Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Галузь знань 12 – Інформаційні технології

Спеціальність 123 –Комп'ютерна інженерія

на тему «Метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень»

КвРКІП. 170374.17.17.13 ПЗ

Виконав: студент 2 курсу, група КІ2м-21-1 _____ Космина Ю.І.
Підпис Ініціали, прізвище

Керівник доктор техн. наук, професор _____ Лисенко С.М.
Науковий ступінь, вчене звання Підпис Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КІІС, д.т.н., проф.
Т.О. Говорущенко
11 05 2023 р.

Хмельницький, 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Комп'ютерної інженерії та інформаційних систем

Освітній рівень магістр

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

Освітня програма освітньо-наукова програма «Комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

"01" 09 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Кобелю Костянтину Олександровичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень

Керівник проекту (роботи) Лисенко С.М., д.т.н., професор

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 09.01.2023 р. № 1

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2023 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Огляд методів виявлення вторгнень в інтернеті речей

Модель функціонування системи керування іот-інфраструктурою під час несанкціонованих вторгнень

Метод синтезу апаратно-програмних засобів керування іот-інфраструктурою під час несанкціонованих вторгнень

Реалізація апаратно-програмних засобів керування іот-інфраструктурою під час несанкціонованих вторгнень

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КІІС		
Антиплагіат	Нічепорук А.О., доцент кафедри КІІС		

7. Дата видачі завдання «_06_»_____09_____2022р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики КвРМ з керівником	05.09.2022	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	05.10.2022	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	05.11.2022	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	05.12.2022	виконано
5	Робота над науковою статтею	05.01.2023	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2022	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	05.04.2023	виконано
8	Оформлення пояснювальної записки згідно вимог	15.04.2023	виконано
9	Попередній захист ДРМ	18.04.2023	виконано
10	Захист ДРМ на засіданні ЕК	До 10.05.2023	

Студент _____ Ю.І. Космина
Підпис Ініціали, прізвище

Керівник роботи _____ С.М. Лисенко
Підпис Ініціали, прізвищ

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Автор роботи: Космина Ю.І..

Керівник роботи: Лисенко С.М..

Пояснювальна записка: 79 с., 15 рис., 7 табл., 2 дод., 80 джерел.

МЕТОД, СИНТЕЗ АПАРАТНО-ПРОГРАМНИХ ЗАСОБІВ, КЕРУВАННЯ ІОТ-ІНФРАСТРУКТУРОЮ, FPGA, GPU, CPU, НЕСАНКЦІОНОВАНІ ВТОРГНЕННЯ.

Об’єктом дослідження є керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Предметом дослідження є метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Метою кваліфікаційної роботи магістра є підвищення ефективності керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Для розв’язання поставлених задач використовуються основні положення теорії комп’ютерних мереж та систем, системного аналізу, моделювання, методів аналізу даних, теорії математичної статистики, теорії дискретної математики.

Наукова новизна отриманих результатів:

− У дослідженні удосконалено метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, який на відміну від відомих застосовує апаратне рішення Hybrid Deep-GAN, і який може ефективно виявляти вторгнення, і який забезпечує підвищення ефективності керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Практична значимість отриманих результатів. В результаті виконаного наукового дослідження буде розроблено апаратно-програмні засоби засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

IoT – Інтернет Речей

БД - база даних

ОС - операційна система

ПЗ - програмне забезпечення

ВСТУП

Керування Інтернету Речей (IoT) інфраструктурою під час несанкціонованих вторгнень є складним завданням, яке вимагає поєднання технічних та організаційних заходів.

Основні технічні заходи для забезпечення безпеки IoT-інфраструктури включають такі аспекти.

Криптографічний захист даних: IoT-пристрої повинні використовувати криптографічні методи для захисту даних, переданих між пристроями та збережених на серверах.

Безпека мережі: IoT-мережа повинна бути захищена від несанкціонованого доступу за допомогою засобів, таких як брандмауери, ідентифікація та аутентифікація користувачів, контроль доступу тощо.

Оновлення програмного забезпечення: IoT-пристрої повинні мати можливість оновлювати своє програмне забезпечення для усунення виявлених уразливостей та покращення безпеки.

Моніторинг та виявлення інцидентів: система повинна мати можливість моніторити та виявляти потенційні загрози безпеці, такі як несподіване збільшення трафіку, відправка підозрілих повідомлень тощо.

Машинне навчання є потужним інструментом для виявлення несанкціонованих вторгнень у системи. Це може бути досягнуто за допомогою навчання алгоритмів класифікації на базі зразків (supervised learning) та без навчання на базі зразків (unsupervised learning).

Одним з найпоширеніших методів машинного навчання для виявлення вторгнень є аналіз аномалій, де методі алгоритми навчаються розрізняти нормальний і аномальний трафік на основі вхідних даних, таких як логи веб-сервера або мережеві дані.

Під час виконання роботи, система порівнює вхідні дані з навчальними даними і повертає повідомлення про вторгнення, якщо вхідні дані відрізняються від нормального шаблону.

Інший підхід - це використання навчання з учителем, де алгоритми навчаються на основі попередньо класифікованих прикладів вторгнень та їх характеристик.

Ці алгоритми можуть виявляти нові види вторгнень, які не були раніше відомі.

Машинне навчання також може бути використане для аналізу журналів системного адміністрування та моніторингу мережі для виявлення незвичайної поведінки та інших показників, що можуть вказувати на потенційне вторгнення.

Однак, для виявлення несанкціонованих вторгнень, може знадобитися великий обсяг даних, що може бути важко отримати. Також, враховуючи те, що хакери можуть використовувати нові технології та алгоритми, системи виявлення вторгнень потребують постійного оновлення та розширення, щоб залишатися ефективними.

FPGA (Field Programmable Gate Array) - це програмований кристал, що дозволяє створювати власні логічні схеми та призначення. FPGA може бути використаний для реалізації засобів машинного навчання.

Апаратна реалізація засобів машинного навчання з FPGA має кілька переваг порівняно з програмним виконанням. FPGA може забезпечувати значно вищу швидкість та низьку затримку, оскільки він працює в реальному часі та не потребує перетворення програмного коду на машинний код. FPGA також може забезпечувати значно більшу обчислювальну потужність, оскільки це програмований кристал з великою кількістю логічних елементів, які можна згрупувати для реалізації великих обчислювальних блоків.

Апаратна реалізація засобів машинного навчання з FPGA може бути використана для швидкої обробки даних в режимі реального часу. Наприклад, вона може бути використана для обробки даних з сенсорів, таких як камери, для

виявлення облич та інших об'єктів. FPGA також може бути використаний для реалізації нейронних мереж та інших засобів машинного навчання, що вимагають значної обчислювальної потужності.

Метою є підвищення ефективності керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Поставлена мета досягається розв’язанням таких основних задач:

– дослідити методи синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень;

– проаналізувати сучасні програмно-технічні засоби керування IoT-інфраструктурою під час несанкціонованих вторгнень

– розробити модель функціонування програмно-технічних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень;

– розробити метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень;

– реалізувати метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Об'єкт дослідження – керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Предмет дослідження – модель, метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Наукова новизна отриманих результатів:

Удосконалено метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, який на відміну від відомих застосовує апаратне рішення Hybrid Deep-GAN, і який може ефективно виявляти вторгнення, і який забезпечує підвищення ефективності керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Набули подальшого розвитку програмно-технічні засоби керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Практична цінність отриманих результатів. В результаті виконаного наукового дослідження буде розроблено апаратно-програмні засоби засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Для розв’язання поставлених задач використовуються основні положення теорії комп’ютерних мереж та систем, теорії архітектури комп’ютерних систем, системного аналізу, моделювання, методів аналізу даних, теорії математичної статистики, теорії дискретної математики.

За темою кваліфікаційної роботи магістра опублікована одна стаття у фаховому науковому виданні ВОТТП [1].

1 ОГЛЯД МЕТОДІВ ВИЯВЛЕННЯ ВТОРГНЕНЬ В ІНТЕРНЕТІ РЕЧЕЙ

1.1 Необхідність забезпечення безпеки Інтернету речей

Розвиток різних технологічних галузей, таких як датчики, автоматична ідентифікація та відстеження, вбудовані обчислення, бездротовий зв’язок, широкосмуговий доступ до Інтернету та розподілені послуги, збільшив потенціал інтеграції розумних об’єктів у нашу щоденну діяльність через Інтернет. Конвергенція Інтернету та розумних об’єктів, які можуть спілкуватися та взаємодіяти один з одним, визначає Інтернет речей (IoT). Ця нова парадигма визнана одним із найважливіших гравців у галузі інформаційно-комунікаційних технологій (ІКТ) на наступні роки [1]. За даними Gartner Inc., IoT може налічувати 26 мільярдів одиниць до 2020 року. Cisco Systems передбачила, що IoT створить $14,4 трильйона в результаті поєднання збільшення доходів і зниження витрат для компаній з 2013 по 2022 рік [2-5].

Багато областей застосування, такі як логістика, промислові процеси, громадська безпека, домашня автоматизація, моніторинг навколишнього середовища та охорона здоров’я, можуть мати значні переваги з системами IoT [6]. Однак інтеграція об’єктів реального світу з Інтернетом створює загрози кібербезпеці для більшості наших щоденних дій. Атаки на критичні об'єкти інфраструктури, такі як електростанції та транспортні системи, можуть мати жахливі наслідки для цілих міст і країн.

Побутова техніка також може бути основною мішенню, загрожуючи безпеці та приватності сімей [7], тести, проведені з трьома популярними пристроями розумного дому, показали різні вразливості, пов’язані з конфіденційністю користувачів, відсутністю шифрування та автентифікації. Через різні стандарти та задіяні комунікаційні стеки, обмежену обчислювальну потужність і велику кількість взаємопов’язаних пристроїв традиційні контрзаходи безпеки не можуть ефективно працювати в системах IoT. З цієї причини розробка спеціальних рішень

безпеки для IoT має важливе значення, щоб дозволити користувачам і організаціям використовувати всі можливості, які він пропонує [8].

Деякі поточні проекти з підвищення безпеки IoT включають методи забезпечення конфіденційності та автентифікації даних, контролю доступу в мережі IoT, конфіденційності та довіри між користувачами та речами, а також застосування політики безпеки та конфіденційності [9]. Однак навіть із цими механізмами мережі IoT вразливі до численних атак, спрямованих на порушення роботи мережі. З цієї причини необхідна інша лінія захисту, призначена для виявлення нападників. Системи виявлення вторгнень (IDS) виконують цю мету.

IDS є одним із основних засобів захисту традиційних мереж та інформаційних систем. IDS контролює роботу хоста або мережі, сповіщаючи системного адміністратора, коли він виявляє порушення безпеки. Таким чином, IDS зміцнила свою позицію як популярна захисна технологія для традиційних IP-мереж, маючи на ринку кілька рішень.

Незважаючи на зрілість технології IDS для традиційних мереж, поточні рішення є неадекватними для систем IoT через особливі характеристики IoT, які впливають на розвиток IDS. По-перше, важливою проблемою є обробка та сховище мережевих вузлів, на яких розміщені агенти IDS. У традиційних мережах системний адміністратор розгортає агенти IDS у вузлах з більшою обчислювальною потужністю. Мережі IoT зазвичай складаються з вузлів з обмеженими ресурсами. Тому в системах IoT важче знайти вузли з можливістю підтримки агентів IDS. Друга конкретна характеристика пов'язана з архітектурою мережі. У традиційних мережах кінцеві системи безпосередньо підключені до певних вузлів (наприклад, бездротових точок доступу, комутаторів і маршрутизаторів), які відповідають за пересилання пакетів до пункту призначення. мережі IoT, з іншого боку, зазвичай є багатострибковими. Тоді звичайні вузли можуть одночасно пересилати пакети та працювати як кінцеві системи. Наприклад, у системах вуличного освітлення на основі IoT датчики з можливістю зв’язку малого радіусу дії розгортаються на опорах освітлення [10-12]. Потім дані, зібрані

датчиком, пересилаються через датчики, розташовані на різних опорах освітлення, до досягнення шлюзу в Інтернеті. Така архітектура ставить перед IDS нові виклики. Остання характеристика пов’язана з конкретними мережевими протоколами. Мережі IoT використовують протоколи, які не використовуються в традиційних мережах, наприклад IEEE 802.15.4, IPv6 через бездротову персональну мережу з низьким енергоспоживанням (6LoWPAN), протокол маршрутизації IPv6 для мереж з низьким енергоспоживанням і мережами з втратами (RPL) і протокол обмежених додатків (CoAP). Різні протоколи привносять початкові вразливості та нові вимоги до IDS.

1.2 Виявлення вторгнень

Виявлення вторгнень — це діяльність із виявлення дій, які зловмисники здійснюють проти інформаційних систем. Ці дії, відомі як вторгнення, спрямовані на отримання несанкціонованого доступу до комп’ютерної системи. Зловмисники можуть бути зовнішніми або внутрішніми. Внутрішні зловмисники — це користувачі всередині мережі з певним ступенем легітимного доступу, які намагаються підвищити свої привілеї доступу, щоб зловживати несанкціонованими привілеями. Зовнішні зловмисники — це користувачі за межами цільової мережі, які намагаються отримати несанкціонований доступ до системної інформації [13-14].

Типовий IDS складається з датчиків, механізму аналізу та системи звітності. Датчики розгортаються в різних точках мережі або хостах. Їх завдання полягає в зборі даних мережі або хосту, таких як статистика трафіку, заголовки пакетів, запити на обслуговування, виклики операційної системи та зміни файлової системи. Датчики надсилають зібрані дані в механізм аналізу, який відповідає за дослідження зібраних даних і виявлення поточних вторгнень. Коли механізм аналізу виявляє вторгнення, система звітування генерує сповіщення для адміністратора мережі.

IDS можна класифікувати як мережеві IDS (NIDS) і Host-based IDS (HIDS). Мережевий IDS (NIDS) підключається до одного або кількох сегментів мережі та відстежує мережевий трафік на наявність зловмисних дій. Host-based IDS (HIDS) підключається до комп'ютерного пристрою та відстежує зловмисну діяльність, що відбувається в системі. На відміну від NIDS, HIDS аналізує не лише мережевий трафік, але й системні виклики, запущені процеси, зміни файлової системи, міжпроцесний зв'язок і журнали програм.

Підходи IDS також можна класифікувати як засновані на сигнатурах, на основі аномалій або на основі специфікацій.

1.3 Інтернет речей

IoT — це концепція, яка об'єднує різноманітні додатки на основі конвергенції розумних об'єктів та Інтернету, встановлюючи інтеграцію між фізичним і кіберсвітом. Ці програми можуть варіюватися від простого приладу для розумного будинку до складного обладнання для промислового підприємства. Хоча додатки IoT мають дуже різні цілі, вони мають деякі спільні характеристики. Загалом, операції IoT включають три окремі фази: фазу збору, фазу передачі та фазу обробки, управління та використання [15].

На етапі збору основною метою є збір даних про фізичне середовище. Для досягнення цієї мети об'єднано сенсорні пристрої та технології для зв'язку на короткій відстані. Пристрої фази збору зазвичай невеликі та обмежені в ресурсах. Комунікаційні протоколи та технології для цієї фази розроблені для роботи на обмежених швидкостях передачі даних і на коротких відстанях, з обмеженою ємністю пам'яті та низьким споживанням енергії. Через ці характеристики фазові мережі збору часто називають LLN (мережі з низьким енергоспоживанням і втратами). Рішення для контролю помилок, контролю доступу до середовища, маршрутизації та адресації в LLN можуть відрізнятися від тих, що використовуються в звичайному Інтернеті.

Фаза передачі має на меті передати дані, зібрані під час фази збору, програмам і, як наслідок, користувачам. На цьому етапі такі технології, як Ethernet, WiFi, коаксіальний гібридний оптоволоконний зв'язок (HFC) і цифрова абонентська лінія (DSL), поєднуються з протоколами TCP/IP для створення мережі, яка з'єднує об'єкти та користувачів на великих відстанях. Шлюзи необхідні для інтеграції протоколів LLN на етапі збору зі звичайними протоколами Інтернету, що використовуються на етапі передачі.

На етапі обробки, керування та використання програми збирають дані для отримання корисної інформації про фізичне середовище. Ці програми можуть приймати рішення на основі цієї інформації, керуючи фізичними об'єктами, щоб діяти на нихнавколишнє середовище. Ця фаза також включає проміжне програмне забезпечення, яке відповідає за полегшення інтеграції та зв'язку між різними фізичними об'єктами та мультиплатформенними програмами.

Різні альянси, консорціуми, групи особливих інтересів та організації з розробки стандартів запропонували величезну кількість комунікаційних технологій для IoT, що може стати серйозною проблемою для наскрізної безпеки в додатках IoT [16]. Найпопулярніші технології для IoT включають IEEE 802.15.4, Bluetooth Low Energy (BLE), WirelessHART, Z-Wave, LoRaWAN, 6LoWPAN, RPL, CoAP і MQTT (Message Queue Telemetry Transport).

IEEE 802.15.4 — це стандарт, запропонований Інститутом інженерів з електротехніки та електроніки (IEEE) для рівнів керування фізичним і середнім доступом низькошвидкісних бездротових персональних мереж. Завдяки стандарту IEEE 802.15.4 пристрої можуть працювати зі швидкістю передачі даних від 20 до 250 кбіт/с і на відстані від 10 м до 100 м. Керування доступом до середовища використовує технологію множинного доступу з визначенням несучої з уникненням зіткнень (CSMA/CA) [17,18].

Internet Engineering Task Force (IETF) запропонувала стандарти для роботи на основі IEEE 802.15.4 і полегшення інтеграції між LLN та Інтернетом. 6LoWPAN стандарт [19] має на меті адаптувати пакет IPv6 для IEEE 802.15.4, оскільки перший

має заголовок 40 байтів, а останній дозволяє лише 127 байт на кадр, включаючи заголовок і корисну інформацію. 6LoWPAN забезпечує взаємодію між вузлами IPv6 і LLN, але шлюз між цими двома мережами все ще потрібен. Робоча група IETF Routing over Low Power and Lossy Networks (ROLL) запропонувала протокол маршрутизації для LLN під назвою RPL [20]. Він представляє топологію сенсорної мережі як орієнтовані на призначення ациклічні графіки (DODAG) для пошуку найкращих шляхів відповідно до цільової функції та деяких показників. Він підтримує трафік «точка-точка», «точка-багато точок» і «точка-точка».

Спільнота IoT також запропонувала протоколи для прикладного рівня. CoAP і MQTT є двома найбільш обговорюваними протоколами додатків для IoT. Робоча група IETF Constrained RESTful Environments (CoRE) запропонувала CoAP бути протоколом передачі (наприклад, Hypertext Transfer Protocol - HTTP) для LLN. CoAP дозволяє транзакції запитів/відповідей у LLN, як вони відбуваються в традиційному Інтернеті, уможливлюючи передачу зібраних даних від пристроїв користувачам [21]. MQTT — це протокол повідомлень, заснований на шаблоні публікація-підписка. OASIS (Організація з удосконалення стандартів структурованої інформації), некомерційний міжнародний консорціум, стандартизував MQTT у 2013 році. Він був розроблений як легкий протокол, придатний для мереж із ненадійними або низькою пропускною спроможністю. У процесі публікації-підписки MQTT беруть участь три компоненти: передплатник, брокер і видавець. Видавець надсилає дані брокеру. У брокера є список передплатників, які отримують дані, що їх цікавлять, надіслані видавцями [22, 23].

IEEE 802.15.4, 6LoWPAN, RPL, CoAP і MQTT — це стандарти, призначені для адресації певних рівнів стеку протоколів LLN. Однак існують також стандарти IoT, які визначають вертикально інтегровані архітектури, такі як BLE, WirelessHART, Z-Wave і LoRaWAN.

BLE був розроблений Bluetooth Special Interest Group як еволюція технології Bluetooth для пристроїв з низьким енергоспоживанням. Завдяки BLE пристрої можуть працювати зі швидкістю 1 Кбіт/с у діапазоні 2,4 ГГц. Відстань між двома

вузлами BLE до 100 м. Нижні рівні стеку протоколів BLE включають фізичний рівень, відповідальний за передачу бітів і модуляцію, і канальний рівень, відповідальний за контроль доступу до середовища та встановлення з'єднання. Коли рівень зв'язку встановлює з'єднання, пристрої можуть виконувати роль головного або підлеглого. Пікосережа BLE складається з набору підлеглих пристроїв, підключених до одного головного. Протокол керування та адаптації логічного зв'язку (L2CAP) працює поверх рівня зв'язку. BLE L2CAP є спрощеною версією традиційного Bluetooth L2CAP, головним чином відповідальним за мультиплексування даних з верхніх рівнів.

Профіль (GATT) і Загальний профіль доступу (GAP). GATT дозволяє виявлення послуги та обмін характеристиками між двома пристроями. GAP визначає деякі можливі режими роботи для пристроїв BLE [24, 25].

WirelessHART є результатом зусиль HART Communication Foundation щодо перетворення протоколу HART (Highway Addressable Remote Transducer) на бездротове рішення. І HART, і WirelessHART були розроблені для керування промисловими процесами. WirelessHART організовано відповідно до структури з п'яти рівнів: фізичного, канального, мережевого, транспортного та прикладного. Фізичний рівень визначається відповідно до фізичного рівня стандарту IEEE 802.15.4. Канальний рівень реалізує контроль доступу до середовища, який базується на техніці множинного доступу з тимчасовим поділом (TDMA), і виправлення помилок. Мережевий рівень є ядром WirelessHART і відповідає за маршрутизацію, контроль топології, наскрізну безпеку та керування сеансами. Мережевий рівень WirelessHART підтримує розгортання сітчастих мереж із самовідновленням і самоорганізацією. Окрім мережевого рівня, транспортний рівень забезпечує наскрізну надійність і контроль потоку. Нарешті, прикладний рівень покладається на командно-відповідні програми, що дозволяють обмінюватися даними між пристроями та шлюзом [26,27].

Z-wave — це архітектура протоколу з низьким енергоспоживанням для автоматизації будинків і малого бізнесу. Його розробила ZenSys, а рекламує Z-

Wave Alliance. Пристрої Z-wave працюють в діапазоні 900 МГц. Швидкість передачі даних становить до 40 Кбіт/с, а максимальна відстань між двома вузлами становить близько 30 м. Рівень керування доступом до середовища Z-wave базується на техніці CSMA/CA і має додатковий механізм повторної передачі для надійності. Мережа Z-wave має два типи пристроїв: контролери та підлеглі. Контролери надсилають команди та запити для підлеглих пристроїв, які виконують команди або надсилають відповіді контролерам. Маршрутизація в мережах Z-wave виконується контролерами, які ведуть таблицю з інформацією про всю топологію [28, 29].

LoRaWAN — це технологія, розроблена некомерційною організацією LoRa Alliance. На відміну від таких технологій, як IEEE 802.15.4, BLE, WirelessHART і Z-Wave, які спрямовані на роботу на коротких відстанях, LoRaWAN — це технологія для глобальних мереж малої потужності (LPWAN). У мережах LoRaWAN кінцеві пристрої підключаються до центрального мережевого сервера через шлюз. Кінцеві пристрої безпосередньо підключаються до шлюзів через бездротові канали з одним стрибком, тоді як шлюзи використовують традиційні IP-мережі для підключення до центральних серверів. Один кінцевий пристрій може передавати дані для кількох шлюзів, а мережевий сервер відповідає за відкидання надлишкових пакетів. Швидкість передачі даних на термінал коливається від 0,3 Кбіт/с до 50 Кбіт/с. Подолана відстань у містах може коливатися від 2 км до 5 км, а в сільській місцевості – від 10 км до 15 км [30, 31]).

За останні роки було опубліковано кілька оглядових статей про IDS для технологій, пов’язаних з IoT, таких як мобільні спеціальні мережі (MANET) [32, 34], бездротові сенсорні мережі (WSN) [35-37], хмарні обчислення [38] та кіберфізичних систем [39].

В [40] зауважте, що застосування досліджень дротових мереж до бездротових мереж є непростим завданням через фундаментальні архітектурні відмінності, особливо відсутність фіксованої інфраструктури. Автори стверджують, що тип реакції на вторгнення для бездротових ad hoc мереж залежить від типу вторгнення,

мережевих протоколів і програм, що використовуються, а також від впевненості в доказах. Деякі ймовірні відповіді включають повторну ініціалізацію каналів зв'язку між вузлами, ідентифікацію скомпрометованих вузлів і реорганізація мережі для припинення скомпрометованих вузлів та ініціювання запиту на повторну автентифікацію для всіх вузлів у мережі. Автори також представляють детальне обговорення семи пропозицій IDS для MANET відповідно до наступних методологій: розподілене виявлення аномалій та виявлення на основі мобільних агентів. В обох випадках агент IDS працює на кожному мобільному вузлі та виконує локальний збір даних і локальне виявлення. Різниця між цими двома методологіями полягає в глобальному виявленні: розподілене виявлення аномалій використовує інформацію з сусідніх вузлів для створення механізму спільного виявлення, тоді як виявлення на основі мобільних агентів використовує технологію мобільних агентів для виявлення вторгнень і реагування на них.

В [41] представлено дослідження про мережеву інфраструктуру для IDS у MANET. Автори описують три архітектури для IDS у MANET: розподілені та кооперативні системи виявлення вторгнень (плоска мережева інфраструктура), ієрархічні системи виявлення вторгнень (багаторівнева мережева інфраструктура) та мобільний агент для систем виявлення вторгнень (плоска та багаторівнева мережева інфраструктура). Через природу MANET, автори повідомляють, що майже всі досліджені IDS структуровані для розподілу та мають кооперативну архітектуру. Автори також представляють таксономію виявлення неправильної поведінки вузлів у MANET, що стосується архітектури, типу збору даних, розподілу даних, спостереження, виявлення неправильної поведінки, покарання та виявлення маршруту.

В [42] представенлено огляд методів виявлення вторгнень для MANET, зосереджуючись на алгоритмах виявлення. Автори вводять дерево класифікації методів виявлення вторгнень за характером механізму обробки, задіяного в методі виявлення. Методи виявлення вторгнень поділяються на статистичні, евристичні, правила, стани, підписи, репутацію, інформацію про маршрутизацію, перехресні

рівні та теорію графів. Для кожного досліджуваного методу виявлення вторгнень автори пропонують детальну класифікацію системи відповідно до методу виявлення (зловживання, на основі аномалії, специфікація або гібрид), архітектури (автономна, розподілена та кооперативна, на основі мобільного агента та ієрархічний IDS), час виявлення (у режимі реального часу або офлайн), протокол маршрутизації, тип атакуваних атак, продуктивність, ефект мобільності, надійності, гнучкості, масштабованості, швидкості та надійності. Крім того, вони перераховують проблеми дослідження та висвітлюють відкриті проблеми у виявленні вторгнень для MANET. Одна важлива проблема пов'язана з динамічним середовищем. І нав'язлива, і доброякісна поведінка користувачів, систем або мережі змінюються з часом. IDS повинна бути самокерованою та самостійно налаштованою, щоб справлятися з безперервним мінливим динамічним середовищем і швидше реагувати на динамічні зміни джерел апаратного та програмного забезпечення в мережі. І нав'язлива, і доброякісна поведінка користувачів, систем або мережі змінюються з часом. IDS повинна бути самокерованою та самостійно налаштованою, щоб справлятися з безперервним мінливим динамічним середовищем і швидше реагувати на динамічні зміни джерел апаратного та програмного забезпечення в мережі. І нав'язлива, і доброякісна поведінка користувачів, систем або мережі змінюються з часом. IDS повинна бути самокерованою та самостійно налаштованою, щоб справлятися з безперервним мінливим динамічним середовищем і швидше реагувати на динамічні зміни джерел апаратного та програмного забезпечення в мережі.

