МАГІСТЕРСЬКА РОБОТА

МР. ІПм - 33.00.00.000 ПЗ

Група ІПм-23-1

Вацик Олексій

2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Вацик Олексій Сергійович

(прізвище, ім'я, по батькові)

УДК 004.942

(індекс)

МАГІСТЕРСЬКА РОБОТА

Концептуальні моделі та методи машинного навчання для застосунків Інтернету речей

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Вацик О.С.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Яцишин Микола Миколайович, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту

Завідувач кафедри

доц. Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Нормоконтроль

доц. Вовк Р.Б.

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**
доц. В.В. Бандура
“ 04 ” вересня 2024 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Вацику Олексію Сергійовичу

(прізвище, ім’я, по-батькові)

1. Тема магістерської роботи “ Концептуальні моделі та методи машинного навчання для застосунків Інтернету речей ”

керівник проекту (роботи) Яцишин Микола Миколайович, к.т.н., доцент

затверджені наказом закладу вищої освіти від “ 22 ” листопада 2024 р. № 781/7

2. Строк подання студентом проекту (роботи) 15 грудня 2024 р.

3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі побудови та функціонування інформаційних технологій машинного навчання

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

1. Дослідження предметної області застосування концепцій Інтернету речей
2. Методи та моделі застосування машинного навчання в області IoMT
3. Моделі та алгоритми сегментації процесів в системі IoMT в контексті машинного навчання
4. Імплементація загорткових мереж та методів машинного навчання для Інтернету речей

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1. Типова структура IoMT (рис. 1.1)
2. Візуальне представлення машини опорних векторів (SVM) (рис. 1.2)
3. Проста CNN для класифікації захворювань (рис. 1.3)
4. Етапи обробки в CNN, які використовуються для процесів класифікації/сегментації (рис. 1.4)
5. Таксономія PUF (рис. 2.1)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата
Перевірка на плагіат	доц., к.т.н. Вовк Р.Б.	

7. Дата видачі завдання 04 вересня 2024 р.

Керівник _____
(підпис)

Завдання прийняв до виконання_____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	15.09.2024	виконано
2	Аналіз концепцій та алгоритмів предметної області	29.09.2024	виконано
3	Дослідження предметної області застосування концепцій Інтернету речей	15.10.2024	виконано
4	Методи та моделі застосування машинного навчання в області IoMT	08.11.2024	виконано
5	Моделі та алгоритми сегментації процесів в системі IoMT в контексті машинного навчання	20.11.2024	виконано
6	Імплементація загорткових мереж та методів машинного навчання для Інтернету речей	01.12.2024	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.12.2024	виконано

Студент – магістр _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Магістерська робота: 79 с., 22 рис., 5 табл., 51 джерел.

Тема: Концептуальні моделі та методи машинного навчання для застосунків Інтернету речей

Об'єкт дослідження: системи Інтернету речей (IoT), зокрема Internet of Medical Things (IoMT), та їх обчислювальні й безпекові аспекти.

Мета роботи: дослідження та впровадження методів машинного навчання, зокрема згорткових нейронних мереж і фізично неклонованих функцій (PUF), для підвищення ефективності й безпеки систем Інтернету речей (IoT).

Предмет дослідження: методи машинного навчання та архітектури, такі як згорткові нейронні мережі і фізично неклоновані функції, для покращення обробки даних та забезпечення безпеки в системах IoT.

Результати дослідження

В роботі представлено новий підхід до навчання без учителя з використанням автоенкодерів, який забезпечує надійну обробку даних та підвищену безпеку у системах IoT.

Висновок

Запропоновано інноваційні архітектури на основі фізично неклонованих функцій для захисту пристроїв IoT від атак на основі машинного навчання та оозроблено методики використання загорткових нейронних мереж для обробки даних у системах IoT з обмеженими ресурсами.

ІНТЕРНЕТ РЕЧЕЙ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, ФІЗИЧНО НЕКЛОНОВАНІ ФУНКЦІЇ, МАШИННЕ НАВЧАННЯ, ОБМЕЖЕНІ ОБЧИСЛЮВАЛЬНІ РЕСУРСИ, БЕЗПЕКА ІОТ, АВТОЕНКОДЕРИ, АТАКИ НА ОСНОВІ МАШИННОГО НАВЧАННЯ.

ABSTRACT

Master Thesis: 79 pp., 22 fig., 5 tab., 51 sources.

Thesis Subject: Conceptual models and methods of machine learning for Internet of Things applications

Research object: Internet of Things (IoT) systems, in particular Internet of Medical Things (IoMT), and their computing and security aspects.

The purpose of the work: research and implementation of machine learning methods, in particular convolutional neural networks and physically uncloned functions (PUF), to improve the efficiency and security of Internet of Things (IoT) systems.

Research subject: machine learning techniques and architectures, such as convolutional neural networks and physically uncloned functions, to improve data processing and ensure security in IoT systems.

Research results

The work presents a new approach to learning without a teacher using autoencoders, which provides reliable data processing and increased security in IoT systems.

Conclusion

Innovative architectures based on physically non-cloned functions are proposed to protect IoT devices from attacks based on machine learning, and methods of using convolutional neural networks for data processing in IoT systems with limited resources are developed.

INTERNET OF THINGS, CONVOLUTIONAL NEURAL NETWORKS, PHYSICALLY UNCLONED FUNCTIONS, MACHINE LEARNING, LIMITED COMPUTING RESOURCES, IOT SECURITY, AUTOENCODERS, MACHINE LEARNING ATTACKS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

IoT - Internet of Things

FPGA - Field Programmable Gate Array

ML - Machine Learning

DL - Deep Learning

CNN - Convolutional Neural Network

RNN - Recurrent Neural Network

PUF - Physical Unclonable Function

LSTM - Long Short-Term Memory

KNN - K-Nearest Neighbors

PCA - Principal Component Analysis

HMM - Hidden Markov Model

K-Means - K-Means Clustering

ANN - Artificial Neural Network

RMSE - Root Mean Squared Error

FNN - Feedforward Neural Network

ReLU - Rectified Linear Unit

SGD - Stochastic Gradient Descent

IoMT - Internet of Medical Things

FL - Federated Learning

DRL - Deep Reinforcement Learning

ФНФ - фізично неклонованиафункція

ВСТУП

Актуальність теми.

Швидке зростання Інтернету речей (IoT) та його медичної підгалузі — Internet of Medical Things (IoMT) — є важливою складовою сучасних інформаційних технологій. У зв'язку зі стрімким розвитком систем, які пов'язані з охороною здоров'я, інфраструктурою та критичними послугами, забезпечення надійної обробки даних та безпеки стає першочерговим завданням. IoMT-екосистеми використовуються для збору та аналізу чутливих даних, зокрема фізіологічних показників пацієнтів, що вимагає розробки передових технологій для збереження конфіденційності та цілісності інформації.

Водночас використання IoT пристроїв часто обмежується невеликими обчислювальними потужностями та енергоресурсами, що створює виклики для обробки великих обсягів даних і застосування традиційних методів машинного навчання. Крім того, сучасні системи IoT є вразливими до різних видів атак, зокрема тих, що базуються на машинному навчанні, що може призвести до порушення безпеки, зламів пристроїв та підробки даних.

Для ефективного функціонування IoMT необхідні інноваційні методи обробки й захисту даних, які можуть працювати на обмежених ресурсах. Зокрема, використання загорткових нейронних мереж і фізично неклонованих функцій (PUF) відкриває нові можливості для створення стійких і безпечних рішень. Актуальність теми також зумовлена необхідністю впровадження децентралізованих навчальних систем, таких як федеративне навчання, які зберігають конфіденційність даних і знижують навантаження на мережу, що особливо важливо в умовах сучасних кіберзагроз.

Отже, розвиток технологій для ефективного управління й захисту даних в IoT і IoMT системах є критично важливим для створення безпечного

та продуктивного цифрового середовища, що робить це дослідження значущим і необхідним.

Метою дослідження є дослідження та впровадження методів машинного навчання, зокрема згорткових нейронних мереж і фізично неклонованих функцій (PUF), для підвищення ефективності й безпеки систем Інтернету речей (IoT).

Об'єкт дослідження - системи Інтернету речей (IoT), зокрема Internet of Medical Things (IoMT), та їх обчислювальні й безпекові аспекти.

Предмет дослідження - методи машинного навчання та архітектури, такі як згорткові нейронні мережі і фізично неклоновані функції, для покращення обробки даних та забезпечення безпеки в системах IoT.

Задачі дослідження

- Проаналізувати структуру та особливості екосистеми IoMT та виклики, що стоять перед сучасними системами IoT.

- Дослідити алгоритми машинного навчання та глибокі нейронні мережі, адаптовані до роботи на платформах з обмеженими ресурсами.

- Розробити архітектури на основі фізично неклонованих функцій (PUF) для автентифікації IoT-пристроїв.

- Визначити основні загрози, пов'язані з атаками на основі машинного навчання, та розробити захисні методи.

- Впровадити загорткові нейронні мережі та автоенкодери для обробки даних у системах IoT, забезпечивши надійне навчання ознак.

- Оцінити ефективність запропонованих методів щодо безпеки й обчислювальної продуктивності.

Методи дослідження:

- Аналіз літературних джерел для оцінки поточного стану технологій IoT і IoMT.

- Розробка та імплементація нейронних мереж і фізично неклонованих функцій.

- Моделювання та симуляція для тестування безпеки й ефективності алгоритмів.

- Статистичний аналіз для оцінки результатів експериментів.

Наукова новизна отриманих результатів

Запропоновано інноваційні архітектури на основі фізично неклонованих функцій для захисту пристроїв IoT від атак на основі машинного навчання та оозроблено методики використання загорткових нейронних мереж для обробки даних у системах IoT з обмеженими ресурсами.

Практичне значення результатів

Розроблені методи та моделі можуть бути впроваджені в сучасні системи IoT для забезпечення надійного захисту даних, ефективного використання обчислювальних ресурсів та покращення загальної продуктивності. Запропоновані рішення мають потенціал для застосування в медичних системах, де критичними є безпека й ефективність роботи з великим обсягом даних.

Структура магістерської роботи. Робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 79 сторінок, і містить 22 рисунки, 5 таблиць, список використаних джерел із 51 найменування.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ КОНЦЕПЦІЙ ІНТЕРНЕТУ РЕЧЕЙ

1.1. Постановка проблеми дослідження

Штучний інтелект і всюдисущі сенсорні системи останнім часом досягли величезного прогресу, що призвело до революційного впливу на такі сфери, як охорона здоров'я, розваги та транспорт через колективну екосистему під назвою Інтернет речей. Поява 5G і вдосконалених бездротових мереж ще більше прискорить дослідження та розробку інструментів глибокого навчання, сенсорних систем і обчислювальних платформ, забезпечуючи покращену затримку мережі та пропускну здатність. Хоча в Інтернеті речей було досягнуто величезного прогресу, поточна робота в основному зосереджена на створенні надійних додатків, які використовують дані, зібрані через всюдисущі сенсорні вузли, для забезпечення діючих правил і шаблонів. Такі інфраструктури за своєю суттю не враховують проблеми, пов'язані з масштабом, такі як конфіденційність, безпека даних і здатність забезпечити повне занурення. Це особливо важливо, оскільки через дещо обмежений обсяг обчислювальних ресурсів периферійні вузли IoT самі не обробляють спостережувані дані. Замість цього вони передають зібрані дані на більш потужні сервери для обробки. Така передача інформації може створювати навантаження на мережу, створюючи проблеми безпеки, такі як підслуховування та атаки типу "людина посередині".

У цій роботі ми вирішуємо ці проблеми, використовуючи машинне навчання як інструмент, розробляючи легкі алгоритми з урахуванням конфіденційності, одночасно оцінюючи здійсненність примітивів безпеки апаратного забезпечення, таких як фізичні неклоновані функції (PUF) для безпеки вузлів Інтернету речей. Точніше, ми розробляємо алгоритми для

безперервного моніторингу активності з датчиків без будь-яких позначених даних, що є першим кроком до парадигми децентралізованого навчання.

Запропонована структура за своєю суттю зберігає конфіденційність шляхом обмеження обсягу даних, що передаються через мережу, забезпечуючи при цьому зворотний зв'язок у реальному часі. По-друге, ми аналізуємо властивості різних підходів глибокого навчання щодо енергоспоживання, обсягу пам'яті та затримки та надаємо оптимізацію часу проектування, щоб забезпечити реалізацію на платформах з обмеженим обчислювальним ресурсом. Нарешті, ми оцінюємо доцільність використання автентифікації на основі PUF для периферійних вузлів IoT, досліджуючи їх сприйнятливість до атак машинного навчання. Ми показуємо, що потужні архітектури PUF чутливі до неінвазивної атаки клонування на основі машинного навчання. Ми також пропонуємо імовірнісну дискримінаторну модель для посилення безпеки протоколу автентифікації на основі PUF шляхом виявлення можливих випадків атак клонування та посилення автентифікації на основі PUF.

У поєднанні ці підходи пропонують шлях вперед для розробки структури IoT для безперервного моніторингу активності, яка може масштабуватися до мільйонів вузлів, забезпечуючи при цьому конфіденційність і безпеку спостережуваних даних. Ми показуємо, що запропонований алгоритм моніторингу активності може ефективно розпізнавати та сегментувати дії з потокових даних без будь-яких позначених даних на обмежених платформах із затримкою, близькою до реального часу. Наші вдосконалення часу розробки для алгоритмів глибокого навчання можуть призвести до 11-кратного зниження енергоспоживання порівняно з іншою платформою FPGA із збільшеним об'ємом пам'яті в 96 разів, зберігаючи найсучаснішу точність класифікації. Завдяки широким експериментам ми показали, що потужні архітектури PUF можна успішно клонувати, включно з тими, що зашифровані за допомогою двох різних протоколів шифрування в DES і AES і з різним ступенем обфускації.

Запропонований дискримінатор може відрізняти клоновані PUF-пристрої від автентичних PUF із середньою точністю 96,01% і може використовуватися для швидкої автентифікації мільйонів вузлів IoT віддалено з хмарного сервера.

1.2. Опис структури та особливостей екосистеми Internet of Medical Things (IoMT)

Екосистема Інтернету речей (IoT) зросла в геометричній прогресії завдяки конвергенції різних технологій, таких як глибоке навчання, сенсорні системи та досягнення в обчислювальних платформах, таких як програмовані вентильні матриці (FPGA) і графічні процесори (GPU). Очікується, що поява технології 5G і перспективи вищої пропускної здатності підвищать пов'язаність сучасної екосистеми Інтернету речей. Ці досягнення призвели до розробки повсюдних сенсорних вузлів, які дозволяють збирати фізіологічні дані для розумної охорони здоров'я. Під загальною назвою «Інтернет медичних речей» (IoMT) розумна охорона здоров'я стала життєздатним варіантом покращення якості життя різними способами, такими як здоровий спосіб життя, віддалений доступ до медичної допомоги для сільських районів і медичне обслуговування вдома, щоб назвати декілька.

Типову структуру IoMT показано на рисунку 1.1. Можна побачити, що дані, зібрані через сенсорні вузли, передаються через безліч серверів, маршрутизаторів і мережевих шлюзів, перш ніж вони будуть оброблені в автономному режимі, і можна буде отримати діючі правила та програми. Застосування пристроїв IoT варіюється від переносних комп'ютерних пристроїв, біоімплантованих пристроїв для моніторингу життєво важливих функцій організму для безпосередньої взаємодії людини, а також для «розумних» пристроїв, з якими ми взаємодіємо щодня. З такою поширеною природою «розумних» пристроїв характер даних, що збираються та

обробляються, може бути все більш приватним і вимагати заходів для забезпечення цілісності та безпеки даних [1, 2].

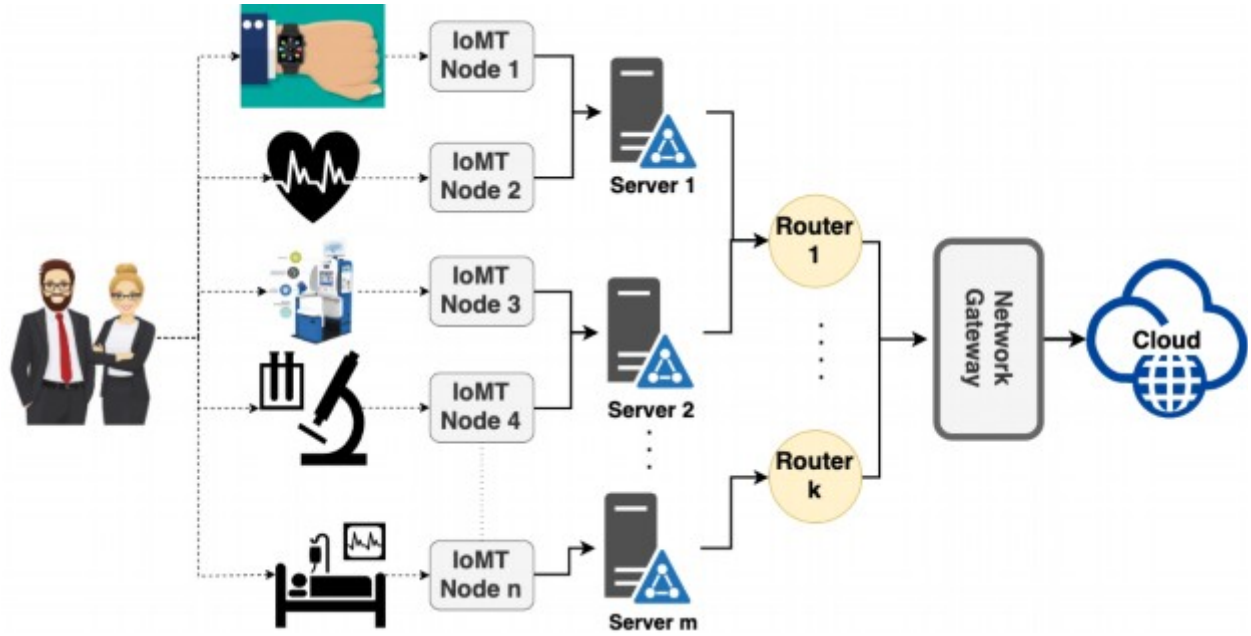

Рис. 1.1. Типова структура IoMT — це сукупність сенсорних вузлів і серверів обробки даних, які збирають і обробляють конфіденційні дані

Через дещо обмежений обсяг обчислювальних ресурсів самі периферійні вузли IoT не обробляють таку інформацію. Натомість вони використовуються як агенти збору даних, які передають зібрані дані на більш потужні периферійні сервери для обробки інформації. Ця передача інформації часто здійснюється через бездротові мережі. Це створює певні проблеми безпеки, такі як підслуховування та атаки типу "людина посередині", які є досить поширеними проблемами безпеки та, отже, вимагають надійних протоколів безпеки для забезпечення цілісності переданих даних. Хоча ми досліджували інфраструктурні аспекти екосистеми Internet of Medical Things, було відносно менше робіт над реальними інтелектуальними алгоритмами, які можуть використовувати переваги бездротового зв’язку та розподілену навчальну платформу для інтелектуальної діагностики здоров’я та моніторингу.

Поточна робота над інфраструктурою IoMT здебільшого зосереджена на створенні надійних програм, які використовують дані, зібрані через повсюдні сенсорні вузли, для надання діючих правил і шаблонів. Вони роблять такі базові припущення:

1) вузли даних захищені та не сприйнятливі до будь-яких вторгнень,

2) фокус зосереджений на отриманні вищої точності ціною вищих вимог до обчислень,

3) вузли даних не є обчислювально обмежений.

