People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University of Abdelhamid Mehri – Constantine 2

Faculty of New Technologies of Information and Communication (NTIC)

Department of Fundamental Computing and its Applications (IFA)

MASTER'S THESIS

to obtain the diploma of Master degree in Computer Science

Option: Networking and Distributed Systems (RSD)

Critical Infrastructure Protection Against Massive Cyber Attacks

---

Realized by:

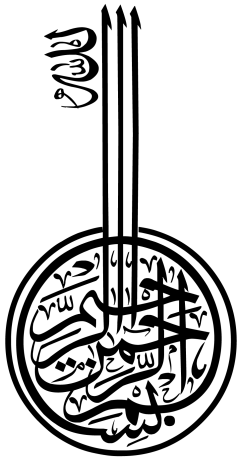
CHEKAIM Badreddine

ZEGHAD Selma

Under supervision of:

Dr. DJENNA Amir

June 2024



Acknowledgments

First and foremost, we thank the Almighty God who gave us the courage, power, strength, and patience to complete this modest work. We extend our heartfelt appreciation to our coach, Dr. Amir DJENNA, for his invaluable guidance, assistance, and unwavering patience throughout this project. His availability and insightful advice have been instrumental, and we hope this work serves as a testament to his exceptional character. We also wish to express our gratitude to all the professors who have imparted their knowledge and supported us during our academic journey. Additionally, we are thankful to the jury members for kindly agreeing to evaluate this work.

Dedication

To those who have guided and supported me,
Words cannot fully express my gratitude,
Love, respect, and appreciation...
I dedicate this work to you.

I thank my very dear family, who have always been there for me. Their unconditional support and encouragement have been of great help. To my beloved mother, father, sisters, brothers, and extended family, including Zeghad and Bouab, your unwavering belief in me has been a source of strength and inspiration. I am profoundly grateful for your love, guidance, and sacrifices throughout my journey. Your presence has been a constant reminder of the importance of perseverance and determination. This accomplishment is as much yours as it is mine, and I carry it with immense pride knowing that I have your unwavering support behind me.

I also want to extend my gratitude to my dear friends Soumia, Chaima, Nida, Rihame, and others who have been by my side through thick and thin. Your encouragement, laughter, and shared moments of joy have made this journey all the more memorable. Your friendship has been a cherished gift, and I am thankful for your unwavering support and belief in me.

Selma Zeghad

From: **Chekaiem Badraldine**

To whom I hold dear,

First and foremost, to my beloved mother, who sacrificed so much for me. She spared no effort in making me happy always.

As we walk the paths of life, someone remains in our minds with every step we take. For me, that person is my dear father, with his kind face and good deeds. He has supported me throughout his life.

To all my dear teachers and friends, who never hesitated to extend a helping hand to me and stood by me in numerous ways,

I present this research to you, and I hope it meets your expectations.

ملخص

التكامل المتزايد لتقنية إنترنت الأشياء في أنظمة التحكم الصناعي قاد إلى تعزيز قدرات وكفاءة أنظمة البنية التحتية الحيوية بشكل كبير. ومع ذلك، فقد أدى هذا التكامل أيضًا إلى ظهور ثغرات جديدة، خاصة في مواجهة الهجمات الموزعة لحرمان الخدمة. تركز هذه الأطروحة على استكشاف تقنيات الدفاع ضد الهجمات الموزعة لحرمان الخدمة في أنظمة البنية التحتية الحيوية. تقدم الدراسة تحليلًا شاملًا لأنواع الهجمات وعملياتها، وتأثيرها على بيئات إنترنت الأشياء الصناعية، والاستراتيجيات الحالية للتخفيف منها. تعتمد الأبحاث على نماذج متقدمة للتعلم الآلي، بما في ذلك الشبكات العصبية التلافيفية والنماذج الهجينة والأساليب القائمة على التعلم الفيدرالي، لاكتشاف وتخفيف هذه الهجمات بفعالية. تظهر النتائج التجريبية فعالية هذه النماذج في تعزيز أمن ومرونة أنظمة البنية التحتية الحيوية.

الكلمات المفتاحية: إنترنت الأشياء، إنترنت الأشياء الصناعية، أنظمة التحكم الصناعي، الهجمات الموزعة لحرمان الخدمة، البنية التحتية الحيوية، التعلم الفيدرالي.

Abstract

The increasing integration of Internet of Things (IoT) technology into Industrial Control Systems (ICS) has significantly enhanced the capabilities and efficiency of critical infrastructure systems. However, this integration has also introduced new vulnerabilities, particularly to Distributed Denial of Service (DDoS) attacks. This thesis focuses on detecting and defending against DDoS attacks in critical infrastructure systems. It presents a comprehensive study of the types and processes of DDoS attacks, their impact on Industrial Internet of Things (IIoT) environments, and the existing mitigation strategies. The research employs advanced machine learning models, including Convolutional Neural Networks (CNN), hybrid CNN-LSTM and CNN-LSTM-GRU models, and federated learning approaches, to effectively detect and mitigate these attacks. Experimental

results demonstrate the efficacy of these models in enhancing the security and resilience of critical infrastructure systems.

Keywords: Internet of Things (IoT), Industrial Internet of Things (IIoT), Industrial Control Systems (ICS), Distributed Denial of Service (DDoS) , Critical Infrastructure, Federated Learning.

Résumé

L'intégration croissante de la technologie de l'Internet des Objets (IoT) dans les systèmes de contrôle industriel (ICS) a considérablement amélioré les capacités et l'efficacité des systèmes d'infrastructures critiques. Cependant, cette intégration a également introduit de nouvelles vulnérabilités, notamment aux attaques par déni de service distribué (DDoS). Cette thèse se concentre sur la détection et la défense contre les attaques DDoS dans les systèmes d'infrastructures critiques. Elle présente une étude complète des types et processus des attaques DDoS, leur impact sur les environnements de l'Internet Industriel des Objets (IIoT), et les stratégies d'atténuation existantes. La recherche utilise des modèles avancés d'apprentissage automatique, y compris les réseaux de neurones convolutifs (CNN), des modèles hybrides CNN-LSTM et CNN-LSTM-GRU, et des approches d'apprentissage fédéré, pour détecter et atténuer efficacement ces attaques. Les résultats expérimentaux démontrent l'efficacité de ces modèles pour améliorer la sécurité et la résilience des systèmes d'infrastructures critiques.

Mots clés : Internet des Objets (IoT), Internet Industriel des Objets (IIoT), Systèmes de Contrôle Industriel (ICS), Attaques par Déni de Service Distribué (DDoS), Infrastructures Critiques, Apprentissage Fédéré.

Table of Contents

List of Figures

List of Tables

List of Algorithms

General Introduction

Project Background

In today's technological landscape, Internet of Things (IoT) integration has become essential. The ongoing industrial revolution plays a pivotal role in driving advancements in current technologies. The Industrial Internet of Things (IIoT) specifically refers to the incorporation of IoT technology into Industrial Control Systems (ICS), which serve as the backbone of a nation's critical infrastructure. Industries such as smart manufacturing, oil and gas exploration, water distribution, and treatment heavily depend on these systems. The application of IoT in ICS enhances network intelligence, leading to optimized and automated industrial operations. However, these advancements come with a downside, as they introduce new vulnerabilities that can be exploited by malicious threats against these critical systems. Industrial IoT (IIoT) systems are vulnerable to various types of attacks that can compromise their integrity, disrupt operations, and pose risks to critical infrastructure. Cybersecurity risks include malicious actors exploiting vulnerabilities in connected devices to launch sophisticated attacks such as ransomware or man-in-the-middle (MitM) attacks. One significant risk to IIoT networks is distributed denial of service (DDoS) attacks. These attacks flood target networks with large volumes of traffic or requests, resulting in disruptions to operations and compromising the availability of crucial services. The financial and reputational consequences of such attacks can be severe. To address this growing concern, organizations must prioritize security and adopt comprehensive strategies. These strategies should include measures such as network segmentation, device hardening, traffic monitoring, anomaly detection, and scalable infrastructure design. Additionally, engaging with specialized DDoS mitigation services can provide an additional layer of protection. By taking proactive measures and implementing robust security practices, businesses can enhance the resilience of their IIoT networks and protect themselves against the potentially devastating impact of DDoS attacks.

Problem

In the ever-evolving landscape of Industry 5.0, the integration of Industrial Internet of Things (IIoT) technology has undeniably transformed industrial environments, bringing about increased flexibility, efficiency, and the ability to gather precise and continuous data for improved monitoring and prevention of hazardous situations. However, these advantages are accompanied by challenges, particularly in the realm of IoT device security. Manufacturers often prioritize low-cost IoT devices, potentially sacrificing security features. The inherent limitations of these devices, including processing power and resource constraints, make it challenging to implement robust security measures without impacting performance and escalating costs.

The fifth industrial revolution marks a new era where humans and machines collaborate closely. It emphasizes customization and the deployment of cooperative robots, liberating workers to focus on tasks that add significant value for customers. This evolution transcends traditional manufacturing, enhancing resilience, prioritizing human well-being, and championing sustainability. In the context of Industry 5.0, security in manufacturing is based on confidentiality, integrity, and availability. It's essential to protect the confidentiality of data exchanges, guarantee the reliability and precision of information throughout the entire product lifecycle, and safeguard against disruptions from cyber or physical threats.

When employing numerous smart devices within industrial settings, vulnerabilities can arise. One prevalent form of cyber assault is known as a Distributed Denial of Service (DDoS) attack. Visualize these devices as akin to individual soldiers, relatively low in computational prowess and often situated in less secure locations. Adversaries can exploit these vulnerabilities, commandeering these devices to inundate the system with an overwhelming volume of data, rendering it incapable of functioning effectively.

In summary, while IIoT technology brings numerous advantages to industrial environments, addressing security concerns and designing robust systems are essential to fully harnessing its potential and ensuring the reliability, integrity, and availability of production systems in the Industry 5.0 era. For this reason, the goal of this research is to create a distributed scheme that can detect and block DDoS attacks in the IIoT environment by locating and neutralizing the attacks near their sources. The development of DDoS mitigation methods in a distributed and coordinated manner is essential to address the limitations of existing solutions, such as inadequate detection efficiency, a high false alert rate, significant time delays, and substantial computational requirements and costs.

Proposed Solutions

To address the intricate security requirements inherent in Industrial Internet of Things (IIoT) systems, we advocate for the utilization of deep learning methodologies to augment the capabilities for detecting distributed denial of service (DDoS) attacks. Our methodology encompasses an exhaustive exploration of diverse neural network architectures, encompassing traditional convolutional neural networks (CNNs) as well as hybrid models incorporating long-short-term memory (LSTM) networks in conjunction with CNNs and CNNs combined with gated recurrent units (GRUs) and LSTM. The effectiveness of each model is meticulously assessed based on accuracy and loss metrics, employing the Edge-IIoTset-2022 dataset for comprehensive training and evaluation. This innovative approach presents a pioneering solution to the multifaceted security challenges prevalent within IIoT systems by harnessing the computational power of deep learning architectures. Our research endeavors contribute significantly to the advancement of a resilient security framework finely tailored to the intricate dynamics of IIoT environments. This framework not only serves to safeguard critical data but also fosters seamless information interchange among interconnected systems. Moreover, our methodology imparts notable enhancements in the real-time detection of DDoS attacks, thereby fortifying the resilience and integrity of IIoT infrastructures.

Document Plan

Our thesis is structured into two principal chapters. Initially, the "State of the Art" chapter conducts an extensive exploration of the domain relevant to critical infrastructure, with a particular focus on the Industrial Internet of Things (IIoT) and its susceptibility to Distributed Denial of Service (DDoS) attacks. This chapter encompasses a comprehensive review of pertinent literature, an examination of methodologies, and an analysis of the challenges faced in safeguarding IIoT environments against DDoS threats. Following this, the "Contributions" chapter articulates our distinctive contributions towards the mitigation of DDoS attacks within IIoT frameworks. It provides in-depth explanations, empirical evidence, and comparative analyses to substantiate the effectiveness of our proposed methodologies. The thesis culminates with an all-encompassing summary that consolidates our scholarly exploration, emphasizing the principal findings, their implications, and prospective avenues for future research in the defense against DDoS attacks in IIoT systems.

Chapter 1

State of the Art

Introduction

In today's interconnected world, where smart devices and machines seamlessly coordinate complex industrial tasks, Industrial Internet of Things (IIoT) security is a strong shield against digital dangers and malicious actors. Like we secure our homes with locks and alarms, IIoT security includes various tools, protocols, and practices to safeguard industrial machinery and systems from cyber threats. Additionally, Deep Learning represents a sophisticated aspect of artificial intelligence that demands our attention. To fully understand, We'll dive into its details in this chapter.

