novel approach to Bayesian online changepoint detection. / Boulder: University of Colorado,. 2008. 30 p.

34.Multifractal detrended fluctuation analysis of nonstationary time series. Physica A. 2002. № 316. P. 87–114.

35.Sheluhin O., Atayero A., Garmashev A. Detection of Teletraffic Anomalies Using Multifractal Analysis. International Journal of Advancements in Computing Technology. 2001. Vol. 3. № 4. P. 174-182.

36.Multifractal analysis of soil surface roughness. Vadose Zone Journal. 2007. № 7(2). P. 512–520.

37.Sheluhin O. I., Atayero A. A. Detection of DoS and DDoS Attacks in Information Communication Networks with Discrete Wavelet Analysis. International Journal of Computer Science and Information Security. 2012. P. 53

38.Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent AUV navigation system. 2004. Vol. 12. № 12. P. 1531–1539.

39.Kalman R. E. A new approach to linear filtering and prediction problems // Journal of basic Engineering. 1960. Vol. 82. № 1. P. 35–45.

40.TON_IOT DATASETS. – URL: https://ieeedataport.org/documents/toniot-datasets (дата обращения: 12.01.2024).

41.Grouped Convolutional Neural Networks for Multivariate Time Series. URL: https://arxiv.org/pdf/1703.09938.pdf (дата звернення: 12.01.2024).

42.Nanduri A., Sherry L. Anomaly detection in aircraft data using Recurrent Neural Networks (RNN). Integrated Communications Navigation and Surveillance (ICNS), 2016. IEEE, 2016. P. 521-528.

43.Filonov P., Lavrentyev A., Vorontsov A. Multivariate Industrial Time Series with Cyber-Attack Simulation: Fault Detection Using an LSTM-based Predictive Data Model. NIPS Time Series Workshop, 2016. P. 242-250

44.Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. KDD: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018. P. 387–395

45.Tharini C., Ranjan V. An Energy Efficient Spatial Correlation Base Data Gathering Algorithm for Wireless Sensor Networks. International Journal of Distributed and Parallel Systems (IJDPS), v.2, №3, May, 2011. 57 p.

46.Boulder Tanenbaum A., Wetherall D. Computer Networks. 5th ed. Prentice Hall, 2010. 960 p.

47.Stankovic J.A. Realistic applications for wireless sensor networks. Theoretical Comput. Sci. 2011. P. 835–863.

48. Recommendation Y.2069. Framework of the WEB of Things. ITU-T, July 2012, Geneva. URL: https://www.itu.int/rec/T-REC-Y.2069-201207-I/en (date of access: 12.03.2024).

49.Recommendation Y.2060. Overview of Internet of Things. ITU-T, February 2012, Geneva. URL: https://www.itu.int/rec/T-REC-Y.2060-201206-I (date of access: 12.03.2024).

50.May M. Design of a Wireless Sensor Node Platform. Waikato: University of Waikato. 2012. P. 212-250

51.Markovich N. M., Krieger U. R. Statistical Analysis and Modeling of Peer-to-Peer Multimedia Traffic. Lecture Notes in Computer Science. 2011. Vol. 5233. P. 70–97.

52.Kellmereit Daniel The Silent Intelligence: The Internet of Things. Publisher: DND Ventures LLC, 2013. P. 130-140

ДОДАТОК А

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Тези наукових доповідей:

1. Denys Zymbytskyi, Oleksandr Laptiev. The role and impact of artificial intelligence in ensuring information security. X international conference Information Technology and Implementation (Satellite). Kyiv, 2023. С. 152-154