В [43] представлено таксономію IDS для WSN щодо техніки виявлення: виявлення неправильного використання, виявлення аномалій та виявлення на основі специфікацій. Вони також забезпечують детальне обговорення механізмів IDS, що стосуються структури WSN, висвітлюючи різні життєво важливі сфери, які наразі недостатньо розвинені. Деякі з тем включають відсутність реальних реалізацій схем IDS у мережах WSN та розробку механізмів IDS, які відповідають баченню IoT. Вони також дійшли висновку, що, незважаючи на те, що за останні

роки сфера IDS для WSN значно просунулася вперед, все ще існують різні галузі досліджень (наприклад, архітектури IDS, баланс між точністю та споживанням ресурсів, краща інтеграція базових механізмів), які потребують мати подальший розвиток.

В [44] проведено огляд літератури щодо IDS для WSN. Вони представляють короткий огляд IDS, запропонованих для MANET, і досліджують їх застосовність до WSN. На думку авторів, деякі IDS будуть застосовані безпосередньо (дві пропозиції), деякі будуть застосовані зі значними модифікаціями (сім пропозицій), а решта не будуть застосовані до WSN (вісім пропозицій) просто через особливі вимоги до дизайну WSN. . Автори також пропонують порівняння IDS, запропонованих для WSN, відповідно до архітектури мережі та техніки виявлення. Нарешті, у роботі підкреслюється енергоспоживання IDS через вимоги до низького енергоспоживання WSN.

В [45] подано кілька вторгнень, які впливають на доступність, конфіденційність і цілісність Cloud Computing. Автори узагальнюють і класифікують IDS, що використовуються в хмарі, за трьома категоріями: технологія IDS (система виявлення вторгнень на основі хосту (HIDS), система виявлення вторгнень на основі мережі (NIDS), система виявлення вторгнень на основі гіпервізора та розподілена система виявлення вторгнень (DIDS).)), техніку виявлення та мережеве позиціонування. Вони також обговорюють переваги та недоліки кожної пропозиції та визначають труднощі для того, щоб зробити Cloud Computing надійною платформою для надання послуг IoT. Більшість запропонованих методів виявлення вторгнень у хмарі не можуть впоратися з повторюваними атаками в цьому середовищі, такими як інсайдерські атаки та атаки на віртуальну машину чи гіпервізор.

В [46] подано кіберфізичну систему (CPS) — це великомасштабні, географічно розосереджені, об’єднані, гетерогенні, життєво важливі системи, які містять датчики, виконавчі механізми, а також компоненти керування та мереж. Автори представляють таксономію сучасних IDS для CPS на основі двох

параметрів дизайну: техніка виявлення та матеріал аудиту (на базі хоста чи мережі). По-перше, вони забезпечують комплексний аналізпро відмінності між традиційними IDS і IDS для CPS, які включають роботу з моніторингом фізичних процесів, складними атаками та застарілими технологіями. Потім автори підсумовують наявну роботу в IDS для проектування CPS з точки зору застосування CPS, типу атаки, функцій аудиту та якості набору даних. Автори також перелічують дослідницькі проблеми та висвітлюють майбутні тенденції в області IDS для CPS.

Хоча ці роботи в основному зосереджені на розробці IDS для кількох пов'язаних з IoT елементів, жодна з них не містить дослідження методів IDS, специфічних для парадигми IoT. У цій оглядовій роботі обговорено стратегії розміщення та методи виявлення IDS, розроблених спеціально для IoT. Ми також представляємо загальні загрози безпеці IoT і те, як IDS можна використовувати для їх виявлення. Крім того, ми представляємо огляд загальних стратегій перевірки, які використовуються в методах виявлення вторгнень для IoT, і обговорюємо відкриті питання досліджень і майбутні тенденції.

1.4 Виявлення вторгнень в Інтернеті речей

У цьому розділі ми проводимо огляд літератури щодо пропозицій IDS щодо IoT. Кожна робота була класифікована за такими атрибутами: стратегія розміщення IDS, метод виявлення, загроза безпеці та стратегія перевірки. атрибут).

1.5 Стратегії розміщення IDS

Перш ніж розпочати обговорення стратегій розміщення IDS у мережах IoT, необхідно представити огляд архітектури мереж IoT та основних елементів, які є її частиною.

В останні роки дослідники показали різні архітектури для IoT [47-49], які тісно пов’язані з етапами збору, передачі та обробки, управління та використання. Хоча ці пропозиції дещо відрізняються в деяких аспектах, вони подібним чином організовують сценарії IoT у трьох широких доменах: фізичному домені, мережевому домені та домені додатків. Фізична область пов’язана з фазою збору та включає пристрої, які сприймають і діють над фізичним середовищем, часто створюючи LLN. Мережевий домен, який спирається на фазу передачі, об’єднує звичайні мережеві рішення та протоколи для передачі даних із фізичного середовища до програм і користувачів. Прикордонний маршрутизатор обов’язково розміщується між фізичним і мережевим доменами для інтеграції протоколів LLN із звичайними протоколами мережевого домену. Нарешті, домен програми включає інтерфейси, які дозволяють користувачам обробляти об’єкти у фізичному домені.

У мережах IoT IDS можна розмістити на межовому маршрутизаторі, на одному або кількох виділених хостах або на кожному фізичному об’єкті. Перевагою розміщення IDS у прикордонному маршрутизаторі є виявлення атак вторгнень з Інтернету на об’єкти у фізичному домені. Однак IDS у прикордонному маршрутизаторі може спричинити витрати зв’язку між вузлами LLN і прикордонним маршрутизатором через часті запити IDS про стан мережі. Розміщення IDS у вузлах LLN може зменшити накладні витрати на зв’язок, пов’язані з моніторингом мережі, але вимагає від них більше ресурсів (обробка, зберігання та енергія) [50]. Це може бути проблемою через обмеження ресурсів вузлів LLN.

Розподіл агентів IDS між деякими виділеними вузлами може бути рішенням для задоволення вимог щодо зменшення трафіку моніторингу та збільшення потужності обробки. Однак це рішення вимагає організації мережі в різних регіонах, що може бути проблемою.

1.6 Розподілене розміщення IDS

У цій стратегії розміщення IDS розміщуються в кожномуфізичний об'єкт ЛЛН. IDS, розгорнуту в кожному вузлі, необхідно оптимізувати, оскільки ці вузли обмежені в ресурсах. Щоб вирішити цю проблему в [51, 52] запропоновані розподілені легкі IDS. О та ін. визначив легкий алгоритм для зіставлення сигнатур атаки та корисного навантаження пакетів. Вони запропонували дві методики: допоміжний перехід і раннє рішення, метою яких є зменшення кількості матчів, необхідних для виявлення атак. Вони порівняли свій підхід з алгоритмом Wu-Manber (WM), який є одним із найшвидших алгоритмів зіставлення шаблонів. За словами авторів, запропонований метод є швидшим, ніж алгоритм Wu-Manber, працює на платформі з обмеженими ресурсами. Лі та ін. у свою чергу запропонував легкий метод, який відстежує енергоспоживання вузла для виявлення вторгнень. Зосередившись лише на одному параметрі вузла, автори намагалися мінімізувати обчислювальні ресурси, необхідні для виявлення вторгнень.

У розподіленому розміщенні вузли також можуть відповідати за моніторинг своїх сусідів. Вузли, які контролюють своїх сусідів, називаються сторожовими. В [53] запропонував рішення під назвою INTI (виявлення вторгнень атак Sinkhole на 6LoWPAN для Інтернету речей), яке об’єднало концепції довіри та репутації зі сторожовими системами для виявлення та пом’якшення атак. По-перше, вузли класифікуються як головні, асоційовані або членські вузли, що утворюють ієрархічну структуру. Роль кожного вузла може змінюватися з часом через реконфігурацію мережі або атаку.

Потім кожен вузол відстежує вищестоящий вузол, оцінюючи його вхідний і вихідний трафік. Коли вузол виявляє атаку, він транслює повідомлення, щоб попередити інші вузли та ізолювати зловмисника. Автори не обговорювали вплив рішення на вузли малої потужності.

1.7 Централізоване розміщення IDS

У централізованому розміщенні IDS розміщується в централізованому компоненті, наприклад, у прикордонному маршрутизаторі або виділеному хості. Усі дані, які вузли LLN збирають і передають в Інтернет, перетинають прикордонний маршрутизатор, а також запити, які клієнти Інтернету надсилають до вузлів LLN. Таким чином, IDS, розміщений у прикордонному маршрутизаторі, може аналізувати весь трафік, який обмінюється між LLN та Інтернетом [54]. Однак аналізу трафіку, який проходить через прикордонний маршрутизатор, недостатньо для виявлення атак, які включають лише вузли в межах LLN. Потім дослідники повинні розробити IDS, які можуть контролювати трафік, що обмінюється між вузлами LLN, не ігноруючи вплив, який ця діяльність моніторингу може мати на роботу вузлів з низькою пропускною здатністю. Крім того, централізована IDS може мати труднощі з моніторингом вузлів під час атаки, яка компрометує частину мережі.

В [55] запропоновано рішення для аналізу пакетів, які проходять через прикордонний маршрутизатор між фізичним і мережевим доменом. Робота була зосереджена на атаках ботнетів, що пояснює їх вибір для моніторингу лише трафіку прикордонного маршрутизатора. В [56] також використано централізоване розміщення, але вони враховували захист IDS від атак DoS (відмова в обслуговуванні). Таким чином, автори вирішили розгорнути механізм аналізу IDS і систему звітності IDS на потужному виділеному хості. Вони розгорнули датчики IDS у LLN, які відповідали за фіксування мережевого трафіку та надсилання цих даних механізму аналізу IDS. Виділений хост IDS є дротом, підключеним до датчиків IDS, що дозволяє уникнути передачі даних IDS і звичайних мережевих даних в одній бездротовій мережі. Таким чином, якщо DoS-атака погіршить якість бездротової передачі, це не вплине на передачу даних IDS.

В [57] запропоновано централізований підхід, у якому IDS розміщується в прикордонному маршрутизаторі. Метою запропонованого рішення є виявлення

атак у фізичному домені. Потім замість моніторингу трафіку, що перетинає прикордонний маршрутизатор, автори запропонували протокол серцевого ритму. Відповідно до запропонованого протоколу прикордонний маршрутизатор регулярно надсилає ехо-запити ICMPv6 до всіх вузлів LLN і очікує відповіді для виявлення атак або проблем з доступністю. Незважаючи на те, що рішення створює додатковий трафік у мережі, автори показали в експериментах, що вузлам LLN не потрібно буде виділяти додаткову пам’ять для запуску алгоритму пульсу, а накладні витрати енергії були мінімальними.

1.8 Гібридне розміщення IDS

Гібридне розміщення IDS поєднує концепції централізованого та розподіленого розміщення, щоб скористатися їхніми сильними сторонами та уникнути недоліків.

В [58] вибрані вузли в мережі розміщують IDS. Ці вибрані вузли (сторожові собаки) спрямовані на виявлення вторгнень шляхом підслуховування обмінюваних пакетів у своєму сусідстві. Сторожовий таймер вирішує, чи скомпрометовано вузол відповідно до набору правил. Кожен сторожовий таймер має певний набір правил, оскільки кожен компонент у мережі може мати різну поведінку. Наприклад, прикордонний маршрутизатор зазвичай має більшу кількість повідомлень, ніж звичайний вузол. Перевага цього підходу полягає в тому, що він дозволяє створювати різні правила для кожної області мережі.

В [59] також дотримався підходу організації мережі в регіонах. Вони використали гібридне розміщення, побудувавши магістраль із вузлів монітора. Завдяки мінімальній кількості вузлів моніторингу, які охоплюють всю мережу, вузол моніторингу вислуховує зв’язок від своїх сусідів і визначає, чи вузол скомпрометований. Перевага цього рішення полягає в тому, що він не створює додаткових накладних витрат на зв’язок, оскільки вузли моніторингу лише фіксують передачі між своїми сусідами. У [60] організовано мережу в невеликі

кластери з однаковою кількістю вузлів. Кожен кластер має голову кластера, яка є вузлом, який безпосередньо спілкується з усіма членами кластера. Екземпляр IDS розміщується в кожній головці кластера, який відстежує членів кластера, перехоплюючи їхній зв'язок. Члени кластеру повинні повідомляти відповідну інформацію про себе та інших сусідів голові кластера. Незважаючи на те, що автори вважали, що головка кластера може бути більш потужним вузлом, вони вирішили розробити легке рішення IDS.

У другому підході до гібридного розміщення модулі IDS розміщуються як у прикордонному маршрутизаторі, так і в інших вузлах мережі. Головною відмінністю цього підходу від першого є наявність центральної складової. Модулі IDS у прикордонному маршрутизаторі відповідають за завдання, які потребують більшої ємності ресурсів, у той час як модулі IDS у звичайних вузлах зазвичай легкі. В [61] запропонован IDS під назвою SVELTE. Під час роботи прикордонний маршрутизатор розміщує інтенсивні модулі IDS, такі як той, що відповідає за виявлення вторгнень шляхом аналізу мережевих даних RPL. Вузли мережі відповідають за легкі завдання, такі як надсилання мережевих даних RPL до прикордонного маршрутизатора та сповіщення прикордонного маршрутизатора про зловмисний трафік, який вони отримують.

В [62] мережеві вузли відповідають за виявлення змін у своєму сусідстві та надсилання інформації про сусідів до централізованих модулів, які розгорнуті в прикордонному маршрутизаторі. Централізовані модулі, у свою чергу, відповідають за зберігання та аналіз цих даних для виявлення вторгнень і ідентифікації можливих зловмисників. Хоча опис IDS може вказувати на архітектуру, яка вимагає інтенсивного обміну трафіком для виявлення вторгнень, результати показали, що накладні витрати на енергію, накладні витрати на пакети та споживання пам'яті були адекватними середовищу з обмеженими вузлами.

В [63] запропонував IDS, який також розподіляє різні обов'язки між прикордонним маршрутизатором і вузлами мережі, змушуючи їх працювати разом. Модуль IDS у вузлі стежить за сусідами вузла, виявляючи можливі вторгнення.

Коли виявляється подія, вузол надсилає сповіщення до модуля IDS на прикордонному маршрутизаторі. Потім модуль прикордонного маршрутизатора співвідносить сповіщення від різних вузлів, щоб прийняти остаточне рішення щодо вторгнення. Thanigaivelan та ін. класифікували свій IDS як розподілений IDS. Однак центральна роль прикордонного маршрутизатора в прийнятті остаточного рішення щодо виявлення вторгнення робить запропонований IDS гібридним підходом.

1.9 Висновки та постановка задачі

Інтернет речей є важливою сферою для забезпечення безпеки як пристроїв, так і додатків, яка пропонує широкий технічний прогрес на вищому рівні в спостереженні за величезними обсягами даних. Він увійшов в нові додатки і пристрої разом з електронікою, датчиками, виконавчими механізмами, протоколами, програмним забезпеченням, щоб поліпшити об'єднання, компіляцію і передачу даних. Мережа Інтернету речей не має наскрізної надійної системи безпеки. Зловмисники можуть отримати контроль над пристроями; майже 85% пристроїв Інтернету речей не захищені від різноманітних кібервторгнень. Крім того, вони піддаються численним загрозам, таким як DoS, DDoS, "воронка", "червоточина", крадіжка даних/ідентифікаційних даних, затримка пристрою та інші сучасні інтелектуальні атаки. Для того, щоб захистити важливі системи, засновані на безпеці, від зловмисників, необхідно добре продумати систему безпеки, щоб виявити всі форми відомих і невідомих загроз.

В даний час доведено, що нейронні мережі мають здатність реконструювати свій власний код, щоб розвиватися, захищатися і відновлюватися від вторгнень. Система виявлення вторгнень за допомогою нейронної мережі з метою дослідження мережевого трафіку та системної інформації від шкідливих дій та видачі попередження є основним механізмом відновлення. Аналіз та генерування відповідей на атаки з різних рівнів. Адаптуємо до моделі систему запобігання

вторгненням з інтегрованою системою виявлення вторгнень для ухилення зловмисника від виконання шкідливої дії.

Звичайні системи виявлення вторгнень не підходять, оскільки багато ексклюзивних характеристик Інтернету речей щодо розміру, низької оброблювальної здатності, типу захисту конфіденційності, прямий метод застосування виявлення вторгнень не призведе до успіху. Тому що в звичайних алгоритмах система може реалізувати систему виявлення вторгнень тільки на центрах обробки даних / вузлах, які мають справу з інформацією з абсолютної моделі Інтернету речей. Але було помічено, що більшість мереж Інтернету речей, які залежать від централізованої системи для встановлення рішення для виявлення вторгнень, є вразливими для порушень безпеки. Існує ймовірність того, що ці централізовані системи виявлення вторгнень можуть бути скомпрометовані або піддані впливу. На противагу цьому, в рамках виявлення вторгнень, кожен пристрій повинен містити базу даних, яка має низьку частину стану мережі. Крім того, такі додатки, як охорона здоров'я, розумний будинок, розумні міста, промислова автоматизація, в яких модель, клієнт може не розкривати життєво важливу ексклюзивну інформацію з контролерами системи, в такому випадку доведено, що система виявлення вторгнень з центром обробки даних некомпетентна.

Тому в цьому дослідженні необхідним є розроблення методу синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, що використовує апаратне рішення на основі GAN (Generative Adversarial Networks), яка полягає в побудові розподіленої моделі разом з методами глибокого навчання.

2 МОДЕЛЬ ФУНКЦІОНУВАННЯ СИСТЕМИ КЕРУВАННЯ ІОТ-ІНФРАСТРУКТУРОЮ ПІД ЧАС НЕСАНКЦІОНОВАНИХ ВТОРГНЕНЬ

2.1 Математичний апарат побудови моделі функціонування системи керування IoT-інфраструктурою під час несанкціонованих вторгнень

Мережі Інтернету речей з'являються в результаті значного зростання сучасних комунікаційних технологічних застосувань.

Мережа, сформована з сенсорних вузлів з обмеженими ресурсами, складністю, відкритими функціями бездротової передачі даних, робить їх вразливими до загроз безпеці.

Ефективна система виявлення вторгнень допомагає виявити атаки і виконати важливі контрдії, щоб забезпечити безпечне і надійне функціонування.

Однак, зважаючи на широке розповсюдження інтернету речей, система виявлення вторгнень повинна працювати в дискретній формі, що зменшує навантаження на адміністратора.

Для того, щоб подолати ці проблеми, пропонується розробити метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Він використовуватиме розподілену генеративну змагальну мережу (D-GAN) з покращеною оптимізацією - розподіленим глибоким навчанням на основі штучної нейронної мережі (EWO-HDL+ANN).

При цьому GAN може виявляти внутрішні атаки, а D-GAN здатна ефективно виявляти як внутрішні, так і зовнішні атаки.

Для цього використовується передача за подібністю підпростору.

Після цього попередньо оброблені дані подаються на етап вилучення необхідних ознак.

Для вилучення ознак застосовується модифікований аналіз головних компонент (Modified Principal Component Analysis, MPCA), який використовується для вилучення нових ознак, які є просвітницькими.

Потім виконується вибір ознак за допомогою алгоритму оптимізації Enhanced Whale.

Метод використовується для вибору значущих і зайвих ознак з набору необхідних даних.

Це покращує точність класифікації завдяки найбільшому значенню придатності.

Потім оцінюється виявлення вторгнень за допомогою алгоритму HDL+ANN, який використовується для потужного виявлення атак.

Експериментальний висновок доводить, що представлений метод EWO-DDL+ANN забезпечує покращену систему виявлення вторгнень з огляду на більшу точність, достовірність, пригадування, f-міру та низьку частоту помилкових спрацьовувань.…

Інтернет речей є важливою сферою для забезпечення безпеки як пристроїв, так і додатків, яка пропонує широкий технічний прогрес на вищому рівні в спостереженні за величезними обсягами даних.

Він увійшов в нові додатки і пристрої разом з електронікою, датчиками, виконавчими механізмами, протоколами, програмним забезпеченням, щоб поліпшити об'єднання, компіляцію і передачу даних [1].

Мережа Інтернету речей не має наскрізної надійної системи безпеки. Зловмисники можуть отримати контроль над пристроями; майже 85% пристроїв Інтернету речей не захищені від різноманітних кібервторгнень.

Крім того, вони піддаються численним загрозам, таким як DoS, DDoS, "воронка", "червоточина", крадіжка даних/ідентифікаційних даних, затримка пристрою та інші сучасні інтелектуальні атаки.

Для того, щоб захистити важливі системи, засновані на безпеці, від зловмисників, необхідно добре продумати систему безпеки, щоб виявити всі форми відомих і невідомих загроз [2].

В даний час доведено, що нейронні мережі мають здатність реконструювати свій власний код, щоб розвиватися, захищатися і відновлюватися від несанкціонованих вторгнень.

Система виявлення вторгнень за допомогою нейронної мережі з метою дослідження мережевого трафіку та системної інформації від шкідливих дій та видачі попередження є основним механізмом відновлення. Аналіз та генерування відповідей на атаки з різних рівнів.

Адаптуємо до моделі систему запобігання вторгненням з інтегрованою системою виявлення вторгнень для ухилення зловмисника від виконання шкідливої дії [3].

Було доведено, що звичайні системи виявлення вторгнень не підходять, оскільки багато ексклюзивних характеристик Інтернету речей щодо розміру, низької оброблювальної здатності, типу захисту конфіденційності, прямий метод застосування виявлення вторгнень не призведе до успіху.

Тому що в звичайних алгоритмах система може реалізувати систему виявлення вторгнень тільки на центрах обробки даних / вузлах, які мають справу з інформацією з абсолютної моделі Інтернету речей.

Але було помічено, що більшість мереж Інтернету речей, які залежать від централізованої системи для встановлення рішення для виявлення вторгнень, є вразливими для порушень безпеки.

Існує ймовірність того, що ці централізовані системи виявлення вторгнень можуть бути скомпрометовані або піддані впливу [4].

На противагу цьому, в рамках виявлення вторгнень, кожен пристрій повинен містити базу даних, яка має низьку частину стану мережі.

Крім того, такі додатки, як охорона здоров'я, розумний будинок, розумні міста, промислова автоматизація, в яких модель, клієнт може не розкривати життєво важливу ексклюзивну інформацію з контролерами системи, в такому випадку доведено, що система виявлення вторгнень з центром обробки даних некомпетентна [5].

В роботі було залучено апарат для класифікації даних, що грунтується на схемі GAN (Generative Adversarial Networks), яка полягає в побудові розподіленої моделі разом з методами глибокого навчання.

В роботі для вирішення задачі побудови схеми функціонування програмно-технічних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень пропонується залучити модель неконтрольованого навчання.

Така модель називається генеративним моделюванням.

Вона застосовується для:

- виконання автоматичного виявлення збору даних;

- аналізу та вивчення вхідних даних з метою використання моделі для створення/виведення інноваційної вибірки, яка потенційно буде отримана з реальної бази даних [6].

Генеративні змагальні мережі (GAN) - це тип моделі глибокого навчання, що складається з двох нейронних мереж, генератора і дискримінатора, які навчаються разом у змагальному процесі.

Генеративні змагальні мережі вчаться генерувати реалістичні дані, такі як зображення, шляхом створення зразків з випадкового вхідного шуму.

Мережа дискримінатора навчається розрізняти реальні дані від згенерованих мережею-генератором.

Генеративна змагальна мережа навчається генерувати дані, які можуть обдурити дискримінаторну мережу, змусивши її повірити, що це справжні дані.

Процес навчання ШНМ є ітеративним, де мережа-генератор генерує дані, а мережа-дискримінатор їх оцінює.

Зворотний зв'язок від дискримінаторної мережі використовується для покращення генераторної мережі, яка, в свою чергу, генерує більш реалістичні дані для оборобки.

Цей процес триває доти, доки генераторна мережа не видаватиме дані, які неможливо відрізнити від реальних даних дискримінантної мережі.

ШНМ використовуються в різних додатках, таких як синтез зображень і відео, доповнення даних і виявлення аномалій.

Однією з проблем при навчанні GAN є пошук правильного балансу між мережами генератора і дискримінатора.

Якщо дискримінаторна мережа занадто сильна, вона може легко ідентифікувати згенеровані дані, а генераторна мережа не зможе вдосконалюватися.

Якщо мережа-генератор занадто сильна, вона може створювати дані, які занадто схожі на навчальні дані, і їм бракує різноманітності.

Генеративні змагальні мережі - це розумний спосіб навчання винахідницької моделі шляхом структурування проблеми як керованого навчання з двома підмоделями.

Такою є модель-генератор, яка навчається створювати нові зразки, та моделлю-дискримінатором, яка намагається розпізнати зразок як унікальний або репродукцію.

Ці моделі навчаються одночасно в змагальній грі з нульовою сумою, доки модель не обманює на півперіоді, тобто модель-генератор не створює перспективні зразки [7].

Використання генеративних змагальних мереж (Generative Adversarial Networks, GAN) у додатках машинного навчання та штучного інтелекту має кілька переваг.

GAN можуть генерувати реалістичні та різноманітні дані: GAN можуть генерувати дані, які візуально схожі на реальні дані і мають широкий спектр варіацій.

Це може бути корисно в таких додатках, як синтез зображень і відео, де генерування різноманітних і реалістичних даних є критично важливим.

GAN можуть вивчати складні розподіли даних: ШНМ можуть вивчати складні розподіли даних, які може бути важко змоделювати за допомогою традиційних статистичних методів.

GAN можуть навчатися на неструктурованих і багатовимірних даних, таких як зображення і аудіо, і генерувати нові дані, які відображають основний розподіл даних.

GAN можуть виконувати аугментацію даних: ГВС можна використовувати для доповнення існуючих наборів даних, що може бути корисним у таких додатках, як розпізнавання і класифікація об'єктів.

Генеруючи нові дані, GAN можуть підвищити надійність і точність моделей машинного навчання.

GAN можуть виявляти аномалії: GAN можна використовувати для виявлення аномалій у даних, таких як шахрайські транзакції або медичні аномалії.

Навчаючи GAN на нормальних даних, він може навчитися ідентифікувати аномальні точки даних, які відхиляються від нормального розподілу.

GAN можна використовувати в неконтрольованому навчанні: GAN можна навчати на немаркованих даних, що може бути корисно в тих випадках, коли марковані дані є дефіцитними або дорогими.