1.1 Project Context and Area

Regarding the threats facing the Industrial Internet of Things (IIoT) and Industrial Control Systems (ICS), it's like navigating a digital battlefield. The ICS CERT landscape report by Kaspersky [2] shows that 34 percent of Industrial Internet of Things (IIoT) and Industrial Control Systems (ICS) computers were detected and blocked in the first half of 2023. The second quarter of 2023 saw the highest quarterly level of threats globally since 2019, impacting 26.8 percent of ICS computers. The report also indicates an increase in cyber threat detections in high-income countries.

As we can observe in Figure 1.2 [2], among various industries, building automation stands out as the most vulnerable sector to cyber threats, representing 38.5 percent of computer attacks. Furthermore, the energy, oil, and gas industries witnessed a significant 36 percent increase in threats. On the other hand, the engineering, ICS integration, manufacturing, and energy sectors observed a general rise in the number of blocked malicious objects during the first half of 2023.

The Internet of Things (IoT) and the Industrial Internet of Things (IIoT) have revolu-

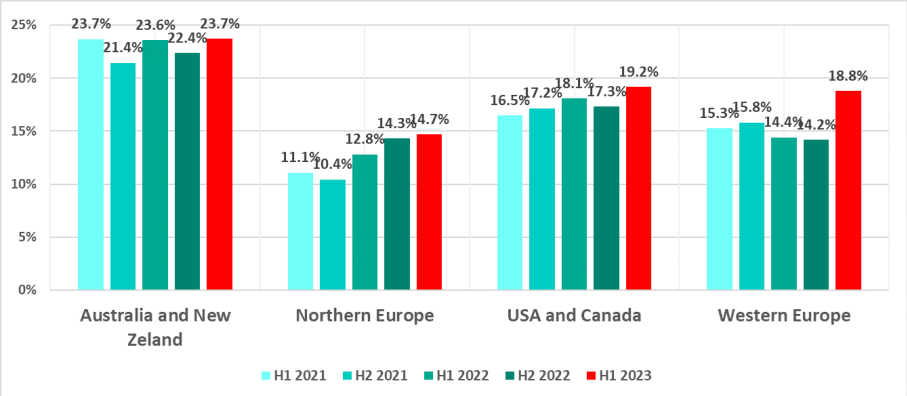


Figure 1.1: Percentage of ICS computers on which malicious objects were blocked in selected regions[2].

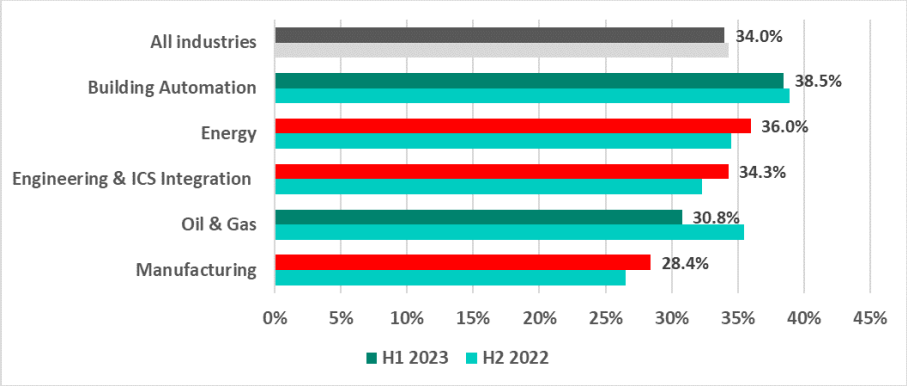


Figure 1.2: Percentage of ICS computers on which malicious objects were blocked in selected industries[2].

tionized connectivity, but they also pose cybersecurity challenges. What is IoT? What is the concept of the Industrial Internet of Things? And what is the architecture of the IIoT and industrial communication layers? What are the security requirements of the IIoT? In the next section, we will examine all of these questions.

1.2 Internet of Things

The Internet of Things (IoT) is a concept that encompasses all objects currently connected to the Internet and how they communicate with each other or with people. While wireless sensors and smart home devices are often the first things people think of when considering IoT, it extends far beyond that. Primarily, IoT revolves around vast amounts of data and the methods for processing and communicating this data across networks. Billions of devices are now interconnected, generating terabytes of data daily. The correct architecture is essential for managing such immense volumes of data.

1.2.1 IoT architecteur

There isn't a universally agreed-upon architecture for IoT; instead, various researchers have proposed different architectures[11]. In Figure 1.3, it is possible to see some of the most common IoT architectures used. Instead, there are several models with different numbers of layers, each suited to various needs. The 3-layer model is the simplest, focusing on data collection, transmission, and application services. The 4-layer model adds a processing level for data management. The most complex model, the 5-layer model, includes a business layer for strategic analysis and decision-making. These structures are flexible to accommodate the wide array of IoT applications and to prevent reliance on a single vendor's platform.

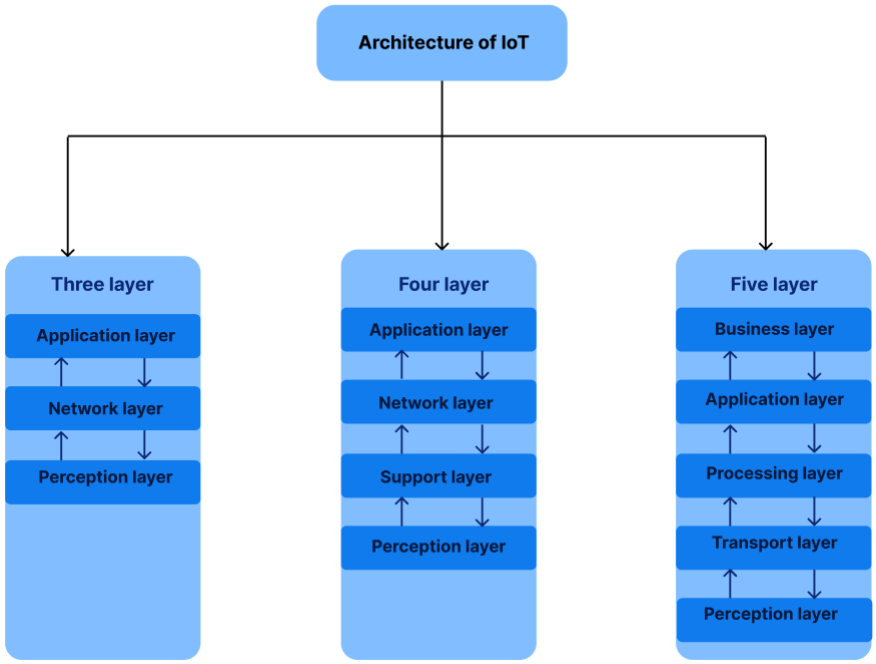


Figure 1.3: The layered architectures of IoT (three, four and five layers).

1.2.2 The Benefits of IoT

The Internet of Things (IoT) offers a wide range of benefits across various industries and aspects of daily life. Here are some key advantages of IoT:

Efficiency and Automation: The IoT can automate and optimize industrial processes, reducing operational costs. In residential contexts, IoT devices can automate tasks, making homes more energy-efficient and convenient.

Data Collection and Analysis: IoT devices generate large amounts of data that can be used for data analytics to make informed decisions, predict trends, and optimize processes.

Cost Savings: Organizations can reduce energy consumption, maintenance, and operational efficiency costs by automating processes and optimizing resource usage.

Enhanced Productivity: IoT can significantly enhance productivity in industrial settings by improving production processes, reducing downtime, and optimizing resource utilization.

Improved Quality of Life: IoT devices enhance healthcare by remotely monitoring patients, providing timely alerts, and enabling personalized services, while smart city applications improve urban quality of life through intelligent traffic and waste management.

Innovation and New Business Models: IoT offers new opportunities for innovation and revenue streams for businesses.

Real-time Monitoring and Control: IoT devices can monitor equipment, assets, and processes in real-time, enabling immediate response to issues and the implementation of preventive measures.

Remote Management and Maintenance: IoT facilitates remote management and maintenance of devices and systems, reducing the need for on-site interventions and enhancing scalability.

While the benefits of IoT are substantial, widespread adoption raises concerns about security, privacy, and standardization. Addressing these challenges is crucial to fully realizing the potential of the Internet of Things.

The factors driving the popularity of consumer IoT devices, such as wireless connectivity, cloud computing, affordable sensors, and advanced AI, are also influencing the evolution of the Industrial Internet of Things (IIoT), extending IoT beyond just home devices like voice-activated speakers and smart thermostats. This technological advancement is reshaping major sectors such as manufacturing, energy, mining, and transportation, with a projected multi-trillion-dollar impact on the overall economy[12].

1.3 Industrial Internet of Things (IIoT)

The fusion of advanced technology with industrial practices within the Industrial Internet of Things (IIoT) is sparking a transformative wave, boosting productivity, honing operational efficiency, and unlocking new possibilities. In our forthcoming discussion, we'll examine the profound impact of IIoT in diverse industries.

1.3.1 Definitions

A definition for the IIoT would be, "The Industrial Internet of Things (IIoT) is the use of Internet of Things (IoT) technologies in manufacturing."[4]. The Industrial Internet of Things (IIoT) integrates IoT into industrial sectors, focusing on machine-to-machine communication, big data, and machine learning to enhance efficiency and reliability in operations like robotics, medical devices, and software-defined production processes.

IIoT goes beyond typical consumer devices, and the ability to connect physical devices is often associated with IoT. What makes it different is the intersection between information technology (IT) and operational technology (OT). OT refers to the interconnection of operational processes and industrial control systems (ICS), including human-machine interface (HMI), supervisory control and data acquisition (SCADA) systems, a distributed control system (DCS), and a programmable logic controller (PLC)[1].In summary, the convergence of IT and OT in the IIoT empowers industries with advanced capabilities, delivering improved system integration, operational optimization, and enhanced management of physical infrastructures.

1.3.2 Industrial Revolutions

The evolution of industrial revolutions has significantly shaped manufacturing, beginning with steam-powered mechanization in Industry 1.0 and advancing through the integration of electricity, automation, and electronics. In the fourth industrial revolution, Industry 4.0, the Industrial Internet of Things (IIoT) transforms cyber-physical systems and production processes using big data and analytics.[1]. However, the vision of Industry 5.0 aims to merge human expertise with resilient, intelligent, and precise machines. Many scientific pioneers believe that Industry 5.0 has the potential to revitalize the foundational aspects of the manufacturing industry.[32].

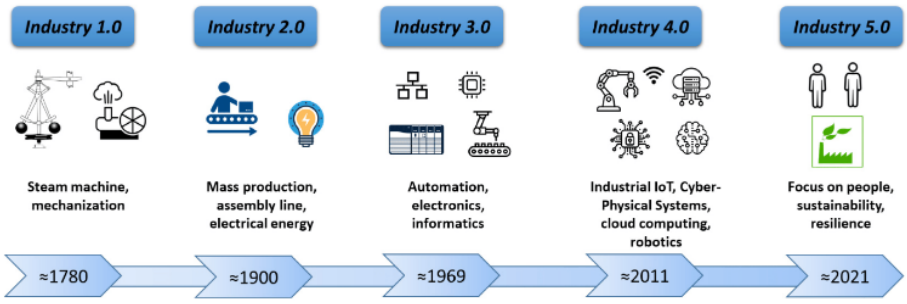


Figure 1.4: Timeline of industrial revolutions from Industry 1.0 to Industry 5.0[17].

The Industrial Internet of Things (IIoT) is transforming sectors with advanced technology to optimize operations, predict maintenance, and enable real-time decision-making.

1.3.3 IIoT Architecture

The Industrial Internet of Things (IIoT) constitutes a unified system characterized by the intelligence and interconnectedness of diverse elements, including devices, sensors, actuators, processors, network devices, and transceivers, within the broader IoT framework. The basic three-layer architecture connects devices through physical, network, and application layers. Adding a support/middleware layer enhances data exchange and control. A five-level architecture includes a business layer, which is applied to manage the complete IoT system, business applications, profit models, and users' private data. Figure 1.5[5] represents the layered IIoT architecture. This includes the physical components in the first layer, communication devices in the second layer, the data storage unit in the support layer, and the interactive layer in the fourth level. A detailed description of each layer is given below.