Ці міркування призвели до створення вузькоспеціалізованих, ресурсомістких програм, які не усвідомлюють комунікаційних, обчислювальних і архітектурних міркувань структури IoMT. В ідеалі додатки повинні обмінюватися перевагами точності на основі контексту та доступності ресурсів для більш ефективного використання різних даних і обчислювальних вузлів, щоб краще адаптувати розподіл робочого навантаження в периферійних середовищах.

1.3. Виклики та поточні тенденції для Інтернету речей

Одним із головних обмежень поточних програм машинного навчання в рамках IoMT є те, що вони не обізнані з ресурсами та за своєю суттю не зберігають конфіденційність. Дані збираються на периферійних вузлах, передаються через мережу (дротову або бездротову) і обробляються в автономному режимі на великих серверах, що потребують інтенсивних обчислень. Одним із важливих кроків до забезпечення безпеки та конфіденційності даних є зменшення обсягу даних, що передаються через бездротовий спектр. Цей підхід вимагає, щоб дані оброблялися на самих крайових вузлах, забезпечуючи при цьому передачу на обчислювальний сервер лише нових даних. Більшість успішних програм базуються на підходах до глибокого навчання, які вимагають великих обсягів позначених даних і платформ з високою інтенсивністю обчислень. Однак обчислювальна

обмеженість пристроїв на межі обмежує ступінь ефективної обробки даних, особливо враховуючи обчислювальну складність сучасних підходів до машинного навчання.

Важливим аспектом використання архітектур глибокого навчання на крайніх пристроях є вихід за рамки розробки апаратних прискорювачів. Це вимагає глибшого розуміння різних аспектів дизайну нейронних мереж та їх впливу на SWAP (розмір, вага, площа та потужність). Сучасні підходи до забезпечення глибокого навчання на апаратних платформах з обмеженими ресурсами зосереджені насамперед на зменшенні затримки та потужності. Вони роблять сильні припущення щодо доступності таких ресурсів, як обчислювальна потужність, пам’ять і розміри. Це створює ряд проблем:

1) моделі глибокого навчання вимагають значних навчальних ресурсів, таких як обчислювальна потужність і позначені навчальні дані,

2) уможливлення логічного висновку та навчання моделей глибокого навчання на обмежених платформах вимагає компромісу між точністю та SWAP, що може не бути варіантом для критично важливих програм, таких як аналітика охорони здоров’я в режимі реального часу,

3) розподіл даних зібраних/спостережуваних даних може бути динамічним і може не відповідати розподілу даних навчання, а отже, може погіршити продуктивність моделей глибокого навчання.

Нейронні мережі зазвичай реалізуються на відносно необмежених пристроях, таких як графічні процесори (GPU) робочої станції, тому методи проектування та оптимізації погано масштабуються для додатків з низьким енергоспоживанням і обмеженими ресурсами в Інтернеті речей (IoT). Незважаючи на те, що зусилля були спрямовані на створення різних структур прискорення на обмежених платформах, отримання енергоефективних реалізацій без значного зниження точності залишається мистецтвом, ніж наукою. Використання альтернативних обчислювальних платформ і ефективних методологій навчання може допомогти зменшити цю залежність від обчислювальних ресурсів.

Поточні додатки машинного навчання для IoMT в основному вирішувалися через контрольоване навчання [3, 4]. Ці підходи, хоч і добре працюють, вимагають великої кількості експертно-анотованих, позначених даних, отримання яких може бути дуже дорогим. Крім того, збір такої кількості даних передбачає захоплення та передачу приватної конфіденційної інформації, такої як характеристики руху, місцезнаходження, частота серцевих скорочень тощо, і може спричинити серйозні проблеми з безпекою в разі порушення безпеки. З прогресом у машинному навчанні та апаратній безпеці цілісність даних може бути порушена, а постійний збір дуже конфіденційної інформації може спричинити проблеми. Децентралізоване навчання [5] стало життєздатною альтернативою, коли кожна модель на периферійних пристроях надсилається на віддалений сервер. Потім віддалений сервер знаходить загальну модель за допомогою різних методів усереднення моделі, і модель на пристрої на крайових вузлах одночасно оновлюється новою загальною моделлю.

Цей підхід вимагає швидких і ефективних алгоритмів навчання, які можуть працювати в масштабі та на пристрої, включно з обчислювально обмеженими крайовими вузлами. Однак це викликає певні проблеми з безпекою:

1) було показано, що знання параметрів моделі та вагових коефіцієнтів може уможливити атаки білого ящика,

2) атака «людина посередині» може непомітно порушити ваги це може повністю змінити характеристики продуктивності моделі,

3) поєднання моделей у різних обчислювальних архітектурах може призвести до заплутаних характеристик точності для стиснутих моделей, що, у свою чергу, може погіршити їх продуктивність.

Крім того, анотацію таких великомасштабних даних може бути важко отримати, і вона не сприяє масштабуванню для мільйонів користувачів. Іншою альтернативою є використання самоконтрольованих і неконтрольованих підходів для вивчення надійних представлень із потокових

даних без необхідності зберігати та передавати дані. Завдяки надійним представленням модель можна налаштувати за допомогою менших анотованих даних. Поєднання можливостей навчання репрезентації самоконтрольованих мереж і можливостей агрегації моделей децентралізованого навчання пропонує шлях вперед для обробки великих обсягів даних на межі, зберігаючи при цьому конфіденційність особи, яка використовує пристрої IoMT, і гарантуючи цілісність машини. рамка навчання.

Забезпечення цілісності та безпеки зібраних даних [1, 2] вимагає надійних протоколів безпеки для забезпечення цілісності даних, що передаються. Протоколи безпеки, такі як автентифікація вузлів , мають бути досить легкими, але високобезпечними, щоб забезпечити виконання цих протоколів на вузлах IoT з обмеженим живленням. Протоколи автентифікації можуть варіюватися від дуже простих, таких як фізичне зберігання секретного ключа на кремнієвих пристроях, до складних алгоритмів на основі криптографії, які можуть потребувати значних вимог до потужності та площі пристрою. Проте було показано, що найпростішу автентифікацію, тобто фізичне зберігання секретного ключа на вузловому пристрої, можна обійти за допомогою фізичних атак і атак на бокових каналах [6]. Відновлення секретного ключа за допомогою таких фізичних атак може скомпрометувати всю мережу IoT і, отже, порушити цілісність і анонімність переданих даних. Оскільки потреба в легких , але безпечних протоколах автентифікації зростає зі швидким зростанням використання вузлів IoT, фізично неклоновані функції (PUF) [7] з’явилися як життєздатний варіант безпеки вузлів IoT [8].

Незважаючи на те, що моделі PUF надзвичайно складні та безпечні, їх можна клонувати за допомогою складних математичних моделей і криптоаналізу. Попередні роботи показали, що їх можна клонувати за допомогою моделей машинного навчання. Однак вони вимагають, щоб вузол IoMT був фізично доступний, а внутрішні знання про вузол були відомі

апріорі. Інше припущення в поточних роботах полягає в тому, що протоколи для аутентифікації на основі PUF надсилаються через канал зв'язку у вигляді звичайного тексту, тобто жодне шифрування не маскує прямий зв'язок між характеристиками запиту та відповіді PUF у вузлі даних. З огляду на те, що більшість, якщо не весь, зв'язок у бездротовому каналі зашифровано за допомогою певного методу хешування або шифрування, це дуже вагоме припущення. Отже, існує сильна потреба оцінити життєздатність використання протоколів автентифікації на основі PUF для безпеки вузлів у структурі IoMT і сформулювати відповідний механізм захисту для виявлення та автентифікації скомпрометованих крайових вузлів.

1.4. Аналіз алгоритмів машинного навчання в контексті IoMT

Проблема розпізнавання та сегментації стану сну була значною мірою вирішена за допомогою навчання під наглядом [3, 4]. Ці підходи, хоч і добре працюють, вимагають великої кількості експертно-анотованих, позначених даних, отримання яких може бути дуже дорогим. Крім того, збір такої кількості даних передбачає захоплення та передачу приватної конфіденційної інформації, такої як характеристики руху, місцезнаходження, частота серцевих скорочень тощо, і може спричинити серйозні проблеми з безпекою в разі порушення безпеки. З прогресом у машинному навчанні та апаратній безпеці цілісність даних може бути порушена, а постійний збір дуже конфіденційної інформації може спричинити проблеми.

В роботі [9] запропонували алгоритм для сегментації даних журналу життя на події. Lifelogging — це процес використання архітектури Microsoft SenseCam для збирання послідовності зображень щосекунди разом з іншими сенсорними даними, такими як показання акселерометра, показання термометра, показання інфрачервоного датчика та датчиків світла. Це дозволило витягти мультимодальну функцію з досвіду користувача, який збирає дані. Автори запропонували метод сегментації даних журналу життя

на події на основі візуального вмісту, закодованого MPEG-7. Кожне зображення з даних SenseCam обробляється для отримання набору функцій, що складається з макета кольорів, структури кольорів, масштабованого кольору та гістограми країв.

Використовуючи ці дескриптори MPEG-7 та інші візуальні функції, вони сегментують дані на події, виконуючи три кроки:

1) обчислюють подібність між сусідніми блоками зображень,

2) визначають порогове значення, вивчаючи значення подібності, які відповідають різним подій,

3) видалити межі послідовних подій, щоб сформувати цілісну подію.

Міра подібності обчислюється за допомогою методу нормалізації MinMax на векторі ознак, що складається з вилучених візуальних ознак, а також інших даних датчика. Імовірність межі події обчислюється шляхом порівняння міри подібності з суміжними зображеннями за допомогою техніки підрахунку піків. Було випробувано різні методи вимірювання відстані, такі як Евклідова, Манахеттенська та гістограма перехрестя, причому гістограма перетину дала найкращі результати.

Автори [10] були одними з перших, хто використовував дані на основі споживчих пристроїв для розробки системи безперервного моніторингу з кількох датчиків, таких як акселерометр, гіроскоп, датчик гравітації, дані компаса, а також дані датчиків світла. Спочатку дані про рух користувача аналізуються та класифікуються за шістьма категоріями на основі вектора ознак руху, обчисленого за допомогою середнього значення, дисперсії, перекосу, ексцесу та ШПФ (швидке перетворення Фур'є). Контекст дій користувача класифікується за допомогою безлічі класифікаторів SVM.

Машини опорних векторів (SVM) — це тип керованого алгоритму машинного навчання, який використовується для завдань класифікації та регресії. Вони широко використовуються в різних сферах, включаючи розпізнавання образів, аналіз зображень і обробку природної мови.

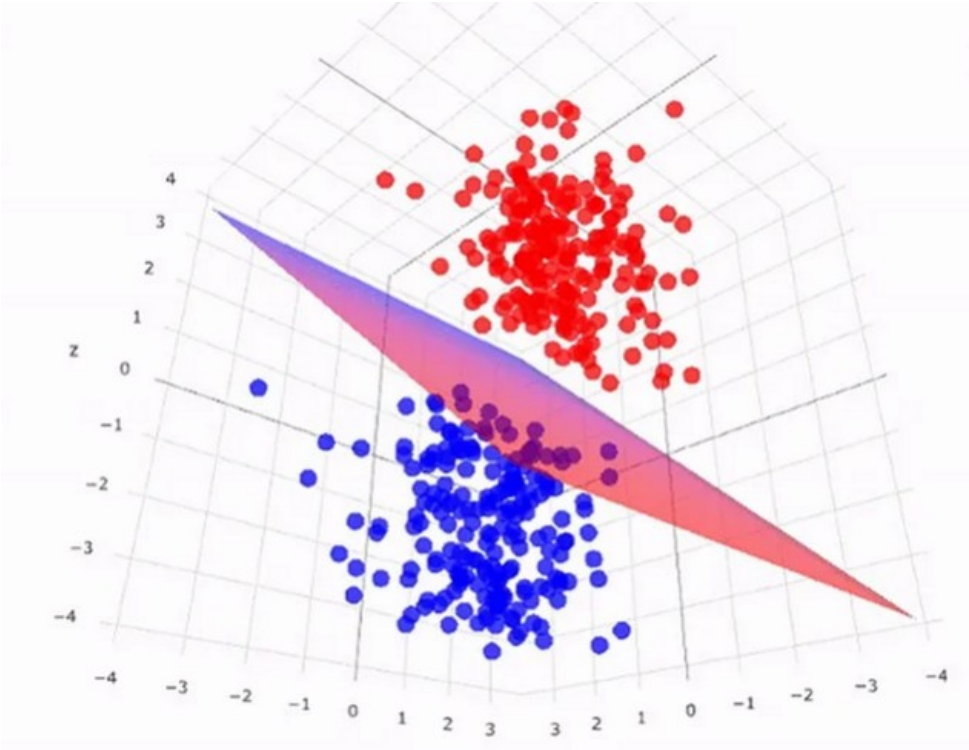

Рис. 1.2. Візуальне представлення машини опорних векторів (SVM)

SVM працюють, знаходячи оптимальну гіперплощину, яка розділяє точки даних на різні класи. Гіперплощина — це межа рішення, яка розділяє точки даних на різні класи у просторі великої розмірності. У двовимірному просторі гіперплощина — це просто лінія, яка розділяє точки даних на два класи. У тривимірному просторі гіперплощина — це площина, яка розділяє точки даних на два класи. Подібним чином у N-вимірному просторі гіперплощина має (N-1)-виміри. Його можна використовувати для прогнозування нових точок даних, оцінюючи, на яку сторону гіперплощини вони потрапляють. Точки даних з одного боку гіперплощини класифікуються як такі, що належать до одного класу, тоді як точки даних з іншого боку гіперплощини класифікуються як належні до іншого класу.

Виходячи з контексту користувача, обчислюється важливість тимчасового вікна, а відео використовується для запису ймовірної події для подальшого виявлення події. Таке використання даних датчиків є одним із видів застосування для виявлення подій і дозволяє швидко контролювати події в режимі реального часу.

Деякі з перших робіт для моніторингу активності використовували комбінацію ручних функцій, а не автоматизоване вилучення ознак для представлення даних. Нейронні мережі використовувалися переважно як механізм класифікації [11, 12].

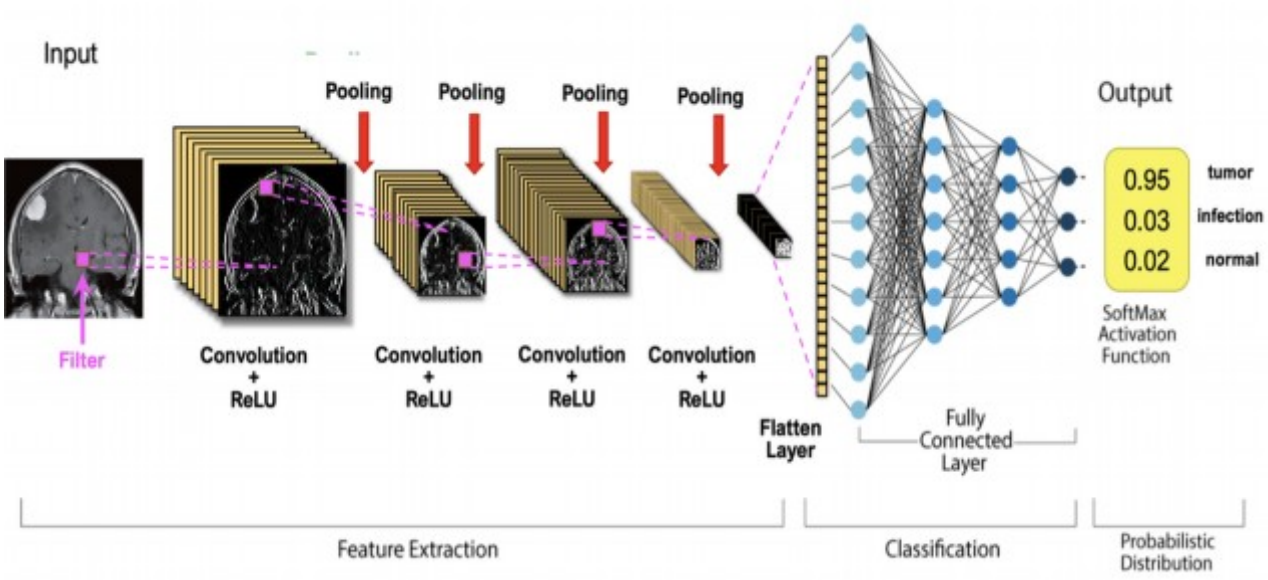

Рис. 1.3. Проста CNN для класифікації захворювань

Пізніше [13] з більшим успіхом було досліджено використання більших, глибоких мереж для автоматизованого виділення ознак і класифікації, ціною обчислювальної складності. Успіхи в згорткових нейронних мережах (CNN) і рекурентних нейронних мережах (RNN) призвели до появи безлічі фреймворків, які вирішують проблему безперервного моніторингу активності, хоча й у спосіб постобробки на серверах з більшою обчислювальною потужністю, ніж типовий край IoT. вузол.

На рисунку 1.4 показано типові кінцеві етапи обробки в CNN, які використовуються для класифікації/сегментації зображення. Векторизовані карти об’єктів пропускаються через кілька повністю пов’язаних шарів для створення числового вихідного вектора (Z). Функція Softmax перетворює ці вихідні числа на ймовірності.

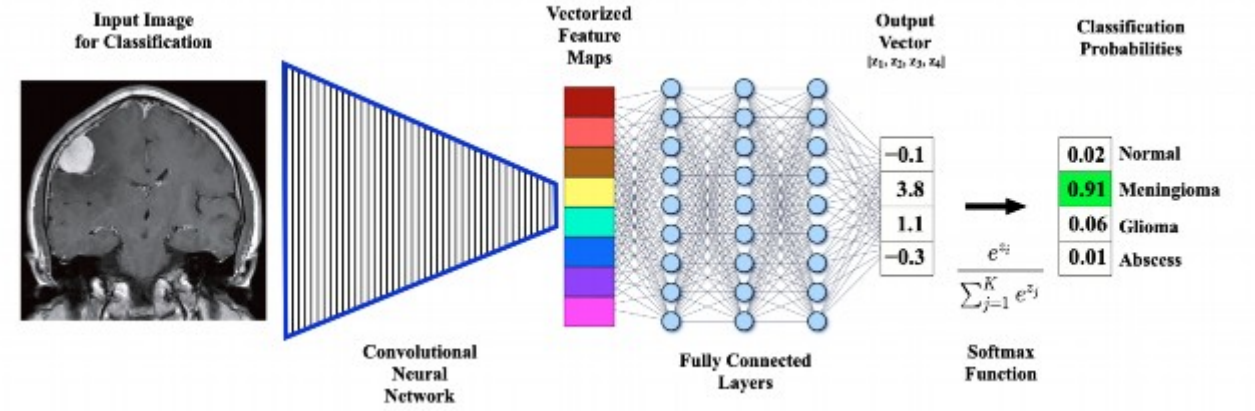

Рис. 1.4. Кінцеві етапи обробки в CNN, які використовуються для процесів класифікації/сегментації

Для більш детального огляду використання мереж глибокого навчання для безперервного моніторингу активності за допомогою повсюдних датчиків споживчого класу можна використати дослідження [14].

1.5. Аналіз впровадження глибоких нейронних мереж на платформах обмеженими обчислювальними можливостями

Хоча було досягнуто прогресу в успішному впровадженні CNN та інших алгоритмів машинного навчання на ПЛІС [15, 22], зусилля були в основному зосереджені на створенні платформ і фреймворків, які дозволяють плавний перехід від традиційного дизайну на основі CPU/GPU до платформи FPGA. Було кілька різних підходів до вирішення проблеми виконання CNN та інших алгоритмів машинного навчання традиційних платформ FPGA, таких як Xilinx Stratix і Virtex7, а також FPGA Altera Arria. Одним із підходів є розробка обчислювально менш інтенсивних варіацій CNN, таких як двійкові нейронні мережі (BNN) [23].