Навчаючись на основі розподілу даних, GAN можуть генерувати нові дані і виявляти в них приховані закономірності.

Незважаючи на свої переваги, генеративні змагальні мережі (GAN) також мають певні обмеження та проблеми.

Нестабільність навчання: GAN може бути важко навчити, а процес навчання може бути нестабільним.

Може бути складно знайти правильний баланс між мережами генератора і дискримінатора, що може призвести до колапсу режиму або коливань.

Збій режиму: збій режиму відбувається, коли генераторна мережа виробляє обмежені або повторювані дані, замість того, щоб досліджувати весь діапазон базового розподілу.

Це може призвести до відсутності різноманітності у згенерованих даних.

Метрики оцінювання: об'єктивно оцінити ефективність роботи ГАН може бути складно.

Традиційні показники оцінки, такі як точність і втрати, можуть неточно відображати якість і різноманітність згенерованих даних.

Обчислювальні ресурси: GAN вимагають значних обчислювальних ресурсів, включаючи високопродуктивні обчислення і великі обсяги пам'яті.

Це може зробити їх дорогими і трудомісткими для навчання вхідних даних системи.

Конфіденційність даних: GAN можна використовувати для створення синтетичних даних, які візуально схожі на реальні дані.

Це може викликати занепокоєння щодо конфіденційності та безпеки даних, оскільки може бути складно відрізнити реальні дані від синтетичних.

Надмірна адаптація: GAN можуть надмірно використовувати навчальні дані, що може призвести до поганого узагальнення нових даних.

Це може бути складно діагностувати і вирішити, оскільки процес навчання є неконтрольованим.

Загалом, GAN є потужним інструментом для генерування реалістичних і різноманітних даних, вивчення складних розподілів даних і підвищення точності та надійності моделей машинного навчання.

Рисунок 2.1 демонструє загальну організацію функції захоплення GAN, яка виконується в колекторі даних для отримання пакетів та вилучення ознак для подальшої їх обробки.

2.2 Застосування GAN

GAN використовує ідею змагального навчання для навчання моделі обробляти складні багатовимірні дані, що генеруються в мережі Інтернету речей. GAN мають генератор B разом з мережею дискримінатора P.

Рисунок 2.1 – Організація функції захоплення GAN, яка виконується в колекторі даних для отримання ознак

Генератор B навчається генерувати фальшиві зразки, оновлюючи шумові змінні s в генераторний зразок B (s), щоб зрадити дискримінатор, в той час як дискримінатор P навчається використовувати переваги ймовірності передбачення того, чи є його вхідні дані навчальними прикладами або генераторними зразками B (s).

Генератор перетворює зловмисну версію вхідних даних і надсилає їх до системи виявлення для сортування та до дискримінатора.

Мета генератора - обманути систему виявлення вторгнень, а мета дискримінатора - імітувати систему виявлення вторгнень, класифікуючи вхідні дані як правдиві або помилкові, і надавати відповідь генератору.

Таким чином, ідентифікатор вторгнення в мережі є збирачем даних, який збирає мережеву інформацію в реальному часі шляхом скупчення пакетів в мережі. Ці пакети обробляються раніше, ніж надходять до набору даних.

Функція перехоплення пакетів виконується всередині збирача даних для отримання пакетів і вилучення функцій.

Зменшення/вилучення ознак реконструює багатопросторовий простір до меншого простору, таким чином оновлення буде лінійним/нелінійним.

Це також дозволяє відкинути зайві значення, що значно спрощує розроблювану систему.

Під час навчання, вибір необхідних ознак або вилучення ознак досліджують можливий набір ознак з вхідної вибірки, і це є завданням навчання класифікатора [8].

Метод вилучення ознак намагається замінити вхідні ознаки новим набором ознак, тоді як схема виділення ознак, здається, призначена для кращого виявлення ознак з фактичних вхідних даних.

Виділення ознак відіграє важливу роль у виявленні вторгнень, що переконливо свідчить про збільшення корисності навчання, підвищення результату узагальнення та покращення візуалізації даних.

На етапі відбору ознак обчислюються ознаки з лінійними кореляціями, нелінійні кореляції для визначення релевантності між мітками класів у даних, пов'язаних з категорією, і надлишкові ознаки для вихідного класу, що містить об’єкти.

Наближена стратегія пошуку Маркова-бланкета, адаптована для виявлення відповідної ознаки, яка містить найбільшу кількість інформації, пов'язаної з вихідним класом [9].

Схема ідентифікації абсолютних міток виражається як класифікація. Вона виконується для сортування даних залежно від навчання та значень необхідних характеристик.

Це також називається категоризацією класів/прогнозуванням/керованим навчанням.

Модель використовується для передбачення міток класів і тестування створеної моделі на тестових зразках, відповідно передбачається точність політики класифікації.

Методи машинного навчання використовуються для ретельного аналізу та перевірки величезних мережевих даних на наявність аномальних/точних даних [10].

Оскільки дані генеруються з різних джерел, трафік в мережі величезний.

Для створення перенесення за подібністю підпросторів потужно використовуються методи машинного навчання, подібні до кластеризації, класифікації.

А також методи, засновані на глибокому навчанні (Deep Learning — DL), приділяють увагу успішному просуванню великої кількості наборів даних перенесення за подібністю підпросторів через Інтернет речей [11].

Основна ідея цієї науково-дослідної роботи полягає в розробці моделі для виявлення вторгнень в мережі Інтернету речей.

Через складну та мінливу в часі динамічну природу мережі Інтернету речей, зразки мережевих вторгнень об'єднуються у велику кількість отриманих нормальних зразків.

Через це навчальні зразки моделі є неадекватними в системі виявлення вторгнень, а результати виявлення просуваються з високим рівнем помилкових виявлень.

Схема визначення абсолютних міток виражається як класифікація. Вона виконується для сортування даних залежно від навчання та значень необхідних характеристик.

Цей процес називається категоризацією класів/прогнозуванням/керованим навчанням.

Модель використовується для передбачення міток класів і тестування створеної моделі на тестових зразках, відповідно передбачається точність політики класифікації.

Методи машинного навчання застосовуються для ретельного аналізу та перевірки величезних мережевих даних на наявність аномальних/точних даних [10].

Оскільки дані генеруються з різних джерел, трафік в мережі величезний. Методи машинного навчання, подібні до кластеризації, класифікації, використовуються для створення потужної системи виявлення вторгнень.

А також методи, засновані на глибокому навчанні (DL), приділяють увагу для успішного просування великої кількості наборів даних системи виявлення вторгнень через Інтернет речей [11].

Основна ідея дослідження роботи полягає в розробці моделі виявлення вторгнень в мережі Інтернету речей.

Через складну та мінливу в часі динамічну природу мережі Інтернету речей, зразки мережевих вторгнень об'єднуються у велику кількість нормальних зразків даних.

Через це навчальні зразки моделі є неадекватними в системі виявлення вторгнень, а результати виявлення проходять з високим рівнем помилкових виявлень.

Багато досліджень і схем знаходяться в тренді, хоча точність систем виявлення вторгнень на основі P-GAN не значно покращилася.

Велика кількість існуючих методів має обмеження, такі як обчислювальні витрати і неточний результат класифікації системи виявлення вторгнень.

Для вирішення вищезазначених проблем, тепер, IWO-DDL+ANN представлені для підвищення продуктивності всієї системи Інтернету речей P-GAN.

Найважливішою частиною цього дослідження є побудова моделі P-GAN, попередня обробка, вилучення ознак, вибір ознак та метод ідентифікації.

Підхід використовується, щоб запропонувати надзвичайно правильний результат системи виявлення вторгнень, забезпечуючи роботу для компетентного підходу до заданої бази даних.

У запропонованій моделі виявлення вторгнень HDGAN спочатку виконується попередня обробка зібраних даних з системи пристроїв Інтернету речей для зменшення шумових зразків

Наступним кроком є підсилення міноритарних зразків за допомогою алгоритму TBSS.

В цьому випадку мова йде про те, що модель GAN повністю вивчає особливості міноритарних зразків і значно скорочує час навчання моделі та вилучення ознак за допомогою застосування модифікованого аналізу головних компонент.

Таким чином, застосовуючи вибір обгортки ознак за допомогою методів поліпшеної оптимізації китів (IEWO).

Було задіяно розподілену мережу GAN, яка забезпечує кращу точність виявлення та класифікацію вторгнення.

2.3 Висновок

В розділі описано передумови побудови моделі функціонування системи керування IoT-інфраструктурою під час несанкціонованих вторгнень

Також в розділі описано основні аспекти функціонування генеративних змагальних мереж (GAN), можуть використовуватися для здійснення висновку щодо наявності вторгнення в IoT-інфраструктуру.

Однією з головних переваг GAN для задач класифікації є те, що вони можуть генерувати синтетичні дані, які можна використовувати для доповнення навчальних даних.

Генеруючи нові дані, GAN можуть збільшити розмір навчальних даних і підвищити надійність і точність моделей машинного навчання.

Описано, що однією перевагою GAN для класифікації є те, що вони можуть вивчати основний розподіл даних, що може покращити здатність моделі узагальнювати нові дані. GAN можуть вивчати складні розподіли даних і генерувати дані, які відображають основні характеристики даних.

Однак є й певні недоліки використання GAN для класифікації. Однією з головних проблем є нестабільність навчання, яка може призвести до колапсу режиму або коливань. Це може призвести до низької продуктивності та недостатньої різноманітності згенерованих даних.

Таким чином, ГСН мають потенціал для покращення продуктивності та точності моделей машинного навчання для задач класифікації, зокрема для застосування виявлення вторгнень в IoT-інфраструктуру і стати основаою для розроблення метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень.

3 МЕТОД СИНТЕЗУ АПАРАТНО-ПРОГРАМНИХ ЗАСОБІВ КЕРУВАННЯ ІОТ-ІНФРАСТРУКТУРОЮ ПІД ЧАС НЕСАНКЦІОНОВАНИХ ВТОРГНЕНЬ

3.1 Основи методу синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень

У дослідженні запропонована метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, який на відміну від відомих застосовує апаратне рішення Hybrid Deep-GAN, і який може ефективно виявляти вторгнення, з якими стикаються мережі IoT через динамічну природу мереж Інтернету речей.

Запропонований GAN займає важливе місце в області глибокого навчання, і основною проблемою в цій області є розробка системи виявлення вторгнень, яка забезпечує рішення для порушень безпеки.

Таким чином, було побудовано модель Deep-GAN для виявлення всіх невідомих атак.

Через велику кількість різнорідних даних, що генеруються з мережі, обов'язковою є попередня обробка, яка тут здійснюється за допомогою алгоритму TBSS, який ефективно заповнює відсутні дані.

А потім метод MPCA, адаптований для виділення ознак, щоб виділити значущі ознаки.

А також алгоритм оптимізації Enhanced Whale орз метою відбору найбільш придатних ознак.

Нарешті, метод HDNN+ANN застосовано для точної класифікації вторгнень на заданому наборі даних.

В кінцевому результаті зазначено, що інноваційний метод EWO-HDNN+ANN забезпечує високу продуктивність щодо точності, достовірності, запам'ятовування, f-міри та пропонує нижчу частоту помилкових спрацьовувань,

зменшує обчислювальну складність у порівнянні з іншими алгоритмами машинного навчання.

У цьому дослідженні розроблення методу синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень пропонується застосування PP-GAN з IWO-DDL+ANN для виявлення вторгнень через Інтернет речей.

Метод включає в себе побудову моделі P-GAN, попередню обробку даних, виділення та вибір ознак, класифікацію та оцінку результатів.

Загальний дизайн представленого методу зображено на рис. 3.1.

P-GAN побудовано для виявлення вторгнень з використанням Інтернету речей. GAN є дуже впливовим і досвідченим методом глибокого навчання.

Тут GAN апроксимує генеративну модель, використовуючи змагальний процес. Зазвичай GAN має дві автономні моделі, такі як B - генератор і P - дискримінатор.

Генеративна структура оцінює розподіл даних *r(l)* на основі фактичного простору даних a.

Метою B є створення нової змагальної вибірки B(s), яка досягається з подібного розподілу a.

Тут дискримінатор моделі P є наслідком ймовірності *P(x),* стосовно наведеного прикладу x - це реальний набір даних, сформований B.

Основна мета B - підвищити ймовірність того, що P помилково сприйме створені дані як реальні, а мета P - виконати протилежну задачу.

Отже, P і B будуть досягати *minmax* еквіваленту і врешті-решт досягнуть елітного результату. Функція цінності *W (B;P)* описується таким чином:

$$\min_{B}\max_{P} W(B,P) = E_{a\sim r_{data}}(a)[log\ \ P(a)] + E_{s\sim r_s(s)}[\log(1 - P\big(B(s)\big))]. \quad (3.1)$$

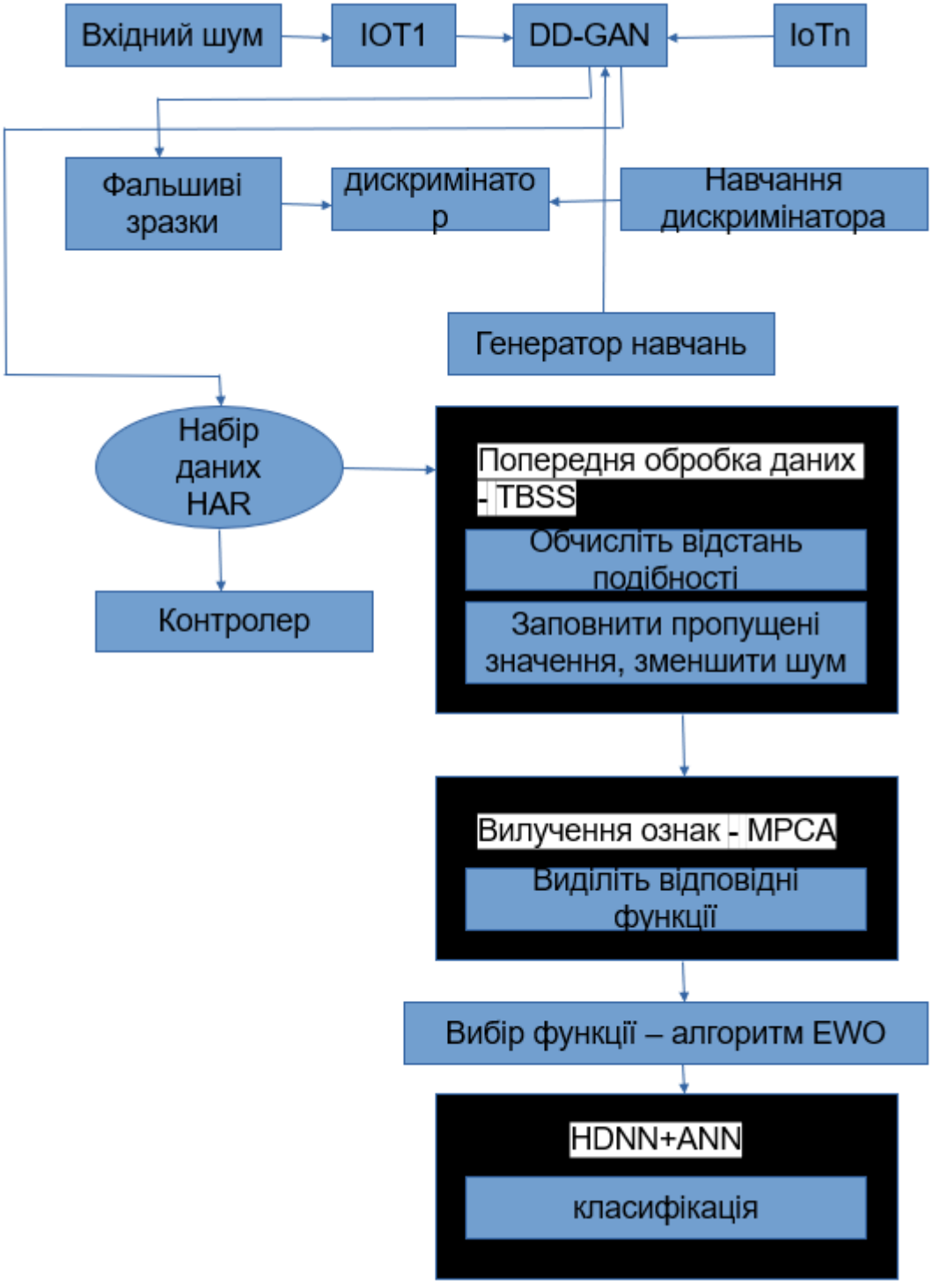


Рисунок 3.1 - Гібридне розподілене виявлення вторгнень на основі глибокого навчання в інфраструктурі Інтернету речей

Надати перевагу Інтернету речей, який містить множину *M* з *j* Інтернету речей, де кожен захисник Інтернету речей і має набір попередньо переданих точок

даних P_i, які є результатом розподілу r_{data} *i (a),* де *a*може бути часовим рядом, військовими даними, економічними звітами або базою даних спостережень за станом здоров'я.

Вважається, що P_i складається з точок даних зі справжнього стану захисника Інтернету речей, коли в Інтернеті речей немає атак.

Також нехай $P_1 \cup P_2 \cup \ldots \cup P_j = P$, де P представляє всі доступні дані з розподілом r_{data}.

Тут кожен захисник Інтернету речей і намагається дізнатися розподіл генератора rl_i в доступній йому базі даних P_i так, щоб $rl_i = r_{data}$, і використовує цей розподіл для виявлення атаки в системі.

Вторгнення в системі - це будь-яка дія зловмисника, яка спрямована на те, щоб захисник Інтернету речей не наближався до відповідних точок даних з метою не допустити подальшого розподілу даних r_{pi}.

Насправді, коли захисник Інтернету речей здійснює розподіл власного фактичного стану, він без особливих зусиль не надає перевагу точці даних, яка відрізняється від нормального розподілу станів.

Загалом, штучні нейронні мережі складаються зі штучних нейронів та функції активації, які відображають вхідні дані на вихід.

Для кожного захисника Інтернету речей *i* визначається додаткова штучна нейронна мережа, яка називається дискримінатор $P_i(x,\theta_{pi})$, який отримує точку даних а і виробляє значення між 0 і 1.

Якщо результат дискримінатора ближче до 1, отримана точка даних знаходиться в нормальному стані, в іншому випадку це вторгнення на захисника Інтернету речей *i*.

У той час як всі генератори захисників Інтернету речей мають намір зменшити функцію цінності, згадану в (1), дискримінатор намагається збільшити її значення.

Отже, життєздатний результат для дискримінатора і генератора може бути досягнутий з наступної задачі на мінімакс:

$$\{P_i^*, B_i^*\}=argmin_{B_i} \arg\max P_i W_i(P_i, B_i) \quad (3.2)$$

Тут, розподілена система виявлення вторгнень на основі GAN розвиває структурний дизайн методів класифікації об’єктів.

Зусилля розподіленої GAN полягає у встановленні дискримінатора на кожному захиснику інфраструктури Інтернету речей далі, ніж розподіл баз даних між собою.

Це робиться для того, щоб кожен дискримінатор захисника інфраструктури Інтернету речей міг розрізняти, коли нова точка даних розширює загальний розподіл даних, r_{data}.

У P-GAN на етапі навчання він забезпечує роботу центрального блоку, який включає генератор B_{φ}, де φ - це вага генератора штучних нейронних мереж.

Крім того, кожен захисник Інтернету речей містить дискримінатор, визначений як $P_{\theta i}$, де θ_i - вага кожного дискримінатора створеної штучної нейронної мережі.

У моделі кожен Захисник Інтернету речей пов'язаний щонайменше з одним Захисником Інтернету речей у мережі Інтернету речей, так що граф зв'язків Захисника Інтернету речей повинен описувати цикл.

Крім того, на етапі навчання всі захисники Інтернету речей підключені до центрального блоку.

GAN компетентний у класифікації наслідків на основі навченої вибірки, але через неконтрольоване навчання та нестабільне навчання його стає дуже складно навчати та генерувати вихідні дані.

3.2 Попередня обробка даних з використанням перенесення за подібністю підпросторів

Попередня обробка даних у запропонованому методі синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень з використанням перенесення за подібністю підпросторів (Transfer by Subspace Similarity , TBSS).

Алгоритм поліпшеного перенесення за подібністю підпросторів адаптовано для попередньої обробки даних, що значно підвищує точність виявлення вторгнень для ексклюзивної бази даних розпізнавання людської активності (Human Activity Recognition, HAR).

Перенос за подібністю підпростору гнучкий для виявлення активності в реальному часі.

Основною метою є розробка алгоритму для роботи з цими послідовними даними з метою покращення якості інтелектуального аналізу даних.

Це здійснюється шляхом включення відсутніх даних, точного налаштування шуму та значної модифікації волатильності для даної бази даних.

Для цього необхідно передати дані:

$$P \text{ до } \hat{P},\ \hat{p} = (a\ n\ e,\ cm),\ \hat{p} \in \hat{P}. \tag{3.3}$$

Звідси покращене перенесення за подібністю підпростору - це досвідчена стратегія кластеризації.

Вона працює, щоб розділити ідентичні дані в наборі даних на певні групи відповідно до різних міток.

Це створює колекцію τ на основі різних класів:

$$\tau = \left\{T_0^{C_x}\right\}_{x=1}^{m} \tag{3.4}$$

Методи вибірки випадковим чином відбирають дані, і ми можемо побудувати піднабори даних.

Це здійснює передумови для визначення важливої інформацію між ними як нові ознаки, використовуючи функцію Transfer by Subspace Similarity (перенесення за схожістю підпростору) на рис. 3.2.

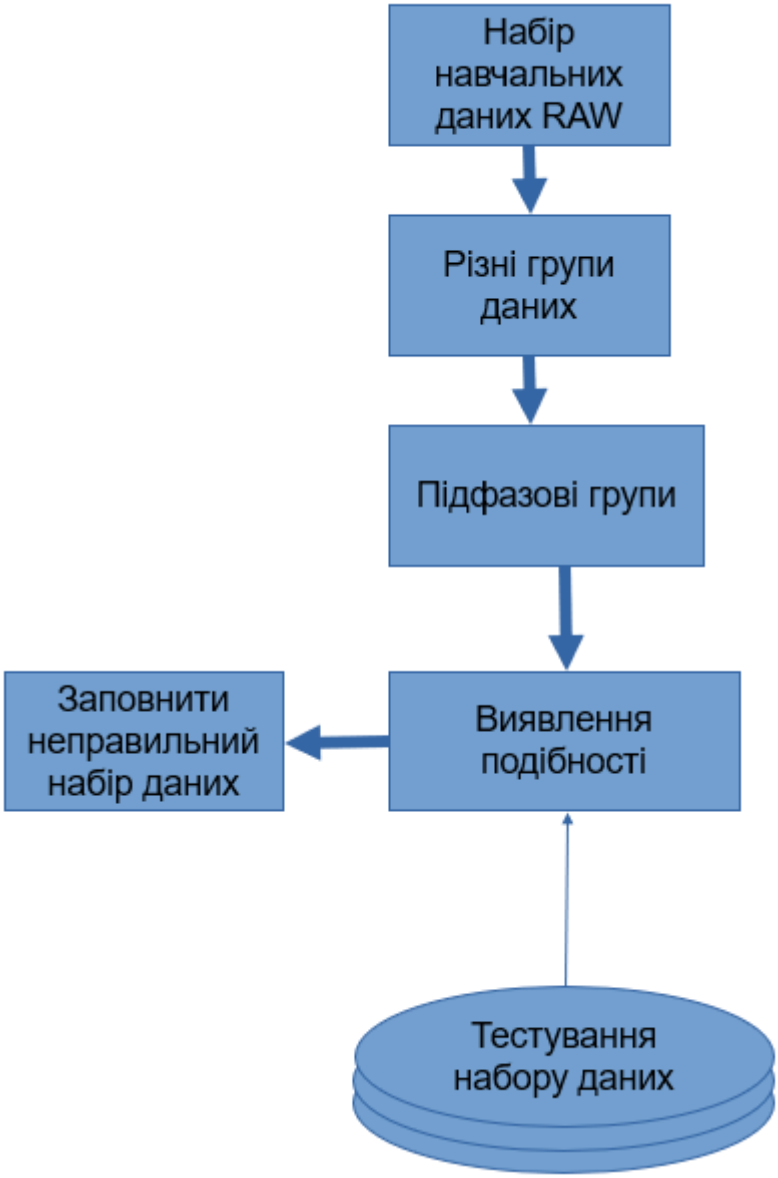


Рисунок 3.2 - Схема покращеного алгоритму перенесення за подібністю підпросторів

Перевага перенесення за подібністю підпросторів полягає в обчисленні ймовірності між кожною схожістю підпросторів та простоті реалізації.

Наступним кроком методу є необхідність визначити витрати часу о на основі часу, витраченого на побудову підпростору, та часу, витраченого на розподіл вихідного простору.

Якщо вихідний простір великий, то час дискретизації може бути більшим, оскільки необхідно забезпечити рівень покриття для вихідного простору.

Якщо ж вихідний простір менший, то час вибірки та витрати часу будуть меншими.

Таким чином, метод попередньої обробки використовується для підвищення точності виявлення вторгнень за допомогою алгоритму перенесення за подібністю підпростору.

3.3 Виділення ознак за допомогою модифікованого аналізу головних компонент

У цій роботі алгоритм модифікованого аналізу головних компонент спроектовано для вилучення ознак, в якому основна увага приділяється зменшенню кількості ознак.

Аналіз головних компонент спрямований (Principal Component Analysis — MPCA) на зменшення простору даних високої розмірності, тобто експериментальних змінних, до простору ознак низької розмірності, тобто автономних змінних, які є важливими для переконливого визначення даних.

Це може бути застосовано у випадку, коли між попередніми змінними існує високий зв'язок.

Під час видалення другорядних компонент аналіз головних компонент може зменшити кількість ознак і помістити набір даних у підпростір з низькою розмірністю [22-25].

Аналіз головних компонент є типовим методом дослідження багатовимірних даних, який адаптовано для лінійного вилучення ознак.

Тому було обрано процес вилучення ознак за допомогою методу головних компонент.

Аспектами цього методу є експлойти як вектори ознак, які добре використовуються для символізації набору даних системи виявлення вторгнень в IoT-інфраструктурі.

Стандартний алгоритм аналізу головних компонент може бути застосований для вилучення ознак з малих наборів даних і ігноруватиме важливу інформацію про ознаки. Метод аналізу головних компонент не дає гарантії того, що дані, пов'язані з відповідними класами, будуть якісно стиснуті.

Щоб запобігти вищезазначеним проблемам, планується модифікований аналіз головних компонент.

Метод модифікованого аналізу головних компонент зменшить вплив власних векторів після масивних власних значень шляхом стандартизації j-го елемента y_{ij}, і-ої ознаки вектора у відносно його стандартного відхилення, $\sqrt{\lambda_j}$.