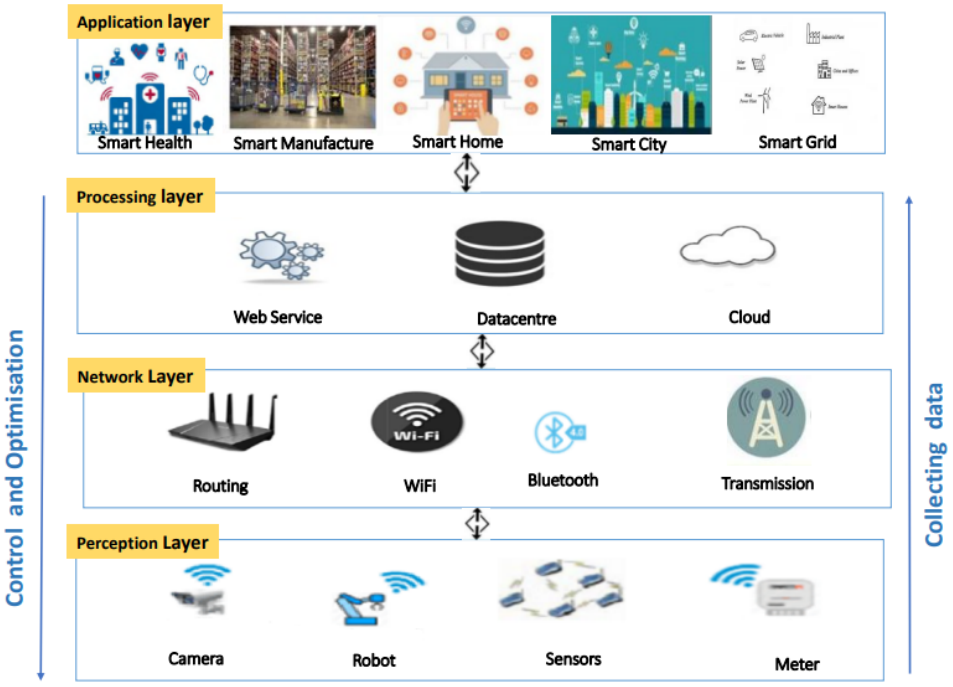


Figure 1.5: IIoT layered architecture, technology[5].

Perception Layer: The Perception Layer, comprising physical and sensor devices for environmental data collection, is highly sensitive and vulnerable to security threats like node capture, eavesdropping, replay attacks, fake node inclusion, and timing attacks.[10; 23]

Network Layer: The Network Layer enables data transmission among smart devices, network equipment, and servers via wired or wireless media, acting as a crucial link between presentation and application layers. Highly sensitive, it's vulnerable to severe

threats like Denial of Service (DoS) and Man-in-the-Middle (MITM) attacks, along with various other network-related vulnerabilities, posing potential risks[10; 23].

Application Layer: Defines a range of applications utilized to control and monitor connected devices. Positioned as an intermediary layer between the connected devices and the user, it serves as a mediator, facilitating communication between end nodes and the network [23].

Support Layer: The security of the three-level architecture becomes compromised when information is directly transmitted to the network layer, creating vulnerabilities to various threats. A new support layer has been introduced to address these shortcomings and enhance protection against risks, resulting in a four-level architecture. In this setup, data from the perception layer undergoes authentication via pre-shared secret keys and passwords before reaching the network layer. However, it's important to note that the support layer remains vulnerable to attacks like DoS, malicious insider actions, and unauthorized access.[10; 23].

In 2017, the Industrial Internet Consortium introduced the Industrial Internet Reference Architecture (IIRA) as a working framework for IIoT. This architecture comprises five key domains: control, information, operation, application, and business. Illustrated in Figure 1.6[23], it is organized into three tiers: the edge tier for physical and control devices, the platform tier for information and operations management, and the enterprise tier for controlling the application interface. The control domain orchestrates communication between the physical system and input devices (sensors, actuators), collecting and transmitting data to the information domain. The information domain serves as a platform for data transformation and distribution. The operational domain manages metadata and oversees applications and portals for interaction, while the application domain provides logic and rules for accessing information and controlling data flow from the business domain[23].

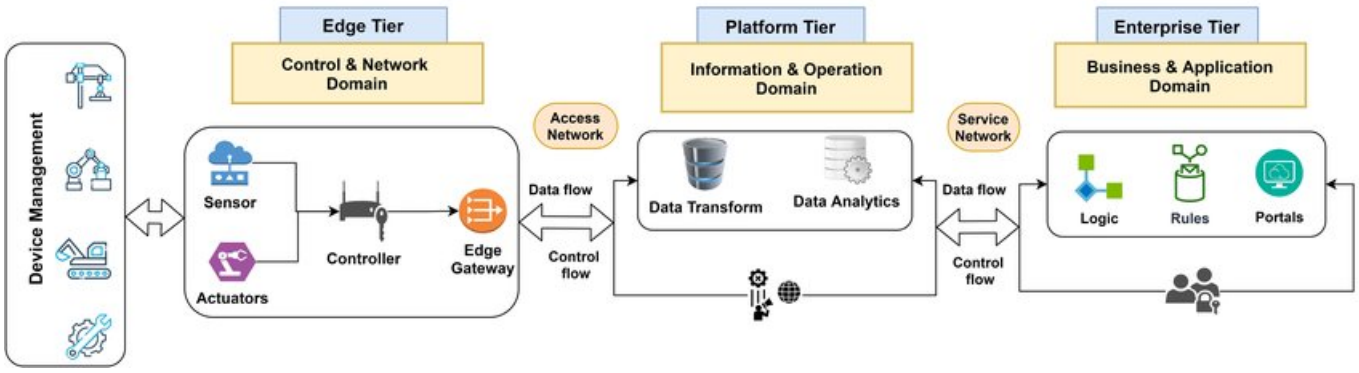


Figure 1.6: Three Tier IIoT System architecture[23].

1.3.4 Industrial Communication Layer

The intricate nature of the IIoT ecosystem underscores the importance of addressing security requirements for industrial connectivity and communication protocols within a three-tier architecture. An abstract three-tier IIoT architecture is introduced, systematically categorizing the key components prevalent in most IIoT developments (refer to figure 1.7) [36].

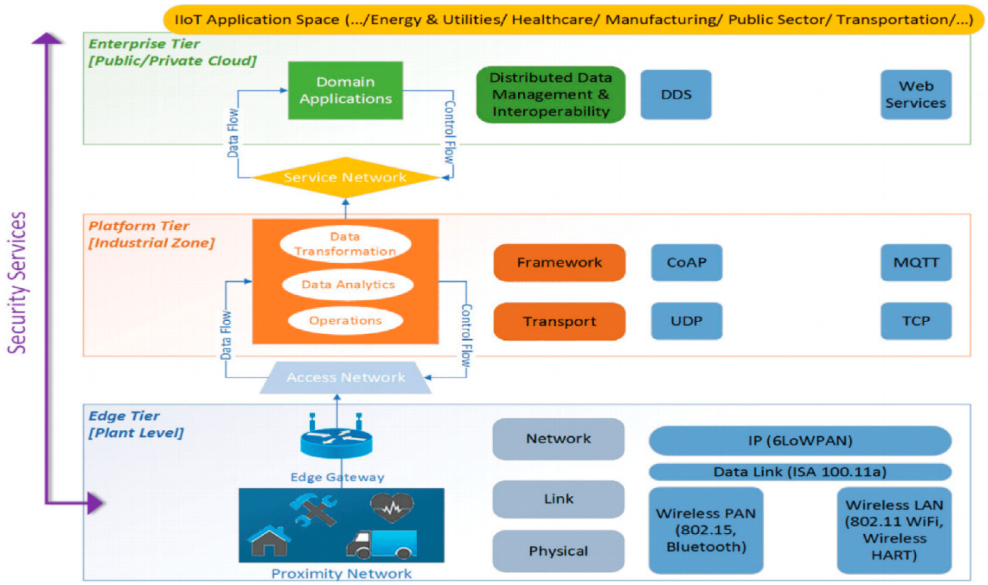


Figure 1.7: Three-tier architecture of IIoT connectivity and communications standards[36].

The Edge Tier comprises end-points and edge-based gateway devices connecting sensor devices, actuators, and control systems, enabling internal and layered communications with the Platform Tier. The Platform Tier, linked to the internet or mobile network, hosts service-based and middleware applications such as analytics and data transformation. The Enterprise Tier interfaces with the Internet-based service network, supporting high-level services like enterprise applications, cloud computing, domain services, and hosting. End users engage with the network through specially designed interfaces.[36].

T. Gebremichael et al. introduced a framework that outlines connectivity protocols per level alongside security measures to ensure the secure deployment of devices within IIoT networks. This architectural approach facilitates the distribution of security requirements across various network segments and establishes backup protections against large-scale breaches [36]. Specifically, within the Edge Tier, connectivity protocols such as Bluetooth [IEEE 802.15.1] (WPAN), ZigBee 802.15.4 (WPAN), IEEE 802.15.4 (WPAN), NB-IoT (WWAN), WirelessHART (WLAN), LoRaWAN (WWAN), ISA100.11a (Data Link), and 6LoWPAN (Network Layer) are categorized and explained. Meanwhile, the Platform Tier encompasses connectivity protocols like CoAP and MQTT[36].

1.4 Industrial Internet of Things (IIoT) Security: IIoT Security Requirements

In this section, we outline the overarching security criteria that need to be met by every communication system, encompassing IIoT environments.

1.4.1 CIA Triad

The CIA triad, a renowned information security model, serves as a foundational framework for security goals. It comprises three essential requirements:

Confidentiality: This involves safeguarding information in any format through methods like access control, encryption, network isolation, and privacy[6].

Integrity: Aims to ensure consistency, authenticity, and accuracy in IIoT entities, fostering trust with other entities[6].

Availability: ensures the continuous and efficient operation of the system at all times, employing methods such as decentralization and redundancy[6].

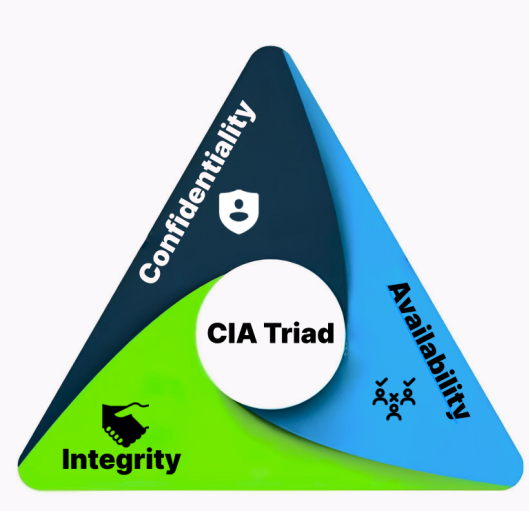


Figure 1.8: CIA triad.

1.4.2 Authentication

IIoT environments require lightweight authentication mechanisms to accommodate power, storage, and processing limitations. Blockchain-based authentication offers a balance between lightweight solutions and data integrity, which is crucial for edge computing environments.[6].

1.4.3 Access Control and Authorization

Access control ensures that IIoT devices are granted appropriate permissions to access network resources based on their privileges. Administrators typically manage an authorization database to determine access levels[13].

1.4.4 Resilience and Maintainability

IIoT systems need to maintain operations even under adversarial conditions. This can be achieved through diversity, redundancy, or system hardening. Additionally, maintaining software on devices is crucial to protect against cyberattacks[38].

1.4.5 Privacy

Protecting data privacy is essential in IIoT environments, where large amounts of data are generated and stored. Encryption methods like attribute-based searchable encryption can help securely store and retrieve data without compromising user privacy[8].

1.4.6 Security Monitoring

IDSs are well-known tools that provide dynamic security monitoring of system behavior. They can spot and handle threats that aim at networks[39]. Any network, especially those with IIoT devices, needs to watch over its communications, detect intrusions, and react to them. IDSs are important because some old and vulnerable devices (i.e., those that cannot be easily updated to fix known flaws) may join the network and need constant security monitoring. These devices may be attacked by a DDoS and become part of a botnet that can harm other IIoT devices in the network.

1.4.7 Secure Data Sharing

IIoT systems often involve distributed computing and storage, raising concerns about data security, privacy, and scalability. Effective protocols, such as secure peer-to-peer and group communication frameworks, are essential for safeguarding confidential data and facilitating safe data sharing among IIoT devices.

1.5 IIoT Attack Categories

This section explores the complexities of the three layers within the IIoT layer architecture: perception, network, and application. Each layer exhibits distinct technologies and

characteristics, which are pivotal in the operation of IIoT systems. Furthermore, it examines the security challenges encountered by IIoT applications at each layer, shedding light on the inherent vulnerabilities within these domains. Additionally, Table 1.2 offers insight into common attacks aimed at the three layers of IIoT, providing a comprehensive overview of the potential threats confronting industrial IoT ecosystems..

Layer	Attack Type	Description
Perception	Node Capture Attacks	Physically obtaining, replacing, or modifying IIoT nodes or hardware[6].
	Jamming Attacks	Disrupting or intercepting IIoT device communication by interfering with wireless access[22].
	Sleep Deprivation Attacks	Preventing IIoT devices from entering sleep mode by inserting looping code or hardware modifications[6].
	Replay Attacks	Exploiting authentication systems within the IIoT environment[24].
Network	Eavesdropping Attacks	Unauthorized interception of message exchanges between IIoT devices[34].
	Sybil and ID Cloning Attacks	Spoofing legitimate node identities to access more devices in the network.
	Wormhole Attacks	Creating virtual long-distance tunnels to force network traffic through[25].
	Denial of Service (DoS) Attacks	Sabotaging bandwidth or resources to disrupt IIoT services.
	Man in the Middle Attacks	Intercepting and altering communications between legitimate IIoT nodes.
Application	Malicious Code Injection Attacks	Exploiting debug module vulnerabilities to inject harmful code into IIoT devices[3].
	Cross-Site or Malicious Scripts Attacks	Exploiting vulnerabilities via malicious scripts on visited websites to infect IIoT systems.
	Malware Injection Attacks	Targeting service requests to inject malware into IIoT devices or networks.
	Data Distortion Attacks	Intercepting and distorting wireless packets transmitted between IIoT entities.
	SQL Injection Attacks	Exploiting application vulnerabilities to modify SQL queries and gain unauthorized database access.
	Ransomware Attacks	Hijacking IIoT devices or files and demanding payment for access restoration.
	Side-Channel Attacks	Using publicly available data to infer confidential information on edge computing infrastructure.
	Authorization and Authentication Attacks	Utilizing fake credentials to gain unauthorized access to protected resources in edge computing environments.