Деякі запропонували використовувати векторизацію для забезпечення уніфікованого представлення всіх операцій для оптимізації процесу

обчислень [22], а також для забезпечення квантування параметрів моделі [24]. Деякі застосували підхід, заснований на оптимізації, такий як реконфігуровані конструкції для потокових шляхів даних для різних рівнів у CNN [15], нові оптимізації для операцій згортки [20] та оптимізації циклів в операціях згортки [21]. Такі роботи здебільшого зосереджені на реалізації та оптимізації згорткових нейронних мереж на традиційних вбудованих платформах, апаратні характеристики яких відповідають масштабу ЦП, а також кількох графічних процесорів середнього рівня, які мають робочу пам’ять від 1 ГБ DDR3 RAM до навіть 48 ГБ ОЗП. Ці підходи виконують навчання мережі «ex-situ», не знаючи про цільову платформу та її обчислювальні обмеження.

З іншого боку, існує сімейство підходів, які націлені на процес навчання конкретної мережі, зважаючи на будь-які майбутні обчислювальні обмеження. Такі підходи в основному базуються на квантуванні та пропонують спосіб «стиснути» або зменшити розмір моделі без шкоди для точності мережі. Деякі загальні підходи полягають у зменшенні параметрів за допомогою скорочення [25], дистиляції знань [26, 27] і квантування параметрів [28]. Поняття навчання з урахуванням квантування [29, 30, 31] також набуло обертів, коли параметри моделі налаштовуються за допомогою ваг змішаної точності. Цей процес навчання дозволяє моделі вивчати компактні функції та пов’язані параметри, які інакше можуть бути втрачені під час квантування або скорочення після навчання.

Висновки до розділу

Перший розділ висвітлив важливі аспекти предметної області, пов’язаної з використанням концепцій Інтернету речей (IoT), зокрема в контексті екосистеми Internet of Medical Things (IoMT). Визначено основні проблеми, пов’язані з безперервним зростанням IoT-технологій у медичній сфері, що зумовлює потребу в ефективному управлінні величезними

обсягами даних, забезпеченні високого рівня безпеки та надійності систем, а також адаптації обчислювальних моделей для роботи на пристроях з обмеженими ресурсами.

Проаналізовано структуру та унікальні особливості екосистеми Internet of Medical Things, яка включає в себе різноманітні сенсори, медичні пристрої, програмні рішення та обчислювальні ресурси. Зазначено важливість забезпечення надійної комунікації та обміну даними між компонентами системи для підтримки безперервного моніторингу пацієнтів та автоматизованої обробки даних. Описано актуальні виклики, зокрема питання кібербезпеки, енергоефективності та забезпечення якості обслуговування (QoS) у IoT-системах. Розглянуто алгоритми машинного навчання, які сприяють ефективній обробці даних у IoMT, включаючи класифікацію, прогнозування та виявлення аномалій. Зокрема, обговорено переваги та обмеження різних методів, таких як SVM, нейронні мережі та методи глибокого навчання.

Загалом, проведене дослідження показало, що успішне впровадження машинного навчання та глибоких нейронних мереж у IoMT вимагає інноваційних рішень для подолання обмежень ресурсів і забезпечення високого рівня безпеки, а також інтеграції новітніх технологій для покращення якості обслуговування і доступності медичних послуг.

РОЗДІЛ 2. МЕТОДИ ТА МОДЕЛІ ЗАСТОСУВАННЯ МАШИННОГО НАВЧАННЯ В ОБЛАСТІ IoMT

2.1. Представлення концепції фізично неклонованих функцій

Фізично неклоновані функції (PUF) - це фізичні об'єкти, які використовують мікроскопічні варіації у фізичних властивостях для створення унікального "відбитка пальця". Ці варіації виникають в процесі виробництва і є непередбачуваними та неконтрольованими.

Фізично неклоновані функції [7] або фізичні випадкові функції — це втілена версія фізичних функцій, яка відображає зовнішній стимул (завдання) на випадкову, але повторювану відповідь. Фізична функція характеризується притаманною випадковістю, яка виникає під час виробничого процесу, і її майже неможливо відтворити, враховуючи поліноміальну кількість ресурсів. Характеристики моделі PUF найкраще виражаються через колекцію пар виклик-відповідь (CRP) і, отже, складають основу більшості, якщо не всіх, протоколів безпеки на основі PUF. PUF можна класифікувати на два типи на основі кількості дійсних, а саме слабкі і сильні [33]. PUF вважається слабким, якщо він має фіксований невеликий набір дійсних CRP, доступ до яких вважається обмеженим. Сильні PUF, з іншого боку, використовують велику кількість властивої непередбачуваності і, отже, мають велику кількість CRP. Також вважається, що вони мають незахищений фізичний інтерфейс і частіше використовуються в програмах безпеки.

Протягом багатьох років було представлено та оцінено багато моделей PUF. Таксономія PUF проілюстрована на рисунку 2.1. Загалом їх можна розділити на дві великі групи – моделі на основі затримки часу та моделі на основі пам’яті. Моделі, засновані на затримці часу, включають PUF кільцевих генераторів і PUF арбітрів або APUF і їх варіації, такі як PUF арбітра прямого зв’язку. Такі моделі PUF можуть генерувати підписи в режимі реального часу, специфічні для чіпа, без потреби у дорогій пам’яті

для зберігання ключів і, таким чином, особливо сприяють автентифікації пристрою, інтелектуальній власності та збереженню конфіденційності даних. З іншого боку, моделі PUF на основі пам’яті використовують варіації між узгодженими кремнієвими пристроями елементів пам’яті, щоб охарактеризувати притаманну випадкову функцію. Деякі поширені бістабільні елементи пам’яті, які використовуються для функцій PUF, це SRAM, засувки та тригери.

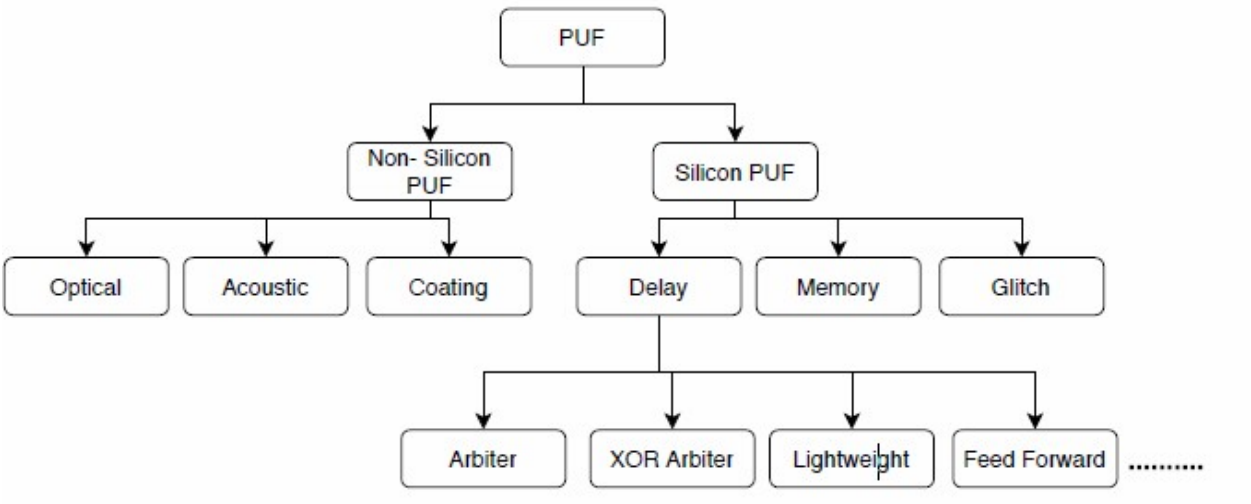

Рис. 2.1. Таксономія PUF

2.1.1. Надійні архітектури PUF

Потужний PUF може підтримувати велику кількість складних CRP з фізичним доступом до PUF для запиту, так що зловмисник не може створити правильну відповідь за обмежених ресурсів і часу [33]. У той час як слабкий PUF має лише кілька CRP, що ускладнює атаку та методи прогнозування. Вони в основному використовуються для зберігання секретних ключів, тому невідомі громадськості. У цій роботі ми розглядаємо сильний PUF. Кількість CRP сильних PUF може експоненціально зростати залежно від кількості модульних блоків, доступних для генерації відповідей на велику кількість відповідних викликів. Помилка через шум у відповіді PUF може бути мінімізована за допомогою допоміжних даних [36, 37]. Існують різні типи

міцного PUF, деякі з яких Arbiter PUF, XOR Arbiter PUF, легкий PUF і FeedForward PUF.

Arbiter PUF [32] є стандартною конфігурацією PUF, що складається з n послідовностей ступенів мультиплексорів. Два сигнали надсилаються одночасно через ці n ступенів, як показано на рисунку 2.2. Шлях визначається зовнішніми бітами для n етапів. Останній етап, що складається з довільного елемента фіксатора, визначає, який сигнал надійшов першим: верхній чи нижній, і його відповідний вихід, який дорівнює одиниці або нулю. Тут зовнішній біт вважається викликом, а результатом є відповідь Y.

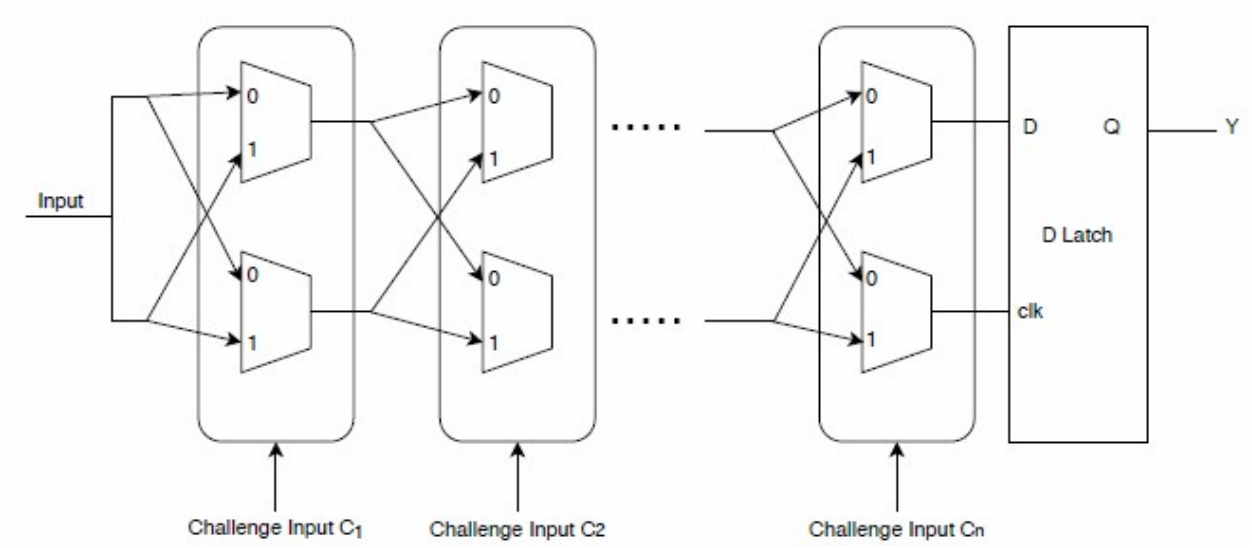

Рис. 2.2. Архітектура арбітражної фізично неклонованої функції

XOR арбітражна PUF [38] складається з n окремих арбітражних PUF з k стадіями. Однаковий виклик застосовується до обох арбітражних PUF, а відповіді об'єднуються операцією XOR для отримання єдиної вихідної відповіді. 2-стадійний XOR арбітражний PUF показано на рисунку 2.3. Залежно від n стадій, можуть бути різні XOR арбітражні PUF, що позначаються як n-XOR арбітражний PUF.

Легковажна PUF [39] подібна до XOR арбітражної PUF, має n паралельних стадій арбітражної PUF з k стадіями, кожна з яких виробляє

вихідний сигнал. Ці вихідні сигнали об'єднуються операцією XOR для отримання багатобітового виходу. Також, зовнішні біти або виклики можуть застосовуватися до n стадій арбітражної PUF.

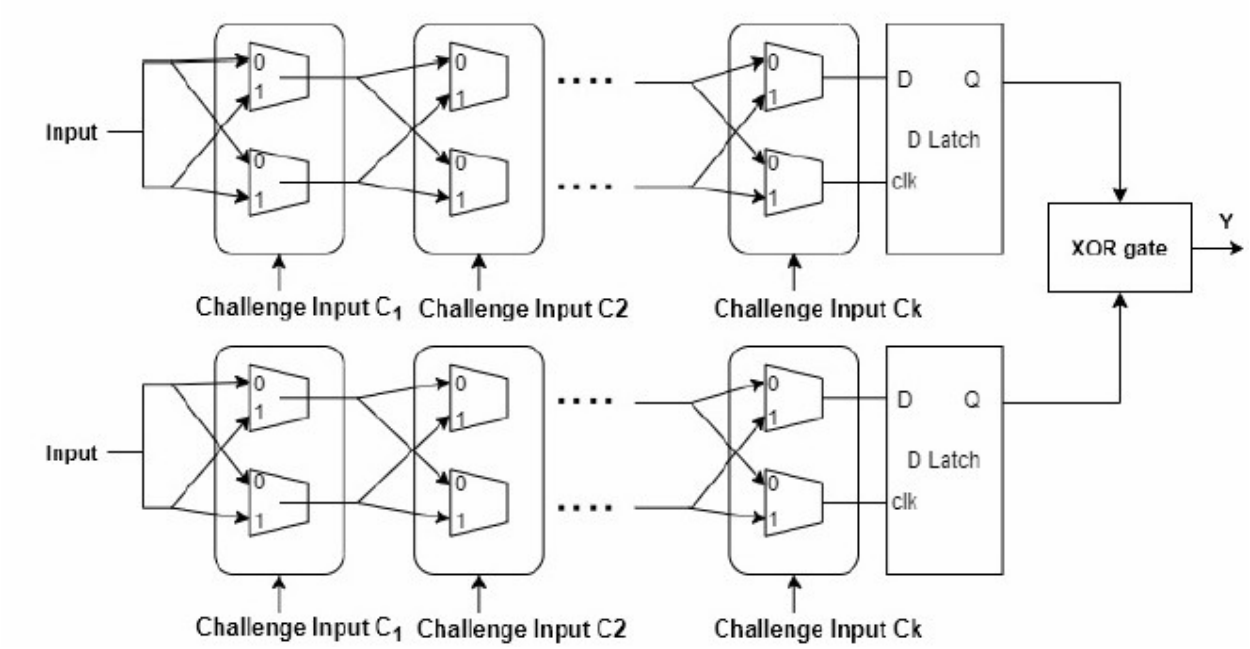

Рис. 2.3. Структура 2-XOR арбітражної фізично неклонованої функції

Притаманні CRP сильних PUF можуть експоненціально зростати залежно від кількості модульних блоків, доступних для генерації відповіді з великими можливостями відповідних викликів. Помилка, викликана шумом у реакції PUF, може бути мінімізована за допомогою допоміжних даних [36, 37]. Для повноти ми припускаємо, що такий механізм виправлення помилок, що включає варіації температури, напруги та старіння, вже присутній у PUF, який потрібно клонувати. Надійний PUF не містить схеми захисту від зчитування, припускаючи, що зловмисник повинен перерахувати велику кількість CRP. Отже, це робить інвазивну атаку нездійсненною, водночас спонукаючи зловмисника застосовувати методи, засновані на ML, щоб бути успішним за межами базової складності сильних PUF. Для детального аналізу конструкцій та опису міцних PUF можна скористатись [34].

Лінійна адитивна поведінка Arbiter PUF (APUF) зробила його ідеальною мішенню для ML-атаки. Отже, більш висока нелінійність у певній архітектурі PUF може покращити унікальність і випадковість із посиленим захистом від атак моделювання. Іншими підходами до стійких до ML PUF були рандомізовані тести [40], обфускація [41, 42] і тести на основі підрядків [43]. Також були запропоновані рандомізовані тести [40] на PUF і обфускацію PUF [41]. В [43] автор представив структуру перевірки-верифікатора для успішної автентифікації на основі підмножини підрядка відповіді. В [44] запропонували використовувати алгоритми пакетування та посилення ML для підвищення точності класифікаторів за умови достатньої ентропії каскадних PUF.

Більшість робіт, що описують ML-стійкі PUF, використовують чітко визначену архітектуру та достатньо великі CRP для процесу навчання. Випадковість і унікальність, навпаки, суттєво погіршуються, коли CRP, які не належать до оригінальних CRP для конкретного PUF, використовуються як випадок, який ми розглядаємо в цій роботі.

2.1.2. IoT безпека на основі фізично неклонованої функції

Фізичні неклоновані функції (PUF) все частіше пропонуються як основа для безпеки вузлів у структурі IoT [45, 46]. Безпека вузла IoT на основі PUF в основному була реалізована двома способами - автентифікація на основі CRP і генерація ключів на основі PUF [38]. В останньому випадку відповідь PUF зазвичай використовується для створення секретних ключів для використання в традиційній криптографії. Відповідь PUF на заданий виклик (оброблений через схему виправлення помилок) зазвичай хешується для генерації секретних ключів. Перший підхід, тобто автентифікація на основі CRP, ширше використовується, особливо з сильними моделями PUF, для створення надійних протоколів автентифікації. Отриманий протокол автентифікації передбачає оцінку ідентичності моделі PUF центральним сервером автентифікації шляхом застосування набору попередньо

визначених зовнішніх викликів і перевірки отриманої відповіді. CRP збираються на етапі реєстрації перед розгортанням, а отримана база даних формує основу автентифікації під час розгортання.

2.1.3. Протоколи шифрування для автентифікації вузла IoT

З використанням CRP для автентифікації вузлів IoT зросла потреба в протоколах шифрування через необхідність додаткового захисту від протоколів підслуховування. Використання протоколів шифрування в комунікації та автентифікації вузлів IoT зазнало суттєвого зростання. Підсумовуючи, використовувані протоколи шифрування — це стандарт шифрування даних (DES) і розширений стандарт шифрування (AES). Незважаючи на те, що криптоаналіз DES був успішним, для цього все ще потрібен надзвичайний обсяг обчислень і доступ до даних, тоді як не було успішної атаки на 128-бітний протокол шифрування AES. Незважаючи на те, що протоколи шифрування широко використовуються у зв’язку між вузлами Інтернету речей, для їх роботи потрібна певна схожість обчислень. Отже, були запропоновані інші протоколи для подолання такої обчислювальної потужності, такі як обфусцований CRP [42] і підрядок, щоб назвати декілька. У цій роботі ми розглядаємо протоколи шифрування AES і DES як механізми шифрування, що використовуються для шифрування CRP у рамках IoT.