В результаті новий вектор ознак y_i' модифікується як

$$y_i' = [\frac{y_{i0}}{\lambda_0}, \frac{y_{i1}}{\lambda_1}, \dots . \frac{y_{i(r-1)}}{\lambda_{r-1}}] \tag{3.5}$$

Однорідні вектори ознак використовуються для побудови нового підпростору ознак. У цьому процесі вектори ознак спочатку нормалізуються за допомогою квадратного кореня з наступних власних значень, а потім обчислюється відстань між навчальними та тестовими ознаками

Зазвичай, лінійне перетворення (аналіз головних компонент) можна подати у вигляді наступного рівняння:

$$C = NA \tag{3.6}$$

Тут *N* - матриця перетворення, *A* - вихідний вектор і *C* - перетворений вектор для визначення матриці перетворення *N*, наступне рівняння:

Застосовується, тут матриці *I,Z,U* та λ - квадратна матриця з одиницею на діагоналі, коваріаційна матриця вихідного зображення, власні вектори та власні значення. U_j та λ_j *(j=1,2,...x)* можна обчислити за рівнянням (2), причому власні значення впорядковані як $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$.

Власні вектори *U* можна подати як $U=[U_1, U_2,.....,U_m]$.

У модифікованому аналізі головних компонент навчальні вибірки, які асоціюються з певним додатком, вибираються з набору даних системи виявлення вторгнень, а перетворена матриця *N'* була отримана з цих навчальних вибірок.

Її можна описати як послідовне рівняння:

$$C = N'A \tag{3.7}$$

$$W_M = b_1u_1 + b_2u_2 + \cdots . + b_Mu_M \tag{3.8}$$

$$Z = \sum_{i=0}^{1} b_1u_1; 1 < M \tag{3.9}$$

На відміну від рівнянь (7) і (8), варіація присутня в матриці перетворення і, головним чином, присутня у вибірках для обчислення коваріаційної матриці, тут ми беремо одиницю з навчальних вибірок; наступна одиниця - з усього набору мовних даних для розпізнавання.

Основною метою модифікованого аналізу головних компонент є обчислення трьох матриць подібності, що використовують вимірювання подібності на основі трьох параметрів, а саме: взаємної інформації, кутової інформації та ядра Гауса.

Модифікований аналіз головних компонент об'єднує метод обгортки та метод прямого відбору з домінуючою дискримінативною здатністю для класифікації зразків.

Можна адаптувати модифікований аналіз головних компонент до функцій, в яких кількість навчальних вибірок є меншою за розмірність даних.

Модифікований аналіз головних компонент забезпечує вищу точність класифікації та результат кластеризації порівняно з аналізом головних компонент.

Модифікований аналіз головних компонент - це математичний алгоритм, який використовує лінійну корекцію для з'єднання даних з простору великої розмірності до простору малої розмірності. Простір низької розмірності фокусується на власних векторах коваріаційної матриці.

У цій роботі було адаптовано модифікований аналіз головних компонент для вилучення цінних значущих ознак вторгнення для набору даних розпізнавання людської активності шляхом зменшення похибки та декореляції ознак.

В результаті, модифікований аналіз головних компонент плідно зменшує розмірність набору даних за рахунок включення координат з високим значенням дисперсії та ухиляється від даних з низькою дисперсією, отримуючи вхідні дані з нормальними параметрами, а дані про втручання включають такі характеристики, як середнє значення та стандартне відхилення.

Середнє = сума кількості даних / загальна кількість даних (3.10)

Середньоквадратичне відхилення: також називається середньоквадратичним відхиленням, оскільки воно є квадратним коренем із середнього квадратичного відхилення від середнього арифметичного.

$$\sigma = \sqrt{(\sum(a - \ddot{a})/m)} \tag{3.11}$$

Метод модифікованого аналізу головних компонент використовується для вилучення інформативних ознак із заданого набору даних, а також для зменшення розмірності цінних ознак.

Було доведено, що модифікований аналіз головних компонент займає менше часу для вилучення ознак з нелінійної комбінації змінних, таких як мережеві дані Інтернету речей, і використовує мінімальну кількість ознак порівняно з алгоритмом аналізу головних компонент, на відміну від алгоритму автоматичного кодування, який займає більше часу для вилучення ознак.

Також було помічено, що модифікований аналіз головних компонент був протестований з наборами даних Інтернету речей, в яких він не усуває помітні ознаки, які вимірюються як головна ознака для точного обчислення прогнозу.

3.4 Виділення ознак за допомогою алгоритму EWO

Для ефективного виділення ознак у наборі даних системи виявлення вторгнень у роботі було використано WOA алгоритм (Whale Optimization Algorithm, WOA).

Метою виділення ознак є виявлення релевантних ознак з даних для визначення появи та відсутності ознак вторгнення та ознак, що не пов'язані з вторгненням.

WOA алгоритм використовується для численних оптимізаційних задач для пошуку оптимального рішення і відбору значущих ознак.

Алгоритм WOA важливий для вибору релевантних ознак і точних, що продуктивність алгоритму може бути покращена для отримання більш високих результатів, і застосовується для вибору ознак в наборах даних системи виявлення вторгнень для точного виявлення.

Алгоритм базується на стратегії полювання горбатих китів і дає можливість створити алгебраїчну модель полювання.

Кити використовують метод бульбашкової сітки для полювання на дрібну рибу, кружляючи навколо неї та створюючи велике коло з бульбашок, які приваблюють рибу до поверхні.

Потім кити переслідують рибу та споживають її.

Полювання включає три етапи: обхід, експлуатацію та розвідку.

Якість експлуатації та розвідки визначається цінністю зібраної риби.

На етапі обходу кити визначають місцезнаходження риби і оточують її. Початкове розташування визначається довільно, а потім інші агенти переглядають свою позицію, щоб знайти оптимальне розташування до цілі.

На етапі роботи китовий алгоритм оптимізації піддається навчанню з використанням техніки навчання з переносом імітуючи поведінку найкращого рішення, що допомагає зменшити різноманітність популяції.

Однак на етапі дослідження цього алгоритму виявлено, що процедура навчання на випадковій особині має невелику ефективність та не може ефективно перетворювати дані між групами, що впливає на швидкість збіжності такої процедури.

Це призводить до низької швидкості збіжності звичайного алгоритму оптимізації кита.

Для вирішення цієї проблеми пропонується вдосконалена версія Китового методу, яка покращує швидкість збіжності шляхом усунення локальних оптимумів та фокусування на глобальному оптимумі.

Моделювання навколишнього середовища здійснюється за допомогою оптимізації рою тварин, що забезпечує змішування найкращих рис та інших партнерів та підвищує якість особини.

Метод динамічного соціального пошуку полягає в визначенні соціального рейтингу та соціальної сили кожного індивіда, а також в побудові соціальної мережі [27].

Щоб покращити комунікацію між групами, цей метод використовується для побудови динамічного сусідства китів, а також розроблено нову ідею щодо оновлення сусідства з метою збільшення різноманітності популяції та точності оцінок.

Рисунок 3.3. - Блок-схема алгоритму оптимізації

Положення кита та його оточення описується такими рівняннями

$$\vec{Q} = |\vec{C}s\vec{A} * (t) - \vec{A}(t)|$$

$$\vec{A}(t+1) = \vec{X}(t) - \vec{T} \times \vec{Q} \quad (3.13)$$

Через $\vec{T}$ та $\vec{C}$ подано векторні коефіцієнти, t позначає поточну ітерацію, через $\vec{A}$ - вектор положення, а $\overrightarrow{A}^*$ - найкращий розв'язок (положення), що починається випадковим чином. Коефіцієнти вектора $\vec{T}$ та $\vec{C}$ обчислюються за допомогою рівнянь (14) і (15)

$$\vec{T} = 2\vec{a} \times \vec{p} - \vec{a} \quad (3.14)$$

$$\vec{C} = 2s\vec{p} \quad (3.15)$$

Компоненти $\vec{a}$ зменшуються від 2 до 0 на кожній ітерації лінійно і символізують довільне значення між 0 і 1

Алгоритми WOA використовують бульбашкову сітку для створення ворожого середовища для своєї здобичі та полювання на неї [28-29]. Ці кити оточують здобич, набагато меншу, ніж вони, та можуть змінювати своє місцезнаходження, щоб знайти оптимальний кут нападу. Числовий параметр WOA вказаний у формулах.

$$A(t+1) = A(t) - T.|C.A^*(t) - A(t)| \text{ якщо } r < 0.5 \quad (3.16)$$

$$A(t+1)|C.T(t) - A(t)|.e^{bl} \cos(2\pi t) + A^*(t) \text{ якщо } r \geq 0.5, \quad (3.17)$$

де A - вектор розташування всіх китів; t - час та індекс ітерації; A* - оптимальний розв'язок, отриманий на даний момент; T=2a. (p-a);C=2.p; a - вектор коефіцієнтів, які лінійно згущуються від 2 до 0 через ітерації; p - випадковий вектор зі значеннями, що містяться між 0 та 1; b - постійна величина, яка виражає форму логарифмічної спіралі відповідно до конкретного шляху, і в цій точці значення присвоюється 1; l - випадкове число між - 1 і 1; r - випадкове число між 0 і 1 і застосовується для модифікації між (3.14) і (3.15) при зміні розташування китів; в

рівняннях (3.16) і (3.17) можливості складають 50% - 50%; це призводить до того, що в процесі оптимізації кити віддають перевагу будь-якому шляху випадковим чином з рівною ймовірністю.

Під час фази бульбашкової мережі випадкове значення для T знаходиться між - 1 і 1, хоча у фазі пошуку випадкове значення вектора T може бути як > 1, так і < 1. Метод пошуку представлено у вигляді рівняння (18):

$$A(t+1) = A_{rand} - T.|C.A_{rand} - A(t)| \quad (3.18)$$

Випадковий пошук зі значенням |T| >1 полегшує пошукову операцію і зберігає алгоритм WOA для виконання глобального пошуку. Випадкові рішення створюються на ранній стадії процесу пошуку WOA. Потім ці результати модифікуються для кожної ітерації.

Процес пошуку повторюється до тих пір, поки не буде досягнуто максимальної кількості ітерацій.

Фаза виконання складається з двох етапів: обходу та перегляду позиції по спіралі. Дії з обходу можна придумати, лінійно зменшуючи a ⃗ від 2 до 0 на кожній ітерації.

Перегляд позиції по спіралі: У цьому місці руху кита до риби гвинтоподібний рух китів задається формулою

$$\overrightarrow{A}(t+1) = \vec{P}ae^{b1}acos(2\pi l) + \vec{N} * (t) \quad (3.19)$$

де $\overrightarrow{P'} = |\vec{A} * (t)\_\vec{A}(t)|$ - остання позиція між рибою та китом, b - статичний фактор або константа, що представляє спіральний рух китів, а також випадковий вектор [-1,1].

Крім того, існує ймовірність варіанту: або занурення вглиб через кружляння і форму спіралі точно задається рівнянням 6, або випадковий вектор значення r[0,1].

У фазі дослідження пошук риби вважається глобальним, і кити шукають здобич, піднімаючись до поверхні.

Перевага перемикання між експлуатацією та дослідженням базується на $\vec{T}$, векторі зі значеннями [0,1], де 0 означає дослідження, а 1 - експлуатацію.

$$\vec{Q} = |\vec{C}aA_{rand} - \vec{A}| \quad (3.20)$$

$$\overrightarrow{A}(t+1) = \vec{A}_{rand} - \vec{T}a\vec{Q} \quad (3.21)$$

де $\vec{A}_{rand}$ - остання позиція кита, яка вибирається довільно з-поміж інших китів.

Застосування теорії соціального навчання може допомогти створити асоціацію сусідства для кожного кита, що сприятиме покращенню поведінки в поточному найкращому рішенні, поліпшенню обміну даними між групами та збільшенню здатності процесу виходити з локального оптимуму.

Для поточної популяції:

$$B(t) = \{a_1(t), a_2(t), \dots., a_M(t)\}, \quad (3.22)$$

де M - розмір популяції. Для отримання впорядкованої популяції розраховуються і регулюються всі пристосованості особин від менших до більших

$$B_1(t) = \{a_{(1)}(t), a_{(2)}(t), \dots., a_{(M)}(t)\} \quad (3.23)$$

А соціальна позиція $x_{(i)}(t)$ має вигляд

$$I_{(i)}(t) = \frac{P_{(i)}(t)}{M} \quad i = 1,2, \dots, M \quad (3.24)$$

де $P_{(i)}$- випадкове число, а $I_{(i)}(t)$ з метою істоти покращити асоціацію з іншою істотою, отже, стадія експлуатації процесу принципово залежить від найкращого результату пошуку, а дослідницький потенціал вдосконалюється за рахунок взаємовідносин між групами, формується новий метод пошуку китів на основі динамічної стратегії сусідства в соціумі.

Значення пристосованості:

$$(\mathrm{A}) = \sum_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{yi} - \mathrm{y}', \tag{3.25}$$

де A - параметр моделі; yi - фактичне значення мережевого трафіку, де k - кількість ітерацій.

Мінімальне значення пристосованості може бути представлено як

$$(\mathrm{A})\mathrm{s.t.}\,\mathrm{L\,G} \leq \mathrm{A} < \mathrm{U\,G} \tag{3.26}$$

де A позначається як P-вимірна змінна, P - кількість параметрів, UG - верхня межа, LG - нижня межа.

Продуктивність алгоритму підвищується за рахунок ігнорування локальних оптимумів і більшої концентрації на глобальних оптимумах для збільшення швидкості збіжності та уникнення величезних коливань в кінці кожної ітерації.

Алгоритм 3.1 – Покращена оптимізація WOA

Початок

Встановити положення популяції китів (вторгнення) A

Обчислити придатність кожного кита (вища точність)

Ініціалізуємо популяцію китів, без ітерацій

Встановити a та p, обчислити T та C

Обчислюємо пристосованість кожного кита

Встановити A* як найкращу позицію кита-мисливця

Поставити t = 1

Поки t ≤ max ітерацій do

Для кожного кита-мисливця do

Якщо r < 0.5

If |T| < 1

Змінити поточне положення мисливського кита на (16)

Інакше, якщо |T| ≥ 1

Випадково вибрати іншого пошукового агента (функцію)

Оновити поточну позицію мисливського кита на (17)

Кінець якщо

Кінець якщо r ≥ 0.5

Оновити поточну позицію мисливського кита на (18)

Кінець якщо

Визначити локальний оптимальний розв'язок за допомогою (22) та (23)

Кінець коли

Переглянути A*, якщо є кращий результат

t = t +1

Проаналізуйте нащадків Ui

Якщо Ui краще за Ai, то

Переглянути особину i , Ai = Ui

Якщо Ui краще за A*, то

Переглянути найкращу особину, A* = Ui

Кінець якщо

Кінець якщо

t = t + 1;

Кінець коли

Вивести A* отримати найкращий розв'язок як більш точний

Кінець

Вивести найкращого пошукового агента.

V. Виявлення вторгнень за допомогою гібридного методу згорткової нейронної мережі на основі глибокого навчання зі штучною нейронною мережею (HDLCNN+ANN)

У запропонованому методі було застосовано поєднання методів класифікації HDNN+ANN для підвищення точності виявлення вторгнень для заданого набору даних. Передбачуваний процес глибокого навчання забезпечує вищу точність.

На рис. 3.4 показано алгоритм HDNN+ANN.

3.5 Навчання моделі

DNN використовується для збору інформації через виявлення. DNN має три шари: вхідний, прихований та вихідний.

Вхідний шар збирає вхідні значення, аналізує їх і виробляє "m" входів набору даних.

Цей процес був визначений з урахуванням вагових коефіцієнтів. Ваги - це допомога даних для вирішення проблем в NN [30].

Прихована інформація витягується прихованим шаром з вхідного шару і переходить до вихідного шару, як тільки прихована життєво важлива інформація витягується. DNN адаптовано для виявлення ознак вторгнення.

Навчання набору даних IDS здійснюється за допомогою DNN, а на етапі тестування ознаки класифікуються та ідентифікують вторгнення.

Запропонована HDLCNN містить вхідний, згортковий та класифікаційний шари. Представлений підхід має очевидні переваги для аналізу даних більшої розмірності.

Він використовує метод спільного використання параметрів, який застосовується у згорточних шарах для управління та зменшення кількості факторів.

Базова архітектура DLCNN показана на рисунку 3.5.

Вхідний шар приймає ознаки вторгнення з навчальних прикладів і перетворює дані в комбіновану форму для правильної передачі даних наступним шарам.

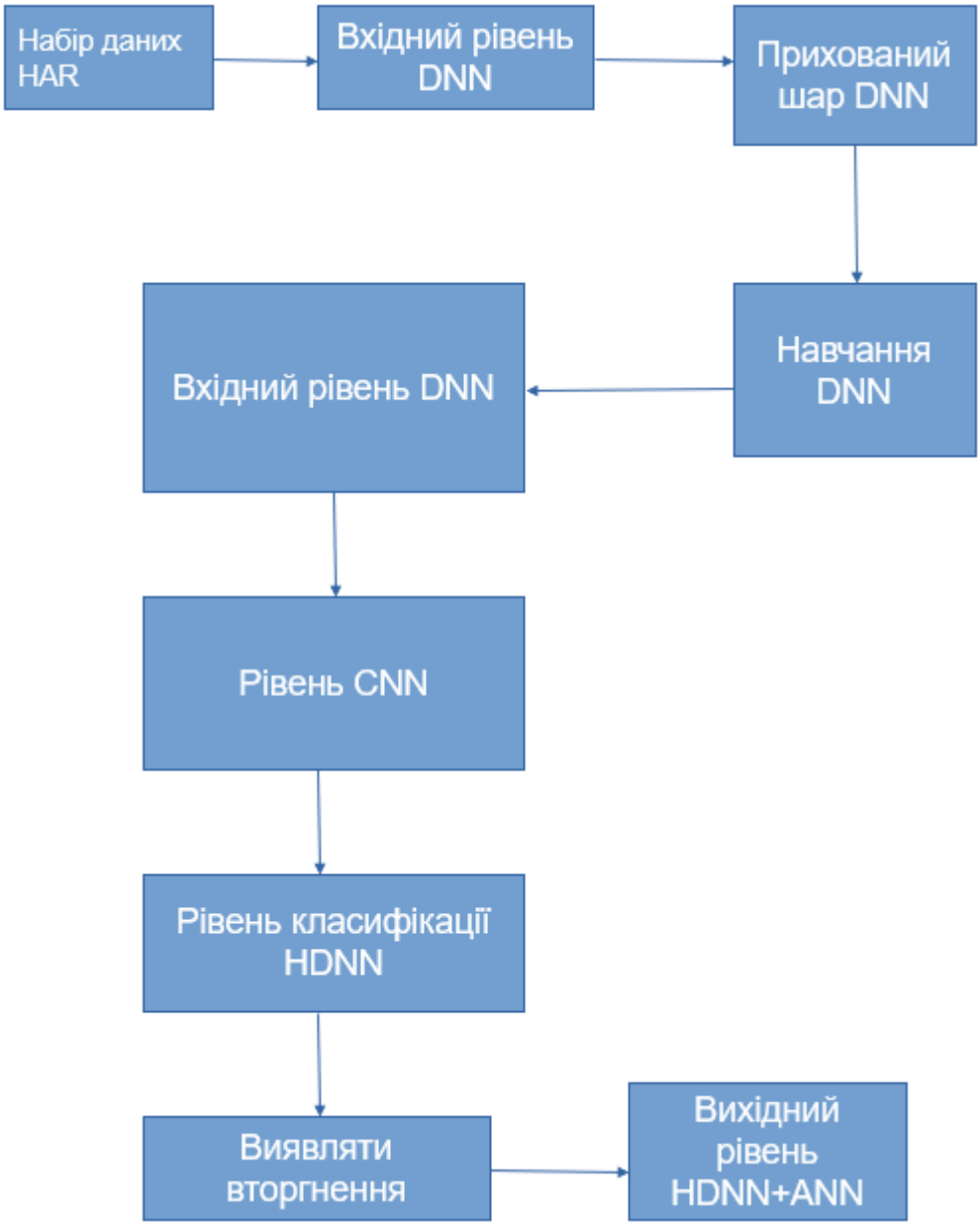


Рисунок 3.4 - Модель HDNN+ANN

Тут початкові параметри визначаються як масштаб локальних сприйнятливих полів і різнорідних фільтрів

Шар згортки зазвичай використовується для виділення важливих ознак і зменшення обчислювальної складності мережі.

Алгоритм 3.1 - Алгоритм виявлення вторгнень - HDNN+ANN

Введіть набір даних IDS

Для кожної вхідної ознаки описати ознаку вторгнення ∈ набору даних IDS do

Для нейронів вхідні ознаки do

Навчити ШНМ на заданому наборі даних

Гібридний DLCNN з ШНМ у P-GAN

Перетворення вхідних даних на шари згортки та класифікації

Виявлення ознак вторгнення

Відбір більш інформативних та релевантних ознак

Проведення навчання та тестування для заданої бази даних

Копіювання попередньо визначеної мітки ознаки вторгнення для кожного об'єкта, як зазначено у вхідній базі даних

Виявлення додаткових точних втручань

3.6 Тестування моделі

Гібридний P-GAN не потребує центрального блоку, оскільки всі дискримінатори на кожному IoTD зможуть ідентифікувати атаку в системі. P-GAN поєднує в собі глибоке навчання з моделлю GAN, тому кожен IoTD обробляє свої фактичні дані в режимі реального часу за допомогою свого індивідуального дискримінатора та будь-якого сусіднього дискримінатора.

Дієвий дискримінатор дасть 0,5 для стандартної точки даних про стан.

Отже, для виявлення вторгнення в систему, вихід дискримінатора можна порівняти з 0,5, і коли вихід близький до 0,5, IoTD залишається в стані відсутності вторгнення.

Проте, коли вихід наближається до 0/1, IoTD знаходиться під загрозою. Метод допомагає системі IoT ідентифікувати загрозу в системі без залежності від центрального блоку, оскільки кожен IoTD може перевірити дані свого сусіда.

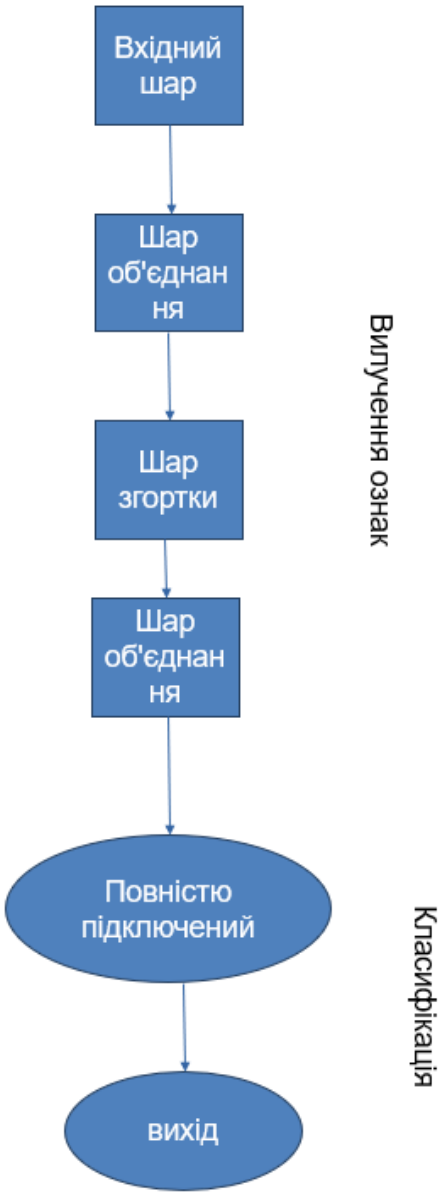


Рисунок 3.5. - Архітектура DLCNN

3.7 Результати експерименту

У цій роботі було використано набір даних розпізнавання щоденної активності, зібраний за допомогою в IoT-інфраструктурах - IoTD. Такі бази даних є приватними і стосуються таких аспектів як інформація щодо різних даних ioT.

База даних складається з 125443 семплів. База даних розділена на навчальну та тестову у співвідношенні 4:1. Для проведення експериментів та визначення результатів використовується програма на мові Python.

У цій роботі існуючі алгоритми, такі як централізований GAN і D-GAN з ANN, оцінюються за допомогою запропонованого алгоритму D-GAN з алгоритмом EWO-HDNN+ANN. Визначено такі показники ефективності, як правильність, точність, відкликання, f-міра, FPR та обчислювальна складність.

Достовірність визначається наступним чином:

$$\text{достовірність} = \frac{T_p + T_n}{(T_p + T_n + F_p + F_n)}$$

де Tp - істинно позитивний, Tn - істинно негативний, Fp - хибно позитивний і Fn - хибно негативний.

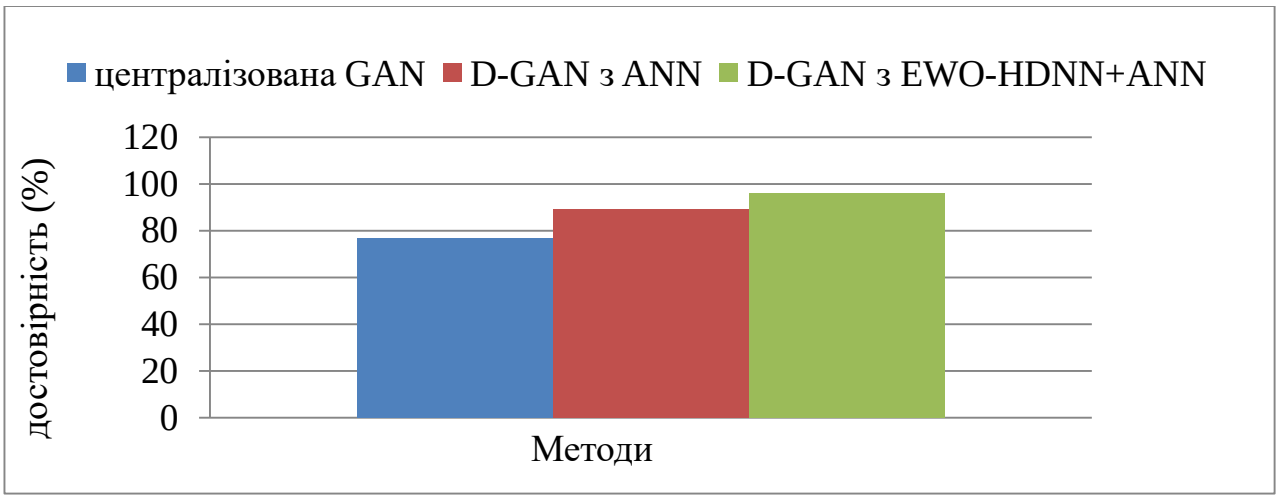


Рисунок 3.6 – Достовірність виявлення

Точність обчислюється за формулою:

$$\text{Точність} = \frac{\text{істино позитивний}}{\text{істино позитивний} + \text{хибно позитивний}} \quad (3.28)$$

Точність розглядається як обчислення правильності або якості.