Table 1.1: IIoT Security Attacks by Layer

1.6 Distributed Denial of Service(DDoS) on IIoT

DDoS attacks on IIoT (Industrial Internet of Things) devices are a growing threat that can cause serious damage to systems and services. These attacks occur when numerous infected computers, known as bots, flood a target service, such as a website or a system, with excessive data, as depicted in Figure 1.7. This overwhelms the service, making it slow or unavailable. Attackers may have various motives, including financial gain, competition, or political reasons. The data sent by the bots appears similar to normal data from regular users, making it challenging to distinguish them. While there are ways to prevent, detect, or stop DDoS attacks, none are foolproof, and some DDoS attacks remain unstoppable.

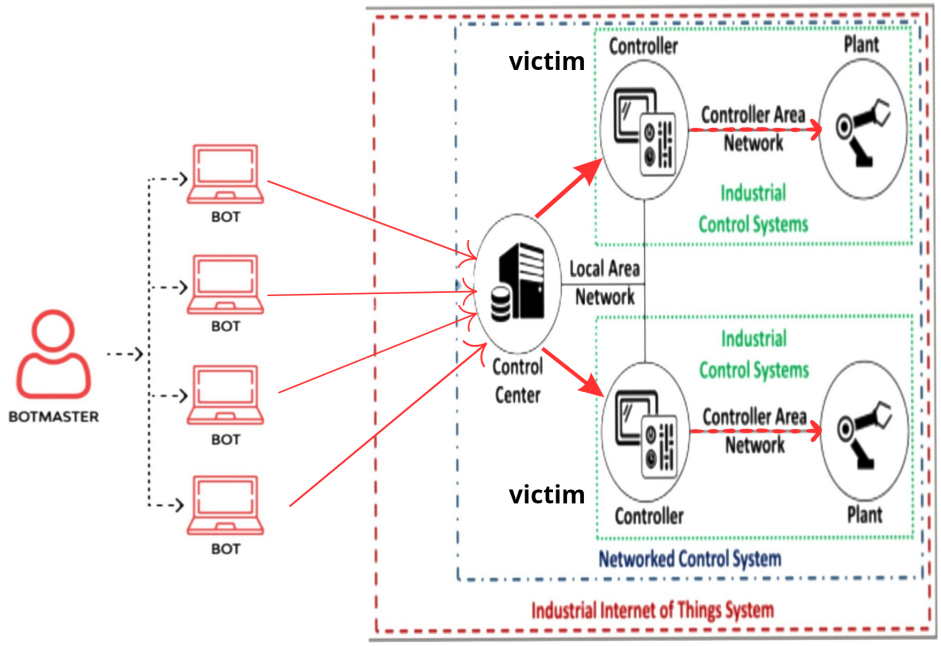


Figure 1.9: DDoS attack on IIoT system.

1.6.1 Types of DDoS Attacks

Different types of DDoS attacks can affect IIoT devices. Some common types are:

1.6.1.1 UDP flood attacks

UDP flooding is a common attack method that utilizes the Internet protocol UDP (User Datagram Protocol), differing from TCP in its lack of a handshake for communication establishment, progress, and completion verification. In this attack, the attacker floods the target server with a large number of UDP packets directed to random ports. The server must respond and check for a listening application on the target port. If none is detected, it sends ICMP packets to indicate unavailability and comply with communication rules. This

overload drains the targeted server's internet connection and other resources. Effectively mitigating UDP floods is challenging as merely limiting the number of processed packets may not fully address the impact on server communication lines[28]. This challenge has contributed to the prevalence of attacks and made defense efforts challenging for server owners.

1.6.1.2 TCP flood attacks

A TCP SYN flood attack exploits the TCP protocol's feature for secure data transmission by inundating a target server with SYN (synchronization) requests but never completing the handshake. Normally, TCP involves a three-way handshake: SYN, SYN-ACK, and ACK. In this attack, the sender sends SYN requests but doesn't respond with the final ACK message after receiving the SYN-ACK from the receiver. This leaves the connection half-open, consuming server resources and preventing it from accepting new connections, potentially leading to malfunction or crash[31].

1.6.1.3 ICMP flood attacks

An ICMP flood attack, also known as a Ping flood attack, is a method of launching a DoS attack by inundating a target device with a large number of ICMP echo requests. ICMP is commonly used for network connectivity testing, but excessive echo requests can overwhelm the target, slowing it down or rendering it unreachable, thereby impacting network performance. The attacker requires knowledge of the target's IP address or network structure to execute this attack, and tools like blind ping can assist in finding target devices. If multiple devices are used to send ICMP echo requests simultaneously, it can escalate into a DDoS attack, generating even more traffic and causing greater damage[9].

1.6.1.4 HTTP flood attacks

HTTP Flood Attacks are a type of DDoS attack targeting web servers with a massive influx of HTTP requests. This flood of traffic overwhelms the server's processing capabilities, depleting CPU, memory, and bandwidth resources, resulting in degraded website performance or complete unavailability. Variants include Layer 7 HTTP Flood, targeting specific URLs; Slowloris Attack, maintaining numerous open connections to exhaust server resources; and HTTP POST Flood, bombarding the server with numerous POST requests. While each variant operates differently, they share the goal of disrupting normal web service operations.

1.6.2 DDoS Attack Process

The DDoS attack process consists of three phases, as shown in Figure 1.10:

Phase 1: Target acquisition The attacker chooses a target based on various motives, such as political or financial gains, personal grudges, cyber warfare, etc. The attacker then collects security details and other information about the target to plan the next phases.

Phase 2: Groundwork The attacker scans the Internet for vulnerable systems and compromises them using various exploits. These compromised systems, also known as bots or zombies, form a large network called a botnet. The attacker controls the botnet through multiple layers of intermediary systems, called stepping stones, to hide their identity.

Phase 3: Attack The attacker sends commands to the botnet through different network channels to launch an attack. The bots flood the target with a large amount of traffic, disrupting its normal functioning and denying service to legitimate users. The attack lasts until the attacker stops it or the target mitigates it.

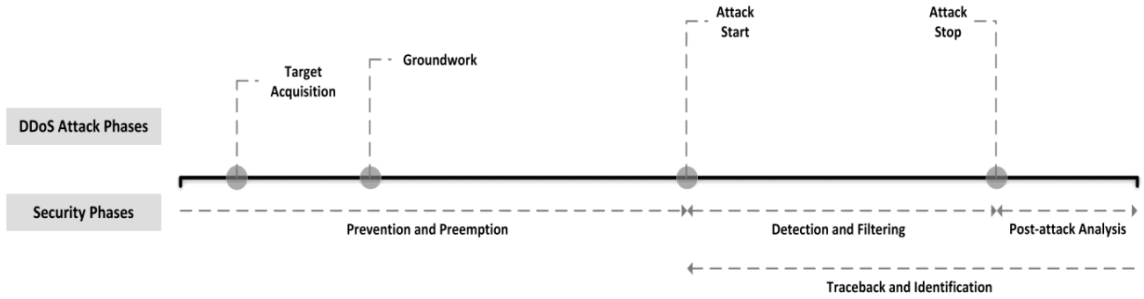


Figure 1.10: Different attack and security phases[35].

1.6.3 DDoS threats

DDoS attacks pose a serious risk for organizations across various industries and sizes. The following are some of the possible outcomes of a successful attack:

Financial losses: A DDoS attack can cause reduced productivity, service interruptions, violations of service level agreements, and expenses for mitigation and recovery.

Operational disruption: A DDoS attack can hinder an organization from performing its essential functions or limit the access of customers to its services.

Reputational damage: DDoS attacks can result in customer attrition who opt for competitors if they experience difficulties accessing the website of an organization or question its ability to deliver products and services.

1.7 Comprehensive Approaches to DDoS Mitigation in IIoT

This section examines a holistic strategy that integrates three formidable defense pillars: intrusion detection systems (IDS), deep learning, and federated learning. Through the synergistic amalgamation of these techniques, our objective is to fortify the resilience of IIoT networks and ensure continuous operations.

1.7.1 Intrusion detection system for IIoT

Intrusion Detection Systems (IDS) play a crucial role in safeguarding critical infrastructure by meticulously monitoring and analyzing activities within computer systems or networks to detect potential security breaches. The continuous operation of critical infrastructures (CIs) is essential for societal well-being, and IDSs serve as a fundamental defense mechanism for CIs, prioritizing readiness and proactive security measures.

1.7.1.1 Definition of Intrusion detection system

An intrusion detection system (IDS) is software designed to monitor network traffic for signs of malicious activities or policy breaches. It actively observes a network or system, promptly alerting administrators upon detecting suspicious transactions. By learning from patterns, the IDS constructs a predictive model, essentially a classifier, adept at discerning between legitimate ('good') connections and potentially harmful ('bad') ones, such as intrusions or attacks[18].

1.7.1.2 Working of Intrusion Detection System(IDS)

An intrusion detection system (IDS) monitors network traffic for suspicious activity, analyzing data for irregular patterns or abnormal behavior. It compares network activity against predefined rules and patterns to identify potential attacks or intrusions. When a match is detected, the IDS alerts the system administrator, who investigates and takes necessary actions to mitigate damage or intrusion, ensuring network security.

1.7.1.3 Intrusion detection system classification

IDS can be classified from the perspective of its deployment or detection methods. A classification taxonomy is provided in Figure 1.11.

Host-based IDS: A host-based intrusion detection system (HIDS) is a cyber security measure that functions at the level of individual hosts to identify and thwart unauthorized access or malicious actions[7].

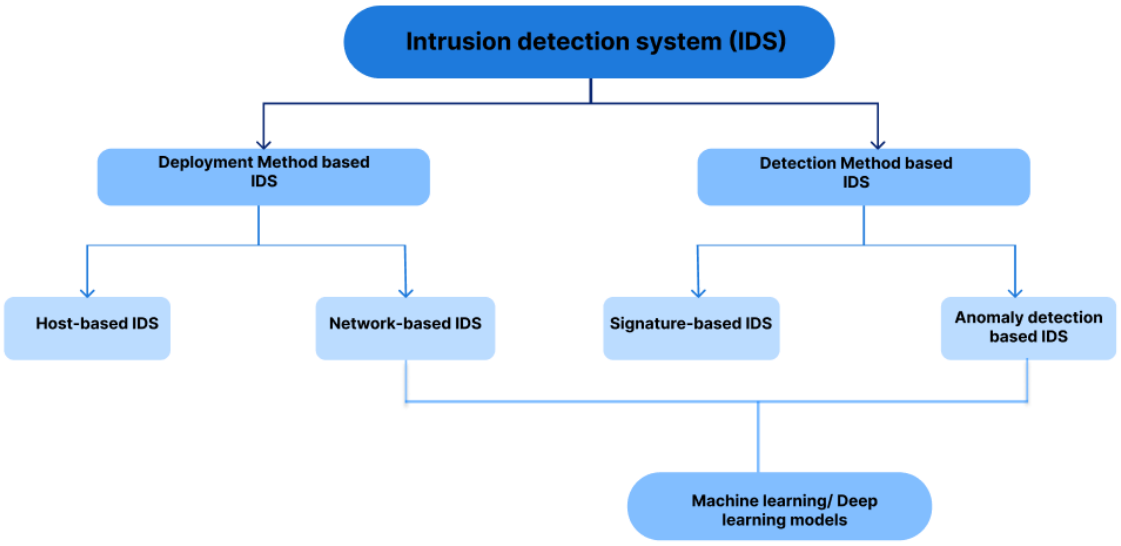


Figure 1.11: Intrusion detection system classification taxonomy.

Network-based IDS: A network-based intrusion detection system (NIDS) is a security solution that scrutinizes network traffic for any signs of suspicious activities or irregularities. Unlike host-based IDS (HIDS), which focuses on monitoring activities on individual devices, NIDS specifically examines the traffic moving across the network[7].

Signature-based IDS: Signature-based IDS effectively identifies known attacks through predefined patterns but cannot detect unknown or newly released threats[7].

Anomaly detection-based IDS: This detection method categorizes network activity as normal or anomalous based on rules or heuristics, rather than patterns or signatures. Effective implementation requires a deep understanding of the network's typical behavior[7].