2.1.4. Атаки на основі машинного навчання на моделі PUF

Широке впровадження моделей PUF у автентифікацію вузлів IoT призвело до збільшення підходів, які намагаються перевірити їх ефективність шляхом атаки або клонування моделі PUF. Клонування моделі PUF зазвичай включає підгонку складної математичної функції для фіксації кореляції між вхідним викликом і відповідною відповіддю PUF. Було декілька підходів, зокрема використання моделей машинного навчання та фізичного моделювання. Враховуючи зростання популярності автентифікації на основі PUF, було багато спроб перевірити ефективність підходу, насамперед за

допомогою математичного моделювання характеристичної функції PUF. В [39] запропонували атаку на основі ML на сильні PUF на основі прогнозної моделі. Автори змогли клонувати функціональність основного PUF, враховуючи модель PUF, оцінюючи параметри моделі за допомогою логістичної регресії (LR) з RProp і стратегіями розвитку (ES). Хоча метод був досить успішним у клонуванні, зловмисник повинен знати базову архітектуру PUF і відповідну функцію підпису. Хоча розумно припустити, що CRP можна отримати шляхом підслуховування або інших інтерфейсів [33], не завжди можливо визначити базову модель PUF без фізичного доступу до PUF. Хоча представлені атаки працюють краще за певного розміру PUF і складності архітектури, зловмисник повинен мати уявлення про базову архітектуру PUF, щоб згенеровані зразки клонів відповідали статистиці реальних CRP.

Інший тип підходу [46, 47] передбачає фізичний доступ до моделі PUF, окрім знань про архітектуру та модель PUF. Зазвичай вони передбачають використання підходів машинного навчання для моделювання реакції PUF шляхом використання фізичних характеристик, отриманих за допомогою підходів побічних каналів. Останнім часом зусилля були перенесені на комбінований ML і бічний канал (час і потужність), щоб представити покращену гібридну поверхню атаки [6]. У [39] була запропонована ML-атака без використання математичної моделі з використанням PAC (Probably Approximately Correct) . Автори представили, що впливовий біт, якщо він присутній у стабільній відповіді PUF, може передбачити майбутню відповідь, що відповідає виклику з низькою ймовірністю.

2.2. Представлення структури моніторингу діяльності на основі машинного навчання

У цьому розділі ми пропонуємо структуру моніторингу діяльності, засновану на самоконтрольованому навчанні репрезентації та алгоритмі

імовірнісної сегментації. Ми демонструємо , що фреймворк можна використовувати для виконання неконтрольованої сегментації стану сну. Ми обговорюємо, як цю структуру можна розширити для інших завдань, таких як моніторинг повсякденної діяльності на основі даних акселерометра.

Сон є життєво важливим процесом для підтримки здоров'я та благополуччя. Нестача сну, як за тривалістю, так і за якістю, є поширеною проблемою, і вражає понад багато мільйонів людей у всьому світі. Наслідки поганого або недосипу можуть мати серйозний вплив на повсякденне, здорове функціонування людського організму та можуть призвести до серйозних проблем, таких як ожиріння, діабет і хвороби серця. Однак, враховуючи важливість сну, основним способом перевірки та вимірювання якості сну є полісомнограма (PSG), дорогий та інвазивний тест, який може вимагати величезної кількості ресурсів, таких як лабораторія сну, лікар та обладнання для моніторингу фізіологічних знаки і доступні лише для оцінки обраних кількох ночей. Тривалий безперервний моніторинг не зовсім варіант. Однак поява парадигми Інтернету речей (IoT) уможливила розгортання недорогих споживчих пристроїв, які можуть збирати неінвазивну інформацію, таку як рух, частота серцевих скорочень тощо. Ці пристрої пропонують альтернативу тесту PSG і, отже, може забезпечити тривалий постійний моніторинг сну для підтримки здоров'я.

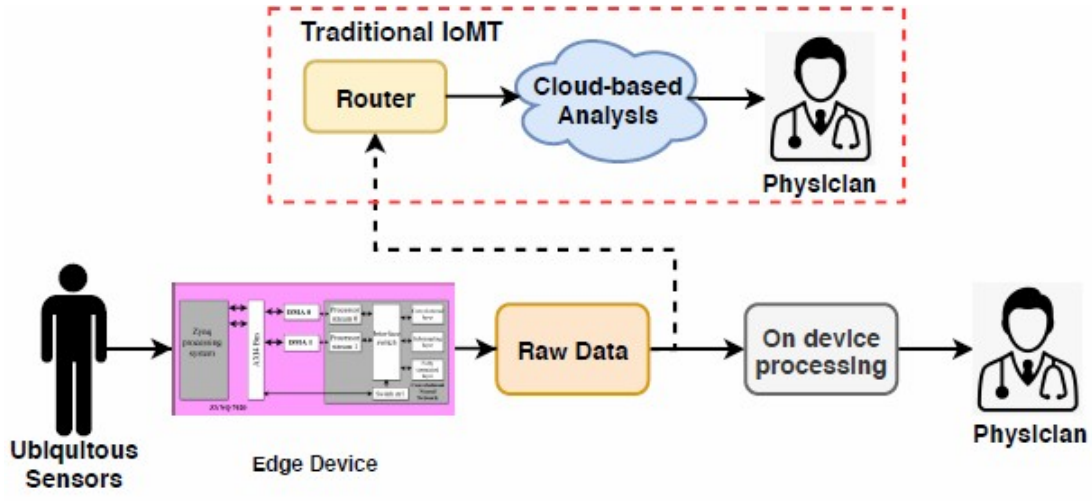

Рис. 2.4. Структура IoMT для безперервного моніторингу сну

Проте є деякі серйозні проблеми з постійним моніторингом стану сну. По-перше, дані часових рядів можуть мати як локальні, так і глобальні закономірності, які необхідно зафіксувати в навчених представленнях. По-друге, обсяг даних для аналізу може бути величезним, оскільки дані (акселерометр, частота серцевих скорочень тощо) збираються на дуже високих частотах і можуть призвести до сотень показань за хвилину. Нарешті, кожен із цих зразків даних має бути позначений експертами в цій галузі, що може бути дуже дорогим і важко отримати.

Однак у традиційних структурах IoT або, скоріше, IoMT (Інтернет медичних речей), датчики діють як вузли збору даних, які передають зібрані дані через безліч серверів, маршрутизаторів і мережевих шлюзів, перш ніж вони будуть оброблені в автономному режимі та діючі правила та можна отримати заявки. Ця передача інформації часто здійснюється через бездротові мережі. Це створює певні проблеми безпеки, такі як підслуховування та атаки типу "людина посередині", які є досить поширеними проблемами безпеки і, отже, вимагають надійних протоколів безпеки для забезпечення цілісності переданих даних.

Проблема розпізнавання та сегментації стану сну була значною мірою вирішена за допомогою навчання з учителем [3, 4]. Ці підходи, хоч і добре працюють, вимагають великої кількості експертно-анотованих, позначених даних, отримання яких може бути дуже дорогим. Крім того, збір такої кількості даних передбачає захоплення та передачу приватної конфіденційної інформації, такої як характеристики руху, місцезнаходження, частота серцевих скорочень тощо, і може спричинити серйозні проблеми з безпекою в разі порушення безпеки. З прогресом у машинному навчанні та апаратній безпеці цілісність даних може бути порушена, а постійний збір дуже конфіденційної інформації може спричинити проблеми. Децентралізоване навчання [5] з’явилося як життєздатна альтернатива, але вимагає швидких і ефективних алгоритмів навчання, які можуть працювати в масштабі та на пристрої, у тому числі на крайових вузлах з обчислювальними обмеженнями.

Крім того, анотацію таких великомасштабних даних може бути важко отримати, і вона не сприяє масштабуванню для мільйонів користувачів. Отже, зростає потреба в самоконтрольованих або неконтрольованих підходах до сегментації стану сну.

У цій роботі ми досліджуємо проблему неконтрольованої сегментації стану сну в потокових даних за допомогою самоконтрольованого навчання. Ми пропонуємо двосторонній підхід до проблеми. По-перше, ми пропонуємо використовувати безперервне прогнозне навчання, щоб навчитися надійному часовому вбудовуванню потокових даних за допомогою самоконтролю. По-друге, ми пропонуємо неконтрольований ймовірнісний алгоритм для сегментації безперервного потоку рухів і даних фотоплетизмографії про частоту серцевих скорочень на його складові стани сну без будь-яких позначених даних.

Існувало декілька підходів до розпізнавання та сегментації стану сну за допомогою даних руху та фотоплетизмографії. Загальний підхід [3, 4] використовує контрольовані алгоритми машинного навчання, які використовують функції контексту, руху та фотоплетизмографії для класифікації попередньо сегментованих епох сну на різні стани сну. Такі підходи розглядають задачу класифікації стану сну , яка припускає, що дані часових рядів уже попередньо сегментовані на епохи, і, отже, завдання обмежене розпізнаванням і не включає сегментацію довгих даних часових рядів. З іншого боку, ми вирішуємо проблему сегментації стану сну і, отже, повинні як тимчасово локалізувати , так і розпізнавати стан сну.

В роботі представляється структура сегментації без нагляду, яка може сегментувати довгі, необрізані послідовності даних про рух і частоту серцевих скорочень без будь-які мітки. Ми показуємо, що підхід до безперервного навчання можна використовувати в децентралізованих структурах навчання для підвищення безпеки та конфіденційності даних, і запропонована структура є достатньо легкою та може бути реалізована на обмежених платформи, не замінюючи точністю на затримку,

використовуючи поточні досягнення апаратних прискорювачів для глибокого навчання.

Решта цього розділу організована таким чином. Спочатку ми представляємо запропоновану структуру сегментації стану сну. Потім ми надаємо кількісний і якісний аналіз нашого підходу на даних акселерометра.

2.3. Моделі та алгоритми сегментації процесів в системі IoMT в контексті машинного навчання без вчителя

У цьому розділі ми представляємо запропоновану структуру сегментації стану сну. Ми представляємо підхід до вивчення репрезентації ознак для захоплення часових репрезентацій та алгоритм неконтрольованої сегментації.

2.3.1. Навчання репрезентації локальних часових ознак

Дані часових рядів, особливо дані про прискорення та фотоплетизмографію частоти серцевих скорочень, можуть бути дуже довгими часовими послідовностями. Тому важливо вивчити надійні представлення локальної часової структури даних. У цій роботі ми називаємо дані, що містяться в одній секунді, як репрезентативні локальної часової структури. Оскільки дані руху (прискорення) і фотоплетизмографії можуть бути дуже стохастичними, інтервал в одну секунду дозволяє нам зафіксувати базові незмінні послідовності шаблони за допомогою необроблених вхідних даних. Ми вивчаємо локальні представлення за допомогою фреймворку автокодувальника, архітектури нейронної мережі, навченої вивчати абстрактні, стислі представлення за допомогою механізму кодування-декодування. Щоб використовувати досягнення в прискоренні згорткової нейронної мережі (CNN) на обмежених платформах [15, 19], ми навчаємо згортову мережу автокодувальника з двома мережами (кодувальником і декодером), які працюють у тандемі, щоб вивчати закодоване представлення

називають латентним простором. Кодер навчається стискати вхідні дані в абстрактне закодоване представлення, яке фіксує базовий шаблон у вхідних даних. Мережа декодера навчена реконструювати оригінальний вхідний сигнал із цього закодованого представлення. Мережа автокодувальника навчена мінімізувати втрати від реконструкції, які оформляються як різниця L2 між реальними та реконструйованими даними. Мережа кодера складається з трьох блоків згортки, кожен з яких складається з двох шарів згортки. Блоки згортки перемежовуються шарами максимального об'єднання та пакетної нормалізації для зменшення розмірності та покращення стабільності навчання. Оскільки дані часового ряду не містять просторового виміру, ми використовуємо одновимірні згортки для захоплення інваріантних до послідовності представлень вхідної послідовності.

2.3.2. Захоплення довгих часових залежностей

У той час як представлення, отримані за допомогою каркаса автокодувальника, охоплюють короткочасні залежності, їх недостатньо для охоплення часових залежностей у вхідних даних фотоплетизмографії, які можуть поширюватися на кілька годин даних. Також неможливо збільшити розмір вхідного вікна для мережі кодера без збільшення кількості параметрів у мережах кодера та декодера, які можуть збільшуватися нелінійно зі збільшенням вхідної розмірності. Таким чином, ми фіксуємо довгострокові тимчасові залежності за допомогою системи прогнозного навчання, використовуючи предиктор на основі довгострокової короткочасної пам'яті (LSTM). Ми навчаємо мережу предикторів безперервно передбачати характеристики вхідних даних на наступному часовому кроці. Потім мережа декодера навчається (точніше, точно налаштовується) для реконструкції прогнозованих даних. Це показано на рисунку 2.5.

Прихований стан мережі предиктора використовується як часове вбудовування для часу t, оскільки він фіксує минулі часові залежності та сильно впливає на прогнози мережі LSTM. Це відрізняється коли навчають

мережу передбачати майбутній простір функцій на відміну від фактичних даних.

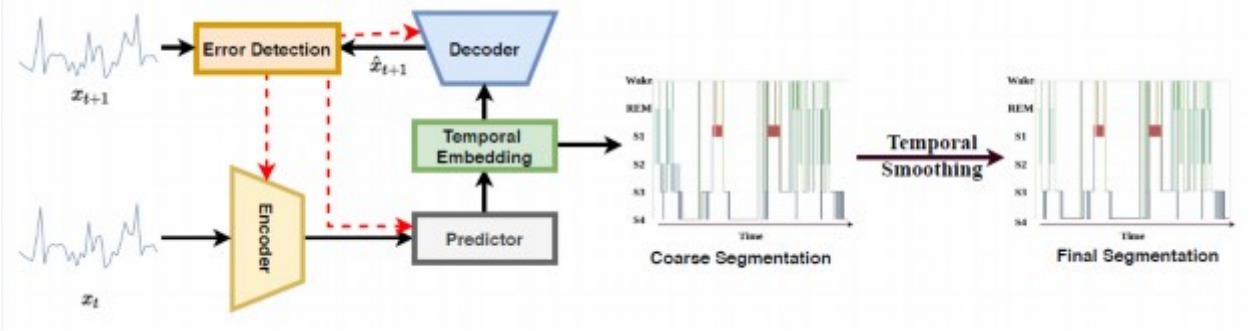

Рис. 2.5. Запропонована структура сегментації сну, що дозволяє зменшення шуму в результатах сегментації та забезпечення узгодженості

Помилка реконструкції (оформлена як помилка суми квадратів між прогнозованими та спостережуваними даними) використовується для навчання кодера, декодера та предиктора мережі. Таким чином, мета навчання визначається

$$\underset{\theta_e,\theta_d,\theta_p}{\operatorname{argmin}} \sum_{i=1}^{n} \|x_{t+1} - \hat{x}_{t+1}\|_{\ell_1}^2$$

Де θ представляють параметри мереж кодера, декодера та предиктора; x_{t+1} є спостережуваними та прогнозованими входами в момент часу t + 1 відповідно.

Причина, чому ми використовуємо прогнозне навчання , хоча доступні багато інших методологій вбудовування, полягає в тому, що підхід до прогнозованого навчання пропонує три основні переваги:

1) він дозволяє навчатися на потокових даних у реальному часі та не вимагає від нас зберігання великих обсягів даних, що може призвести до проблем із безпекою та конфіденційністю,

2) стек передбачень можна зробити таким, щоб бути легким і містити менше параметрів для оптимізації, що є важливим для обробки в режимі реального часу на обмежених платформах, доступних у типовій структурі IoT,

3) підхід передбачуваного навчання природним чином сприяє децентралізованому навчанню [5], коли кожен екземпляр мереж можна навчити окремо та об’єднати пізніше для покращення продуктивності.

Останнім кроком у нашій структурі є сегментація стану сну в потокових даних, яка використовується для сегментації даних часових рядів у сегмент на основі стану сну. Ми використовуємо часові вбудовування, створені через мережу, описану в попередньому підрозділі і забезпечуємо початкову грубу сегментацію, призначаючи кожен часовий крок (інтервали в одну секунду) k кластерам. Кожен кластер представляє кожен стан сну, наприклад неспання, нешвидкі рухи очей (NREM), швидкі рухи очей (REM) тощо. Ми визначаємо це як грубу сегментацію, оскільки вона не враховує часову когерентність, дійсний стан переходів і невизначеності класифікації врахов.

Ми виконуємо цю класифікацію шляхом моделювання моделі Гауса (GMM). Ми оптимізуємо GMM за допомогою очікування-максимізації [79], і отримана класифікація є k-вимірним вектором з ймовірністю того, що вхідні дані належать до кожного з k класів. Це відрізняється від кластеризації k-середніх, яка забезпечує жорстке присвоєння й не повертає жодних ймовірностей, а також дозволяє нам зафіксувати будь-яку невизначеність у класифікації.

Використання грубої сегментації було б наївним підходом, який може призвести до неоднорідних кластерів станів сну. Щоб полегшити це, ми запровадили функцію тимчасового згладжування, яка дозволяє представити узгоджену в часі однорідну сегментацію вхідної послідовності. Заснований на алгоритмі Вітербі, процес згладжування є імовірнісним алгоритмом тимчасового зв’язування, який приймає грубу сегментацію як вхідні дані та

виводить остаточну сегментацію. Позначимо і-й кластер з часу t через dt . Ми обчислюємо показник тимчасової спорідненості між двома часовими кроками dt і jt +1 як

$$S_c(d_t, d_{t+1}) = (1-\beta)E_c(d_t) + \beta E_c(d_{t+1}) + \psi_{d_t,d_{t+1}}$$

де Ес - оцінка класу довіри даного кроку часу, в є фактором тимчасової пам’яті і є ймовірністю перемикання міток між dt і dt+1. Моделюємо φ() як похибка передбачення, визначена в попередньому рівнянні між часовими кроками t і t+1. Часові кроки з максимальною сумарною тимчасовою спорідненістю об’єднуються, щоб сформувати новий кластер, і їх оцінки ймовірності усереднюються. Алгоритм згладжування застосовується n = 15 разів або до тих пір, поки менше ніж 5% кроків часу не змінять свої мітки.

2.3.3. Деталі архітектури та реалізації мережі

Загальна структура запропонованого тимчасового автокодувальника проілюстрована на рисунку 2.6.

Рис. 2.6. Пропонована архітектура мережі автокодера. Фреймворк призначений для захоплення локальних часових шаблонів у вхідних даних

Ми використовуємо досягнення архітектури VGG-Net і використовуємо невеликі фільтри (1-D фільтри з сприйнятливим полем 3 і

висотою 1) для згортання в часі. Ми ділимо мережу на блоки згорткових шарів, які перемежовуються шарами максимального об’єднання.

VGG-Net (або VGG) - це тип згорткової нейронної мережі (CNN), розроблений в Оксфордському університеті групою Visual Geometry Group (звідси і назва VGG). Ця архітектура стала відомою завдяки своїй простоті та ефективності в задачах класифікації зображень, особливо після того, як вона досягла передових результатів на конкурсі ImageNet у 2014 році.