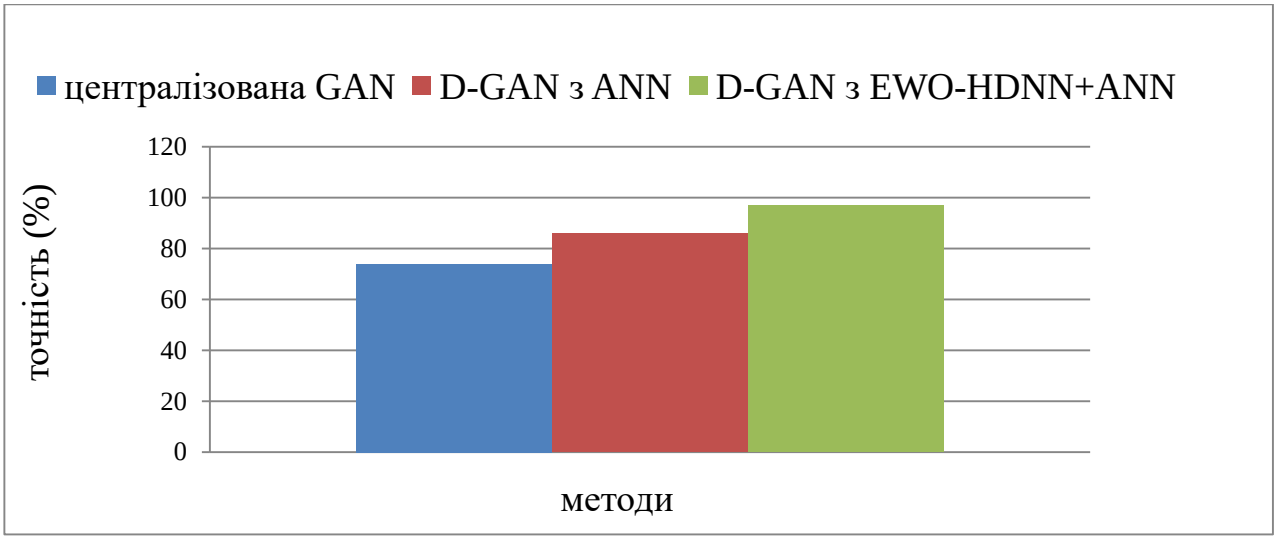


Рисунок 7.7 – очність

Recall обчислюється наступним чином:

$$\text{Recall} = \frac{\text{істино позитивний}}{\text{істино позитивний} + \text{хибно негативний}} \quad (3.29)$$

Recall - це відношення кількості знайденого в пошуку відповідного контенту до загальної кількості поточних релевантних документів.

F-міра є сумішшю точності та запам'ятовування,

$$F = 2.\frac{PR}{P+R} \quad (3.30)$$

Для оцінювання процедур класифікації покладаються на F-міру, оскільки вона є типовою мірою інкапсуляції точності P на додаток до пригадування R.

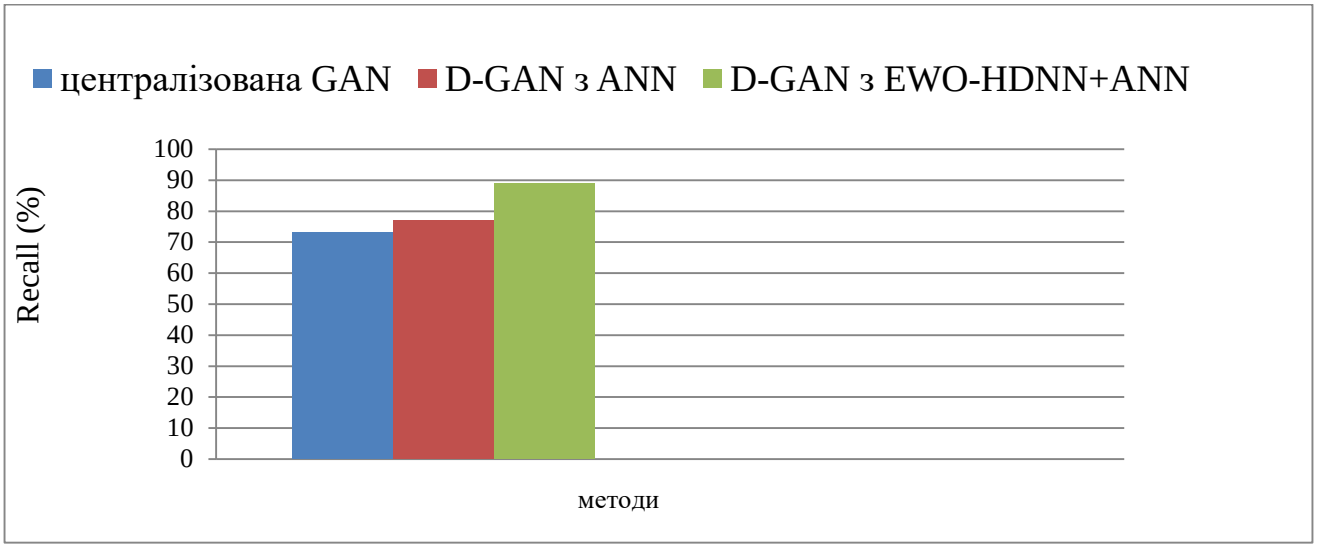


Рисунок 3.8 – Recall

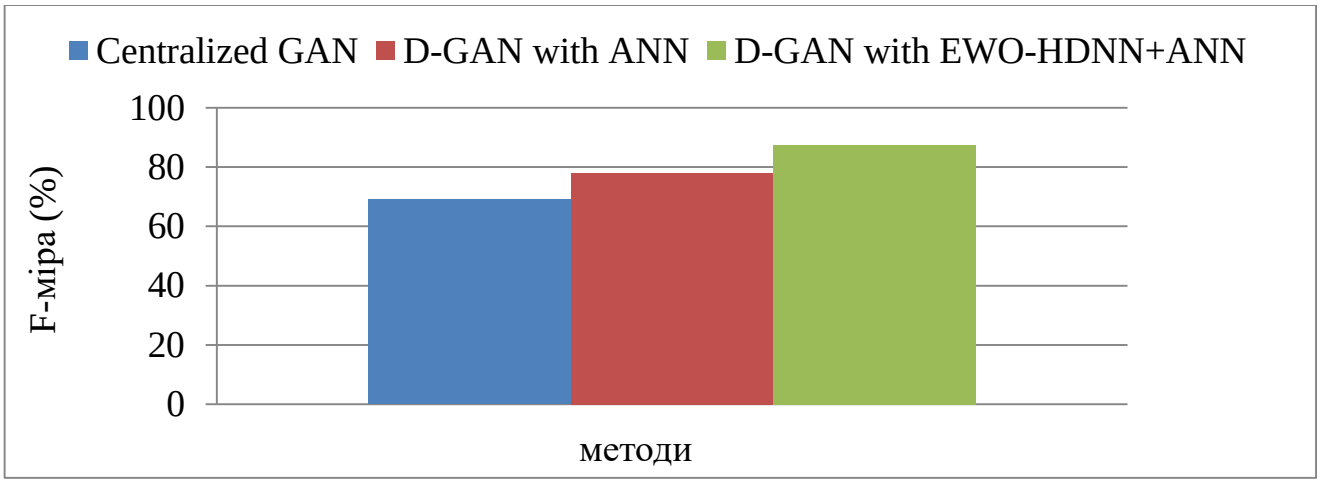


Рисунок 3.9 – F-міра

Частота хибнопозитивних спрацьовувань IDS обчислюється як

$$\frac{FP}{FP+TN} \tag{3.31}$$

Для IDS, FPR - це співвідношення між кількістю стандартних точок даних про стан, помилково класифікованих як вторгнення (FP), та кількістю реальних стандартних точок даних про стан (FP + TN).

Обчислювальна складність. Система є кращою, але запропонований метод пропонує меншу обчислювальну складність, коли зібрані дані попередньо обробляються за допомогою EWO, що зменшує складність.

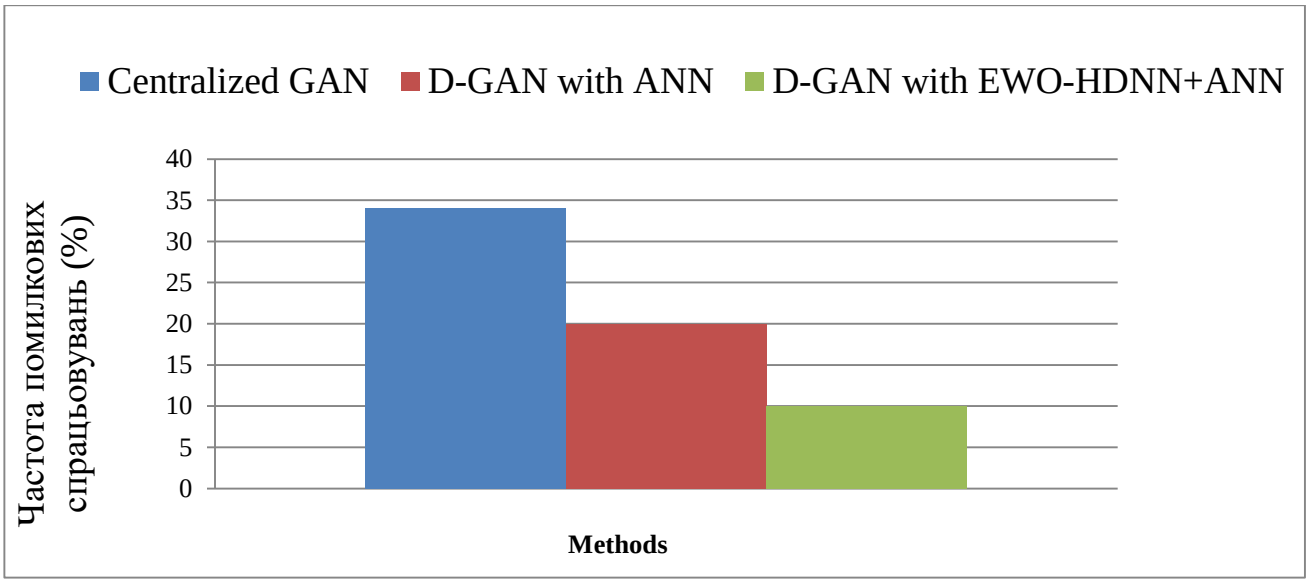


Рисунок 3.10 – Частота помилкових спрацьовувань

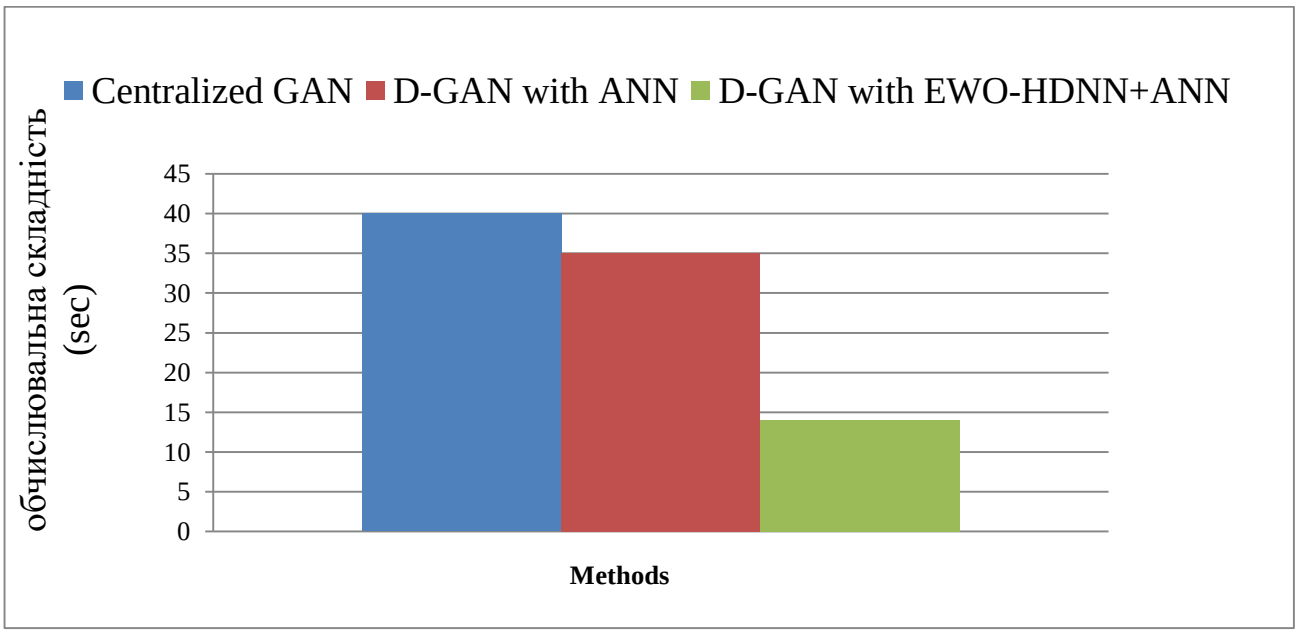


Рисунок 3.11 – Обчислювальна складність

У таблиці 3.1 наведено порівняльні значення для набору даних IDS з використанням існуючих та запропонованих методів

Таблиця 3.1 - Порівняльні значення для набору даних IDS з використанням існуючих та запропонованих методів

Методи	C-GAN	P-GAN з ANN	P-GAN з EWO-HDNN+ANN
Правильність (%)	76	85	97.4
Точність (%)	71	82	96
Відкликання (%)	67	74	88.5
F-міра (%)	69	78	87.4
FPR (%)	34	20	10
Обчислювальна складність (сек)	40	35	14

3.7 Висновки

У дослідженні удосконалено метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, який на відміну від відомих застосовує апаратне рішення Hybrid Deep-GAN, і який може ефективно виявляти вторгнення, і який забезпечує підвищення ефективності керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Запропонований метод застосовує Hybrid Deep-GAN і може ефективно виявляти вторгнення, з якими стикаються мережі IoT через динамічну природу мереж Інтернету речей.

Було доведено, що запропонований GAN займає важливе місце в області глибокого навчання, і основною проблемою в цій області є розробка системи виявлення вторгнень, яка забезпечує рішення для порушень безпеки.

Метод дозволяє виявлення невідомих атак. Через величезну кількість різнорідних даних, що генеруються з мережі, обов'язковою є попередня обробка, яка тут здійснюється за допомогою алгоритму TBSS, який ефективно заповнює відсутні дані.

Метод також застосовує MPCA, адаптований для виділення ознак, щоб виділити значущі ознаки, а також використовує алгоритм оптимізації Enhanced Whale для відбору найбільш придатних ознак.

Нарешті, метод HDNN+ANN застосовано для точної класифікації вторгнень на заданому наборі даних.

В кінцевому результаті зазначено, що метод EWO-HDNN+ANN забезпечує високу продуктивність щодо точності, достовірності, запам'ятовування, f-міри та пропонує нижчу частоту помилкових спрацьовувань, зменшує обчислювальну складність у порівнянні з іншими алгоритмами машинного навчання.

4 РЕАЛІЗАЦІЯ АПАРАТНО-ПРОГРАМНИХ ЗАСОБІВ КЕРУВАННЯ ІОТ-ІНФРАСТРУКТУРОЮ ПІД ЧАС НЕСАНКЦІОНОВАНИХ ВТОРГНЕНЬ

4.1 Вибір типу архітектури та зразків проектування

Розглянемо аспекти проєктування апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, які ґрунтуються на імплементації методу, описаного в Розділі 3.

На рисунку 4.1 представлено архітектуру гібридної апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, в основі яких лежить система машинного навчання.

Вона є основою здійснення висновку про вторгнення і містить три частини, тобто навчальну частину, частину висновків і частину трансплантації моделі. Навчальна частина реалізована на графічному процесорі та відповідає за навчання моделі з бази даних.

Частина логічного висновку побудована за допомогою FPGA і відповідає за логічний висновок.

Оскільки частина висновку FPGA не може безпосередньо завантажити модель, згенеровану частиною навчання GPU, компонент трансплантаціїдодано.

4.2 Частина навчання апаратних засобів

Мета етапу навчання – навчити модель отримувати максимальну точність за мінімальний час. Виходячи з двох причин, ми вибираємо GPU для реалізації навчальної частини. Однією з причин є те, що навчальна частина зосереджена на швидкій і точній моделі побудови. І більшість навчальних робіт виконується лише один раз. Отже, під час цієї фази обчислювальна здатність з високою щільністю є важливішою, ніж затримка обчислення. За однакового рівня ціни графічні процесори мають більше обчислювальних ядер і більш ефективні в навчанні

моделі, тоді як FPGA, ресурси яких обмежені та дорогі, рідко використовуються для навчання моделі. Інша причина полягає в тому, що графічні процесори легше програмувати, ніж FPGA. Оскільки вимоги до навчальної частини зазвичай змінюються, важливіше просте програмування.

На основі CUDA (Compute Unified Device Architecture) багато компаній уже розробили серію фреймворків, таких як TensorFlow, який є фреймворком програмного забезпечення з відкритим кодом для машинного навчання, розробленим Google Brain уже підтримували прискорення GPU.

Ці структури інтегрують функції CUDA та приховують деталі реалізації CUDA, дозволяючи користувачамзосередитися на розробці алгоритмів машинного навчання.

4.3 Проєктування апаратних засобів на основі FPGA

Мета етапу - зробити висновок із моделлю, згенерованою на етапі навчання, з мінімальною затримкою. Фаза висновків часто виконується знову і знову протягом певного досліджуваного періоду. Будь-яка невелика затримка кожного циклу буде накопичуватися до величезної кількості. Тому має сенс зменшити затримку висновків. ПЛІС складаються з логічних елементів (LE), які можуть бути запрограмовані в апаратні логічні схеми. І в більшості реалізацій FPGA немає системи вибору і декодування інструкцій, це змусить FPGA працювати набагато швидше, ніж GPU або CPU.

Однак ПЛІС мають деякі недоліки, такі як обмежені та дорогі ресурси та важка розробка.

Обмежена складним і важким виробничим процесом, ціна FPGA однакового розміру зазвичай дорожчаніж графічні процесори.

Це обмежує розмір площі FPGA, який не може бути занадто великим для проєктування.

FPGA можуть бути розроблені лише за допомогою HDL (мов опису

обладнання), таких як Verilog або VHDL (VHSIC (дуже високошвидкісна інтегральна схема) мова проектування обладнання), які вимагають глибоких знань логічних електронних схем і систем.

Подібно до мов асемблера, для опису навіть невеликої функції потрібна велика кількість речень HDL.

Якщо система занадто складна, як глибока нейромережа, важко використовувати HDL для її розробки.

Тепер труднощі проектування FPGA можна компенсувати за допомогою мов програмування високого рівня, таких як OpenCL (Open Computing Language), які не вимагають великих знань про апаратне забезпечення.

OpenCL – це фреймворк, який можна використовувати для розробки програм, які можна запускати на різних гетерогенних платформах, таких як графічні процесори (графічні процесори) і FPGA (програмовані в польових умовах).ворітні масиви).

4.4 Імплементація методу

Для ефективності більшість моделей машинного навчання не запускатимуться на платформах, які їх створюють. Тому необхідна фаза трансплантації моделі. Хоча вже є деякі інструменти, такі як MMdnn, для перетворення файлу моделі серед основних фреймворків, він не підходить для нашої роботи.

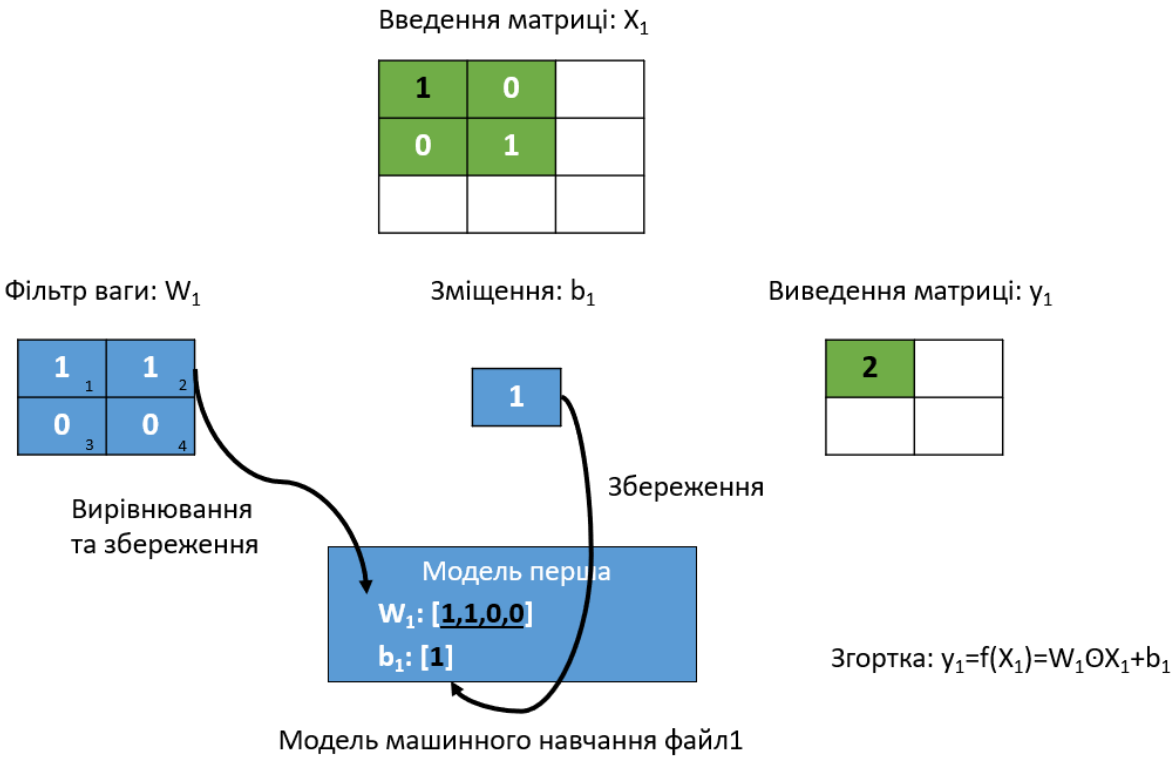
Оскільки фаза висновків реалізована засобами FPGA, всі функції створено з нуля, його фреймворк відрізняється від будь-яких інших основних фреймворків.

Файл моделі машинного навчання записує такі параметри, як ваги та зміщення, створені під час фази навчання моделі, і буде завантажено під час фази висновку. Рисунок 4.1 демонструє перетворення файлу моделі. Він містить операції згортання та зведення. У цьому прикладі десяткові крапки та способи зведення відрізняються. W1 у файлі моделі 1 зведено як [1, 1, 0, 0], тоді як W2 у файлі моделі

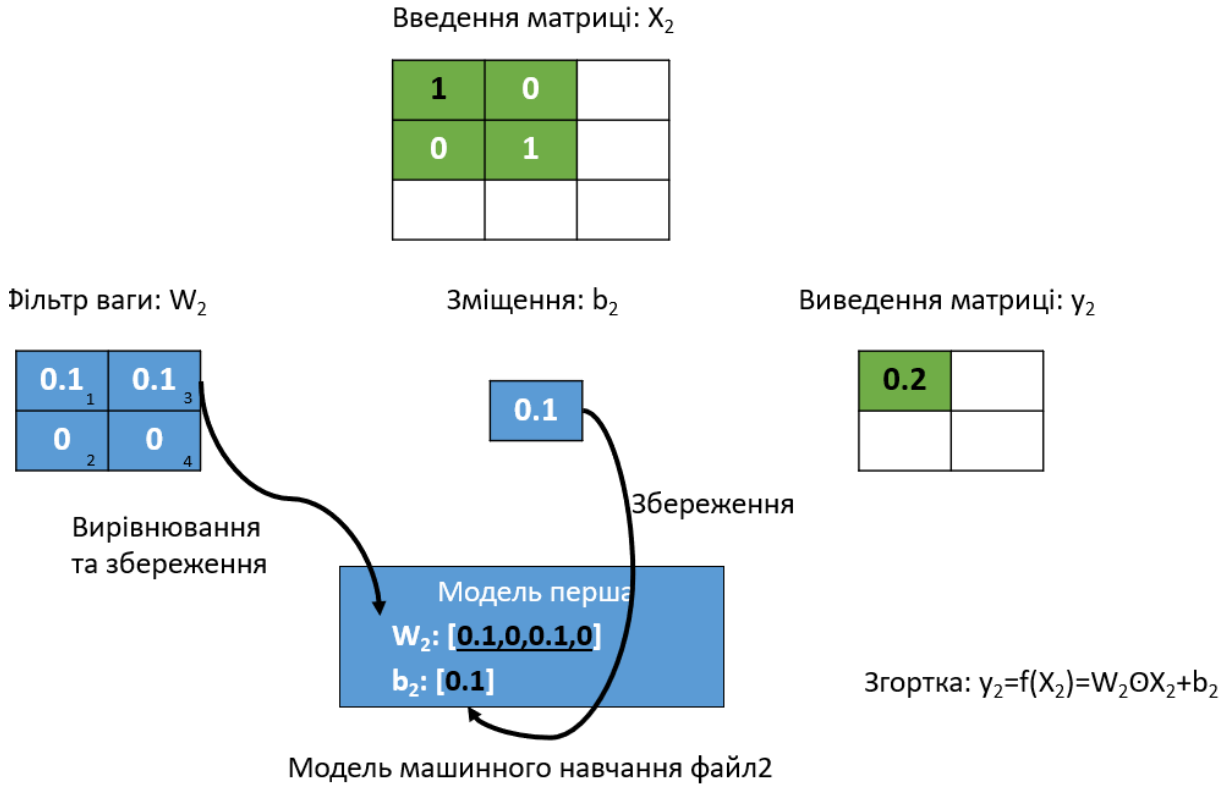
2 зведено як [0,1, 0, 0,1, 0]. Отже, модель 1 рамки 1не можна використовувати безпосередньо на фреймворку 2, він повинен спочатку виконати трансформацію моделі. Для реальної трансформації моделі машинного навчання ситуація була б складнішою, параметри навіть мають чотири виміри. Існує 256 способів зведення чотиривимірного параметра, і лише один спосіб працює для конкретної моделі.

4.5 Застосування гібридної платформи машинного навчання

Щоб перевірити ефективність рішення спроектуваної платформи, було впроваджено алгоритм машинного навчання відповідно до запропонованого методу.



а)



б)

Рисунок 4.1 – Схема застосування моделі

4.6 Апаратне забезпечення

Список пристроїв наведено в таблиці 4.1. Відеокарта GeForce 3040 спілкується з хост-машиною через PCIe gen 3 x16 (із пропускною спроможністю 15760 МБ/с).

Плата FPGA Altera Cyclone V передає дані з хост-машиною за допомогою PCIe gen 3 x8.

Обсяг навчальних даних, який зазвичай величезний і розділений на багато невеликих пакетів, не може бути завантажений у GPU за один раз.

Тому дані будуть часто передаватись між хостом і пристроєм.

З цієї точки зору, пропускна здатність стає вузьким місцем у продуктивності системи, тому краще вибративиберіть графічний процесор для навчання та FPGA для створення висновків.