In essence, IDSs are vital for protecting IoT and IIoT systems by identifying and mitigating security threats. As these systems advance, ongoing research strives to improve intrusion detection abilities and address distinct challenges.

1.7.2 Deep Learning

Deep learning, a subset of machine learning, employs multi-layered neural networks, known as deep neural networks, to emulate the intricate decision-making capabilities of the human brain. The majority of artificial intelligence (AI) applications in our daily lives are driven by some variant of deep learning[21].

1.7.2.1 Deep Learning for IIOT:

Deep learning (DL) is a powerful subset of machine learning (ML) that is transforming manufacturing by using its layered structure to process large datasets. DL is particularly adept at processing complex sensory data, which improves manufacturing efficiency. Its self-learning abilities, along with its skill in recognizing patterns and making decisions, highlight its importance. DL simplifies the feature extraction process by automatically identifying important features from data, eliminating the need for separate feature extraction algorithms. The advantages of DL over traditional ML methods in the context of the Industrial Internet of Things (IIoT) are detailed in Figure 1.12.

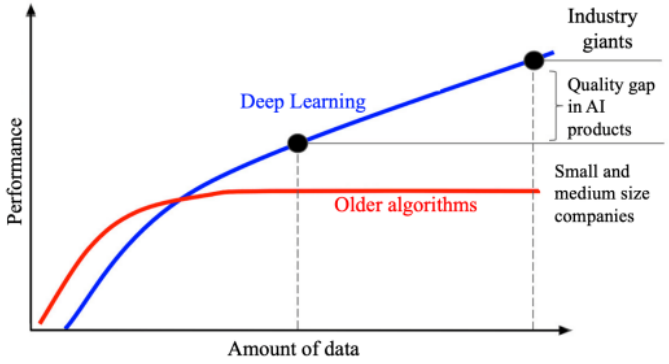


Figure 1.12: Comparison of DL with traditional algorithms for IIoT[27].

1.7.2.2 Deep Learning Methods

Deep learning methods refer to a variety of techniques within the field of artificial intelligence that utilize multi-layered neural networks to process and analyze complex data.
Convolutional Neural Network

Convolutional neural networks (CNNs), or ConvNets, are a type of neural network designed for grid-patterned data like images. They are known for image recognition and classification, as well as learning patterns and features directly from data. CNNs are a specialized category in deep learning with layers that process and simplify high-dimensional data into a more manageable form. Their architecture includes convolutional, pooling, and fully connected layers that filter and reduce data dimensions, as shown in Figure 1.13.

Additionally, CNNs have recently been investigated and applied for analyzing sequential data in one dimension, accommodating diverse Industrial Internet of Things (IIoT) applications[27].

Recurrent Neural Network Recurrent neural networks (RNNs) were initially favored for sequence data modeling until attention mechanisms became popular. They had limi-

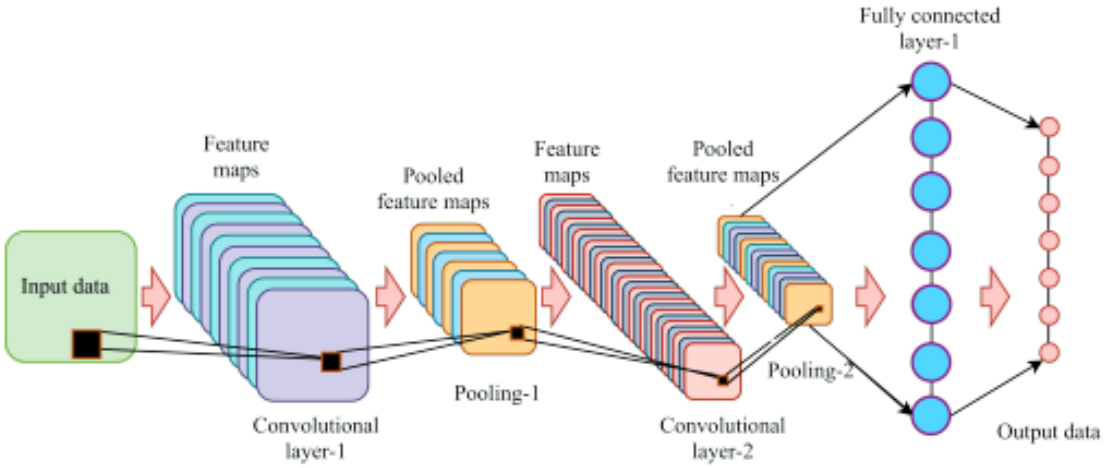


Figure 1.13: The generalized layer-wise structure of a convolutional neural network (CNN)[27].

tations, such as requiring unique parameters for each sequence part and struggling with sequences of different lengths. Advances like long-short-term memory (LSTM) and gated recurrent units (GRU) were developed to overcome these challenges. They handle issues like vanishing gradients well and are adept at identifying time-related patterns, making them essential for analyzing data with temporal connections[37].

Long short-term memory (LSTM)

In 1997, Hochreiter and Schmidhuber developed Long Short-Term Memory (LSTM) networks to improve upon traditional RNNs by addressing the issue of long-term dependencies[20]. LSTMs are designed to maintain information for extended durations, which is a significant advancement over the simpler structure of regular RNNs that typically contain a single tanh layer. The LSTM structure is considerably more intricate, often consisting of four distinct hidden layers, as shown in Figure 1.14.

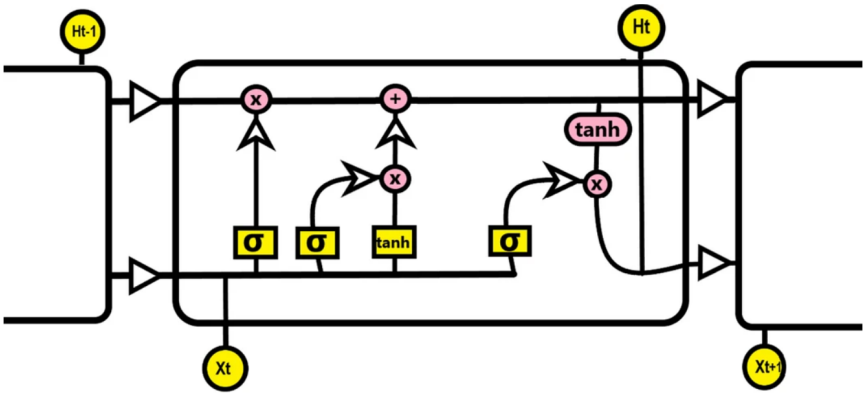


Figure 1.14: Long short-term memory (LSTM) architecture[29].

1.7.2.3 Federated Learning (FL)

Federated Learning (FL) is an effective approach for creating smart IIoT applications with a focus on affordability and data protection. It constructs sophisticated AI models by combining updates from multiple IIoT devices, bypassing the need to access their data directly. By tapping into the computational abilities and varied data of these devices, FL improves the training process and model precision. Additionally, FL offers a strong safeguard against DDoS attacks by facilitating joint model training without data exchange, thus maintaining privacy. In a standard FL-IIoT setup, data clients and an aggregator collectively train a universal model, keeping the actual data decentralized, as shown in Fig. 1.15, boosting efficiency while protecting data privacy.

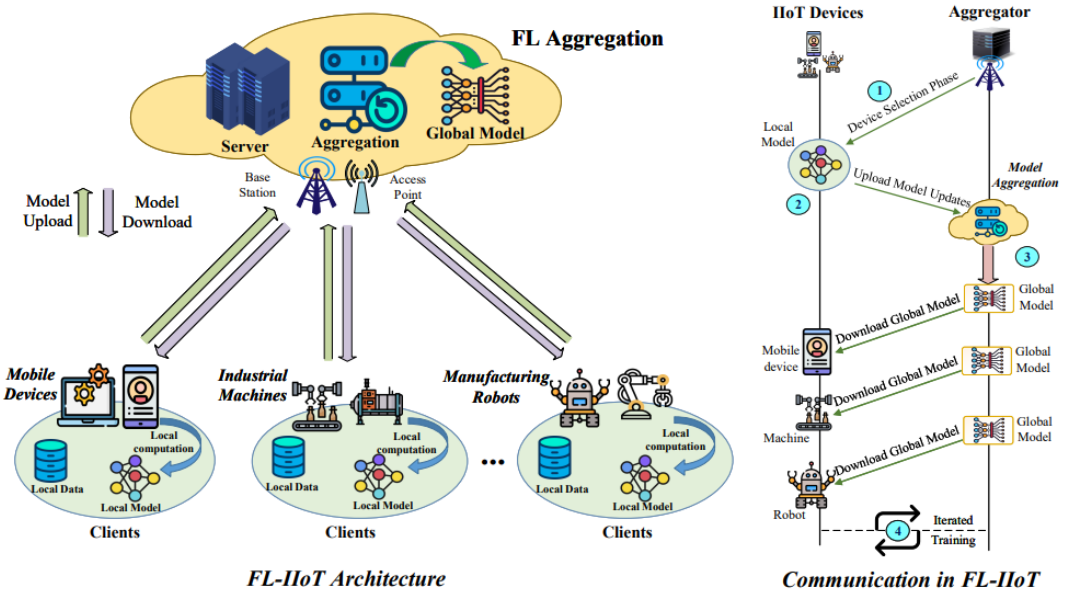


Figure 1.15: The network architecture and communication process for FL-IIoT[33].

FL for IIoT Attack Detection

As industrial devices become increasingly susceptible to cyber threats that can distort AI and ML model predictions, Federated Learning (FL) offers a promising defense. FL enables IIoT devices to collaboratively learn and improve security measures against diverse attacks, maintaining the integrity of industrial processes. In FL-based attack detection, each IIoT device independently trains a neural network to refine the defense model using adversarial examples. These local models send their updates to a central server, where they're combined to form a unified model. This cycle repeats until a robust attack detection model is established, significantly bolstering IIoT network security, as depicted in Figure 1.16.

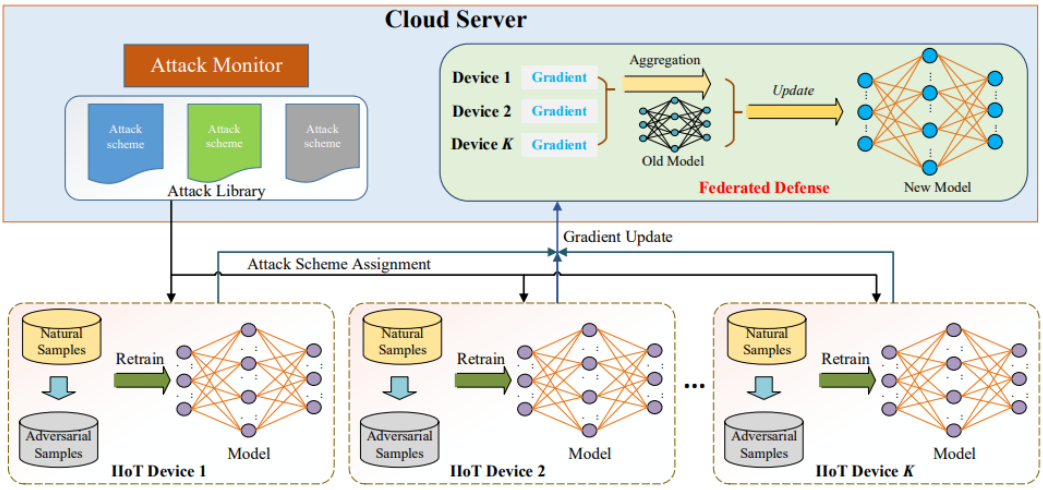


Figure 1.16: Federated attack detection and defense in FL-based IIoT networks[33].

1.8 Related Work

Section 1.8 delves into an extensive examination of research endeavors dedicated to intrusion detection systems (IDS) employing deep learning methodologies in the Industrial Internet of Things (IIoT) domain, with a focus on some distributed denial of service (DDoS) detection. Table 1.2 presents a summary of the discussed literature in this area.

The related work conducted by Kanimozhi et al[26]. focuses on utilizing artificial intelligence (AI) techniques to classify the CSE-CIC-IDS-2018 dataset, a widely recognized benchmark dataset for evaluating intrusion detection systems. Employing AI approaches, such as machine learning algorithms like decision trees, support vector machines, or neural networks, the study aimed to classify network traffic data and detect intrusions effectively. Notably, Kanimozhi et al. achieved a remarkable classification success rate of 99.97%, underscoring the high accuracy of their AI-based approach in identifying and categorizing network intrusions within the CSE-CIC-IDS-2018 dataset.

Sandeep et al[19]. developed a Deep Neural Network (DNN) for Network Intrusion Detection System (NIDS), comprising a sparse auto-encoder for unsupervised feature learning and logistic regression for binary classification on the NSL-KDD dataset (normal/intrusion). Initially taking 115 features as input, the sparse auto-encoder was employed to train and learn new features, subsequently reducing them to 50 and then to 10. These features were then fed into the logistic regression classifier. Performance evaluation was conducted in terms of accuracy, precision, and recall, with the model achieving an overall accuracy of 87.2%.