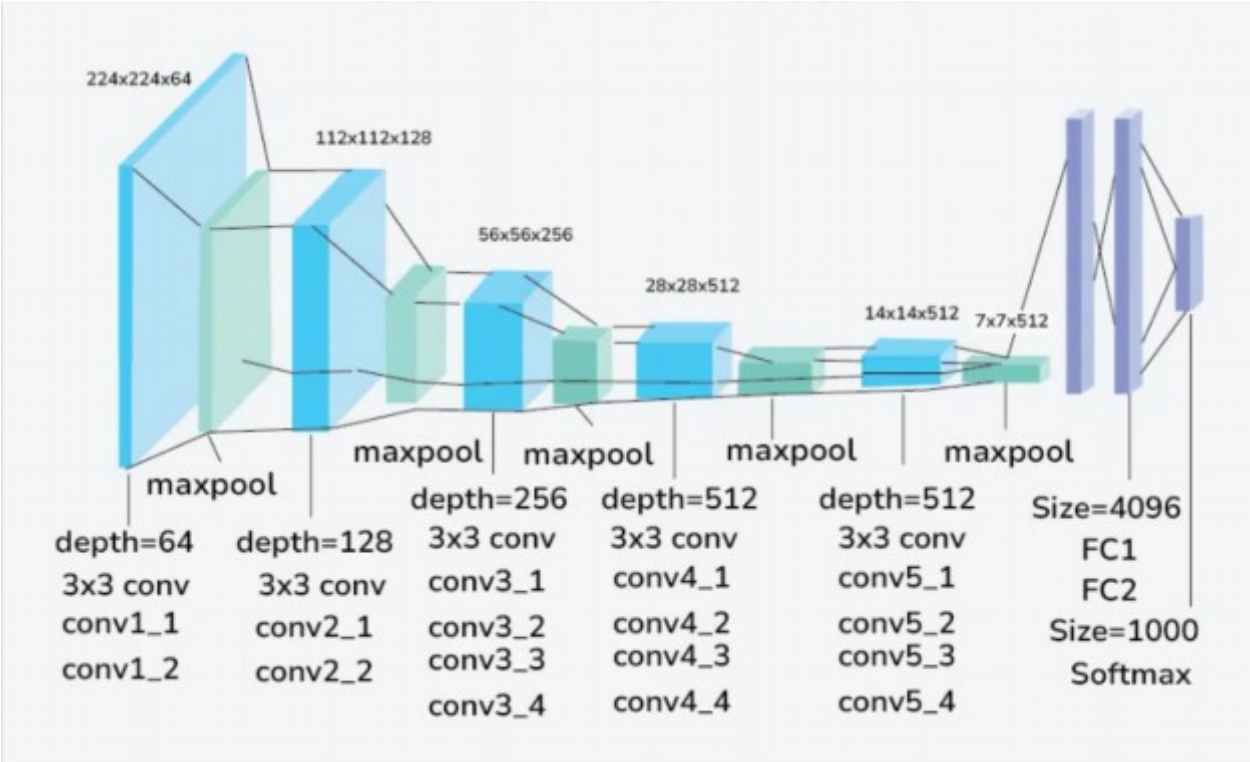

Рис. 2.7. Архітектура VGG

Архітектура мережі тісно ґрунтується на архітектурі VGG-11, яка, як ми вважаємо, підходить до платформ з обмеженими обчисленнями. Ми поступово зменшуємо розмірність вхідних даних за допомогою тимчасового об’єднання, яке реалізується як традиційна операція об’єднання з висотою 1. Мережа декодера є дзеркальним відображенням мережі кодера. Ми попередньо навчаємо мережу кодера та декодера реконструювати вхідний сигнал, який відбирається щосекунди, що становить приблизно 45 показань акселерометра. Ми також випадково додаємо деякий шум у вхід кодера, щоб

допомогти зробити мережу інваріантною до будь-яких тимчасових збурень через помилки читання та калібрування. Це дозволяє мережі вивчати надійні уявлення, які дозволяють більш послідовні прогнози. Мережа предикторів — це мережа LSTM із прихованим розміром 128. Ми не використовуємо жодних механізмів уваги, оскільки метою є авторегресивний прогноз. Ми виявили, що додавання механізмів уваги шкодить часовій здатності передбачати мережу LSTM. Цей ефект може віднести до функції втрат, яка сильно штрафуватиме будь-які відхилення від вхідних даних, тоді як механізм уваги змусить мережу зосередитися на найбільш дискримінаційних частинах.

2.3.4. Реалізація на обмежених платформах у рамках IoT

Хоча можливість сегментації підходу є важливою, ми також повинні дозволити ефективну реалізацію мережі на обмежених платформах, що є типовим для більшості обчислювальних вузлів у рамках IoT. Щоб врахувати обчислювальні обмеження, ми робимо деякі важливі зміни в дизайні та налаштовуємо реалізацію в нашій структурі. Перший і найважливіший — це використання 2-D згорткових ядер у реалізації мережі. Хоча дані є одновимірними, а операції одновимірної згортки ефективніші, ми виявили, що існуюча робота над апаратними прискорювачами для глибокого навчання в основному зосереджена на 2D CNN. Отже, ми апроксимуємо операцію одновимірної згортки, зробивши висоту ядра згортки рівною 1. По-друге, ми виявили, що використання шарів об’єднання має значний вплив на споживання енергії на платформі PYNQ, коли об’єднання операція вимагає доповнення для виконання. Таким чином, ми гарантуємо, що розміри перед шаром об’єднання є такими, що для його виконання не потрібно жодних доповнень. Нарешті, ми спостерігали пряму залежність між затримкою мережі та пропускною здатністю повністю підключених рівнів. Отже, ми наближаємо обчислення повністю зв’язаних шарів шляхом маніпулювання (наприклад, зміни форми) вивчених параметрів повністю зв’язаного шару

(вагова матриця W) у згорткові фільтри. Це дозволило нам зменшити обсяг пам’яті та затримку в 2 і 1,75 рази відповідно. Ми виконуємо таке ж наближення з внутрішньою механікою шару LSTM.

2.4. Оцінка та аналіз пропонованої структури

У цьому розділі ми представляємо оцінку запропонованої структури. Ми проводимо наші експерименти на основі даних набору взятих з ресурсу Kaggle. Дані містять дані акселерометра та фотоплетизмографії, зібрані від 39 учасників, які використовували Apple Watch протягом восьмигодинного періоду сну. Суб'єкти були поміщені на 7-14-денний період амбулаторного обліку. Дані включають дані про рух у вигляді показань акселерометра та частоти серцевих скорочень, отриманих за допомогою фотоплетизмографії, вбудованої в Apple Watch. Ми використовуємо дані акселерометра (в одиницях g) і частоти серцевих скорочень, щоб розділити дані на стани сну, що входять до них. Ми перевіряємо акселерометр щосекунди, що призводить до приблизно 50 показань на секунду. Для ефективного обчислення та підтримки рівності для всіх послідовностей ми відбираємо 48 показань за кожен момент часу. Для моментів часу з менш ніж 48 показаннями ми інтерполюємо показання, щоб отримати задану кількість показань за секунду.

Оцінка виконується за допомогою перехресної перевірки Монте-Карло, як описано в [3]. Ми повідомляємо точність як показник оцінки за крок часу (інтервал в одну секунду). Це відрізняється від протоколів оцінки в [3], де оцінка виконується за епоху сну, яка зазвичай триває 30 секунд і вимагає висококваліфікованих експертних анотацій. Наше налаштування є більш реалістичним налаштуванням оцінки безперервної сегментації стану сну на споживчих пристроях, які не мають таких експертних коментарів. Ми оцінюємо за кількома контрольованими базовими лініями, такими як логістична регресія, випадковий ліс, нейронні мережі, а також неконтрольованими базовими лініями, такими як k -середні. Слід зазначити,

що ми використовуємо ті самі самоконтрольовані функції як вхідні дані для кожного з алгоритмів, параметри яких є єдиною метою навчання на етапі тонкого налаштування.

Таблиця 2.1.

Ефективність різних підходів до завдання сегментації сну

Supervision	Approach	Accuracy (%)	Wake Correct (%)	Sleep Correct (%)
Full	LR	0.71	**0.42**	**0.95**
	RF	0.67	0.38	0.92
	NN	**0.72**	0.41	0.94
None	k-means	0.59	0.3	0.89
	GMM	0.54	0.23	0.87
	GMM + TS	**0.63**	**0.31**	**0.91**

З таблиці 2.1 видно, що найкраща точність для GMM+TS і LR становить 81,8% і 85,6% відповідно. Рівень відкликання становить 80,5% і 83,7% відповідно.

Ми оцінюємо наш підхід та різні базові алгоритми на двох різних завданнях. Перше завдання полягає в сегментації потокових даних на дві категорії - сон та неспання. Тут мета полягає в тому, щоб правильно визначити відрізки часу, коли людина не спить, і коли людина спить. Ми підсумовуємо результати в таблиці 2.1. Ми класифікуємо підходи на дві категорії залежно від того, скільки нагляду (тобто позначених даних) вимагає підхід. Якщо підхід вимагає позначених даних, то він називається таким, що вимагає «повного нагляду», і «без нагляду» в іншому випадку. Видно, що використання нашого тимчасового вбудовування дозволяє повністю контрольованим базовим лініям працювати дуже добре і звужує розрив між повністю контрольованими та неконтрольованими підходами. Видно, що використання тимчасового згладжування значно покращує продуктивність підходу на основі GMM. Наша найкраща продуктивність становить 96,2% точності для пацієнта з ідентифікатором 0, а наша найкраща продуктивність

на набір оцінювання (10 пацієнтів) становить 81,8%, що є найсучаснішим у задачі сегментації сну, особливо на обчислювально обмежених платформах. Коефіцієнт відкликання становить 83,7%. Ми також оцінюємо підходи до набагато складнішого завдання сегментації на три класи - неспання, сон NREM та сон REM. Ми підсумовуємо результати в таблиці 2.2.

Таблиця 2.2.

Ефективність різних підходів до завдання сегментації сну проти NREM проти REM. TS відноситься до тимчасового згладжування

Supervision	Approach	Wake Correct (%)	NREM Correct (%)	REM Correct (%)
Full	LR	**0.57**	**0.51**	**0.53**
	RF	0.53	0.40	0.44
	NN	0.54	0.39	0.5
None	k-means	0.49	0.51	0.37
	GMM + TS	0.50	0.42	0.33
	GMM	**0.51**	**0.44**	**0.43**

Це набагато складніше завдання, оскільки характеристики сну REM та NREM дуже тісно пов'язані та можуть спричинити велику невизначеність у задачі класифікації. Ми отримуємо нашу найкращу точність (78,9%) для пацієнта з ідентифікатором 0, причому наша найкраща продуктивність на набір оцінювання (10 пацієнтів) становить 63,2%, що є найсучаснішим у задачі сегментації неспання проти REM проти NREM сну. Коефіцієнти відкликання становлять 76,9% та 70,3% для задач сегментації неспання проти сну та неспання проти REM проти NREM відповідно. Ми повідомляємо про точність на пацієнта на рисунку 2.8, відсортованому за точністю. Крім того, дані для цього завдання є дуже незбалансованими та вимагають ретельного налаштування процесу навчання для отримання хороших результатів. Ми гарантуємо, що на кожній епосі навчання існує збалансований набір зразків для всіх класів. Видно, що репрезентації, отримані за допомогою

прогнозного навчання, забезпечують хороші результати та ще більше звужують розрив у продуктивності між контрольованими та неконтрольованими методами.

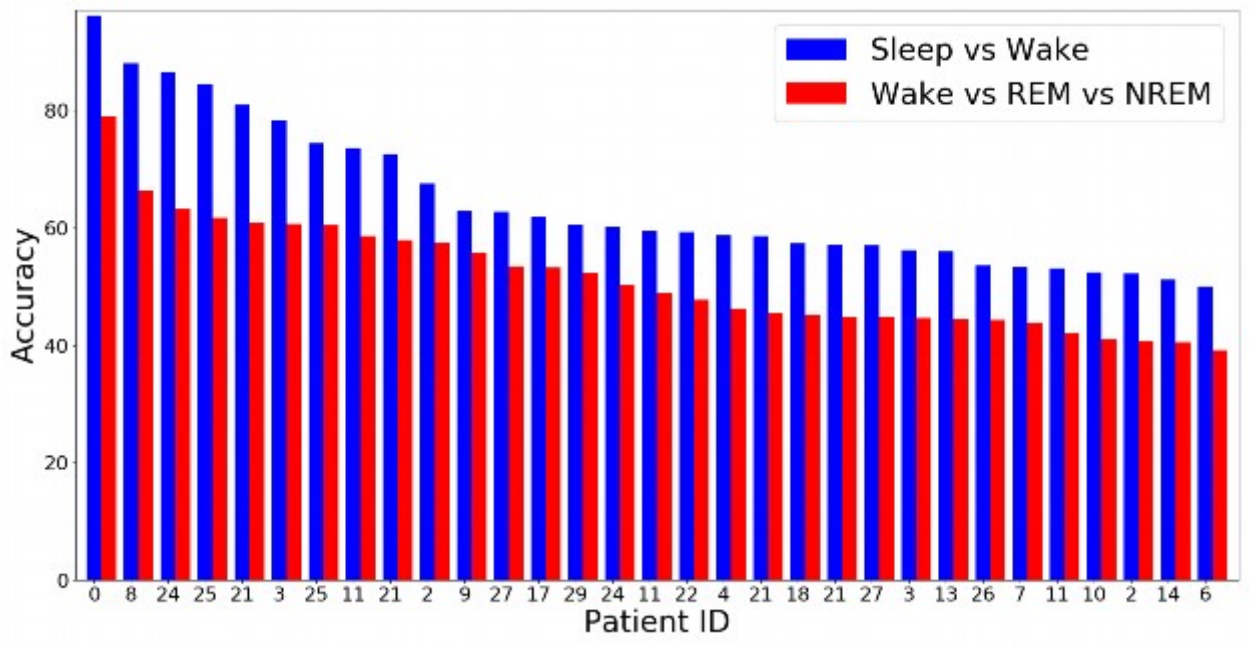

Рис. 2.8. Порівняння точності для окремих пацієнтів для обох завдань

Різниця між контрольованими та неконтрольованими моделями, на яку слід звернути увагу, полягає в тому, що існує значний розрив між контрольованими підходами (приблизно 9% у задачі неспання проти сну та приблизно 6% у задачі неспання проти NREM проти REM). Однак це компроміс між кількістю дорогих, анотованих експертами даних, необхідних для навчання цих моделей. Однак використання самоконтрольованого тимчасового вбудовування може значно зменшити цей розрив у продуктивності. Вихідні дані неконтрольованих моделей можна використовувати для попереднього навчання контрольованих моделей для зменшення надмірної залежності від позначених навчальних даних, і це є активною областю досліджень.

Висновки до розділу

У цьому розділі розглядається поточна робота з проектування та впровадження різних компонентів у структурі Інтернету речей, таких як програми, впровадження крайових вузлів і забезпечення безпеки. Узагальнюються існуючі підходи до безперервного моніторингу активності засобами IoT. Досліджуються різні механізми, що застосовуються для впровадження та прискорення глибоких нейронних мереж на платформах з обмеженими ресурсами. Представляються основні механізми фізично неклонованих функцій (PUF) і різні архітектури. Розглядається поточна робота з інтеграції PUF у протоколи автентифікації у фреймворках IoT та аналізується їх сприйнятливість до різних атак клонування, включаючи атаки на основі машинного навчання.

Описується неконтрольований алгоритм сегментації стану сну, що використовує парадигму самоконтрольованого тимчасового навчання функцій, яка не потребує мічених даних. Рамка прогнозованого навчання сприяє децентралізованому, федеративному навчанню, що допомагає зберігати конфіденційність і зменшувати навантаження на мережу. Показано, що підхід до неконтрольованої сегментації є конкурентоспроможним із повністю контрольованими базовими лініями та перевершує деякі з них. Намічено вдосконалення можливостей навчання функцій системи прогнозованого навчання для зменшення розриву між повністю контрольованими та неконтрольованими базовими лініями й інтеграції в децентралізовану структуру навчання з використанням кількох крайових вузлів IoT.

РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ ЗАГОРТКОВИХ МЕРЕЖ ТА МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ЗАСТОСУНКІВ ІНТЕРНЕТУ РЕЧЕЙ

3.1. Особливості використання загорткових нейронних мереж

У цьому розділі ми аналізуємо властивості різних шарів глибоких нейронних мереж та пропонуємо оптимізації часу проектування для реалізації архітектури, запропонованої в другому розділі, на платформах з обмеженими ресурсами, таких як плата PYNQ. Ми використовуємо унікальні властивості цієї платформи та, завдяки точному вибору дизайну та оптимізаціям, реалізуємо висококомпактні та ефективні версії чотирьох поширених архітектур ЗНМ, які можна розширити на різні моделі глибокого навчання, такі як LSTM, на додаток до ЗНМ.

Реконфігуровані обчислювальні архітектури, такі як програмовані логічні інтегральні схеми (ПЛІС) та обчислювальні системи на основі ПЛІС, стали можливою платформою для навчання на кристалі в периферійних пристроях Інтернету речей. Ці пристрої часто мають низьке енергоспоживання, але мають обмежені можливості паралельної обробки порівняно з більш традиційними платформами машинного навчання, такими як графічні процесори (GPU) робочих станцій або багатоядерні процесори. Тим не менш, мало досліджень у літературі прагнули розробити та оптимізувати складні завдання машинного навчання на платформах з обмеженими ресурсами, і тому більшість досліджень повідомляють про дуже високу продуктивність та точність висновків за рахунок високого енергоспоживання, значних вимог до пам'яті та великого розміру пристроїв [15, 19], що призводить до відносно низької енергоефективності та ефективності використання пам'яті. Замість цього ми пропонуємо оптимізацію дизайну для чотирьох архітектур згорткових нейронних мереж (ЗНМ) та представляємо результати на невеликій гетерогенній

обчислювальній платформі з низьким енергоспоживанням, яка є репрезентативною для систем, які можуть використовуватися в обмежених пристроях Інтернету речей.

ЗНМ є обчислювально та ресурсоємними, містять багато параметрів, що навчаються, часто в діапазоні від десятків до сотень мільйонів, і виконують кілька складних класів завдань над заданим входом, таких як двовимірні згортки та субдискретизація. Враховуючи масштаб обчислювальних вимог та вимог до пам'яті, а також бажання реалізувати глибокі нейронні мережі на кристалі для швидшого часу відгуку системи, зменшення вимог до пропускної здатності мережі та переваг конфіденційності, критично важливо дослідити архітектури мереж та оптимізації в реалізації, що підходить для середовищ з низьким енергоспоживанням та обмеженими ресурсами, без шкоди для точності прогнозування чи класифікації.

Програмовані логічні інтегральні схеми (ПЛІС) піддаються реалізації різних алгоритмів машинного навчання та можуть забезпечити покращення енергоспоживання та продуктивності порівняно з процесорами загального призначення. Однак відображення програмних реалізацій на ПЛІС за допомогою високорівневого синтезу (HLS) є значним викликом. Отримання максимальних переваг у енергоспоживанні та продуктивності часто вимагає додаткових вказівок, які автоматизовані інструменти самі по собі не можуть забезпечити. Середовище PYNQ від Xilinx пропонує альтернативу традиційному робочому процесу ПЛІС, замість того, щоб використовувати гетерогенне середовище обробки для сценаріїв Python. Кілька недавніх робіт досліджували реалізацію алгоритмів навчання та аналітики великих даних на платформі, включаючи Spark, виявлення країв [26, 27], обробку відео [28] та рекурентні нейронні мережі [29].

У цій роботі ми використовуємо унікальні властивості цієї платформи та, завдяки точному вибору дизайну та оптимізаціям, реалізуємо висококомпактні та ефективні версії чотирьох поширених архітектур ЗНМ:

LeNet, AlexNET, VGG-11 та VGG-16. Реалізації перевіряються за допомогою стандартних наборів даних, MNIST [90] та CIFAR-10 [91], а для точної оцінки енергоефективності використовується джерело живлення з високою точністю вимірювання.

Отже, в даному розділі:

- Аналізуються властивості різних типів шарів ЗНМ щодо енергоспоживання, обсягу пам'яті та затримки.

- Описуються чотири оптимізовані реалізації ЗНМ на гетерогенній обчислювальній платформі з низьким енергоспоживанням та демонструє значне покращення енергоспоживання з мінімальним впливом на затримку та збереженням найсучаснішої точності класифікації.

- Представляється оптимізація часу проектування, які дозволяють реалізувати архітектуру ЗНМ з глибиною більше п'яти згорткових шарів, використовуючи не більше 512 МБ доступної пам'яті. Такі оптимізації дозволяють розробникам використовувати значно менше пам'яті, ніж раніше заявлені реалізації, збільшуючи пропускну здатність на менших платформах із суворими обмеженнями щодо площі, потужності та пам'яті.