Таблиця 4.1 – Список апаратних пристроїв

пристрій	Тип	Номер
ЦП	Процесор Intel(R) Xeon(R) E5-1620 v4 @ 3,50 ГГц	1
Пам'ять	Hunix DDR4 32g	4
Твердотільний диск	SSD 480Gb	1
GPU	NVIDIA 3050	1
FPGA	Комплект розробки Intel Altera Cyclone V GX FPGA	1

4.7 Програмні засоби

Програмні середовище показано в таблиці 4.2. Операційна система OpenSuSE Leap15.

Таблиця 4.2 - Програмне середовище

Програмне забезпечення	Версія
OpenSuSE Leap15	15
Python	3.5.2
Tensorflow	1.4.0
CUDA	8.0

4.7.1 Алгоритм використання прецеденту

Оскільки метою є перевірка продуктивності гібридної системи машинного навчання, було застосовано моделі, описані в розділах 2 та 3. які не надто складні для апаратної реалізації засобами FPGA, щоб показати ефект апаратного прискорення.

4.7.2 Навчання системи

На рисунку 4.2 подано потік керування навчанням. Він включає дві основні фази, тобто пряме поширення та зворотне поширення.

Різниця між значеннями, розрахованими шляхом прямого розповсюдження, і значенням мітки передається у зворотне розповсюдження для обчислення ваг і зміщень кожного шару.

Після кількох обчислювальних циклів різниця сходиться в невеликому діапазоні, процес навчання припиняється.

Остаточні ваги та зміщення будуть збережені у файлі моделі.

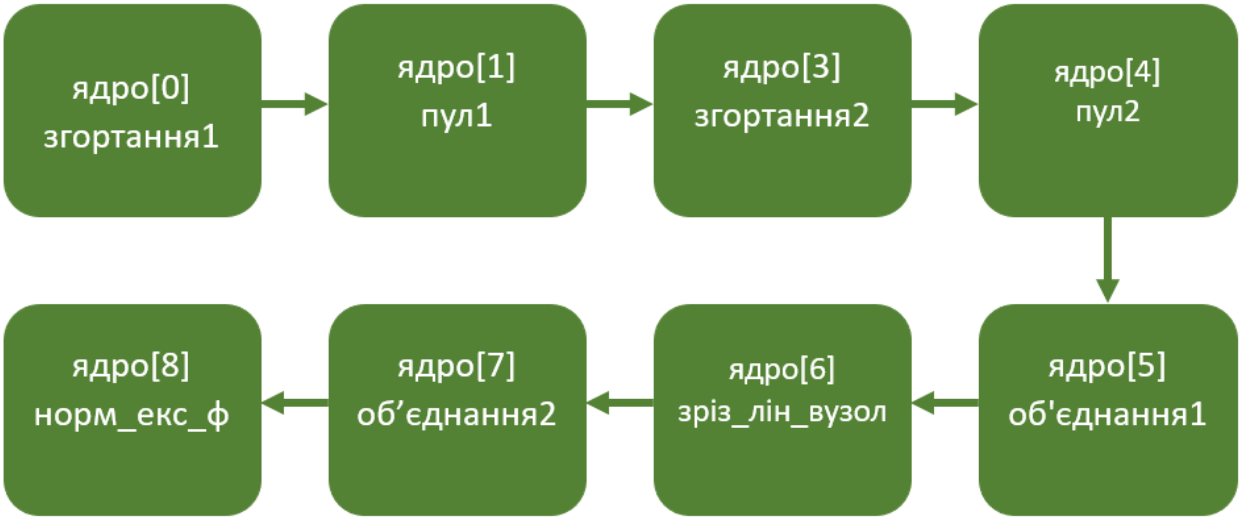


Рисунок 4.2 - Потік керування навчанням

Для прискорення процесу навчання ми використовували відеокарту NVIDIA GeForce 3040. GPU має 3840 ядер CUDA, які можна запрограмувати на виконання паралельних обчислень.

Його обчислювальна продуктивність може досягати 12 TFLOPS.

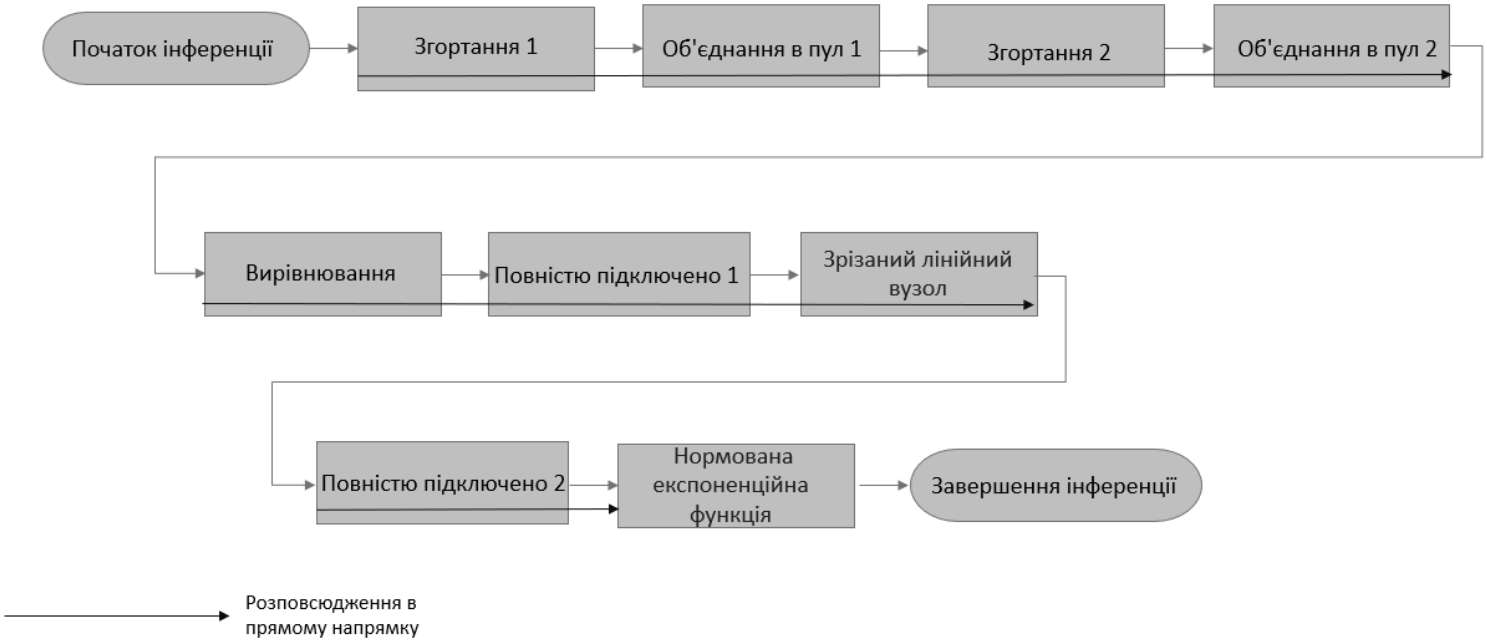


Рисунок 4.5 – Ядра реалізації OpenCL

Щоб розробити комп'ютерну розробку графічної карти NVIDIA, вона повинна використовувати мову програмування CUDA, згадану раніше.

Це означає, що вам слід створити ядро прискорення з нуля.

Завдяки TensorFlow, який містить бібліотеки CUDA, тепер потрібно зосередитися лише на розробці архітектури.

4.7.3 Результати експерименту

Мета експерименту – знайти найкращий пристрій для навчальної моделі, порівнюючи швидкість ЦП і ГП.

Для досягнення цієї мети ми розробили два варіанти використання.

Один має лише CPU-E5-1620, інший має CPU-E5-1620 і GPU-3040.

Було обрано однакові 45120 навчальних прикладів і відповідно розрахували час їх навчання.

Було проведено шість разів однакові тести та обчислили їхні середні значення.

Результати подано в таблиці 4.3.

Можна зробити висновок, що середня швидкість 3050 приблизно в 8,8 разів вище, ніж середня швидкість CPU E5-1620 з такою ж точністю.

Таблиця 4.3 – Час навчання

Експеримент	CPU-E5-1620	GPU		FPGA	
	Навчання час	Точність (%)	Навчання Час	Точність (%)	Прискорення (GPU/CPU)
1	448,60	98,70	50,61	98,80	8,86
2	447,79	98,88	51.17	98,58	8,75
3	448,35	98,90	50,73	98,80	8,84
4	448,62	98,94	50,46	98,88	8,89
5	447,62	98,59	50,94	98,82	8,79
6	447,88	98,94	50,78	98,79	8.82
Середній	448.10	98,80	50,80	98,80	8.80

4.7.4 Висновок у випадку використання

Потік керування фазами висновку має лише пряме поширення.

Він завантажує модель, згенеровану знавчальну частину, а потім робить висновок за допомогою алгоритму прямого поширення.

Ми використовували плату розробки Intel Altera Cyclone V FPGA для реалізації частини висновку. Altera Cyclone V — це найновіший продукт FPGA, виготовлений Intel за технологією 20 нм.

Має високу продуктивність і низьке енергоспоживання. Хоча його 1,5 TFLOPS все ще повільніше, ніж GeForce 3040 12 TFLOPS, його енергоспоживання нижче 100 Вт набагато краще, ніж GeForce 3040 250 Вт.

Було використано OpenCL для розробки FPGA.

Його розробка включає дві частини: хост-програму та ядра. Головна

програма працює на ЦП, а ядра – на ПЛІС.

Рисунок 4.5 показує ядра, реалізовані за допомогою OpenCL. Ядра поєднуються з апаратними логічними схемами окремо та виконуються одне за одним для реалізаціїфункція висновку.

У наступному експерименті було використано центральний процесор, графічний процесор і FPGA, щоб виконати ту саму роботу з висновків і порівняти їхню ефективність.

Щоб уникнути впливу різних смуг пропускання, ми розрахували лише час виведення одного зображення.

Було отримано 6 висновків на пристроях CPU, GPU та FPGA та обчислили середні значення часу відповідно.

Результати дослідів представлені в таблиці 4.4.

Таблиця 4.4 - Результати дослідів

Час експерименту	CPU-E5-1620	GPU-3040		FPGA		
	Висновок Час	Висновок Час	Прискорення (GPU/CPU)	Висновок Час	Прискорення (FPGA/CPU)	Прискорення (FPGA/GPU)
1	3172	616045	0,0051	88,74	35.7	6942,4
2	5564	589114	0,0094	90.12	61.7	6536,7
3	4620	588444	0,0079	94,83	48.7	6205,3
4	3234	598652	0,0054	84,57	38.2	7079,0
5	4037	600288	0,0067	101.08	39.9	5938,8
6	4579	609913	0,0075	108,74	42.1	5609,0
Середнє зеначення	4201	600409	0,0070	94,70	44.4	6341,5

З таблиці 4.4 видно, що середня швидкість Altera Cyclone V приблизно в 44,4 рази вища за середню швидкість CPU E5-1620 і приблизно в 6342 рази швидше, ніж GPU 3040.

Під час роботи Wang та ін. [18], FPGA становить 36,1x швидше ніж центральний процесор.

Ці результати доводять правильність нашого рішення використовувати FPGA для логічного висновку.

А найгіршу поведінку графічного процесора ми пояснюємо тим, що оскільки робоче навантаження (виведення одного зображення) замале порівняно з часом ініціалізації та затримкою графічного процесора, загальний час (час ініціалізації та час виведення) графічного процесора є найдовшим.

4.8 Результати експерименту

З метою перевірки ефективності методу було проведено два експерименти на тих самих прикладах тесту 10K MNIST відповідно.

В експерименті 1 було лише перевірено точність висновку FPGA за допомогою оригінальної моделі, тоді як у експерименті 2 було змінено конфігурацію CNN.

Також було переналаштовано модель за допомогою TensorFlow із графічним процесором, перенесли модель на FPGA.

Остаточно, було перевірено результати логічний висновку на FPGA і Tensorflow.

Таблиця 4.5 представляє статистику точності, зібрану з двох експериментів. Точність експерименту 2 краща, ніж експерименту 1.

В експерименті 2 FPGA зберегла таку ж точність, як і TensorFlow, який довів успішність роботи методу.

Таблиця 4.5 – Точність в експериментах із трансформацією моделі.

Експеримент 1	Точність (%)	Експеримент 2	Точність (%)
N/A	N/A	TensorFlow	98.29
FPGA	99.05	FPGA	98.44

4.9 Висновки

У розділі подано реалізацію методу синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, який на відміну від відомих застосовує апаратне рішення Hybrid Deep-GAN,

Апаратне рішення було здійснено засобами програмування CUDA та FPGA.

Результати експерименту показують, що швидкість навчання графічного процесора в середньому в 8,8 раза вища, ніж у центрального процесора, а швидкість виводу FPGA в середньому в 40,2 раза вища, ніж у центрального процесора, і в середньому в 62 рази швидше, ніж у графічного процесора.

Таким чином, було покращено точність виявлення вторгнень з 99,05% до 99,13% і зберегли точність 99,13% успішно, коли було перенесено модель з платформи GPU на платформу FPGA.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, що використовує апаратне рішення на основі GAN.

У першому розділі досліджено відомі методи синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень.

У другому розділі описано передумови побудови моделі функціонування системи керування IoT-інфраструктурою під час несанкціонованих вторгнень

Також в розділі описано основні аспекти функціонування генеративних змагальних мереж (GAN), можуть використовуватися для здійснення висновку щодо наявності вторгнення в IoT-інфраструктуру.

У третьому розділі удосконалено метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, який на відміну від відомих застосовує апаратне рішення Hybrid Deep-GAN, і який може ефективно виявляти вторгнення, і який забезпечує підвищення ефективності керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Запропонований метод застосовує Hybrid Deep-GAN і може ефективно виявляти вторгнення, з якими стикаються мережі IoT через динамічну природу мереж Інтернету речей.

Було доведено, що запропонований GAN займає важливе місце в області глибокого навчання, і основною проблемою в цій області є розробка системи виявлення вторгнень, яка забезпечує рішення для порушень безпеки.

Метод дозволяє виявлення невідомих атак. Через величезну кількість різнорідних даних, що генеруються з мережі, обов'язковою є попередня обробка, яка тут здійснюється за допомогою алгоритму TBSS, який ефективно заповнює відсутні дані.

Метод також застосовує MPCA, адаптований для виділення ознак, щоб виділити значущі ознаки, а також використовує алгоритм оптимізації Enhanced Whale для відбору найбільш придатних ознак. Для точної класифікації вторгнень на заданому наборі даних застосовано метод HDNN+ANN. Було виявлено, що метод EWO-HDNN+ANN забезпечує високу продуктивність щодо точності, достовірності, запам'ятовування, f-міри та пропонує нижчу частоту помилкових спрацьовувань, зменшує обчислювальну складність у порівнянні з іншими алгоритмами машинного навчання.

У четвертому розділі реалізовано метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, який на відміну від відомих застосовує апаратне рішення Hybrid Deep-GAN.

Апаратне рішення було здійснено засобами програмування CUDA та FPGA.

Результати експерименту показують, що швидкість навчання графічного процесора в середньому в 8,8 раза вища, ніж у центрального процесора, а швидкість виводу FPGA в середньому в 40,2 раза вища, ніж у центрального процесора, і в середньому в 62 рази швидше, ніж у графічного процесора.

Таким чином, було покращено точність виявлення вторгнень з 99,05% до 99,13% і точність 99,13%, коли було перенесено модель з платформи GPU на платформу FPGA.

За темою кваліфікаційної роботи магістра опублікована одна стаття у фаховому науковому виданні ВОТТП [1].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1 Лисенко С.М., Космина Ю.І., Бондарук О.В. Метод синтезу апаратнопрограмних засобів забезпечення стійкості корпоративної комп'ютерної мережі. Вісник Хмельницького національного університету. 2023.№3.

2 A technical overview of LoRa and LoRaWAN, 2015. White paper, LoRa Alliance.

3 Abduvaliyev, A., Pathan, A.S.K., Jianying, Z., Roman, R., Wai-Choong, W., On the vital areas of intrusion detection systems in wireless sensor networks. Commun. *Surv. Tutor. IEEE* 2013. 15 (3), 1223–1237.

4 Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M., Internet ofthings: a survey on enabling technologies, protocols, and applications. *IEEE Commun. Surv. Tutor.* 2015.17 (4), 2347–2376.

5 Amaral, J., Oliveira, L., Rodrigues, J., Han, G., Shu, L., Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks. *In:Communications (ICC), 2019 IEEE International Conference on*, 2014.pp. 1796–1801.

6 Anantvalee, T., Jie, W., A survey on intrusion detection systems in mobile ad hoc networks. *Wirel. Netw. Secur.* 2017.2, 159–180.

7 Ashraf, Q.M., Habaebi, M.H., Autonomic schemes for threat mitigation in internet of things. *J. Netw. Comput. Appl.* 2019. 49, 112–127.

8 Balci, O., Verification, validation, and accreditation, In*: Proceedings of the 30th Conference on Winter Simulation*, WSC '98, IEEE Computer Society Press, Los Alamitos, CA, USA, 1998. pp. 41–4.

9 Bandyopadhyay, D., Sen, J., Internet of things: applications and challenges in technology and standardization. *Wirel. Pers. Commun.* 2021.58 (1), 49–69.

10 Banks, A., Gupta, R., MQTT version 3.1.1, OASIS Standard. 2021.

11 Borgia, E., The Internet of Things vision: key features, applications and open issues. *Comput. Commun.* 2014. 54, 1–31.

12 Bradley, J., Barbier, J., Handler, D., Embracing the Internet of Everything to capture your share of $14.4 trillion, Tech. rep., Cisco White Paper. 2013.

13 Butun, I., Morgera, S., Sankar, R., A survey of intrusion detection systems in wireless sensor networks. *Commun. Surv. Tutor. IEEE* 2018.16 (1), 266–282.

14 Butun, I., Morgera, S.D., Sankar, R.,. A survey of intrusion detection systems in wireless sensor networks. *IEEE Commun. Surv. Tutor.* 2018b 16 (1), 266–282.

15 Cervantes, C., Poplade, D., Nogueira, M., Santos, A., Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things. *In: 2015 IFIP/ IEEE International Symposium on Integrated Network Management (IM)*, 2015. pp. 606–611.

16 Cho, E., Kim, J., Hong, C., Attack model and detection scheme for botnet on 6LoWPAN. In: Hong, C., Tonouchi, T., Ma, Y., Chao, C.-S. (Eds.), Management Enabling the Future Internet for Changing Business and New Computing Services, L*ecture Notes in Computer Science* 2019. 5787. Springer, Berlin, Heidelberg, 515–518.

17 Elejoste, P., Angulo, I., Perallos, A., Chertudi, A., Zuazola, I.J.G., Moreno, A., Azpilicueta, L., Astrain, J.J., Falcone, F., Villadangos, J., An easy to deploy street light control system based on wireless communication and LED technology. *Sensors* 2018. 13 (5), 6492–6523.

18 ETSI, T., 102 690, machine-to-machine communications (M2M); functional architecture., European Telecommunications Standards Institute (ETSI) 2016.20, 332.

19 Farooqi, A., Khan, F., Intrusion detection systems for wireless sensor networks: a survey. *In: Šlęzak, D., Kim, T.-H., Chang, A.-C., Vasilakos, T., Li, M., Sakurai, K. (Eds.), Communication and Networking, Communications in Computer and Information Science* 2019. 56. Springer, Berlin, Heidelberg, 234–241.

20 Filho, H.G.S., Filho, J.P., Moreli, V.L., The adequacy of LoRaWAN on smart grids: a comparison with RF mesh technology, *In: 2016 IEEE International Smart Cities Conference (ISC2)*, 2016. pp. 1–6.

21 Garcia-Morchon, O., Kumar, S., Struik, R., Keoh, S., Hummen, R. Security considerations in the IP-based Internet of Things, *IETF Internet-Draft.* , 2019.

22 Gluhak, A., Krco, S., Nati, M., Pfisterer, D., Mitton, N., Razafindralambo, T. A survey on facilities for experimental Internet of Things research. *IEEE Commun. Mag.* , 2016.49 (11), 58–67.

23 Gomez, C., Paradells, J., Wireless home automation networks: a survey of architectures and technologies. *IEEE Commun. Mag.* 2019. 48 (6), 92–101.

24 Gomez, C., Oller, J., Paradells, J., Overview and evaluation of Bluetooth Low Energy: an emerging low-power wireless technology. *Sensors* 2012. 12 (9), 11734.

25 Granjal, J., Monteiro, E., Silva, J.S., On the effectiveness of end-to-end security for Internet-integrated sensing applications. *In: Green Computing and Communications (GreenCom), 2012 IEEE International Conference on,* 2017. pp. 87–93.

26 Gupta, A., Pandey, O., Shukla, M., Dadhich, A., Mathur, S., Ingle, A., Computational intelligence based intrusion detection systems for wireless communication and pervasive computing networks, *In: Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on,* 2013. pp. 1–7.

27 Han, C., Jornet, J.M., Fadel, E., Akyildiz, I.F., A cross-layer communication module for the Internet of Things. *Comput. Netw.* 2013. 57 (3), 622–633.

28 Hubballi, N., Suryanarayanan, V., False, alarm minimization techniques in signature-based intrusion detection systems: a survey. *Comput. Commun.* 2018.49, 1–17.

29 Isa, M.A.M., Mohamed, N.N., Hashim, H., Adnan, S.F.S., Manan, J.A., Mahmod, R., A lightweight and secure TFTP protocol for smart environment. In: Computer Applications and Industrial Electronics (ISCAIE), *2012 IEEE Symposium,* 2012. pp. 302–306.

30 Khan, R., Khan, S.U., Zaheer, R., Khan, S., Future Internet: the Internet of Things architecture, possible applications and key challenges., *In: FIT*, 2017.pp. 257–260.

31 Kim, A.N., Hekland, F., Petersen, S., Doyle, P., When HART goes wireless: understanding and implementing the WirelessHART standard, *IEEE International Conference on Emerging Technologies and Factory Automation,* 2018. pp. 899–907.

32 Kolias, C., Stavrou, A., Voas, J., Bojanova, I., Kuhn, R. Learning Internet-of-thingssecurity “Hands-on”. *IEEE Secur.* Priv. 20 (February), 2–11. http://dx.doi.org/10.1109/MSP.2016.4, , 2016.

33 Krimmling, J., Peter, S., Integration and evaluation of intrusion detection for CoAP in smart city applications. *Communications and Network Security (CNS)*, 2014 IEEE Conference on, 2014. pp. 73–78.

34 Kumar, S., Dutta, K. Intrusion detection in mobile ad hoc networks: techniques,systems, and future challenges. *Secur. Commun. Netw.* , 2016.9 (14), 2484–2556.

35 Le, A., Loo, J., Chai, K.K., Aiash, M. A specification-based IDS for detecting attacks on RPL-based network topology. *Information*, 2016. 7 (2), 25.

36 Le, A., Loo, J., Luo, Y., Lasebae, A., Specification-based IDS for securing RPL from topology attacks, *In: Wireless Days (WD)*, 2011. 2011 IFIP, pp. 1–3.

37 Lee, I., Lee, K., The internet of things (IoT): applications, investments, and challenges for enterprises. *Bus. Horiz.* 2015. 58 (4), 431–440.

38 Lee, T.-H., Wen, C.-H., Chang, L.-H., Chiang, H.-S., Hsieh, M.-C., A lightweight intrusion detection scheme based on energy consumption analysis in 6LowPAN. In: Huang, Y.-M., Chao, H.-C., Deng, D.-J., Park, J.J.J.H. (Eds.), Advanced Technologies, Embedded and Multimedia for Human-centric Computing, *LectureNotes in Electrical Engineering* 260. Springer, Netherlands, 2014. 1205–1213.

39 Liao, H.-J., Lin, C.-H.R., Lin, Y.-C., Tung, K.-Y., Intrusion detection system: a comprehensive review. *J. Netw. Comput. Appl.* 2013. 36 (1), 16–24.

40 Liu, C., Yang, J., Zhang, Y., Chen, R., Zeng, J., Research on immunity-based intrusion detection technology for the Internet of Things. *Natural Computation(ICNC), 2011 Proceedings of the Seventh International Conference*, 2011. Vol. 1, pp. 212–216.

41 Mahoney, M.V., Chan, P.K. An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection, 2003.220–237.

42 McHugh, J., Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln

Laboratory. *ACM Trans. Inf. Syst. Secur*. 2000. 3 (4), 262–294.

43 Meddeb, A., Internet of things standards: who stands out from the crowd? *IEEE Commun. Mag*. 2016. 54 (7), 40–47.

44 Milenkoski, A., Vieira, M., Kounev, S., Avritzer, A., Payne, B.D., Evaluatingcomputer intrusion detection systems: a survey of common practices. *ACM Comput. Surv*. 2015. 48 (1), 1–41.

45 Miorandi, D., Sicari, S., De Pellegrini, F., Chlamtac, I., Internet of things: vision, applications and research challenges. *Ad Hoc Netw*. 2012.10 (7), 1497–1516.

46 Mishra, A., Nadkarni, K., Patcha, A., Intrusion detection in wireless ad hoc networks. *IEEE Wirel. Commun*. 2004. 11 (1), 48–60.

47 Misra, S., Krishna, P., Agarwal, H., Saxena, A., Obaidat, M., 2011. A learning automata based solution for preventing Distributed Denial of Service in Internet of Things. Internet of Things (iThings/CPSCom), *2011 International Conference on and Proceedings of the 4th International Conference on Cyber, Physical and Social Computing*, pp. 114–122.

48 Mitchell, R., Chen, I.-R., A survey of intrusion detection techniques for cyberphysical systems. *ACM Comput. Surv.* (CSUR) 2014.46 (4), 55.

49 Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., Rajarajan, M.,. A survey of intrusion detection techniques *in Cloud. J. Netw. Comput. Appl*. 2013. 36 (1), 42–57.

50 Nehinbe, J.O., 2011. A critical evaluation of datasets for investigating IDSs and IPSs researches. *2011, Proceedings of the 10th IEEE International Conference on Cybernetic Intelligent Systems*, CIS 2011, pp. 92–97.

51 Notra, S., Siddiqi, M., Gharakheili, H., Sivaraman, V., Boreli, R., 2014. An experimental study of security and privacy risks with emerging household appliances. *Communications and Network Security (CNS)*, 2014 IEEE Conference, pp. 79–84.

52 R. Abdulhammed, M. Faezipour, and K. M. Elleithy, “Network intrusion detection using hardware techniques: A review,” in *Proc. IEEE long Island Systems, Applications and Technology Conference*, 2016.

53 B. Subba, S. Biswas, and S. Karmakar, “A neural network based system for intrusion detection and attack classification,” in *Proc. National Conference on Communication,* 2016.

54 A. Gharib, I. Sharafaldin, A. II. Lashkari, and A. A. Ghorbani, “An evaluation framework for intrusion detection dataset,” in *Proc. International Conference on Information Science and Security,* 2016.

55 S. Yusuf, W. Luk, M. K. N. Szeto, and W. Osborne, “Unite: Uniform hardware-based network intrusion detection engine,” in *Reconfigurable Computing: Architectures and Applications,* 2006, pp. 389-400.