In their research, Ferrag et al[14]. explored various deep learning methods, including recurrent neural networks (RNN), deep neural networks (DNN), restricted Boltzmann ma-

chines (RBM), deep belief networks (DBN), convolutional neural networks (CNN), deep Boltzmann machines (DBM), and deep autoencoders (DA). They applied these methods to the CSE-CIC-IDS2018 and Bot-IoT datasets, comparing their classification performance and execution time. Additionally, the study examined intrusion detection systems based on deep learning methods, categorizing 35 attack detection datasets from existing literature for analysis.

Li et al[30]. propose a collaborative defense approach to tackle distributed denial-of-service (DDoS) attacks in industrial IoT (IIoT) setups. Their method combines fog/edge computing and federated learning. They employ the UNSW NB-15 dataset for training and evaluation, utilizing the FLEAM procedure to train a globally optimized model with datasets from multiple defenders. Their approach achieves a notable detection accuracy of 98%, highlighting its effectiveness against DDoS attacks.

Ferrag et al[16]. extensively explored federated deep learning's role in bolstering IIoT security. They employed RNN, CNN, and DNN architectures, comparing centralized and federated learning using Bot-IoT, MQTTset, and TON-IoT datasets. Results indicated RNN excelled in Bot-IoT (96.76%), while DNN lagged (95.76%). Conversely, DNN outperformed in MQTTset (90.06%), with RNN trailing (89.29%). In TON-IoT, RNN achieved peak accuracy (99.98%), while CNN fell short (98.87%). These findings underscore federated deep learning's efficacy and the importance of considering model variations across diverse IoT datasets, pivotal for enhancing IIoT security.

Author	Year	Methods	Dataset	Comments	Performance Metrics
Kanimozhi et al[26].	2019	SVM, RF, k-NN, ANN, Adaboost, NB	CSE-CIC-IDS2018	the reliance on a single dataset suggests that performance variations may occur in diverse datasets or real-world scenarios.	Accuracy=99.97%, Precision=1, Recall=1, F1-Score = 1
Sandeep et al[19].	2019	DNN	NSL-KDD	Reliance on specific datasets like NSL-KDD, while valuable for research, may not fully represent the complexity of real-world network traffic, potentially limiting the generalizability and real-world effectiveness of the intrusion detection system.	Accuracy=87.2%, Precision=84.6%, Recall=92.8%, Specificity=80.7%
Ferrag et al[14].	2020	DNN, RNN, CNN, RBM, DBN, DBM, DA	CSE-CIC IDS2018 BoT-IoT	While classification success and execution time comparison is informative, considerations like model interpretability, scalability, and robustness to adversarial attacks are also crucial for evaluating the suitability of deep learning methods in intrusion detection.	Accuracy, Time(s)
Li et al[30].	2021	GRU	UNSW NB-15	High accuracy in detecting DDoS attacks. Energy cost not calculated.	Accuracy= 98%
Ferrag et al[16].	2021	CNN, DNN, RNN	Bot-IoT, MQTTset TON_IoT	While the study covers a range of IoT datasets, the results may not fully represent the performance of federated deep learning methods across all possible IIoT environments and scenarios.	Accuracy, Precision, Recall, F1-Score

Table 1.2: Comparison of Different Intrusion Detection Approaches

1.9 Synthesis and Discussion

Section 1.9 offers a thorough examination and synthesis of the reviewed state-of-the-art. It focuses on IIoT security, particularly addressing DDoS attacks and defenses. The section provides a contextual understanding of IIoT within the broader IoT landscape, explaining its architecture and emphasizing the crucial role of the CIA Triad in protecting this interconnected ecosystem. It discusses security challenges, with a specific emphasis on DDoS threats, and highlights the importance of advanced mitigation techniques like deep learning. The conclusion acknowledges previous research endeavors and advocates for ongoing innovation to protect industrial cyberspace from evolving cyber threats. This summary emphasizes the need for continuous research and development in IIoT security to ensure the safety and resilience of industrial systems.

1.10 Conclusion

In conclusion, this chapter provides a comprehensive foundation for exploring IIoT security and mitigating DDoS attacks. It covers essential concepts like IoT, IIoT, and deep learning basics, equipping readers with the necessary knowledge for navigating industrial cybersecurity complexities. Highlighting the architecture of IIoT systems and the crucial role of the CIA Triad, the chapter underscores the challenges posed by DDoS attacks and emphasizes the significance of advanced mitigation techniques such as IDS and federated learning. As the cybersecurity landscape evolves, ongoing research and innovation are vital. Thus, the chapter advocates for sustained efforts to ensure the safety and resilience of industrial systems against emerging threats.

Chapter

2

Contributions

Introduction

Deep learning (DL) techniques are at the forefront of modern intrusion detection systems (IDS), particularly in identifying distributed denial of service (DDoS) attacks within industrial Internet of Things (IIoT) environments. This research builds upon the Edge-IIoTset dataset to develop advanced DL models. Initially, a CNN model was employed, which progressed to incorporate LSTM and GRU architectures, culminating in a hybrid CNN-LSTM-GRU model. This approach aims to enhance the system's ability to detect a broad range of attack patterns while maintaining a negligible false alarm rate. The effectiveness of these models was evaluated using key metrics such as accuracy, recall, and F1 score, crucial for robust DDoS detection. The following chapter will provide a detailed methodology outlining the development of these specialized IDS for DDoS defense.

2.1 Tools and environments

Deep learning requires substantial computational resources, notably Graphics Processing Units (GPUs), to handle intricate calculations. Initially, we set up a local Python [1] development environment using the Anaconda distribution. However, we promptly shifted to utilizing the cloud-based platform Google Colab for improved computational efficiency.

Anaconda : Anaconda [2] simplifies Python package management, especially for data science and deep learning. However, the transition to Google Colab was prompted by its cloud-based environment and free access to potent computational resources like GPUs, eliminating the need for local installations and enhancing collaboration with automatic version control.

Google Colab : Google Colab [3], or "Collaboratory," is a cloud-based service that enables users to write and execute Python code directly through their browsers. It offers

hassle-free access to GPUs and TPUs at no cost, making it a preferred platform for deep learning, data analysis, and educational purposes. Colab facilitates collaborative work by storing notebooks in Google Drive, allowing easy sharing similar to Google Docs.

Scikit-learn : Scikit-learn (also known as sklearn [4]) is a powerful Python library that offers a wide range of tools for machine learning, statistical modeling, and data analysis techniques.

Pandas and Numpy : Pandas [6], a Python library, specializes in data analysis by offering efficient data structures for structured data management. NumPy [7] complements this with extensive support for vast, multi-dimensional arrays and matrices, coupled with an extensive array of mathematical functions for array processing.

Matplotlib : Matplotlib [8] is a versatile Python library for creating a wide array of visualizations.

TensorFlow : TensorFlow [9], an open-source deep learning framework by Google, excels in high-performance numerical computations, catering to various tasks, including deep learning and neural networks. Its adaptability enables the creation of complex models with ease, leading to widespread adoption in academic and industrial domains for research and deployment in production environments.

Keras : Keras [5], renowned as an open-source framework, facilitates the development and training of neural networks with its user-friendly, high-level API. It provides a straightforward platform for setting up and managing neural network layers, making it highly approachable for developers engaged in deep learning projects.

FLWR : Flower [10] is an open-source framework meticulously crafted to simplify and optimize federated learning workflows across clusters of machines. This Python-based platform provides a user-friendly solution tailored for training various models, encompassing sophisticated deep neural networks.

---

[1] Python: `https://www.python.org`

[2] Anaconda: `https://www.anaconda.com/download`

[3] Google Colab: `https://colab.research.google.com/`

[4] Scikit-learn: `https://scikit-learn.org/`

[5] Keras: `https://keras.io/`

[6] Pandas: `https://pandas.pydata.org/`

[7] NumPy: `https://numpy.org/`

[8] Matplotlib: `https://matplotlib.org/`

[9] TensorFlow: `https://www.tensorflow.org/`

[10] FLWR: `https://flower.dev/`

2.2 Problem Formulation and Motivation

The proposed research aims to implement deep learning (DL) and federated learning (FL) methodologies to detect distributed denial of service (DDoS) intrusions. Given the distributed nature of DDoS attacks, which utilize networks of hijacked devices, gathering centralized data poses a significant challenge. FL offers a solution by enabling the training of models on local datasets within individual devices, thereby circumventing the need for data centralization and enhancing privacy. DL techniques are particularly adept at discerning complex patterns and behaviors associated with DDoS attacks. By integrating DL with FL, the system can engage in ongoing learning and adjust to the dynamic nature of DDoS threats, harnessing the power of collective intelligence while safeguarding the privacy of individual devices. Furthermore, FL facilitates a collaborative approach to data processing and analysis among dispersed devices, thereby diminishing the computational burden on single devices and enabling scalable detection of DDoS activities across extensive networks of devices.

2.3 Dataset Presentation

This project introduces the Edge-IIoTset-2022, a comprehensive cybersecurity dataset tailored specifically for IoT and IIoT applications. We have selected this new synthetic real-world dataset for its effectiveness in training deep learning-based intrusion detection systems, which can operate in either centralized or federated learning modes[15]. The dataset is organized into seven layers, each integrating cutting-edge technologies to address the needs of IoT and IIoT applications. It covers a wide range of IoT devices, including digital sensors for temperature, humidity, ultrasonic, water level, pH, soil moisture, heart rate, and flame detection, resulting in diverse data generation. Moreover, the dataset highlights fourteen identified attacks targeting IoT and IIoT connectivity protocols, including DoS/DDoS attacks.

2.3.1 Dataset Description

The Edge-IIoTset dataset can be structured into three main categories:

Normal Traffic: This segment of the dataset comprises normal traffic generated by various sensors within the testbed. Each sensor's normal traffic is stored in ten directories, with data available in both ".pcap" and ".csv" formats. For instance, the "Distance" directory contains two subfiles named"distance.pcap" and "distance.csv".

Attack Traffic: In this category, the dataset incorporates traffic resulting from a predefined set of attacks applied to the normal traffic. Each attack generates its traffic

data, resulting in a total of 14 files. Similar to the normal traffic, the attack traffic data is stored in both ".pcap" and ".csv" formats.

Selected Dataset for ML and DL: This section of the dataset is specifically curated for machine learning (ML) and deep learning (DL) applications. It includes processed subsets of both normal and attack traffic, optimized for ease of manipulation and compatibility with various applications.

2.3.2 Attacks in Edge-IIoTset Datase

The Edge-IIoTset-2022 dataset encompasses fourteen distinct attacks targeting IoT and IIoT applications. These attacks have been identified, analyzed, and classified into five primary threat categories.

DoS/DDoS attacks: involve attackers attempting to disrupt services for legitimate users, either individually or through distributed means. This category encompasses four commonly employed techniques: TCP SYN Flood, UDP flood, HTTP flood, and ICMP flood.

Information gathering: It involves acquiring intelligence about the target victim, typically as the initial step in any successful attack. In our research, we concentrate on three primary tasks frequently conducted by malicious actors in the initial stages of gathering information: port scanning, identifying the operating system through fingerprinting, and assessing vulnerabilities.

Malware attacks: These attacks have garnered considerable attention in recent years due to their widespread impact and consequential losses, posing a serious concern. Our analysis encompasses three types of these attacks: backdoors, password crackers, and ransomware attacks.

Injection attacks: These attacks aim to undermine the integrity and confidentiality of the targeted system. Our approach entails utilizing three distinct methods: XSS, SQL injection, and uploading attacks.

Man-in-the-middle: These attacks aim to intercept and manipulate communication between two parties who believe they are communicating directly. Our focus in executing this attack is on targeting two widely used protocols found in nearly every system: DNS and ARP.

2.4 Data preparation

Data preparation is crucial in deep learning and federated learning, as it ensures data quality and relevance, directly impacting model performance. This involves cleaning, normalizing, transforming data, and, in federated learning, ensuring privacy and security

across decentralized devices. Proper preparation results in more robust, accurate, and efficient models, emphasizing its critical role in these machine learning techniques. The implementation of our project involved several stages. We began with data preprocessing and then moved on to subsequent steps.

2.4.1 Needed libraries:

To start our project, we installed additional libraries that were not pre-installed in Google Colab. These included "numpy," "kaggle," and "tensorflow", etc.

2.4.2 Uploading the dataset:

To initiate our project, we uploaded the "Edge-IIoTset" dataset. For a streamlined process, we crafted a JSON file to facilitate direct dataset uploads from Kaggle. This approach conserves both time and memory by bypassing the need for local downloads. Additionally, we transferred the "DNN-EdgeIIoT-dataset.csv" in a compressed zip format and extracted it to retrieve the necessary file.

2.4.3 Comprehensive Data Overview

A comprehensive data overview combines descriptive statistics and visual displays to enhance understanding and interpretation of the dataset. Figures 2.1 and Table 2.1 illustrate the distribution of attack traffic relative to the percentage of normal traffic.