У цьому розділі ми надаємо короткий вступ до структури згорткових нейронних мереж (CNN) і описуємо проблеми, пов’язані з розробкою та оптимізацією мереж для архітектур з низьким енергоспоживанням і обмеженими ресурсами. Ми починаємо з обговорення згорткових шарів, за яким слідує аналіз методів підвибірки, що використовуються для зменшення розмірності ознак, обговорення використання щільних шарів і закінчуємо коротким описом процесу навчання.

Приклад типового потоку даних CNN показано на рисунку 3.1. Маючи вхідне зображення, карти активації або функції будуються за допомогою згортки вхідних даних і згорткових ядер. Отримані карти функцій надсилаються через шар підвибірки або шар « об’єднання », щоб зменшити його розмірність. Процес повторюється до тих пір, поки об’єкти не досягнуть відповідної розмірності, після чого нелінійність у об’єктах великої

вимірності вловлюється через повністю пов’язані « щільні » шари та використовується для класифікації.

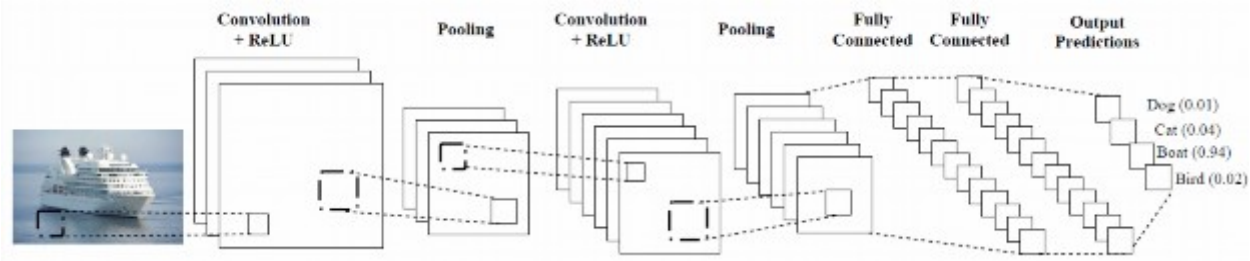

Рис. 3.1. Типова архітектура CNN для завдання розпізнавання об’єктів

3.1.1. Згорткові шари

Згорткові рівні утворюють основну функціональність CNN і є одним із процесів у мережі, які потребують більшого обчислення. Згортковий рівень має набір параметрів, що представляють набір доступних для вивчення ядер фільтрів, який називається його сприйнятливим полем. Кожен фільтр розроблено таким чином, щоб бути малим у просторовому вимірі, але зі значно більшою глибиною, оскільки він поширюється через усе вхідне зображення. Кожен фільтр згортається через просторові розміри (висота та ширина) вхідного зображення для створення карти активації, яка описує реакції фільтра. У міру навчання рівнів мережа має тенденцію вивчати фільтри, чутливі до візуальних особливостей нижчого рівня, таких як край певної орієнтації на початкових рівнях до дуже складних функцій на вищих рівнях мережі. Такі карти активації представляють візуальні особливості, захоплені із зображення на певному рівні. Формально, операція згортки на заданому шарі і та ядрі розміру kx k визначається як

$$x_{ij}^{\ell} = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} W_{m,n} y_{(i+m),(j+n)}^{\ell-1}$$

де xj — вихід у точці (i, j) на карті активації на шарі і, а у e-1 — вихід із шару і — 1.

Для функції активації кожен згортковий шар обчислює лінійні закономірності у вхідних даних за допомогою своєї згорткової операції над простором зображення. Однак важливо ввести нелінійність у функції, які вивчаються зі згорткового рівня. Ця нелінійність зазвичай досягається за допомогою функції активації. Загальні функції активації включають tanh, sigmoid і Rectified Linear Unit (ReLU). Активації ReLU виявилися більш ефективними у забезпеченні швидшого часу навчання в мережах, а також допомагають пом'якшити проблему зникнення градієнта, стан, при якому навчання збільшується через повільне поширення градієнта через шари.

3.1.2. Мережні архітектури

Ми розробили, оптимізували та оцінили чотири широко використовувані архітектури CNN на зростаючих рівнях складності: LeNet, AlexNet, VGG-11 і VGG-16. Кожна з цих мереж дотримується подібної парадигми проектування: комбінація згорткових шарів, за якими слідують шари підвибірки, що закінчується кількома щільними шарами. У цьому розділі ми описуємо архітектуру кожної моделі.

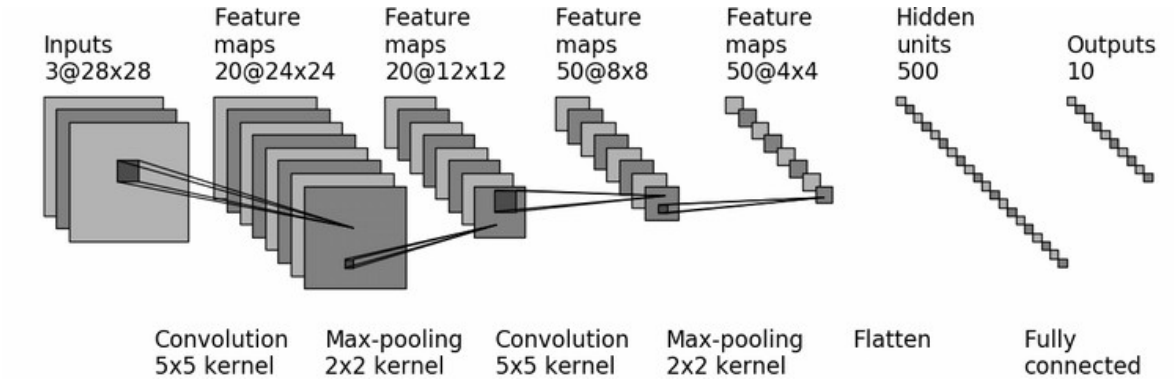

Рис. 3.2. Приклад архітектури LeNet

Архітектура LeNet складається зі згорткового шару з шістьма фільтрами 5 x 5 і шаром максимального об'єднання з кроком 2, за яким слідує інший згортковий рівень із шістнадцятьма фільтрами 5 x 5 і шаром

максимального об’єднання з кроком 2. Три слідують повністю з’єднані шари з сигмовидною активацією для перших двох і softmax для останнього.

AlexNet була першою успішною реалізацією глибокої CNN із загальною кількістю восьми рівнів. П'ять згорткових шарів чергуються з трьома шарами підвибірки, один після перших двох згорткових шарів і інший після трьох останніх згорткових шарів. Розмір ядер варіюється від 11 x 11, 7 x 7 і 3 x 3, з кількістю фільтрів від 96 до 1024. Подібно до LeNet, AlexNet також містить три повністю пов’язані рівні з сигмовидною активацією для перших двох і softmax. наостанок.

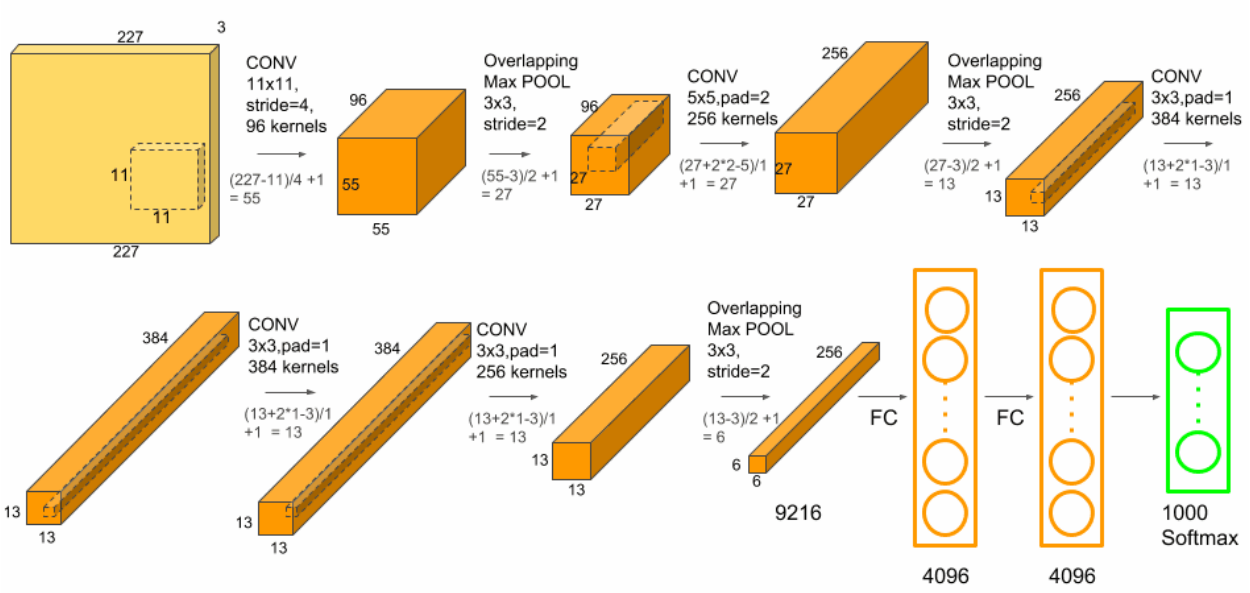

Рис. 3.3. Архітектура AlexNet

VGGNet надав аналіз впливу глибини на точність CNN з використанням малих згорткових ядер малих рецептивних полів (3 x 3) . Ми впровадили та випробували дві версії з 11 та 16 шарами кожна, які називаються VGG-11 та VGG-16, які показали найсучасніші результати з огляду на кількість параметрів та енергоспоживання. Мережі використовують комбінацію двох згорткових шарів, і три набори згорткових шарів, які називаються блоками згорткових шарів, перемежовуються шарами максимального об’єднання для зменшення просторової розмірності.

3.1.3. Процес навчання

Процес визначення параметрів, які можуть реалізувати бажану функціональність та точність класифікації, вказує на фазу навчання або тренування. Навчання мережі здійснюється за допомогою процесу мінімізації помилок, який називається зворотним поширенням. На кожній ітерації помилка обчислюється як різниця між цільовим значенням та прогнозованим значенням, а потім поширюється назад по мережі для покращення точності в наступних ітераціях. Зворотне поширення вимагає приблизно вдвічі більше обчислень прямого процесу як для поширення помилок, так і для оновлення параметрів. Це дозволяє розробникам використовувати подібні конструкції обчислювальних ядер як у прямому, так і у зворотному процесах.

Стандартний процес, який використовується для оптимізації зворотного поширення, називається градієнтним спуском, де градієнт помилки поширюється назад для налаштування ваг таким чином, щоб подальші висновки мали меншу помилку. Підмножина навчальних даних, яка називається міні-партією, вибирається для кроку навчання, і параметри мережі оновлюються на основі її середньої помилки. Крім того, ще один гіперпараметр у цьому називається швидкістю навчання, який дозволяє нам контролювати ступінь поширення помилок і, отже, контролювати ступінь коригування ваг.

У наших експериментах цей процес виконувався на одному графічному процесорі, оскільки нас в першу чергу цікавить енергоспоживання та затримка операцій прямого поширення, що є репрезентативним для мобільного або IoT застосунку.

3.2. Використання архітектур на основі фізично неклонованих функцій в протоколі автентифікації системи IoT

У цьому розділі ми оцінюємо доцільність використання архітектур на основі PUF як основного компонента в протоколі автентифікації. Спочатку

ми досліджуємо наявну роботу щодо використання PUF і основних міркувань безпеки. Потім ми пропонуємо три моделі атак на основі машинного навчання, які намагаються клонувати архітектуру дуги PUF за різними сценаріями, такими як незашифровані, зашифровані та обфусковані CRP, щоб імітувати умови реального світу. Ми виявили, що використання неконтрольованого генеративного попереднього навчання допомагає вловити основний зв’язок між викликом і відповіддю, навіть із введенням шуму через шифрування та обфускацію. Нарешті, ми представляємо захист на основі машинного навчання, який називається дискримінатором, щоб допомогти розрізнити клонований і оригінальний PUF для покращення процесу автентифікації.

Екосистема Інтернету речей (IoT) зростає експоненціально завдяки конвергенції різних технологій, таких як глибоке навчання, сенсорні системи та досягнення в обчислювальних платформах. Очікується, що поява технології 5G та обіцянка вищої пропускної здатності збільшать високу зв'язність сучасної екосистеми IoT. Масова колекція повсюдних та поширених пристроїв в екосистемі IoT була розгорнута в різних середовищах для збору та обробки величезних обсягів даних. Застосування пристроїв IoT варіюється від носимих обчислювальних пристроїв, біоімплантованих пристроїв для моніторингу життєво важливих функцій організму для прямої взаємодії з людиною, а також для "розумних" пристроїв, з якими ми взаємодіємо щодня. З такою високою поширеністю "розумних" пристроїв, характер даних, що збираються та обробляються, може бути все більш приватним та вимагати гарантій для забезпечення цілісності та безпеки даних [1, 2].

З такими надзвичайно приватними даними, вузли IoT повинні бути належним чином автентифіковані перед збором та обробкою таких даних. Протокол автентифікації може бути таким простим, як зберігання секретного ключа на фізичних пристроях на основі кремнію, а також може бути складним протоколом на основі криптографії. Вибір протоколу

автентифікації має наступний набір проблем, які необхідно дослідити та вирішити:

1) Пристрої IoT, як правило, обмежені в ресурсах, тому вимагають високоефективних протоколів безпеки,

2) Їх висока розподіленість може забезпечити легкий фізичний доступ до вузла,

3) Висока зв'язність IoT-фреймворку вимагає швидких та безпечних протоколів безпеки.

Традиційні підходи до криптографії, хоча й ефективні, не виявилися достатньо легкими та швидкими для автентифікації пристроїв IoT. Наприклад, протоколи автентифікації, які вимагають зберігання секретного ключа на кожному вузлі пристрою, хоча й є ефективною стратегією, можуть бути обійдені за допомогою фізичних та бічних канальних атак на вузол пристрою [6] та поставити під загрозу цілісність мережі IoT та пов'язаних з нею даних. Недавні зусилля були спрямовані на використання притаманної випадковості, індукованої в кремнієвих пристроях під час виробничого процесу, як секретного ключа, на відміну від традиційного двійкового ключа, що зберігається в кремнієвих пристроях, який може бути схильний до фізичних атак.

Такі підходи, які називаються фізично неклонованими функціями (PUF), допомогли забезпечити вищий рівень безпеки від прямих фізичних атак. Це усуває необхідність у дорогих заходах фізичного захисту. Фізично неклоновані функції стають все більш популярними та використовуються для автентифікації пристроїв IoT [45, 46] та інших завдань безпеки. PUF використовуються як для розширення існуючих протоколів автентифікації на основі криптографії, так і для нових протоколів, які використовують легку природу та зручність використання архітектур фізично неклонованих функцій.

Пристрої PUF — це легко виготовлені фізичні структури, які використовують стохастичний характер виробничого процесу для створення

фізично неклонованих унікальних ідентифікаторів для кожної виготовленої одиниці. Це зазвичай призводить до односторонньої функції. Враховуючи електронний стимул, відповідь пристрою PUF є непередбачуваною, повторюваною функцією. Ця відповідь ідентифікує кожен пристрій за допомогою унікального підпису. Це в першу чергу пояснюється взаємодією зовнішнього подразника та фізичної структури PUF. Ця взаємодія називається парою виклик-відповідь (CRP), де виклик є зовнішнім стимулом, а реакція PUF називається відповіддю. Через непередбачувану природу PUF, яка може бути дуже чутливою до шуму, схеми корекції помилок [100] використовуються для зменшення невизначеності у відповіді PUF, щоб зробити його більш надійним. PUF з досить великими парами виклик-відповідь називаються сильними PUF і зазвичай вибираються для більшості практичних програм безпеки.

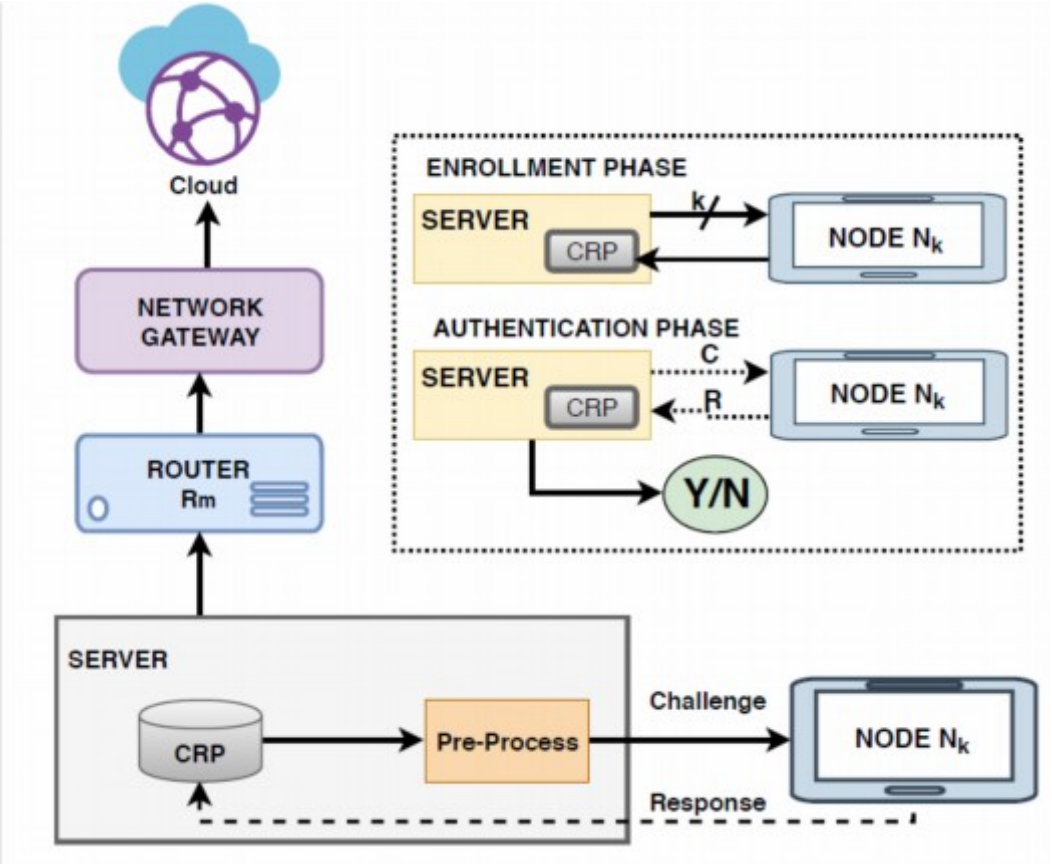

Рис. 3.4. Типова архітектура IoT з фазою реєстрації та фазою автентифікації схеми автентифікації вузла IoT на основі PUF

Використання PUF як основи для автентифікації вузлів IoT останнім часом набуло популярності. Використання PUF для протоколів безпеки IoT зазвичай передбачає етап початкової реєстрації та протокол автентифікації під час фактичного обміну даними. Рисунок 3.4 ілюструє типову архітектуру мережі IoT і загальний протокол реєстрації. На внутрішньому блоці показано фазу реєстрації та фазу автентифікації схеми автентифікації вузла IoT на основі PUF. Блок Pre-Process представляє додатковий процес шифрування та/або обфускації. Типова мережа IoT складається з віддалених вузлів даних з обмеженими ресурсами (Ni ,N2 ,N3 ... Nk), підключених до статичних серверних вузлів (Si, S2, S3 ... Sn), які передають отримані дані в хмару за допомогою маршрутизаторів (Ri, R2, R3 ... R m). Дані передаються з маршрутизаторів у хмару за допомогою мережевого шлюзу. Граничні вузли IoT можуть варіюватися від простих датчиків до складних систем із процесором, пам'яттю, зв'язком тощо. Потужні PUF, реалізовані в складних вузлах IoT, піддаються атакам, що є центром цієї роботи. Коли вузол даних додається до мережі IoT, виконується фаза реєстрації, щоб створити базу даних CRP для PUF у вузлі даних. Ця база даних CRP використовується на етапі автентифікації, коли два вузли, що відповідають одному вузлу сервера, хочуть спілкуватися. Вузол спільного сервера автентифікує обидва вузли даних, генерує пари ключів безпеки та допомагає безпечно обмінюватися ключами. Хоча це практично, фаза реєстрації може бути використана зловмисником для підслуховування та клонування набору CRP, які можуть бути використані для обходу автентифікації на основі PUF і скомпрометувати безпеку вузлів даних. Були спроби які тепер були запропоновані, що екстракція CRPs потім знищується, тобто з'єднує дроти екстракції, тим самим викорінюючи можливість клонування за допомогою цього методу.