56 R. Ptoudfoot, K. Kent, E. Aubanel, and N. Chen, “Flexible software-hardware network intrusion detection system,” in *Proc. International Symposium on Rapid System Prototyping,* 2008, pp. 182-188.

57 A. Das, D. Nguyen, J. Zambreno, G. Memik, and A. Choudhary, “An FPGA-bascd network intrusion detection architecture,” *IEEE Transactions on Information Forensics and Security,* vol. 3, no. 1, pp. 118-132, 2008.

58 A. L. P. de Franca, R. P. Jasinski, V. A. Pcdroni, and A. O. Santin, “Moving network protection from software to hardware: An energy efficiency analysis,” in *Proc. IEEE Computer Society Symposium on VLSI,* July 2014, pp. 456-461.

59 A. L. P. de Franca, R. P. Jasinski, P. Cemin, V. A. Pedroni, and A. O. Santin, “The energy cost of network security: A hardware vs. software comparison,” in *Proc. International Symposium on Circuits and Systems (ISCAS),* 2015, pp. 81-84.

60 S. Shreejith and S. A. Fahmy, “Security aware network controllers for next generation automotive embedded systems,” in *Proc. Design Automation Conference (DAC),* 2015.

61 E. Hodo, X. Bellekens, A. Hamilton, P. L. Dubouilh, E. lorkyase, C. Tachtatzis, and R. Atkinson, “Threat analysis of iot networks using artificial neural network intrusion detection system,” in *Proc. International Symposium on Networks, Computers and Communications,* 2016.

62 M. Idhammad, K. Afdel, and M. Belouch, “DoS detection method based on

artificial neural networks," *International Journal of Advanced Computer Science and Applications,* vol. 8, no. 4, 2017.

63 B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in *Proc. International Conference on Signal Processing and Communication Engineering Systems,* 2015, pp. 92-96.

64 K. Kim, M. E. Aminanto, and H. C. Tanuwidjaja, *Network Intrusion Detection using Deep Learning: A Feature Learning Approach.* Springer Singapore, 2018, pp. 35-45.

65 C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access,* vol. 5, pp. 21954-21961, 2017.

66 Z. Li, Z. Qin, K. Huang, X. Yang, and S. Ye, "Intrusion detection using convolutional neural networks for representation learning," in *Proc. International Conference Neural Information Processing,* 2017, pp. 858-866.

67 T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in *Proc. International Conference on Wireless Networks and Mobile Communications,* 2016, pp. 258-263.

68 T. Tang, S. Zaidi, D. McLernon, L. Mhamdi, and M. Ghogho, "Deep recurrent neural network for intrusion detection in SDN-based networks," in *Proc. International Conference on Network Softwarization,* 2018.

69 S. Shreejith, B. Anshuman, and S. A. Fahmy, "Accelerated artificial neural networks on FPGA for fault detection in automotive systems," in *Proc. Design, Automation Test in Europe Conference Exhibition (DATE),* 2016, pp. 37^2.

70 O. Karan, C. Bayraktar, H. Giimiikaya, and B. Karlk, "Diagnosing diabetes using neural networks on small mobile devices," *Expert Systems with Applications,* vol. 39, no. 1, pp. 54-60, 2012.

71 M. Tavallacc, E. Baghcri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE International Conference on Computational Intelligence for Security and Defense Applications,* 2009, pp. 53-58.

72 M. Abadi *et al.,* “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. [Online]. Available: hltps://www. tcnsorflow.org/

73 K. Bajaj and A. Arora, “Improving the intrusion detection using discriminative machine learning approach and improve the time complexity by data mining feature selection methods,” *International Journal of Computer Applications,* vol. 76, no. 1, pp. 5-11, August 2013.

74 S. I. Venieris, A. Kouris, and C.-S. Bouganis, “Toolflows for mapping convolutional neural networks on FPGAs: A survey and future directions,” *ACM Computing Surveys,* vol. 51, no. 3, pp. 56:1-56:39, 2018.

75 V. Sze, Y. Chen, T. Yang, and J. S. Emer, “Efficient processing of deep neural networks: A tutorial and survey,” *Proceedings of the IEEE,* vol. 105, no. 12, pp. 2295-2329, Dec 2017.

76 K. Guo, S. Zeng, J. Yu, Y. Wang, and H. Yang, “[DL] A survey of FPGA-bascd ncural network inference accelerators,” *ACM Trans. Reconfigurable Technol. Sy st.,* vol. 12, no. 1, pp. 2:1-2:26, Mar. 2019.

77 K. Vipin and S. A. Fahmy, “FPGA dynamic and partial reconfiguration: A survey of architectures, methods, and applications,” *ACM Computing Surveys,* vol. 51, no. 4, pp. 72:1-72:39, Jul. 2018.

78 K. Vipin and S. A. Fahmy, “ZyCAP: Efficient partial reconfiguration management on the Xilinx Zynq,” *IEEE Embedded Systems Letters,* vol. 6, no. 3, pp. 41^44, Sep. 2014.

79 J. Postel, “internet protocol,” *RFC,* vol. 791, pp. 1-51, 1981.

80 S. Shreejith, R. A. Cooke, and S. A. Fahmy, “A smart network interface approach for distributed applications on Xilinx Zynq SoCs,” in *Proc. Field Programmable Logic and Applications (FPL),* 2018, pp. 186-190.

81 Wellem, T., Lai, Y.K., Huang, C.Y., Chung, W.Y.: A hardware-accelerated infrastructure for flexible sketch-based network traffic monitoring. *In: IEEE 17th HPSR.* IEEE. 2016.

ДОДАТОК А
(обов'язковий)
КОПІЯ ПУБЛІКАЦІЇ

УДК 004.93
DOI:

С.М. ЛИСЕНКО, Ю.І.КОСМИНА, О.В. БОНДАРУК
Хмельницький національний університет

МЕТОД СИНТЕЗУ АПАРАТНО-ПРОГРАМНИХ ЗАСОБІВ КЕРУВАННЯ IOT-ІНФРАСТРУКТУРОЮ ПІД ЧАС НЕСАНКЦІОНОВАНИХ ВТОРГНЕНЬ

У цій роботі розглянуто методологію керування Інтернету Речей (IoT) інфраструктурою під час несанкціонованих вторгнень є складним завданням, яке вимагає поєднання технічних та організаційних заходів. Удосконалено метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, який на відміну від відомих застосовує апаратне рішення Hybrid Deep-GAN, і який може ефективно виявляти вторгнення, і який забезпечує підвищення ефективності керування IoT-інфраструктурою під час несанкціонованих вторгнень

Ключові слова: метод, синтез апаратно-програмних засобів, керування IoT-інфраструктурою, FPGA, gpu, cpu, несанкціоновані вторгнення

S. LYSENKO, Y.I. KOSMINA, O.BONDARUK
Khmelnytskyi National University, Khmelnytskyi, Ukraine

METHOD OF SYNTHESIS OF HARDWARE AND SOFTWARE TOOLS FOR MANAGEMENT OF IOT INFRASTRUCTURE DURING UNAUTHORIZED INTRUSIONS

In this paper, based on the results of theoretical and practical studies, a method for synthesizing hardware and software tools for managing IoT infrastructure during unauthorized intrusions using a hardware solution based on GAN is developed. The paper investigates the known methods of synthesizing hardware and software tools for managing IoT infrastructure during unauthorized intrusions. Section 2 describes the prerequisites for building a model of the functioning of the IoT infrastructure management system during unauthorized intrusions. The section also describes the main aspects of the functioning of generative adversarial networks (GANs), which can be used to conclude whether an intrusion into the IoT infrastructure has occurred. The study improves the method of synthesizing hardware and software tools for managing IoT infrastructure during unauthorized intrusions, which, unlike the known ones, uses the GAN hardware solution, and which can effectively detect intrusions, and which provides an increase in the efficiency of managing IoT infrastructure during unauthorized intrusions. The proposed method applies GAN and can effectively detect intrusions faced by IoT networks due to the dynamic nature of IoT networks. It has been proven that the proposed GAN has an important place in the field of deep learning, and the main challenge in this area is to develop an intrusion detection system that provides solutions to security breaches.

The method allows detection of unknown attacks. Due to the huge amount of heterogeneous data generated from the network, pre-processing is mandatory, which is done here using the algorithm, which effectively fills in the missing data. The method also applies MPCA adapted for feature extraction to isolate meaningful features, and uses the Enhanced Whale optimization algorithm to select the most suitable features. To accurately classify intrusions on a given dataset, the method was applied. It was found that the method provides high performance in terms of accuracy, precision, recall, f-measure and offers a lower false positive rate, reduces computational complexity compared to other machine learning algorithms. The paper also presents the implementation of a method for synthesizing hardware and software tools for managing IoT infrastructure during unauthorized intrusions, which, unlike the known ones, uses the GAN hardware solution. The hardware solution was implemented using CUDA and FPGA programming tools. Experimental results show that the learning speed of the GPU is on average 8.8 times faster than that of the CPU, and the output speed of the FPGA is on average 40.2 times faster than that of the CPU, and on average 62 times

faster than that of the GPU. Thus, the intrusion detection accuracy was improved from 99.05% to 99.13% and the accuracy was 99.13% when the model was transferred from the GPU platform to the FPGA platform.

Keywords: method, hardware-software synthesis, IoT infrastructure management, FPGA, GPU, CPU, unauthorized intrusions

1 Вступ

Керування Інтернету Речей (IoT) інфраструктурою під час несанкціонованих вторгнень є складним завданням, яке вимагає поєднання технічних та організаційних заходів. Основні технічні заходи для забезпечення безпеки IoT-інфраструктури включають такі аспекти. Криптографічний захист даних: IoT-пристрої повинні використовувати криптографічні методи для захисту даних, переданих між пристроями та збережених на серверах. Безпека мережі: IoT-мережа повинна бути захищена від несанкціонованого доступу за допомогою засобів, таких як брандмауери, ідентифікація та аутентифікація користувачів, контроль доступу тощо. Оновлення програмного забезпечення: IoT-пристрої повинні мати можливість оновлювати своє програмне забезпечення для усунення виявлених уразливостей та покращення безпеки. Моніторинг та виявлення інцидентів: система повинна мати можливість моніторити та виявляти потенційні загрози безпеці, такі як несподіване збільшення трафіку, відправка підозрілих повідомлень тощо.

Машинне навчання є потужним інструментом для виявлення несанкціонованих вторгнень у системи. Це може бути досягнуто за допомогою навчання алгоритмів класифікації на базі зразків (supervised learning) та без навчання на базі зразків (unsupervised learning). Одним з найпоширеніших методів машинного навчання для виявлення вторгнень є аналіз аномалій, де методі алгоритми навчаються розрізняти нормальний і аномальний трафік на основі вхідних даних, таких як логи веб-сервера або мережеві дані. Під час виконання роботи, система порівнює вхідні дані з навчальними даними і повертає повідомлення про вторгнення, якщо вхідні дані відрізняються від нормального шаблону. Інший підхід - це використання навчання з учителем, де алгоритми навчаються на основі попередньо класифікованих прикладів вторгнень та їх характеристик. Ці алгоритми можуть виявляти нові види вторгнень, які не були раніше відомі. Машинне навчання також може бути використане для аналізу журналів системного адміністрування та моніторингу мережі для виявлення незвичайної поведінки та інших показників, що можуть вказувати на потенційне вторгнення. Однак, для виявлення несанкціонованих вторгнень, може знадобитися великий обсяг даних, що може бути важко отримати. Також, враховуючи те, що хакери можуть використовувати нові технології та алгоритми, системи виявлення вторгнень потребують постійного оновлення та розширення, щоб залишатися ефективними.

FPGA (Field Programmable Gate Array) - це програмований кристал, що дозволяє створювати власні логічні схеми та призначення. FPGA може бути використаний для реалізації засобів машинного навчання.

Апаратна реалізація засобів машинного навчання з FPGA має кілька переваг порівняно з програмним виконанням. FPGA може забезпечувати значно вищу швидкість та низьку затримку, оскільки він працює в реальному часі та не потребує перетворення програмного коду на машинний код. FPGA також може забезпечувати значно більшу обчислювальну потужність, оскільки це програмований кристал з великою кількістю логічних елементів, які можна згрупувати для реалізації великих обчислювальних блоків.

Апаратна реалізація засобів машинного навчання з FPGA може бути використана для швидкої обробки даних в режимі реального часу. Наприклад, вона може бути використана для обробки даних з сенсорів, таких як камери, для виявлення облич та інших об'єктів. FPGA також може бути використаний для реалізації нейронних мереж та інших засобів машинного навчання, що вимагають значної обчислювальної потужності.Метою дослідження є підвищення ефективності керування IoT-інфраструктурою під час несанкціонованих вторгнень.

2 Відомі методи побудови синтезу апаратно-програмних засобів керування іот-інфраструктурою під час несанкціонованих вторгнень

Розвиток різних технологічних галузей, таких як датчики, автоматична ідентифікація та відстеження, вбудовані обчислення, бездротовий зв’язок, широкосмуговий доступ до Інтернету та розподілені послуги, збільшив потенціал інтеграції розумних об’єктів у нашу щоденну діяльність через Інтернет. Конвергенція

Інтернету та розумних об’єктів, які можуть спілкуватися та взаємодіяти один з одним, визначає Інтернет речей (IoT). Ця нова парадигма визнана одним із найважливіших гравців у галузі інформаційно-комунікаційних технологій (ІКТ) на наступні роки [1]. За даними Gartner Inc., IoT може налічувати 26 мільярдів одиниць до 2020 року. Cisco Systems передбачила, що IoT створить $14,4 трильйона в результаті поєднання збільшення доходів і зниження витрат для компаній з 2013 по 2022 рік [2-5].

Багато областей застосування, такі як логістика, промислові процеси, громадська безпека, домашня автоматизація, моніторинг навколишнього середовища та охорона здоров’я, можуть мати значні переваги з системами IoT [6]. Однак інтеграція об’єктів реального світу з Інтернетом створює загрози кібербезпеці для більшості наших щоденних дій. Атаки на критичні об'єкти інфраструктури, такі як електростанції та транспортні системи, можуть мати жахливі наслідки для цілих міст і країн.

Побутова техніка також може бути основною мішенню, загрожуючи безпеці та приватності сімей [7], тести, проведені з трьома популярними пристроями розумного дому, показали різні вразливості, пов’язані з конфіденційністю користувачів, відсутністю шифрування та автентифікації. Через різні стандарти та задіяні комунікаційні стеки, обмежену обчислювальну потужність і велику кількість взаємопов’язаних пристроїв традиційні контрзаходи безпеки не можуть ефективно працювати в системах IoT. З цієї причини розробка спеціальних рішень безпеки для IoT має важливе значення, щоб дозволити користувачам і організаціям використовувати всі можливості, які він пропонує [8].

Деякі поточні проекти з підвищення безпеки IoT включають методи забезпечення конфіденційності та автентифікації даних, контролю доступу в мережі IoT, конфіденційності та довіри між користувачами та речами, а також застосування політики безпеки та конфіденційності [9]. Однак навіть із цими механізмами мережі IoT вразливі до численних атак, спрямованих на порушення роботи мережі. З цієї причини необхідна інша лінія захисту, призначена для виявлення нападників. Системи виявлення вторгнень (IDS) виконують цю мету. IDS є одним із основних засобів захисту традиційних мереж та інформаційних систем. IDS контролює роботу хоста або мережі, сповіщаючи системного адміністратора, коли він виявляє порушення безпеки. Таким чином, IDS зміцнила свою позицію як популярна захисна технологія для традиційних IP-мереж, маючи на ринку кілька рішень. Незважаючи на зрілість технології IDS для традиційних мереж, поточні рішення є неадекватними для систем IoT через особливі характеристики IoT, які впливають на розвиток IDS. По-перше, важливою проблемою є обробка та сховище мережевих вузлів, на яких розміщені агенти IDS. У традиційних мережах системний адміністратор розгортає агенти IDS у вузлах з більшою обчислювальною потужністю. Мережі IoT зазвичай складаються з вузлів з обмеженими ресурсами. Тому в системах IoT важче знайти вузли з можливістю підтримки агентів IDS. Друга конкретна характеристика пов'язана з архітектурою мережі. У традиційних мережах кінцеві системи безпосередньо підключені до певних вузлів (наприклад, бездротових точок доступу, комутаторів і маршрутизаторів), які відповідають за пересилання пакетів до пункту призначення. мережі IoT, з іншого боку, зазвичай є багатострибковими. Тоді звичайні вузли можуть одночасно пересилати пакети та працювати як кінцеві системи. Наприклад, у системах вуличного освітлення на основі IoT датчики з можливістю зв’язку малого радіусу дії розгортаються на опорах освітлення [10-12]. Потім дані, зібрані датчиком, пересилаються через датчики, розташовані на різних опорах освітлення, до досягнення шлюзу в Інтернеті. Така архітектура ставить перед IDS нові виклики. Остання характеристика пов’язана з конкретними мережевими протоколами. Мережі IoT використовують протоколи, які не використовуються в традиційних мережах, наприклад IEEE 802.15.4, IPv6 через бездротову персональну мережу з низьким енергоспоживанням (6LoWPAN), протокол маршрутизації IPv6 для мереж з низьким енергоспоживанням і мережами з втратами (RPL) і протокол обмежених додатків (CoAP). Різні протоколи привносять початкові вразливості та нові вимоги до IDS.

3 Основи методу синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень

У дослідженні запропонована метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, який на відміну від відомих застосовує апаратне рішення Hybrid Deep-GAN, і який може ефективно виявляти вторгнення, з якими стикаються мережі IoT через

динамічну природу мереж Інтернету речей. Запропонований GAN займає важливе місце в області глибокого навчання, і основною проблемою в цій області є розробка системи виявлення вторгнень, яка забезпечує рішення для порушень безпеки. Таким чином, було побудовано модель Deep-GAN для виявлення всіх невідомих атак. Через велику кількість різнорідних даних, що генеруються з мережі, обов'язковою є попередня обробка, яка тут здійснюється за допомогою алгоритму TBSS, який ефективно заповнює відсутні дані, а потім метод MPCA, адаптований для виділення ознак, щоб виділити значущі ознаки. А також алгоритм оптимізації Enhanced Whale орз метою відбору найбільш придатних ознак. Метод HDNN+ANN застосовано для точної класифікації вторгнень на заданому наборі даних. В кінцевому результаті зазначено, що метод забезпечує високу продуктивність щодо точності, достовірності, запам'ятовування, f-міри та пропонує нижчу частоту помилкових спрацьовувань, зменшує обчислювальну складність у порівнянні з іншими алгоритмами машинного навчання. У цьому дослідженні розроблення методу синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень пропонується застосування GAN для виявлення вторгнень через Інтернет речей. Метод включає в себе побудову моделі GAN, попередню обробку даних, виділення та вибір ознак, класифікацію та оцінку результатів. Загальний дизайн представленого методу зображено на рис. 1.

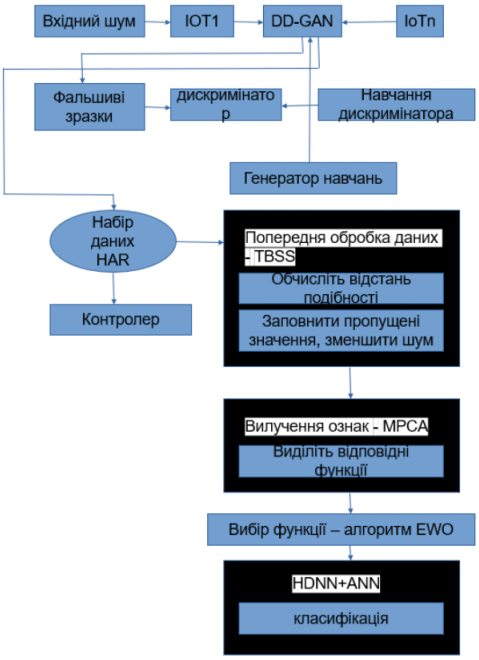


Рисунок 1 - Гібридне розподілене виявлення вторгнень на основі глибокого навчання в інфраструктурі Інтернету речей

P-GAN побудовано для виявлення вторгнень з використанням Інтернету речей. GAN є дуже впливовим і досвідченим методом глибокого навчання. Тут GAN апроксимує генеративну модель, використовуючи змагальний процес. Зазвичай GAN має дві автономні моделі, такі як B - генератор і P - дискримінатор. Генеративна структура оцінює розподіл даних *r(l)* на основі фактичного простору даних a. Метою B є створення нової змагальної вибірки B(s), яка досягається з подібного розподілу a. Тут дискримінатор моделі P є наслідком ймовірності *P(x),* стосовно наведеного прикладу x - це реальний набір даних, сформований B. Основна мета B - підвищити ймовірність того, що P помилково сприйме створені дані як реальні, а мета P - виконати протилежну задачу. Отже, P і B будуть досягати *minmax* еквіваленту і врешті-решт досягнуть елітного результату. Функція цінності *W (B;P)* описується таким чином:

$$\min_B \max_P W(B,P) = E_{a\sim r_{data}}(a)[log\ \ P(a)] + E_{s\sim r_s(s)}[\log(1 - P(B(s)))]. \qquad (1)$$

Надати перевагу Інтернету речей, який містить множину M з j Інтернету речей, де кожен захисник Інтернету речей і має набір попередньо переданих точок даних P_i, які є результатом розподілу r_{data} *i (a),* де *a*може бути часовим рядом, військовими даними, економічними звітами або базою даних спостережень за станом здоров'я. Вважається, що P_i складається з точок даних зі справжнього стану захисника Інтернету речей, коли в Інтернеті речей немає атак. Також нехай $P_1 \cup P_2 \cup \ldots \cup P_j = P$, де P представляє всі доступні дані з розподілом r_{data}. Тут кожен захисник Інтернету речей і намагається дізнатися розподіл генератора rl_i в доступній йому базі даних P_i так, щоб $rl_i = r_{data}$, і використовує цей розподіл для виявлення атаки в системі.

Вторгнення в системі - це будь-яка дія зловмисника, яка спрямована на те, щоб захисник Інтернету речей не наближався до відповідних точок даних з метою не допустити подальшого розподілу даних r_{pi}. Насправді, коли захисник Інтернету речей здійснює розподіл власного фактичного стану, він без особливих зусиль не надає перевагу точці даних, яка відрізняється від нормального розподілу станів. Загалом, штучні нейронні мережі складаються зі штучних нейронів та функції активації, які відображають вхідні дані на вихід.

Для кожного захисника Інтернету речей i визначається додаткова штучна нейронна мережа, яка називається дискримінатор $P_i(x,\theta_{pi})$, який отримує точку даних а і виробляє значення між 0 і 1. Якщо результат дискримінатора ближче до 1, отримана точка даних знаходиться в нормальному стані, в іншому випадку це вторгнення на захисника Інтернету речей i. У той час як всі генератори захисників Інтернету речей мають намір зменшити функцію цінності, згадану в (1), дискримінатор намагається збільшити її значення. Отже, життєздатний результат для дискримінатора і генератора може бути досягнутий з наступної задачі на мінімакс:

$$\{P_i^*, B_i^*\} = argmin_{B_i} \arg\max P_i W_i(P_i, B_i) \quad (2)$$

Тут, розподілена система виявлення вторгнень на основі GAN розвиває структурний дизайн методів класифікації об’єктів. Зусилля розподіленої GAN полягає у встановленні дискримінатора на кожному захиснику інфраструктури Інтернету речей далі, ніж розподіл баз даних між собою. Це робиться для того, щоб кожен дискримінатор захисника інфраструктури Інтернету речей міг розрізняти, коли нова точка даних розширює загальний розподіл даних, r_{data}.

У P-GAN на етапі навчання він забезпечує роботу центрального блоку, який включає генератор B_φ, де φ - це вага генератора штучних нейронних мереж. Крім того, кожен захисник Інтернету речей містить дискримінатор, визначений як $P_{\theta i}$, де θ_i - вага кожного дискримінатора створеної штучної нейронної мережі. У моделі кожен Захисник Інтернету речей пов'язаний щонайменше з одним Захисником Інтернету речей у мережі Інтернету речей, так що граф зв'язків Захисника Інтернету речей повинен описувати цикл. Крім того, на етапі навчання всі захисники Інтернету речей підключені до центрального блоку. GAN компетентний у класифікації наслідків на основі навченої вибірки, але через неконтрольоване навчання та нестабільне навчання його стає дуже складно навчати та генерувати вихідні дані.

Попередня обробка даних з використанням перенесення за подібністю підпросторів

Попередня обробка даних у запропонованому методі синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень з використанням перенесення за подібністю підпросторів (Transfer by Subspace Similarity , TBSS).

Алгоритм поліпшеного перенесення за подібністю підпросторів адаптовано для попередньої обробки даних, що значно підвищує точність виявлення вторгнень для ексклюзивної бази даних розпізнавання людської активності (Human Activity Recognition, HAR). Перенос за подібністю підпростору гнучкий для виявлення активності в реальному часі. Основною метою є розробка алгоритму для роботи з цими послідовними даними з метою покращення якості інтелектуального аналізу даних. Це здійснюється шляхом

включення відсутніх даних, точного налаштування шуму та значної модифікації волатильності для даної бази даних. Для цього необхідно передати дані:

$$P \text{ до } P\hat{}, \hat{}p = (a\ n\ e, cm), \hat{}p \in P\hat{}. \quad (3)$$

Звідси покращене перенесення за подібністю підпростору - це досвідчена стратегія кластеризації. Вона працює, щоб розділити ідентичні дані в наборі даних на певні групи відповідно до різних міток. Це створює колекцію τ на основі різних класів:

$$\tau = \{T_0^{Cx}\}_{x=1}^{m} \quad (4)$$

Методи вибірки випадковим чином відбирають дані, і ми можемо побудувати піднабори даних.

Це здійснює передумови для визначення важливої інформацію між ними як нові ознаки, використовуючи функцію Transfer by Subspace Similarity (перенесення за схожістю підпростору) на рис. 2.

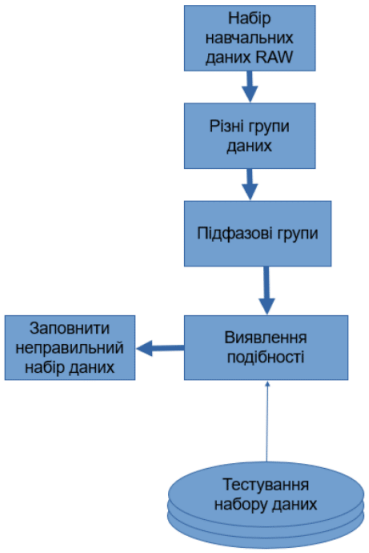


Рисунок 2 - Схема покращеного алгоритму перенесення за подібністю підпросторів

Перевага перенесення за подібністю підпросторів полягає в обчисленні ймовірності між кожною схожістю підпросторів та простоті реалізації. Наступним кроком методу є необхідність визначити витрати часу о на основі часу, витраченого на побудову підпростору, та часу, витраченого на розподіл вихідного простору. Якщо вихідний простір великий, то час дискретизації може бути більшим, оскільки необхідно забезпечити рівень покриття для вихідного простору. Якщо ж вихідний простір менший, то час вибірки та витрати часу будуть меншими. Таким чином, метод попередньої обробки використовується для підвищення точності виявлення вторгнень за допомогою алгоритму перенесення за подібністю підпростору.