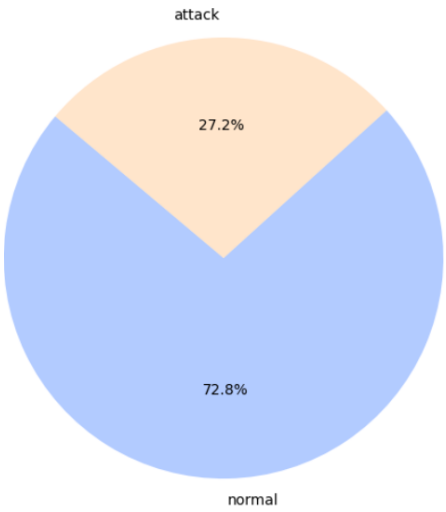


Figure 2.1: Illustration of normal traffic percentage vs attack traffic.

Figure 2.2 and Table 2.2 provide additional insights into various attack types present in the dataset.

Attack_label	Count	%
normal	1615643	72.802914
attack	603558	27.197086

Table 2.1: Distribution of Attack Traffic.

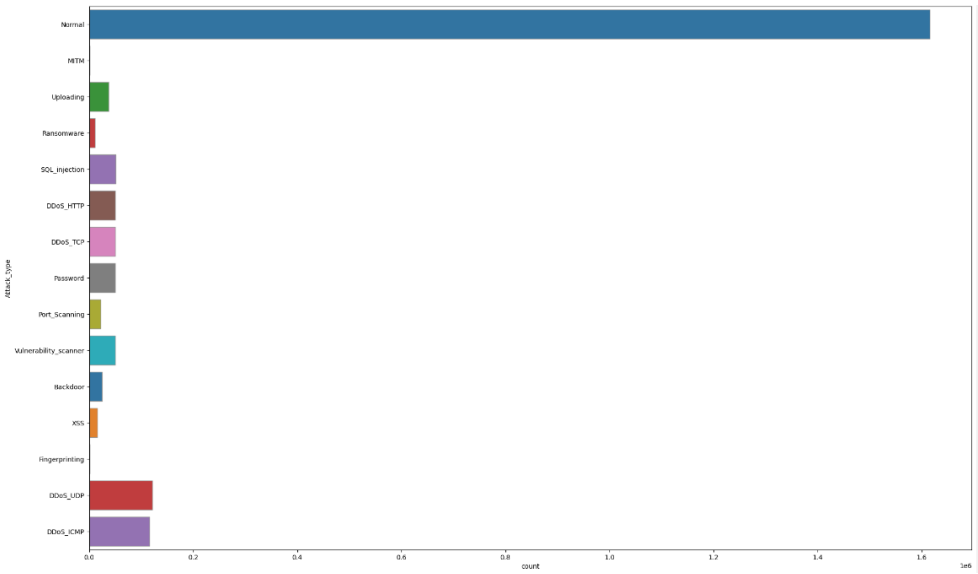


Figure 2.2: Different attack types in the dataset.

Attack_type	Count	%
Backdoor	24862	1.120313
DDoS_HTTP	49911	2.249053
DDoS_ICMP	116436	5.246753
DDoS_TCP	50062	2.255857
DDoS_UDP	121568	5.478008
Fingerprinting	1001	0.045106
MITM	1214	0.054704
Normal	1615643	72.802914
Password	50153	2.259958
Port_Scanning	22564	1.016762
Ransomware	10925	0.492294
SQL_injection	51203	2.307272
Uploading	37634	1.695836
Vulnerability_scanner	50110	2.258020
XSS	15915	0.717150

Table 2.2: Distribution of Attack Types

2.4.4 Data pre-processing:

The pre-processing phase of the Edge-IIoTset dataset holds significant importance in our research endeavor. This crucial stage entails a series of systematic procedures aimed at optimizing the data to seamlessly integrate with our ensemble models. The pre-

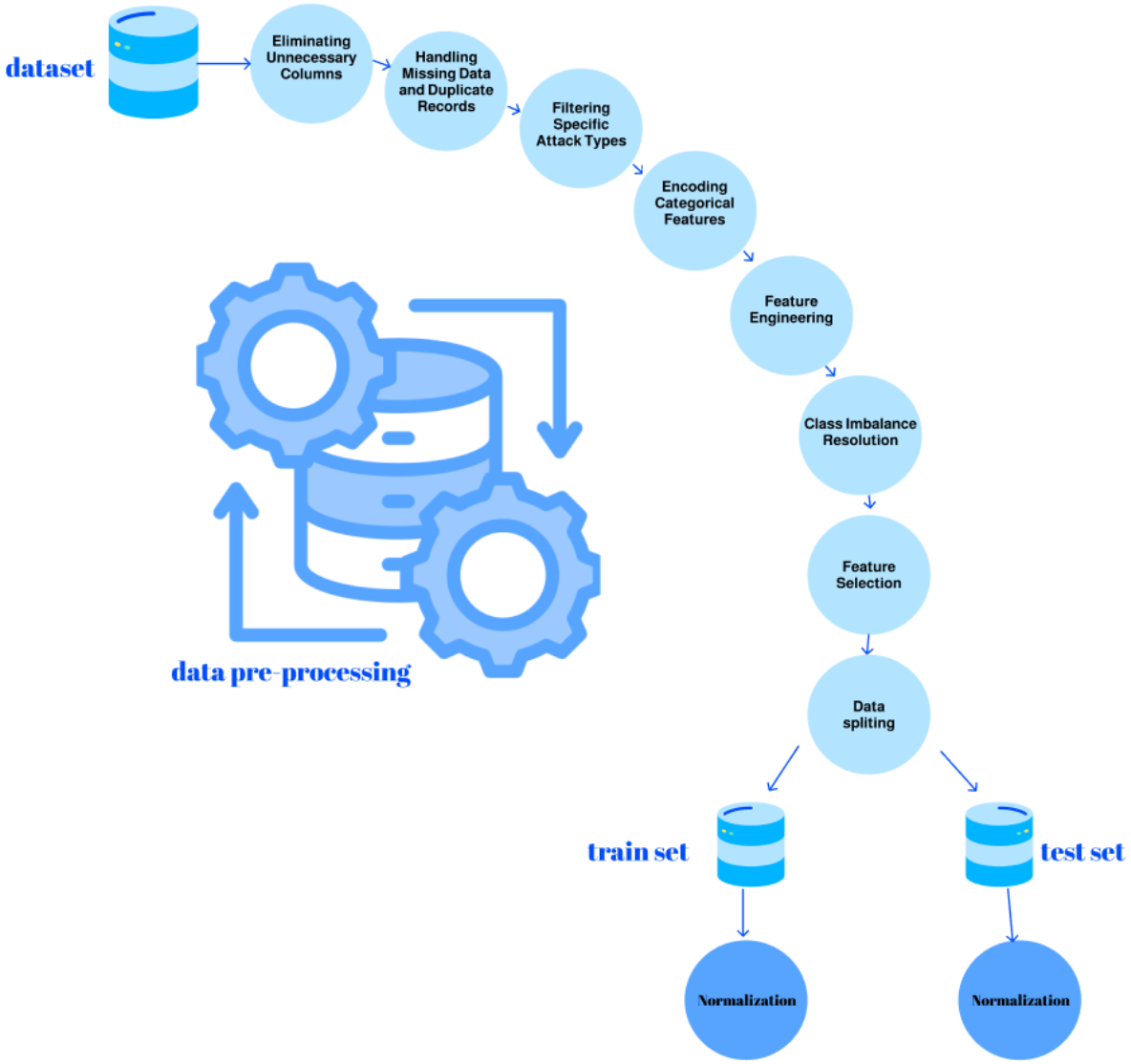


Figure 2.3: Data pre-processing Steps.

processing stage of our study begins with the exclusion of columns deemed irrelevant to our analysis, such as *"frame.time," "ip.src_host," "ip.dst_host," "arp.src.proto_ipv4,"* and *"http.file_data,"* among others, which could potentially skew our results. Subsequently, we address duplicate rows and features with *"NULL"* or *"NUN"* values, as they offer no relevant information and could impede the efficiency of our application.

Additionally, we filter out non-DDoS attack types like *"MITM," "XSS,"* and *"SQL_injection"* to streamline data processing. To bolster data quality, median and standard deviation-

based filtering techniques are applied, followed by the storage of the processed dataset for future analysis.

In tackling class imbalance, we opted for downsampling and feature selection to pinpoint the most influential attributes. Subsequently, the dataset was partitioned into training and testing subsets, where label encoding and feature scaling were employed to standardize the data, as depicted in Figure 2.3. Additionally, we applied the SMOTE technique for more balanced training data. This rigorous approach ensures the attainment of a high-quality dataset, a prerequisite for the effective application of deep learning methodologies and the achievement of precise outcomes.

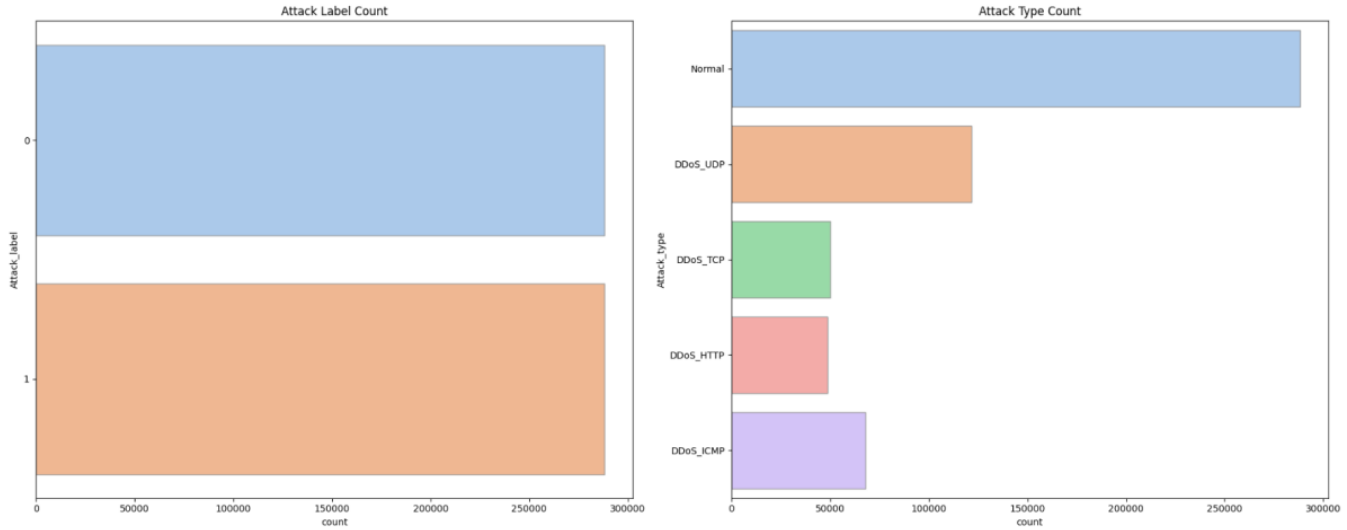
Tables 2.3 and 2.4 provide detailed information about the features, listing those that have been deleted and selected, respectively. Table 2.4 displays the features selected through techniques such as SelectKBest, utilizing the f-classif scoring function based on ANOVA F-values to evaluate feature-target relationships. Features with higher F-values are given priority, indicating their greater relevance for predicting the target variable and suitability for further analysis or modeling. Furthermore, Figure 2.4 offers a visual representation of the dataset's distribution following the pre-processing steps.

Column	Dtype	Column	Dtype
frame.time	object	http.request.uri.query	object
ip.src_host	object	tcp.options	object
ip.dst_host	object	tcp.payload	object
arp.dst.proto_ipv4	object	tcp.srcport	object
arp.src.proto_ipv4	object	tcp.dstport	float64
http.file_data	object	udp.port	float64
http.request.full_uri	object	mqtt.msg	object
icmp.transmit_timestamp	float64		

Table 2.3: The set of features deleted from the dataset.

Feature selected	Type	Feature selected	Type
icmp.checksum	float64	tcp.flags	float64
icmp.seq_le	float64	tcp.flags.ack	float64
tcp.checksum	float64	udp.stream	float64
Attack_label	int64	http.request.method-0	object
http.request.method-0.0	object	http.referer-0	object
http.referer-0.0	object	http.request.version-0	object
http.request.version-0.0	object	dns.qry.name.len-0	object
dns.qry.name.len-0.0	object	mqtt.conack.flags-0	object
mqtt.conack.flags-0.0	object	mqtt.protoname-0	object
mqtt.protoname-0.0	object	mqtt.topic-0	object
mqtt.topic-0.0	object		

Table 2.4: Feature Selection



(a) the distribution of the database on 2 class after prepossessing

(b) the distribution of the database on 5 class after prepossessing

Figure 2.4: The distribution of the dataset after pre-processing.