Дотримуючись протоколів існуючі мережі IoT, що використовують автентифікацію PUF [45 , 46, 47], роблять наступні базові припущення щодо безпеки:

1) клонування архітектури PUF, фізично чи математично, є складною проблемою, особливо якщо базова архітектура невідома,

2) зловмисник має необмежений фізичний доступ до каналу зв'язку,

3) характеристики виклик-відповідь PUF у вузлі даних IoT є неявною властивістю і недоступні для супротивник,

4) зловмисник може отримати доступ до бази даних CRP через атаки зловмисного програмного забезпечення, незважаючи на явне знання секретних ключів.

Враховуючи ці припущення щодо безпеки, мета супротивника стає простою. По суті, він повинен мати можливість підробити серверні вузли, щоб прийняти шкідливий вузол від імені вихідних вузлів даних без фактичного володіння відповідним вузлом. Будь-які фізичні втручання можуть порушити цілісність PUF і, отже, зробити атаку марною. Основна стохастична природа PUF і наведені вище обмеження створюють надійний протокол безпеки, який важко зламати. Однак прогрес у машинному навчанні призвів до переважної більшості неінвазивних атак на захист на основі PUF.

Підходи на основі машинного навчання можна охарактеризувати застосуванням вивченої математичної моделі до зібраної підмножини дійсних CRP. Курування таких даних зазвичай вважається протоколом підслуховування, що не є необґрунтованим припущенням. Попередні роботи, особливо [39], продемонстрували великий успіх у клонуванні PUF, отримавши точність клонування до 99,99%. Такий успіх приходить із застереженням: архітектура, що лежить в її основі, повинна бути відома апріорі через інвазивні фізичні втручання або явні знання про архітектуру.

Сучасні вузли IoT розроблені таким чином, що вони захищені від несанкціонованого доступу, що ускладнює або робить неможливим мікрозондування. Навіть якщо зловмисник успішно здійснив мікрозондування, враховуючи безліч архітектур PUF в літературі, отримати інформацію про базову архітектуру PUF надзвичайно складно. Отже, попередні PUF-атаки на основі машинного навчання з припущенням знання

базової архітектури або непрактичні, або надзвичайно складні для інсценування. Крім того, ці методи припускають, що виклик доступний для зловмисника у вигляді відкритого тексту, тобто до виклику не застосовується шифрування. Враховуючи, що більшість зв'язку через бездротовий канал зашифровано, це дуже сильні припущення, особливо в контексті безпеки вузлів у рамках IoT. У цій роботі ми представляємо атаку на основі ML, яка не вимагає інформації про архітектуру PUF. Ми також пропонуємо протидію цій атаці, яку можна ефективно використовувати для віддаленої оцінки рівня довіри вузла IoT.

Щоб подолати такі обмеження, ми зосереджуємось на атаці, незалежній від архітектури, яка не передбачає попереднього знання архітектури PUF у системі. Ми показуємо, що спостережуваних CRP достатньо для підвищення точності клонування сильного PUF, незалежно від базової архітектури. Атака може симулювати вузол даних на основі PUF, не знаючи базової архітектури PUF. Щоб оцінити ефективність нашого підходу, ми порівняємо з моделлю атаки (наступний розділ), яка використовує поточні досягнення в клонуванні PUF-архітектури. Ми використовуємо специфічне для архітектури клонування через каскадну структуру (1) ідентифікації архітектури PUF, (2) використовуємо специфічні для архітектури моделі клонування та (3) оцінюємо прогноз і точність моделі шляхом поєднання точності класифікації архітектури і точність клонування в середньому гармонічному.

На основі роботи щодо Generative Adversarial Networks (GAN) ми пропонуємо захист на основі машинного навчання, дискримінатор, щоб визначити можливість клонування за допомогою будь-якої неінвазивної атаки на основі машинного навчання. Заходи протидії [41, 43] клонуванню на основі ML зосереджені на створенні складної, стійкої до клонування архітектури PUF. Оскільки ми входимо в більш реалізовану екосистему IoT, складні архітектури PUF можуть не підходити для легких систем IoT. Отже, ми пропонуємо полегшену ймовірнісну ідентифікацію клонування за допомогою машинного навчання. Наскільки відомо авторам, це перший

подібний фреймворк для неінвазивної атаки схем автентифікації мережі IoT на основі PUF і запропонований механізм для диференціації оригінальних PUF від клонованих.

Підводячи підсумок, ми представляємо фреймворк для клонування автентифікації на основі PUF у налаштуваннях IoT без будь-якого фізичного доступу до пристрою та будь-яких попередніх знань про базову архітектуру PUF. Ми також показуємо, що цей підхід можна розширити за допомогою попереднього навчання з шумом, що не передбачається, для роботи з двома стандартними протоколами шифрування та трьома загальними архітектурами PUF, які на практиці формують деякі з найбільш поширених налаштувань автентифікації вузлів.

3.3. Концепція атаки методом повного перебору в системі IoT

У цьому розділі ми описуємо атаку повного перебору, яку ми формулюємо для оцінки ефективності запропонованого рішення. Ми починаємо з мотивації атаки повного перебору та основних припущень, необхідних для застосування моделей клонування. Потім ми продовжуємо обговорення математичних моделей, що використовуються для ідентифікації базової архітектури PUF, та наступних результатів.

На основі моделей запропонованих в [39] ми можемо успішно клонувати моделі сильних ФНФ (фізично неклонованих функцій) з точністю прогнозування 99,9%. Однак, щоб використовувати його неінвазивним способом, нам спочатку потрібно визначити базову архітектуру ФНФ, оскільки підходи в [39] вимагають глибоких знань архітектури ФНФ, таких як тип ФНФ, кількість етапів та кількість вентилів XOR, щоб назвати декілька.

Щоб вирішити цю проблему, ми пропонуємо використовувати модель машинного навчання для ідентифікації архітектури ФНФ шляхом спостереження за парами "запит-відповідь", як показано на рисунку 3.5.

Одним з основних припущень у цьому підході є те, що існує підмножина запитів C, яка є дійсною для всіх архітектур ФНФ у даній мережі, де C - це колекція всіх дійсних пар "запит-відповідь". Враховуючи кількість архітектур ФНФ та їх використання для автентифікації, це не є необґрунтованим припущенням.

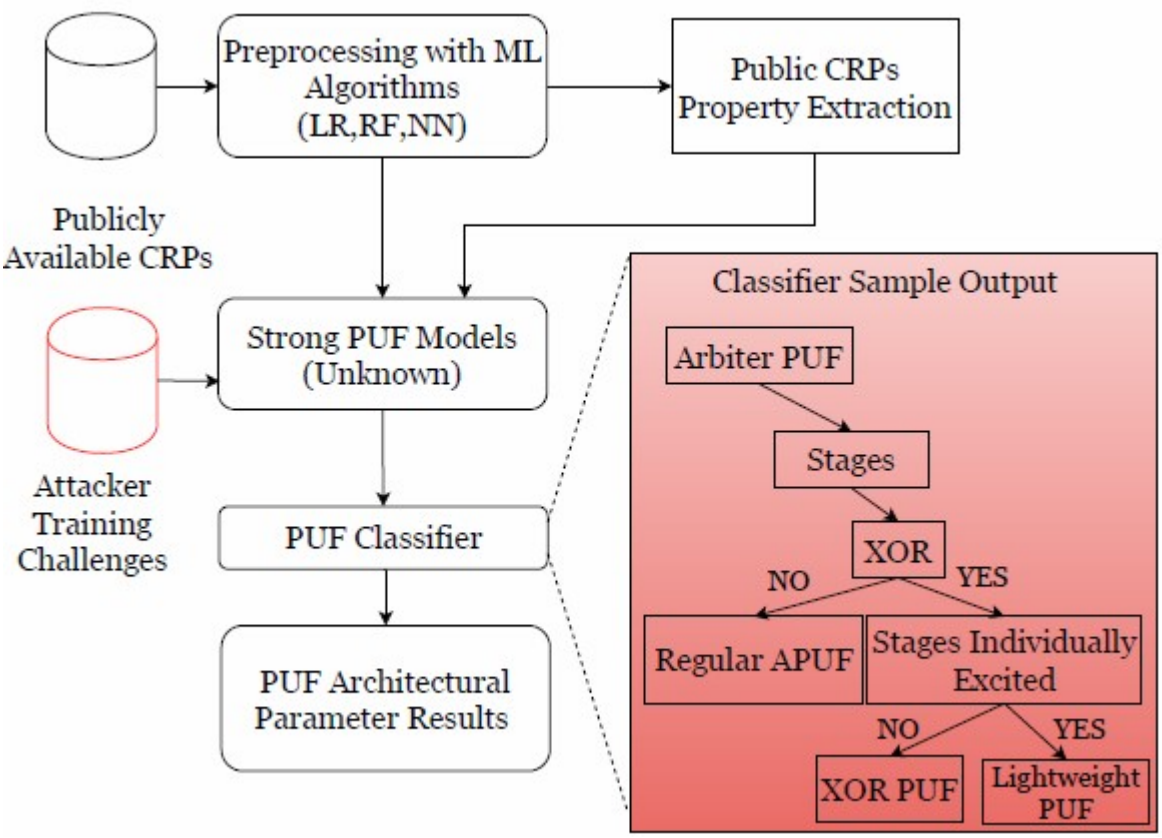

Рис. 3.5. Запропонована модель атаки на архітектуру необізнаних функцій (ФНФ). Атака повного перебору має додатковий процес виявлення архітектури ФІФ

Ми оцінюємо продуктивність запропонованої атаки, щоб визначити архітектуру восьми загальних сильних архітектур PUF. Ми використовуємо фіксовану кількість випадково відібраних 100 CRP для оцінки для кожної архітектури PUF для загальної кількості 800 CRP. Ми повідомляємо про середні результати 5 різних прогонів, причому тестовий набір кожного разу відбирається випадковим чином. Ми куруємо колекцію зі 100 000 CRP для навчання моделі класифікації.

Таблиця 3.1.

Ефективність класифікації та точність клонування при атаці повного перебору

PUF Model	PUF Classification Rate (%)	Cloning Rate (%)
APUF	81.49%	77.42%
3 XOR APUF	76.53%	72.71%
4 XOR APUF	65.01%	61.76%
5 XOR APUF	63.57%	60.39%
6 XOR APUF	61.31%	58.25%
LW 3 XOR APUF	76.91%	73.05%
LW 4 XOR APUF	65.37%	62.10%
LW 5 XOR APUF	59.32%	56.33%

Як видно з таблиці 3.1, ідентифікація архітектури PUF із спостережуваного набору CRP не є тривіальним завданням. Навіть зі 100% точністю клонування для даної архітектури PUF ідентифікація зазначеної архітектури вимагає великого набору CRP для навчання моделі. Максимальна продуктивність, яку ми змогли отримати, була за допомогою моделі логістичної регресії, для зближення якої знадобилося 100 ітерацій, що призвело до максимального рівня класифікації для архітектури Arbiter PUF. Існувала велика плутанина між різними варіантами дизайну кожного типу ППУ. Швидкість передбачення для XOR PUF зменшилася зі збільшенням складності архітектури.

Видно, що ідентифікація архітектури PUF вимагає значних ресурсів навчання – 100 000 CRP при розпізнаванні PUF арбітра із середньою точністю 81,49%. Класифікатор показав найгірші результати на легких PUF, забезпечивши максимальну точність ідентифікації для 3-бітового XOR легкого PUF. Частота ідентифікації також впливала на швидкість прогнозування клонування підходу грубої сили, оскільки кожна неправильно класифікована архітектура PUF впливала на якість клонування. Хоча середня точність клонування може досягати 77,42% (для Arbiter PUF), на практиці ці

цифри можуть ввести в оману. Ефективність двоетапної моделі атаки досить низька, враховуючи можливий розрив між відстанями Хеммінга всередині та між Хеммінгом CRP PUF, цей показник передбачення не можна вважати успішним клонуванням архітектури PUF.

3.4. Використання автоенкодерів для надійного навчання ознак

У той час як модель атаки, представлена в попередньому розділі, може обробляти виклики відкритого тексту, протоколи шифрування, такі як AES і DES, можуть вводити шум у зв'язок між викликом і відповіддю, таким чином приховуючи характерну функцію архітектури PUF. Щоб пояснити це, потрібно або зламати шифрування за допомогою традиційного криптоаналізу, або вивчити надійні представлення, які можуть відокремити шум від важливої інформації в рамках виклику введення. Оскільки обчислювальні ресурси для досягнення першого можуть бути дорогими, ми застосовуємо другий підхід і намагаємося вивчити надійні представлення шляхом попереднього навчання без нагляду за допомогою автоматичного кодувальника з усуненням шумів. У цьому підході ми навчаємо нейронну мережу (багатошаровий перцептрон, MLP) як нашу модель атаки.

Традиційний автокодер — це неконтрольована нейронна мережа, метою якої є вивчення стисненого представлення вхідних даних за допомогою операції каскадного кодування-декодування. Архітектура мережі складається з двох нейронних мереж, мережі кодера та мережі декодера, які працюють разом, щоб дізнатися закодоване представлення або прихований простір. Роль кодера полягає в тому, щоб стискати вхідні дані в представлення нижчих розмірів, яке фіксує базовий шаблон даних, навчаючись ігнорувати якомога більше помилкових шаблонів або шуму . Це стиснуте представлення представляє рівень вузького місця мережі. Роль декодера полягає в тому, щоб навчитися реконструювати оригінальний вхід із цього стисненого представлення. Цей процес представлено на рисунку 3.6

а), де видно, що латентний простір має меншу розмірність порівняно з більшими розмірними входом і виходом. Вхід і вихід каркаса автокодувальника мають однакові розміри. Метою навчання для мережі автокодера є мінімізація втрат при реконструкції, якими зазвичай є втрати L2 або двійкова крос-ентропія.

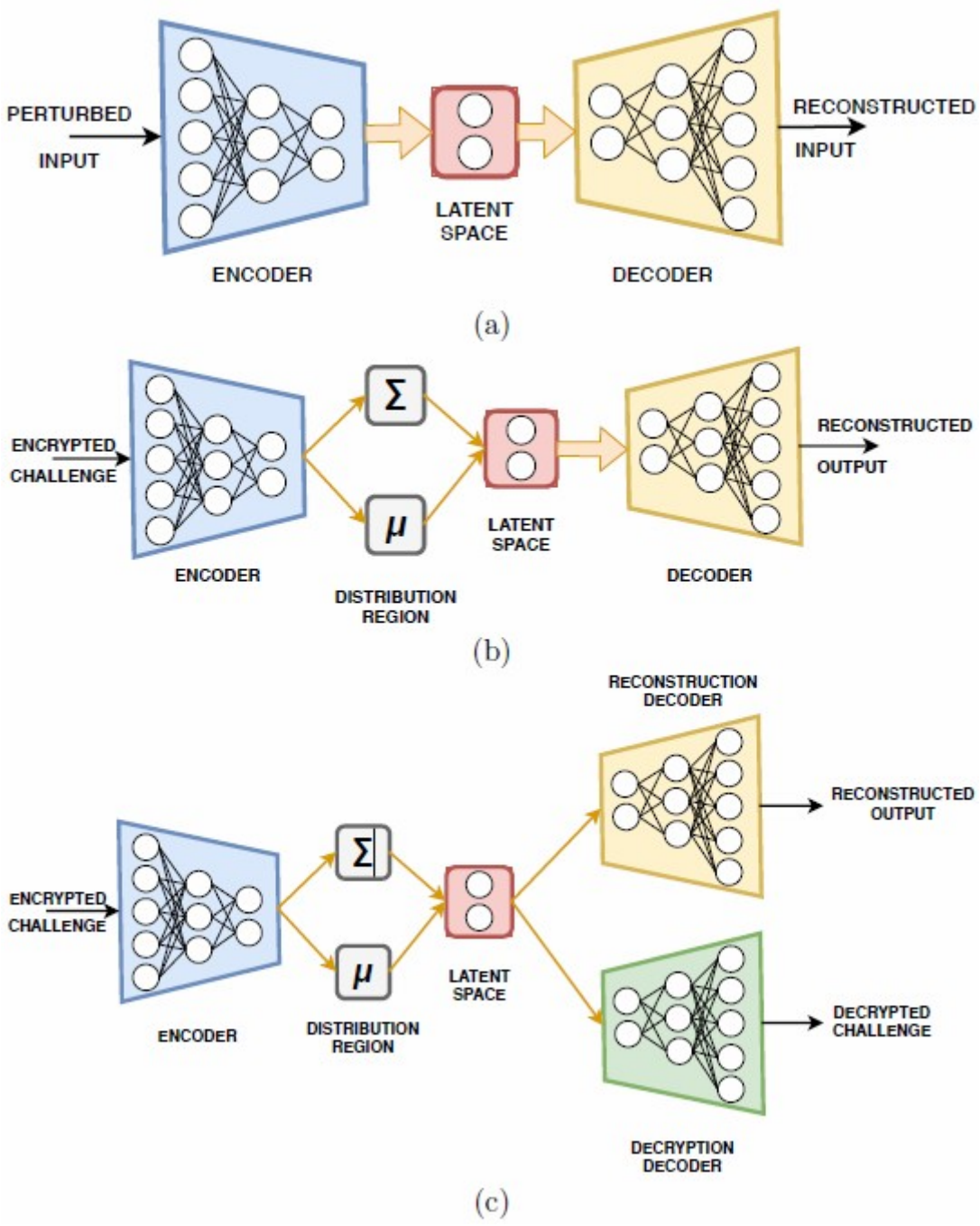

Рис. 3.6. Представлення a) автоенкодера, b) варіаційного автоенкодера та c) запропонованого мультиваріаційного автоенкодера

У той час як автокодери вивчають корисні функції (прихований простір), які можна використовувати для завдань класифікації нижче за потоком, додавання шуму або збурень у вхідних даних може суттєво змінити представлення, якщо їх не додати під час навчання. Щоб врахувати шум, створений через шифрування, ми навчаємо автокодер як шумозаглушувальний автокодер. Ідея полягає в тому, щоб навчити автокодер реконструювати вхідні дані з пошкодженої або випадково збуреної версії вхідних даних. Ця стратегія навчання застосовується, щоб змусити прихований рівень виявити більш надійні функції та запобігти його простому навчанню функції ідентичності. Ми створюємо дешумуючий автокодер, додаючи крок стохастичного пошкодження до вхідних даних. Незважаючи на те, що вхідні дані можуть бути збурені багатьма способами, ми хочемо, щоб наші представлення обробляли внутрішній шум, який застосовувався через бездротовий канал, обфускацію та шифрування.