Виділення ознак за допомогою модифікованого аналізу головних компонент

Алгоритм модифікованого аналізу головних компонент спроектовано для вилучення ознак, в якому основна увага приділяється зменшенню кількості ознак. Аналіз головних компонент спрямований (Principal Component Analysis — MPCA) на зменшення простору даних високої розмірності, тобто експериментальних змінних, до простору ознак низької розмірності, тобто автономних змінних, які є важливими для переконливого визначення даних. Це може бути застосовано у випадку, коли між попередніми змінними існує високий зв'язок. Під час видалення другорядних компонент аналіз головних компонент може зменшити кількість ознак і помістити набір даних у підпростір з низькою розмірністю [18-21]. Аналіз головних

компонент є типовим методом дослідження багатовимірних даних, який адаптовано для лінійного вилучення ознак. Тому було обрано процес вилучення ознак за допомогою методу головних компонент. Аспектами цього методу є експлойти як вектори ознак, які добре використовуються для символізації набору даних системи виявлення вторгнень в IoT-інфраструктурі. Стандартний алгоритм аналізу головних компонент може бути застосований для вилучення ознак з малих наборів даних і ігноруватиме важливу інформацію про ознаки. Метод аналізу головних компонент не дає гарантії того, що дані, пов'язані з відповідними класами, будуть якісно стиснуті. Щоб запобігти вищезазначеним проблемам, планується модифікований аналіз головних компонент. Метод модифікованого аналізу головних компонент зменшить вплив власних векторів після масивних власних значень шляхом стандартизації j-го елемента y_{ij}, і-ої ознаки вектора у відносно його стандартного відхилення, $\sqrt{\lambda_j}$. В результаті новий вектор ознак y_i' модифікується як

$$y_i' = [\frac{y_{i0}}{\lambda_0}, \frac{y_{i1}}{\lambda_1}, \dots \frac{y_{i(r-1)}}{\lambda_{r-1}}] \quad (5)$$

Однорідні вектори ознак використовуються для побудови нового підпростору ознак. У цьому процесі вектори ознак спочатку нормалізуються за допомогою квадратного кореня з наступних власних значень, а потім обчислюється відстань між навчальними та тестовими ознаками. Зазвичай, лінійне перетворення (аналіз головних компонент) можна подати у вигляді наступного рівняння:

$$C = NA \quad (6)$$

де N - матриця перетворення, A - вихідний вектор і C - перетворений вектор для визначення матриці перетворення N, наступне рівняння. Застосовується, тут матриці I,Z,U та λ - квадратна матриця з одиницею на діагоналі, коваріаційна матриця вихідного зображення, власні вектори та власні значення. U_j та λ_j $(j=1,2,...x)$ можна обчислити за рівнянням (2), причому власні значення впорядковані як $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$. Власні вектори U можна подати як $U=[U_1, U_2,.....,U_m]$.

У модифікованому аналізі головних компонент навчальні вибірки, які асоціюються з певним додатком, вибираються з набору даних системи виявлення вторгнень, а перетворена матриця N' була отримана з цих навчальних вибірок. Її можна описати як послідовне рівняння:

$$C = N'A \text{ т} \quad (7)$$

$$W_M = b_1u_1 + b_2u_2 + \cdots . + b_Mu_M \quad (8)$$

$$Z = \sum_{i=0}^{1} b_1u_1; 1 < M \quad (9)$$

На відміну від рівнянь (7) і (8), варіація присутня в матриці перетворення і, головним чином, присутня у вибірках для обчислення коваріаційної матриці, тут береться одиниця з навчальних вибірок; наступна одиниця - з усього набору мовних даних для розпізнавання.

Основною метою модифікованого аналізу головних компонент є обчислення трьох матриць подібності, що використовують вимірювання подібності на основі трьох параметрів, а саме: взаємної інформації, кутової інформації та ядра Гауса. Модифікований аналіз головних компонент об'єднує метод обгортки та метод прямого відбору з домінуючою дискримінативною здатністю для класифікації зразків.

Можна адаптувати модифікований аналіз головних компонент до функцій, в яких кількість навчальних вибірок є меншою за розмірність даних. Модифікований аналіз головних компонент забезпечує вищу точність класифікації та результат кластеризації порівняно з аналізом головних компонент.

Модифікований аналіз головних компонент - це математичний алгоритм, який використовує лінійну корекцію для з'єднання даних з простору великої розмірності до простору малої розмірності. Простір низької

розмірності фокусується на власних векторах коваріаційної матриці. У цій роботі було адаптовано модифікований аналіз головних компонент для вилучення цінних значущих ознак вторгнення для набору даних розпізнавання людської активності шляхом зменшення похибки та декореляції ознак. В результаті, модифікований аналіз головних компонент зменшує розмірність набору даних за рахунок включення координат з високим значенням дисперсії та ухиляється від даних з низькою дисперсією, отримуючи вхідні дані з нормальними параметрами, а дані про втручання включають такі характеристики, як середнє значення та стандартне відхилення. Середньоквадратичне відхилення: також називається середньоквадратичним відхиленням, оскільки воно є квадратним коренем із середнього квадратичного відхилення від середнього арифметичного.

$$\sigma = \sqrt{(\sum(a - \ddot{a})/m)} \tag{10}$$

Метод модифікованого аналізу головних компонент використовується для вилучення інформативних ознак із заданого набору даних, а також для зменшення розмірності цінних ознак. Було доведено, що модифікований аналіз головних компонент займає менше часу для вилучення ознак з нелінійної комбінації змінних, таких як мережеві дані Інтернету речей, і використовує мінімальну кількість ознак порівняно з алгоритмом аналізу головних компонент, на відміну від алгоритму автоматичного кодування, який займає більше часу для вилучення ознак. Також було помічено, що модифікований аналіз головних компонент був протестований з наборами даних Інтернету речей, в яких він не усуває помітні ознаки, які вимірюються як головна ознака для точного обчислення прогнозу.

Висновки

У дослідженні удосконалено метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, який на відміну від відомих застосовує апаратне рішення Hybrid Deep-GAN, і який може ефективно виявляти вторгнення, і який забезпечує підвищення ефективності керування IoT-інфраструктурою під час несанкціонованих вторгнень. Запропонований метод застосовує Hybrid Deep-GAN і може ефективно виявляти вторгнення, з якими стикаються мережі IoT через динамічну природу мереж Інтернету речей. Було доведено, що запропонований GAN займає важливе місце в області глибокого навчання, і основною проблемою в цій області є розробка системи виявлення вторгнень, яка забезпечує рішення для порушень безпеки. Метод дозволяє виявлення невідомих атак. Через величезну кількість різнорідних даних, що генеруються з мережі, обов'язковою є попередня обробка, яка тут здійснюється за допомогою алгоритму TBSS, який ефективно заповнює відсутні дані. Метод також застосовує MPCA, адаптований для виділення ознак, щоб виділити значущі ознаки, а також використовує алгоритм оптимізації Enhanced Whale для відбору найбільш придатних ознак. Нарешті, метод HDNN+ANN застосовано для точної класифікації вторгнень на заданому наборі даних. В кінцевому результаті зазначено, що метод EWO-HDNN+ANN забезпечує високу продуктивність щодо точності, достовірності, запам'ятовування, f-міри та пропонує нижчу частоту помилкових спрацьовувань, зменшує обчислювальну складність у порівнянні з іншими алгоритмами машинного навчання.

Література

1. Kim, A.N., Hekland, F., Petersen, S., Doyle, P. When HART goes wireless: understanding and implementing the WirelessHART standard, IEEE International Conference on Emerging Technologies and Factory Automation, 2018. pp. 899–907.
2. Kolias, C., Stavrou, A., Voas, J., Bojanova, I., Kuhn, R. Learning Internet-of-thingssecurity “Hands-on”. *IEEE Secur. Priv*. 20 (February), 2–11. http://dx.doi.org/ 10.1109/MSP.2016.4, , 2016.

3. Krimmling, J., Peter, S., Integration and evaluation of intrusion detection for CoAP in smart city applications. *Communications and Network Security (CNS)*, 2014 IEEE Conference on, 2014. pp. 73–78.

4. Kumar, S., Dutta, K. Intrusion detection in mobile ad hoc networks: techniques, systems, and future challenges. Secur. Commun. Netw. , 2016.9 (14), 2484–2556.

5. Le, A., Loo, J., Chai, K.K., Aiash, M. A specification-based IDS for detecting attacks on RPL-based network topology. *Information*, 2016. 7 (2), 25.

6. Le, A., Loo, J., Luo, Y., Lasebae, A., Specification-based IDS for securing RPL from topology attacks, In: *Wireless Days (WD)*, 2011. 2011 IFIP, pp. 1–3.

7. Lee, I., Lee, K., The internet of things (IoT): applications, investments, and challenges for enterprises. *Bus. Horiz.* 2015. 58 (4), 431–440.

8. Lee, T.-H., Wen, C.-H., Chang, L.-H., Chiang, H.-S., Hsieh, M.-C. A lightweight intrusion detection scheme based on energy consumption analysis in 6LowPAN. In: Huang, Y.-M., Chao, H.-C., Deng, D.-J., Park, J.J.J.H. (Eds.), Advanced Technologies, Embedded and Multimedia for Human-centric Computing, *LectureNotes in Electrical Engineering* 260. Springer, Netherlands, 2014. 1205–1213.

9. Liao, H.-J., Lin, C.-H.R., Lin, Y.-C., Tung, K.-Y., Intrusion detection system: a comprehensive review. J. Netw. Comput. Appl. 2013. 36 (1), 16–24.

10. Liu, C., Yang, J., Zhang, Y., Chen, R., Zeng, J. Research on immunity-based intrusion detection technology for the Internet of Things. Natural Computation(ICNC), *2011 Proceedings of the Seventh International Conference,* 2011. Vol. 1, pp. 212–216.

11. Notra, S., Siddiqi, M., Gharakheili, H., Sivaraman, V., Boreli, R. An experimental study of security and privacy risks with emerging household appliances. *Communications and Network Security (CNS),* 2014 IEEE Conference, 2014. pp. 79–84.

12. R. Abdulhammed, M. Faezipour, and K. M. Elleithy, "Network intrusion detection using hardware techniques: A review," in *Proc. IEEE long Island Systems, Applications and Technology Conference*, 2016.

13. B. Subba, S. Biswas, and S. Karmakar. A neural network based system for intrusion detection and attack classification. Proc. National Conference on Communication, 2016.

14. A. Gharib, I. Sharafaldin, A. II. Lashkari, and A. A. Ghorbani. An evaluation framework for intrusion detection dataset. *Proc. International Conference on Information Science and Security*, 2016.

15. S. Yusuf, W. Luk, M. K. N. Szeto, and W. Osborne. Unite: Uniform hardware-based network intrusion detection engine. *Reconfigurable Computing: Architectures and Applications*, 2006, pp. 389-400.

16. R. Ptoudfoot, K. Kent, E. Aubanel, and N. Chen, Flexible software- hardware network intrusion detection system. *Proc. International Symposium on Rapid System Prototyping,* 2008, pp. 182-188.

17. A. Das, D. Nguyen, J. Zambreno, G. Memik, and A. Choudhary. An FPGA-bascd network intrusion detection architecture. IEEE Transactions on Information Forensics and Security, vol. 3, no. 1, pp. 118-132, 2008.

18. A. L. P. de Franca, R. P. Jasinski, V. A. Pcdroni, and A. O. Santin. Moving network protection from software to hardware: An energy efficiency analysis. *Proc. IEEE Computer Society Symposium on VLSI,* July 2014, pp. 456-461.

19. A. L. P. de Franca, R. P. Jasinski, P. Cemin, V. A. Pedroni, A. O. Santin. The energy cost of network security: A hardware vs. software comparison. *Proc. International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 81-84.

20. S. Shreejith and S. A. Fahmy. Security aware network controllers for next generation automotive embedded systems. *Proc. Design Automation Conference (DAC),* 2015.

21. E. Hodo, X. Bellekens, A. Hamilton, P. L. Dubouilh, E. lorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of iot networks using artificial neural network intrusion detection system. *Proc. International Symposium on Networks, Computers and Communications,* 2016.

22. M. Idhammad, K. Afdel, and M. Belouch. DoS detection method based on artificial neural networks. *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 4, 2017.

ДОДАТОК Б
(обов'язковий)

ПРЕЗЕНТАЦІЯ

Метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень

Виконав студент групи КІ2м-21-1 Юрій Космина

Науковий керівник – д.т.н. проф. Лисенко С.М.

Хмельницький - 2023

Об'єкт, предмет та мета дослідження

- Метою роботи є підвищення ефективності керування IoT-інфраструктурою під час несанкціонованих вторгнень.
- Об'єктом дослідження є керування IoT-інфраструктурою під час несанкціонованих вторгнень.
- Предметом дослідження є метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Наукова новизна

У дослідженні удосконалено метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, який на відміну від відомих застосовує апаратне рішення Hybrid Deep-GAN, і який може ефективно виявляти вторгнення, і який забезпечує підвищення ефективності керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Практична значимість

В результаті виконаного наукового дослідження буде розроблено апаратно-програмні засоби засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень.

Актуальність

Втрати компаній на відновлення після кібератак зросли в середньому на 50% у 2022 році порівняно з попереднім роком. Витрати від кібератак можуть значно відрізнятися в залежності від розміру компанії, галузі, рівня захисту та інших факторів ці цифри можуть досягнути мільярдів доларів, і тому потрібно покращувати і оптимізовувати систему безпеки.

Метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою з застосуванням PP-GAN і IWO-DDL+ANN

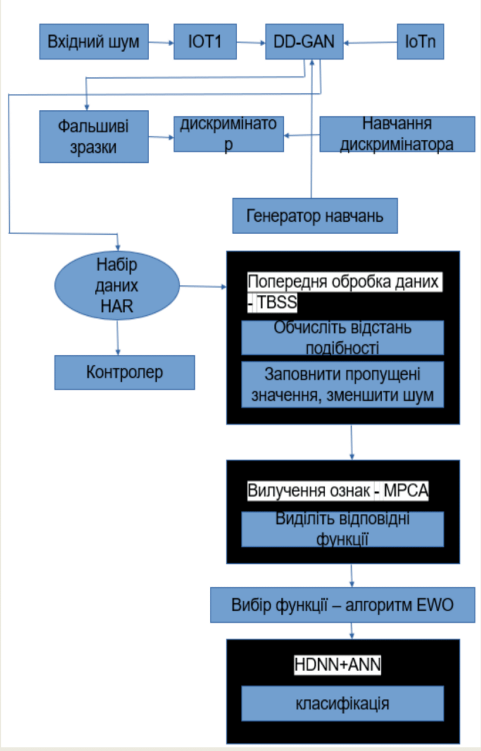
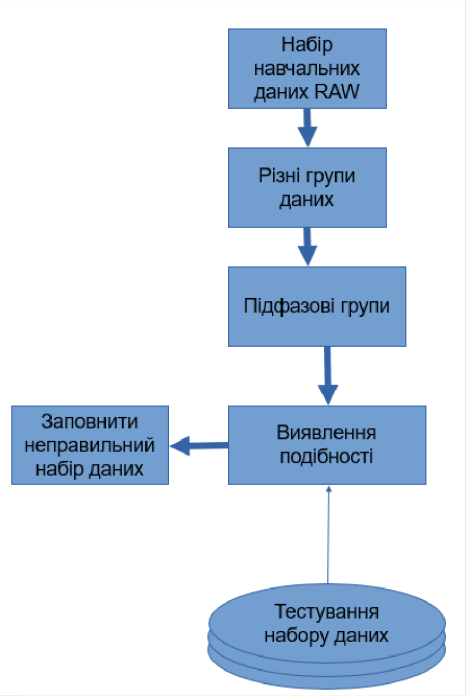
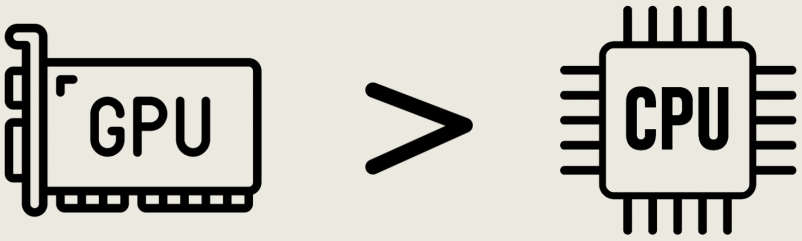


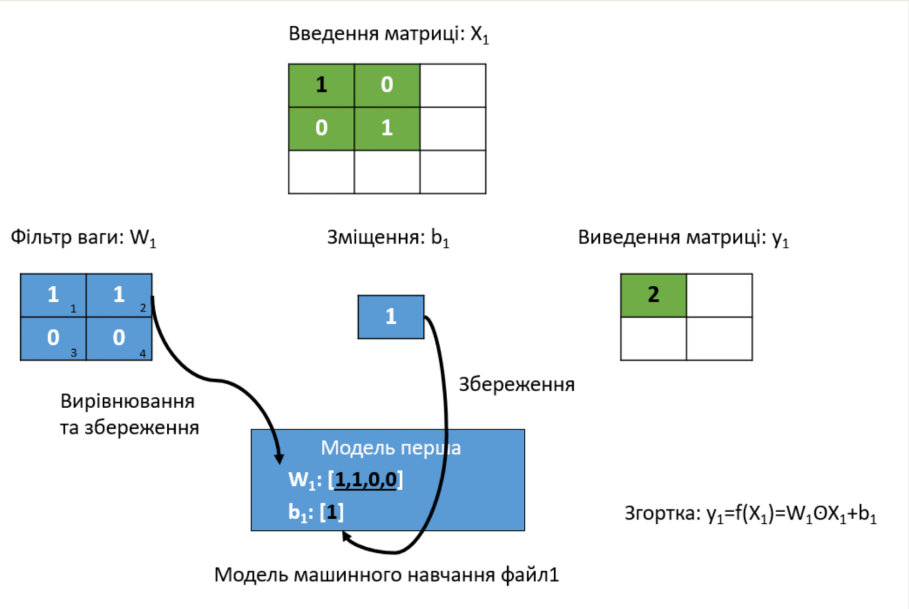
Схема покращеного алгоритму перенесення за подібністю підпросторів



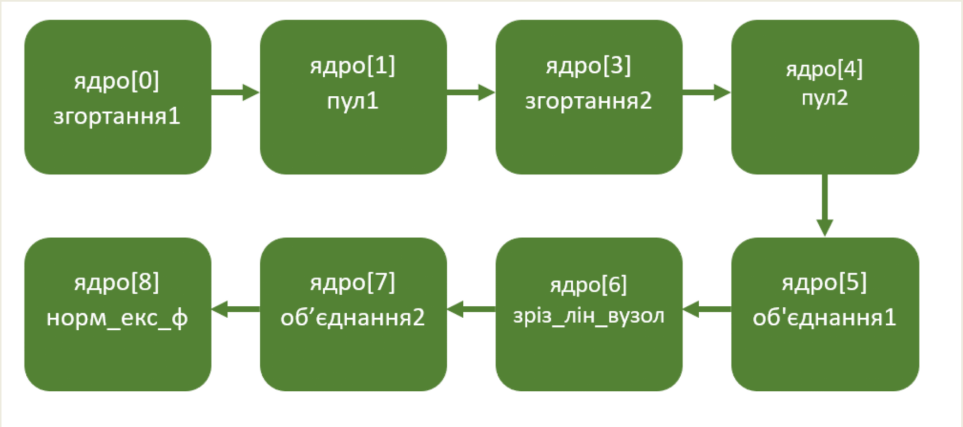
Щоб навчити модель отримувати максимальну точність за мінімальний час. Було обрано GPU для реалізації навчальної частини. Однією з причин є те, що навчальна частина зосереджена на швидкій і точній моделі побудови. І більшість навчальних робіт виконується лише один раз. Отже, під час цієї фази обчислювальна здатність з високою щільністю є важливішою, ніж затримка обчислення. За однакового рівня ціни графічні процесори мають більше обчислювальних ядер і більш ефективні в навчанні моделі.



Реалізація системи машинного навчання



Потік керування навчанням



Щоб знайти найкращий пристрій для навчальної моделі, порівнюючи швидкість ЦП і ГП. Було використано два варіанти налаштувань. Один має лише CPU-E5-1620, інший має CPU-E5-1620 і GPU-3050. Було обрано однакові 45120 навчальних прикладів і відповідно розрахували час їх навчання. Було проведено шість разів однакові тести та обчислили їхні середні значення.

Експеримент	CPU-E5-1620		GPU		FPGA
	Навчання	Точність (%)	Навчання	Точність (%)	Прискорення (GPU/CPU)
	час		Час		
1	448,6	98,7	50,61	98,8	8,86
2	447,79	98,88	51.17	98,58	8,75
3	448,35	98,9	50,73	98,8	8,84
4	448,62	98,94	50,46	98,88	8,89
5	447,62	98,59	50,94	98,82	8,79
6	447,88	98,94	50,78	98,79	82.82
Середній	448.10	98,8	50,8	98,8	80.80

У наступному експерименті було використано центральний процесор, графічний процесор і FPGA, щоб виконати ту саму роботу з висновків і порівняти їхню ефективність.
Щоб уникнути впливу різних смуг пропускання, ми розрахували лише час виведення одного зображення.
Було отримано 6 висновків на пристроях CPU, GPU та FPGA та обчислили середні значення часу відповідно.

Час експерименту	CPU-E5-1620	GPU-3050		FPGA		
	Висновок Час	Висновок Час	Прискорення (GPU/CPU)	Висновок Час	Прискорення (FPGA/CPU)	Прискорення (FPGA/GPU)
1	3172	616045	0,0051	88,74	35.7	6942,4
2	5564	589114	0,0094	90.12	61.7	6536,7
3	4620	588444	0,0079	94,83	48.7	6205,3
4	3234	598652	0,0054	84,57	38.2	7079
5	4037	600288	0,0067	101.08	39.9	5938,8
6	4579	609913	0,0075	108,74	42.1	5609
Середнє зеначення	4201	600409	0,007	94,7	44.4	6341,5

З метою перевірки ефективності методу було проведено два експерименти на тих самих прикладах тесту 10K MNIST відповідно. В експерименті 1 було лише перевірено точність висновку FPGA за допомогою оригінальної моделі, тоді як у експерименті 2 було змінено конфігурацію CNN.
Також було переналаштовано модель за допомогою TensorFlow із графічним процесором, перенесли модель на FPGA.
Остаточно, було перевірено результати логічний висновку на FPGA і Tensorflow.

Експеримент 1	Точність (%)	Експеримент 2	Точність (%)
N/A	N/A	TensorFlow	98.29
FPGA	99.05	FPGA	98.44

Висновок

Результати експерименту показують, що швидкість навчання графічного процесора в середньому в 8,8 раза вища, ніж у центрального процесора, а швидкість виводу FPGA в середньому в 40,2 раза вища, ніж у центрального процесора, і в середньому в 62 рази швидше, ніж у графічного процесора.

Таким чином, було покращено точність виявлення вторгнень з 99,05% до 99,13% і зберегли точність 99,13% успішно, коли було перенесено модель з платформи GPU на платформу FPGA.

UNICHECK

Ім'я користувача:
Кафедра КІ

ID перевірки:
1014987211

Дата перевірки:
09.05.2023 10:19:58 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
09.05.2023 10:21:31 EEST

ID користувача:
100005591

Назва документа: **Космина_Метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під...**

Кількість сторінок: **94** Кількість слів: **17565** Кількість символів: **136435** Розмір файлу: **556.34 KB** ID файлу: **1014678539**

2.15%
Схожість

Найбільша схожість: **1.44% з джерелом з Бібліотеки** (ID файлу: **1014669707**)

0.95% Джерела з Інтернету	135	Сторінка 96
1.83% Джерела з Бібліотеки	122	Сторінка 97

0.3% Цитат

Цитати	3	Сторінка 98
Посилання	1	Сторінка 98

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	19

Tue May 09 09:25:08 EEST 2023, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 31.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 10%**

ID: 113114 Назва: МКР Метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень Додано в БД: 2023-05-09 Автора: Космина Ю.І. Керівники: Лисенко С.М. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	112425	868	36289 (32%)	289 (33%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми
112081	Назва: ЗВІТ з науково-дослідної практики "Метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень" Додано в БД: 2023-03-21 Автора: Космина Ю.І. Керівники: Гнатчук Є.Г. Консультанти: Опоненти:	35168 (31.0%)	281 (32.0%)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувач: Космина Юрій Іванович

Тема: Метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень – ; кількість сторінок записки 79

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано апаратно-програмний засіб для керування IOT-інфраструктурою під час несанкціонованих вторгнень

2. Висновок про відповідність роботи дипломному завданню Кваліфікаційна робота магістра відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено огляд методів виявлення вторгнень в інтернеті речей. Досліджено відомі рішення та засоби в цій сфері. У другому розділі запропоновано модель функціонування системи керування іот-інфраструктурою під час несанкціонованих вторгнень. У третьому розділі удосконалено метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень. У четвертому розділі запропоновано апаратно-програмний засобіб керування іот-інфраструктурою під час несанкціонованих вторгнень

4. Позитивні сторони роботи: Запропонована система синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень, що використовує апаратне рішення на основі GAN.

5. Негативні сторони роботи: В роботі присутні певні логічні помилки щодо опису моделі функціонування системи керування IoT-інфраструктурою під час несанкціонованих вторгнень

6. Оцінка графічного оформлення та пояснювальної записки роботи: _–_____

7. Відгук про роботу в цілому: __В загальному робота виконана на невисокому рівні._____

8. Інші зауваження: ___–_____

9. Оцінка кваліфікаційної роботи:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи вважаю, що робота заслуговує оцінки «задовільно» 3.00 (E)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) ____д.т.н.. професор, Мартинюк В.В., завідувач кафедри_____

"__10 травня__"_____2023р. _____

Завідувачу кафедри КІІС
д-р.техн.наук, проф. Говорущенко Т. О.

Кошина Юлія Анатоліна
ПІБздобувача вищої освіти

ФІТ, 2 курсу, групи КІ2м-21-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

10.05.2023
дата

підпис

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень

Автор: Космина Юрій Іванович

Спеціальність: 123 – Компютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Лисенко Сергій Миколайович доктор техн. наук, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах є збіг зі звітом з науково-дослідної практики автора Юрія Космини "Метод синтезу апаратно-програмних засобів керування IoT-інфраструктурою під час несанкціонованих вторгнень ", який було додано в репозитраій ХНУ 21 березня 2023 року;
2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
3) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на декілька фрагментів речення;
4) в якості запозичень в окремих місцях системою зафіксовано формули, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
5) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості Unicheck, складає 2,15% і адресується до 135 першоджерела; та системою Anti-Plagiarism складає 31%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ С.М. Лисенко

Гарант ОП _____ О. С. Савенко

Завідувач кафедри КІІС _____ Т. О. Говорущенко