Отже, у нашій реалізації ми застосовуємо наступні збурення:

1) випадково маскуємо частину вхідних даних, роблячи їх нульовими,

2) додаємо до вхідних даних випадковий білий шум,

3) додаємо функцію хешування до CRP для імітації методи шифрування.

На кожній навчальній ітерації одне з перерахованих вище збурень застосовується до входу, а вихід мережі декодера порівнюється з вихідним входом.

Через складну природу запропонованої мережі ми представляємо деталі реалізації для розуміння. Мережа кодера — це чотирирівнева мережа повністю з’єднаних шарів. Між кожним наступним шаром є дропаутний шар, який допомагає запобігти переобладнанню. Імовірність випадання кожного рівня становить 50%. Кількість нейронів у кожному шарі зменшується в 0,5 раза, щоб зменшити розмірність оброблених даних. Це відповідає стандартному протоколу в автокодерах, щоб викликати вузьке місце в кінці мережі кодування. Мережа декодування є дзеркалом мережі кодування, де

кількість нейронів збільшується відповідно до вихідних розмірів. Ми навчаємо мережу протягом десяти епох зі швидкістю навчання або 1e-4 за допомогою стандартного оптимізатора Gradient Descent.

Оскільки запропонована архітектура має складну структуру, ми детально описуємо деталі реалізації та стратегію навчання для підходу тут. Кодер складається з чотирьох щільно з'єднаних шарів, кожен з яких перемежовується шаром випадання. Імовірність випадання кожного рівня становить 50%. Ми зменшуємо розмірність вхідних даних на 0,5x на кожному повністю зв'язаному (щільному) шарі. Це відповідає стандартному протоколу в автокодерах, щоб створити вузьке місце в кінці мережі кодування. Кожен із двох декодерів (відновлення та дешифрування) складається з двох повністю з'єднаних рівнів, які збільшують розмірність до початкового розміру та розмірності розшифрованого завдання відповідно. Ми також маємо серію з двох повністю з'єднаних шарів, які приймають латентний простір як вхідні дані та створюють відповідь PUF як вихідні дані. Це єдина частина мережі, яка навчається під наглядом, тобто за допомогою міток і цільових розмірів. Кодер і два декодери навчаються без нагляду.

Оскільки навчальні дані обмежені, більшість нейронних мереж, як правило, переналаштовуються на менші обсяги даних і погано узагальнюють інші, неспостережувані пари виклик-відповідь. Щоб подолати це, ми пропонуємо наступний режим тренувань. Протягом десяти епох ми спочатку наскрізно навчаємо мережу лише з активним декодером реконструкції, тобто спочатку він навчається як традиційний варіаційний автокодер. Протягом наступних десяти епох ми потім навчаємо дешифратор дешифрування протягом десяти епох, одночасно заморожуючи ваги декодера реконструкції. Це частина запропонованого режиму тренувань без нагляду. Потім ми починаємо навчальний процес під наглядом. У цій частині тренінгу ми заморожуємо шари структур декодування та беремо латентний простір, створений мережею кодувальника, і передаємо його на серію повністю з'єднаних шарів і моделюємо відповідь PUF до виклику. Ціллю нейронної

мережі є відповідь PUF. Загалом ми тренуємося протягом 100 епох, причому частини без нагляду та під наглядом чергуються разом.

3.5. Методика захисту на основі машинного навчання

Моделювання внутрішньої випадковості заданої архітектури ФНФ ставить під сумнів цілісність автентифікації на основі пар "запит-відповідь". Отже, критично важливо, щоб ми могли розрізняти оригінальний ФНФ та ворожу атаку. З цією метою ми вводимо математичну модель, яка здатна розрізняти оригінальний та клонований ФНФ, яка називається дискримінаторною моделлю, як показано на рисунку 3.7.

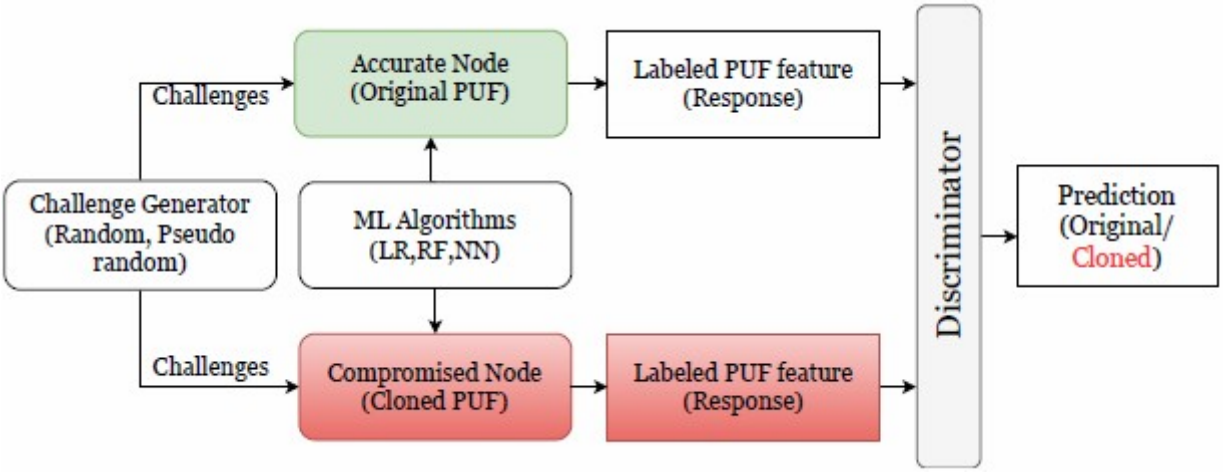

Рис. 3.7. Модель дискримінатора на основі ML для перевірки цілісності PUF

Дискримінатор вирішує, чи кожен екземпляр відповіді належить фактичному ФНФ чи зловмиснику. Як видно на рисунку 3.7, дискримінаторна модель приймає на вхід відповідь оригінального ФНФ разом із відповіддю ФНФ, клонованого за допомогою кількох атак машинного навчання, щоб передбачити, чи є ФНФ оригінальним чи клонованим, і повертає ймовірності. Клонована частина відповіді показана червоним кольором. Вихід цього дискримінатора - це єдине скалярне

значення D(C), яке вказує на ворожу атаку. Значення D(C) - це функція ймовірності, яка відображає задану відповідь (R) на розподіл, що належить або оригінальному ФНФ (^f(C)), або зловмиснику (f(C)) для заданого n-бітового запиту C. Отже, оптимальна дискримінаторна модель задається формулою

$$D^{\star}(C,R) = \frac{p(\hat{f}(C)}{p(f(C)) + p(\hat{f}(C))}$$

де D*(C;R) - це математична модель, яка відображає відповідь R для заданого запиту (C) у простір ймовірностей або оригінального ФНФ (f(.)), або моделі атаки (f(:)). Ми повідомляємо про продуктивність дискримінаторної моделі на різних архітектурах ФНФ у таблиці 3.2. Знову ж таки, ми досліджуємо використання відомих моделей машинного навчання як основу для нашої математичної моделі дискримінатора.

Таблиця 3.2.

Продуктивність дискримінатора для різних архітектур PUF

PUF Model	Discriminator Accuracy (%)
APUF	94.43
3 XOR APUF	98.81
4 XOR APUF	95.99
5 XOR APUF	96.15
6 XOR APUF	100
LW 3 XOR APUF	91.66
LW 4 XOR APUF	94.78
LW 5 XOR APUF	96.31
Average	96.01

Простір пошуку оптимального дискримінатора аналогічно характеризується функцією оптимізації. Однак пошук представлений дискримінатором, щоб розрізнити оригінальну відповідь PUF і атаку клонування.

Висновки до розділу

Отже, в цьому розділі представлено імплементацію загорткових мереж і методів машинного навчання для застосувань Інтернету речей (IoT). Визначено особливості використання згорткових нейронних мереж, детально описано функціонування згорткових шарів та різні мережні архітектури, а також охарактеризовано процес навчання. Особливу увагу приділено викликам і перевагам інтеграції цих методів у системи IoT.

Висвітлено використання архітектур на основі фізично неклонованих функцій (PUF) у протоколах автентифікації для підвищення безпеки IoT. Проаналізовано концепцію атак методом повного перебору, що становить загрозу для цих систем, і запропоновано способи протидії таким атакам. Розглянуто застосування автоенкодерів для надійного навчання ознак, що забезпечує ефективне виявлення аномалій і покращує загальну безпеку.

Також представлено методику захисту на основі машинного навчання, яка забезпечує підвищену стійкість систем IoT до різних атак. Обговорено ефективність цих підходів у підвищенні надійності та функціональності IoT-систем у складних і обмежених середовищах.

ВИСНОВКИ

В магістерській роботі досліджено концептуальні моделі та методи машинного навчання для застосунків Інтернету речей. Проведено комплексне дослідження концепцій Інтернету речей (IoT) із фокусом на Internet of Medical Things (IoMT) та впровадження методів машинного навчання для підвищення ефективності й безпеки таких систем. Робота включає аналіз сучасних викликів і тенденцій у цій предметній області, оцінку алгоритмів машинного навчання та особливостей впровадження глибоких нейронних мереж на обмежених платформах.

Розглянуто надійні архітектури фізично неклонованих функцій (PUF) як основу для автентифікації пристроїв IoT і проаналізовано загрози, пов'язані з атаками на основі машинного навчання. Запропоновано підходи для вдосконалення безпеки IoT, зокрема використання протоколів шифрування і методів захисту, що забезпечують підвищену стійкість до кібератак.

Значна увага приділена розробці та імплементації загорткових нейронних мереж, а також їхньому використанню для обробки даних у межах IoT-додатків. Описано структури та процеси навчання, наведено приклади ефективного захоплення часових залежностей і представлено методи сегментації без учителя для підвищення продуктивності в системах IoMT. Також розроблено підхід до надійного навчання ознак із використанням автоенкодерів.

Оцінка запропонованих рішень підтверджує їхню здатність покращити як безпеку, так і обчислювальну ефективність у системах IoT. Застосовані методи й моделі відкривають нові можливості для децентралізованого навчання та створення стійких до атак архітектур, що є критично важливим у сфері IoMT.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). "A survey on deep learning for big data." Information Fusion, 42, 146-157.
2. N. Cam-Winget, A. Sadeghi, and Y. Jin, “Can IoT be secured: Emerging challenges in connecting the unconnected,” in Proceedings of the 53rd Annual Design Automation Conference DAC. ACM, 2016, p. 122.
3. Deng, L., Li, G., Han, S., Shi, L., & Xie, Y. (2020). "Model compression and hardware acceleration for deep learning on edge devices: A survey." IEEE Signal Processing Magazine, 37(1), 101-110.
4. Shi, W., & Dustdar, S. (2016). "The promise of edge computing." Computer, 49(5), 78-81.
5. O.Walch, Y. Huang, D. Forger, and C. Goldstein, “Sleep stage prediction with raw acceleration and photoplethysmography heart rate data derived from a consumer wearable device,” Sleep, vol. 42, no. 12, p. zsz180, 2019.
6. Chen, M., Mao, S., & Liu, Y. (2014). "Big data: A survey." Mobile Networks and Applications, 19, 171-209.
7. Li, S., Xu, L. D., & Zhao, S. (2018). "5G Internet of Things: A survey." Journal of Industrial Information Integration, 10, 1-9.
8. Khan, M. A., & Salah, K. (2018). "IoT security: Review, blockchain solutions, and open challenges." Future Generation Computer Systems, 82, 395-411.
9. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). "Internet of Things: A survey on enabling technologies, protocols, and applications." IEEE Communications Surveys & Tutorials, 17(4), 2347-2376.
10. S. Ray, S. Bhunia, Y. Jin, and M. Tehranipoor, “Security validation in IoT space,” in 2016 IEEE 34th VLSI Test Symposium (VTS). IEEE, 2016, pp. 1–1.

11. Atzori, L., Iera, A., & Morabito, G. (2010). "The Internet of Things: A survey." Computer Networks, 54(15), 2787-2805.
12. Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2018). "Deep learning for IoT big data and streaming analytics: A survey." IEEE Communications Surveys & Tutorials, 20(4), 2923-2960.
13. Wang, K., Yin, Y., & Wei, G. (2017). "Towards a ubiquitous Internet of Things for smart environments: Challenges and future directions." IEEE Communications Magazine, 55(12), 94-101.
14. P. Dehkordi, A. Garde, W. Karlen, D. Wensley, J. M. Ansermino, and G. A. Dumont, “Sleep stage classification in children using photoplethysmogram pulse rate variability,” in Computing in Cardiology 2014. IEEE, 2014, pp. 297–300.
15. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in Artificial Intelligence and Statistics, 2017, pp. 1273–1282.
16. Yang, Y., Wu, L., Yin, G., Li, L., & Zhao, H. (2017). "A survey on security and privacy issues in Internet-of-Things." IEEE Internet of Things Journal, 4(5), 1250-1258.
17. Guo, L., Wang, X., & Zhou, B. (2019). "Machine learning-based anomaly detection for smart city IoT infrastructure." IEEE Access, 7, 83747-83759.
18. Yu, K., Liang, Y., He, Y., Hossain, M. S., & Alamri, A. (2017). "Blockchain-based IoT infrastructure for reliable and secure data streaming." Internet Technology Letters, 1(1), e12.
19. A. Mahmoud, U. Rührmair, M. Majzoobi, and F. Koushanfar, “Combined Modeling and Side Channel Attacks on Strong PUFs,” Cryptology ePrint Archive, Report 2013/632, 2013, https://eprint.iacr.org/2013/632.
20. Stankovic, J. A. (2014). "Research directions for the Internet of Things." IEEE Internet of Things Journal, 1(1), 3-9.

21. Kumar, N., & Sangaiah, A. K. (2018). "Machine learning for Internet of Things (IoT): Applications, challenges, and future directions." Computer Communications, 137, 62-78.
22. Lu, Y., & Xu, X. (2019). "Internet of Things (IoT) cybersecurity research: A review of current research topics." IEEE Internet of Things Journal, 6(2), 2101-2112.
23. B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, “Silicon physical random functions,” in Proceedings of the 9th ACM Conference on Computer and Sommunications Security. ACM, 2002, pp. 148–160.
24. Pal, D., Funilkul, S., Charoenkitkarn, N., & Kanthamanon, P. (2018). "Internet of Things and smart homes for elderly healthcare: An end-user perspective." IEEE Access, 6, 10483-10496.
25. Nguyen, T. T., & Ioannidis, M. (2017). "A survey of machine learning methods for IoT security." ACM Computing Surveys, 50(6), 1-37.
26. Yaqoob, I., Hashem, I. A. T., Ahmed, A., & Gani, A. (2017). "Internet of Things architecture: Recent advances, taxonomy, requirements, and open challenges." IEEE Wireless Communications, 24(3), 10-16.
27. X. Zhang, A. Ramachandran, C. Zhuge, D. He, W. Zuo, Z. Cheng, K. Rupnow, and D. Chen, “Machine learning on FPGAs to face the IoT revolution,” in Proceedings of the 36th International Conference on Computer-Aided Design. IEEE Press, 2017, pp. 819–826.
28. Rathore, M. M., Paul, A., & Hong, W. H. (2016). "Real-time big data analytics for event detection and decision-making in smart city environments." Journal of Big Data, 3(1), 13.
29. Liu, J., Wang, Q., & Yang, J. (2020). "Deep reinforcement learning for IoT security: Recent advances and challenges." IEEE Network, 34(6), 303-309.
30. He, D., Zeadally, S., & Khan, S. (2015). "An analysis of RFID authentication schemes for Internet of Things in healthcare environment using elliptic curve cryptography." IEEE Internet of Things Journal, 2(1), 72-83.

31. S. I. Venieris, A. Kouris, and C.-S. Bouganis, “Toolflows for mapping convolutional neural networks on FPGAs: A survey and future directions,” arXiv preprint arXiv:1803.05900, 2018.
32. Pan, J., & McElhannon, J. (2018). "Future edge cloud and edge computing for Internet of Things applications: A survey." IEEE Internet of Things Journal, 5(1), 439-452.
33. Ray, P. P., Dash, D., & De, D. (2018). "Edge computing for Internet of Things: A survey." IEEE Internet of Things Journal, 5(4), 2709-2724.
34. Wang, H., Yang, Z., & Zhou, B. (2019). "Machine learning-based energy optimization for smart home IoT devices." Journal of Ambient Intelligence and Smart Environments, 11(3), 247-258.
35. C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, “Optimizing FPGA-based accelerator design for deep convolutional neural networks,” in Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. ACM, 2015, pp. 161–170.
36. Hong, M., Wei, G., & Chen, Z. (2020). "Federated learning for the Internet of Things: A comprehensive survey." IEEE Communications Surveys & Tutorials, 22(3), 2032-2060.
37. Xu, L. D., He, W., & Li, S. (2014). "Internet of Things in industries: A survey." IEEE Transactions on Industrial Informatics, 10(4), 2233-2243.
38. Hameed, S., & Khan, F. (2020). "Using machine learning algorithms for security in the Internet of Things (IoT): A systematic review." Future Internet, 12(7), 110.
39. S. I. Venieris and C.-S. Bouganis, “FPGAConvNet: A framework for mapping convolutional neural networks on FPGAs,” in 2016 IEEE 24th Annual International Symposium on FCCM. IEEE, 2016, pp. 40–47.
40. Pereira, J., & Aguiar, A. (2019). "Machine learning strategies for time series prediction in smart city applications." ACM Transactions on Internet Technology, 19(3), 1-25.

41. A. R. Doherty and A. F. Smeaton, “Automatically segmenting lifelog data into events,” in 2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services. IEEE, 2008, pp. 20–23.
42. P.-Y. Hsu, W.-F. Cheng, P.-J. Hsieh, Y.-L. Lin, and W. H. Hsu, “Real-time instant event detection in egocentric videos by leveraging sensor-based motion context,” in Proceedings of the 23rd ACM International Conference on Multimedia. ACM, 2015, pp. 1275–1278.
43. Sun, Y., Song, H., & Jara, A. J. (2016). "Internet of Things and big data analytics for smart and connected communities." Journal of Network and Computer Applications, 67, 102-111.
44. Patel, H., & Patel, V. (2018). "Edge computing for Internet of Things: Applications, challenges, and future directions." International Journal of UbiComp, 9(3/4), 19-40.
45. Abou-Nassar, E. M., et al. (2020). "Energy-efficient and secure IoT-based healthcare framework using machine learning." Future Generation Computer Systems, 103, 259-278.
46. Kumar, R., & Ahmad, S. (2017). "IoT applications in smart agriculture: A review." IEEE Internet of Things Journal, 4(6), 1-14.
47. C. Zhang, Z. Fang, P. Zhou, P. Pan, and J. Cong, “Caffeine: Towards uniformed representation and acceleration for deep convolutional neural networks,” in ICCAD, 2016 IEEE/ACM International Conference on. IEEE, 2016, pp. 1–8.
48. Khan, M. A., & Rahman, M. (2018). "Data management in Internet of Things (IoT): Security, privacy, and trust." ACM Computing Surveys, 50(3), 32.
49. Moghaddam, M. H. Y., et al. (2019). "Machine learning approaches for IoT: A survey and taxonomy." Sensors, 19(22), 4934.
50. M. N. Aman, K. C. Chua, and B. Sikdar, “Position paper: Physical unclonable functions for IoT security,” in Proceedings of the 2nd ACM

international workshop on IoT privacy, trust, and security. ACM, 2016, pp. 10–13.

51. Y. Ishai, M. Prabhakaran, A. Sahai, and D. Wagner, “Private circuits II: keeping secrets in tamperable circuits,” in Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2006, pp. 308–327.