2.5 Models

In our study, we explored the performance of three different deep-learning models: CNN, CNN-LSTM, and CNN-LSTM-GRU. These models were applied to the preprocessed version of the Edge-IIoT dataset. Our initial investigation focused on binary classification, differentiating between normal network traffic and DDoS attacks. The primary goal was to gauge the models' accuracy in identifying DDoS attacks. Subsequently, we expanded our analysis to include multi-class classification, comprising five unique categories. These categories delineated normal traffic and four types of DDoS attacks. Our overarching objective remained the assessment of the model's effectiveness in detecting various DDoS attacks.

2.5.1 Convolutional Neural Networks (CNN)

While convolutional neural networks (CNNs) are typically employed for image detection, in our study, we utilized a 1D CNN model for both multi-class and binary classification. The 1D CNN operates similarly to 2D or 3D CNNs but was chosen for its superior performance and learning rate. During the development of the 1D CNN model, we adjusted several parameters to achieve optimal results. Figure 2.3 illustrates the architecture of our 1D CNN model for this classification task.To prepare the data for the CNN, we reshaped it to ensure it is in the correct format for input into the 1D CNN model. The following algorithm outlines the steps taken:

Algorithm 2.1 Data Reshaping for 1D CNN

Require: $X\_train\_resampled$, X_test
Ensure: X_train, X_test reshaped for 1D CNN input
1: $X\_train \leftarrow X\_train\_resampled.reshape(X\_train\_resampled.shape[0], X\_train\_resampled.shape[1], 1)$
2: $X\_test \leftarrow X\_test.reshape(X\_test.shape[0], X\_test.shape[1], 1)$
3: **print** $(X\_train.shape)$
4: **print** $(X\_test.shape)$

These steps are crucial to make sure that the data is in the appropriate format for the 1D CNN model to process effectively.

Convolutional Layer (Conv1D): 4 filters, kernel size 3, ReLU activation, input shape `input shape`, with L2 regularization (penalty factor 0.01).

Dropout Layer: Applies a dropout rate of 0.6 to prevent overfitting.

MaxPooling Layer (MaxPooling1D): Pool size of 2 to reduce the output shape by half.

Flatten Layer: Converts the previous layer's output into a one-dimensional vector.

Dense Layer: `num classes` units with softmax activation to output class probabilities.

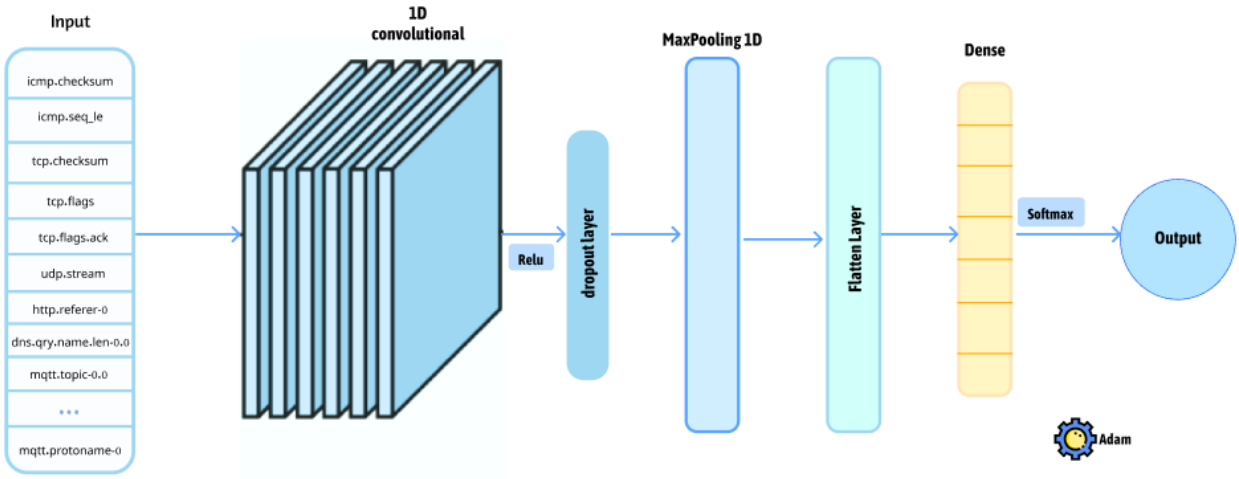


Figure 2.5: Comprehensive CNN Architecture for Multi-Class and Binary Classification.

Overall, the model starts with Conv1D, then Dropout, MaxPooling, and Flatten, and ends with a Dense layer, using Adam optimizer.

2.5.2 Convolutional Neural Network, Long Short-Term Memory(CNN-LSTM)

In this model, we have designed a neural network architecture that combines convolutional layers and LSTM layers to leverage the data's spatial and temporal features. The model is employed for multi-class and binary classification tasks, incorporating several key components that help learn complex patterns from the input data.

Algorithm 2.2 Algorithme of CNN-LSTM

```
model ← Sequential()
model.add(Conv1D(16, kernel_size = 3, activation = 'sigmoid', input_shape = input_shape))
model.add(Dropout(0.2))
model.add(MaxPooling1D(pool_size = 2))
model.add(LSTM(4))
model.add(Dense(8, activation = 'tanh'))
model.add(Dense(num_classes, activation = 'softmax'))
```

The model architecture comprises a sequence of layers, starting with a convolutional layer for spatial feature extraction, followed by dropout for regularization, then max pooling for dimensionality reduction, and an LSTM layer for capturing temporal dependencies. Dense layers are utilized for classification, with softmax activation providing

class probabilities. This architecture synergistically combines convolutional and LSTM layers, making it adept at discerning intricate spatial and temporal patterns in the data.

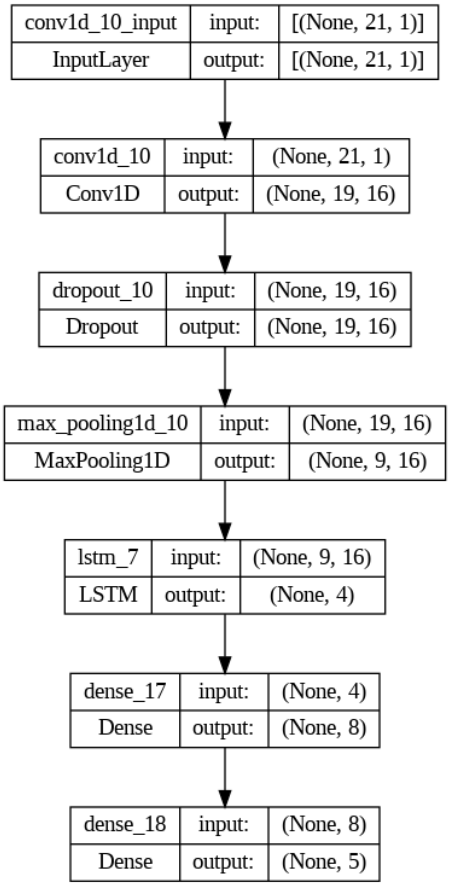


Figure 2.6: The Structure of CNN-LSTM Model.

2.5.3 Integration of CNN, LSTM, and GRU (Gated Recurrent Unit)

The integration of CNNs, LSTMs, and GRUs in our "cnn-lstm-gru-model" represents a sophisticated strategy for analyzing sequential data, especially in the realm of intrusion detection. Each neural network architecture brings unique strengths to the table, collectively enhancing the model's ability to discern intricate patterns within complex data streams.

Firstly, the 1D CNN layer acts as a feature extractor adept at capturing spatial information from the input data. This capability is particularly valuable for identifying local patterns and anomalies within the sequential data. Subsequently, the LSTM and GRU layers come into play, modeling temporal dependencies to capture long-range dependencies and contextual information over time. By doing so, the model gains an understanding of the sequential nature of the data, allowing it to detect patterns that evolve over multiple time steps. Our "cnn-lstm-gru-model" is meticulously designed with a carefully orches-

trated sequential arrangement of these layers, optimizing their interactions to achieve superior performance in intrusion detection. Additionally, we incorporate dropout layers to mitigate overfitting by randomly dropping a fraction of neurons during training, thereby promoting better generalization

Furthermore, we employ regularization techniques, such as kernel regularization, to enhance the robustness of our model. These techniques impose penalties on the magnitude of network weights, preventing them from growing excessively large and aiding in the prevention of overfitting.

To visually represent our model architecture, we include Figure 2.7, which showcases the sequential arrangement of the CNN, LSTM, and GRU layers, along with the dropout and dense layers. This graphical depiction offers a clearer understanding of how each component interacts and contributes to the overall functionality of the model, enhancing the interpretability of our approach.

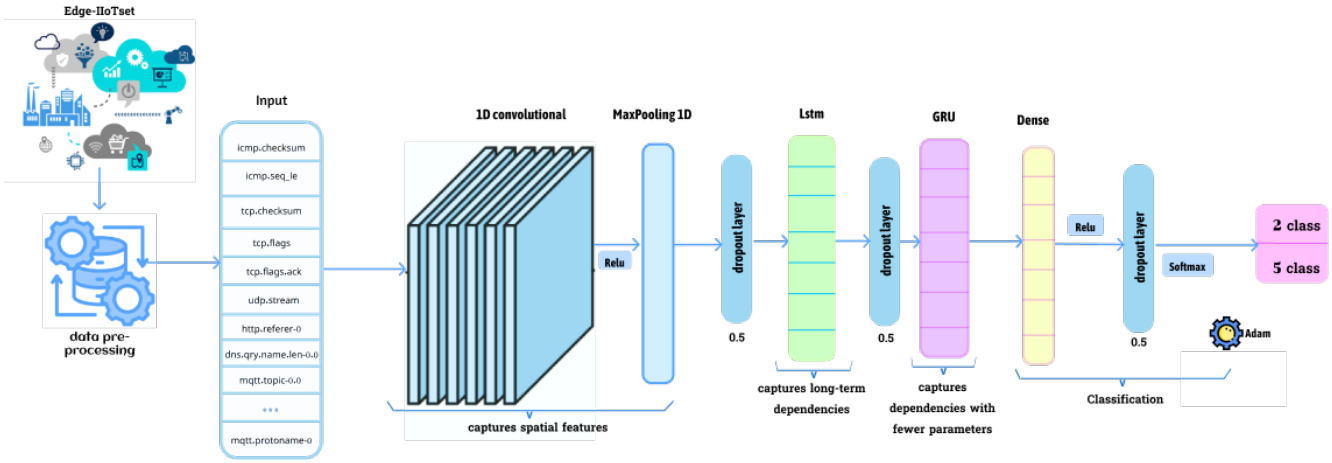


Figure 2.7: The Structure of CNN-LSTM-GRU Model.

2.5.4 Training and Testing Models

Before identifying the best deep learning model, we conducted comprehensive training and testing of precedent models. We partitioned the dataset into 80% for training and 20% for testing, using stratification to maintain consistent class distribution across both sets.

```
# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X_selected, y,
    test_size=0.2, random_state=42)
```

Then proceed to train each model using the following instructions:

```
opt = Adam(learning_rate=0.001)
```

```
model.compile(optimizer=opt,
   loss=tf.keras.metrics.categorical_crossentropy,
               metrics=['accuracy'])
history = model.fit(X_train, y_train,
                    validation_data=(X_test, y_test),
                    epochs=15,
                    batch_size=5000,
                    verbose=1)
```

Model Compilation:

1. The model is configured for training using the compile function.
2. The loss function is set to `'categorical_crossentropy'` for multi-class classification and `'binary_crossentropy'` for binary classification.
3. The Adam optimizer is used for efficient parameter updates.
4. The model's performance will be evaluated based on accuracy during training.

Model Training:

1. The model is trained using the `fit` function.
2. The training data (`X_train`) and labels (`y_train`) are used.
3. The training process consists of 15 epochs (iterations over the training dataset).
4. Each update of the model's weights is performed on a batch of 5000 samples because our training set exceeds 1 million samples.
5. The model's performance is evaluated on the validation data (`X_test` and `y_test`).
6. Training progress is displayed with verbosity set to 1.

2.6 Proposed Intrusion Detection System Based on Deep Learning Models

This section provides a detailed description of the process flow for developing and evaluating an Intrusion Detection System (IDS) using deep learning models. The focus is on using a dataset (DNN-EdgeIIoT-dataset.csv) to train and validate models for binary classification (normal vs. attack) and multi-class classification (normal vs. specific types of attacks). The development and evaluation of an Intrusion Detection System (IDS) using deep learning models begins with the DNN-EdgeIIoT-dataset.csv. The data undergoes pre-processing, including normalization and feature encoding, to prepare it for analysis. The dataset is then split into a train/validation set and a test set. The training set is further divided and labeled for binary classification (normal vs. attack) and multi-class classification (normal vs. specific types of DDoS attacks). Both training and test sets are

normalized to ensure consistent input feature scaling. Various deep learning models such as CNN, CNN + LSTM, and CNN + LSTM + GRU are trained using the prepared data. Finally, these models are evaluated on the test set using metrics like accuracy, precision, recall, and F1-score to assess their performance in detecting both binary and multi-class intrusions, as illustrated in Figure 2.8.

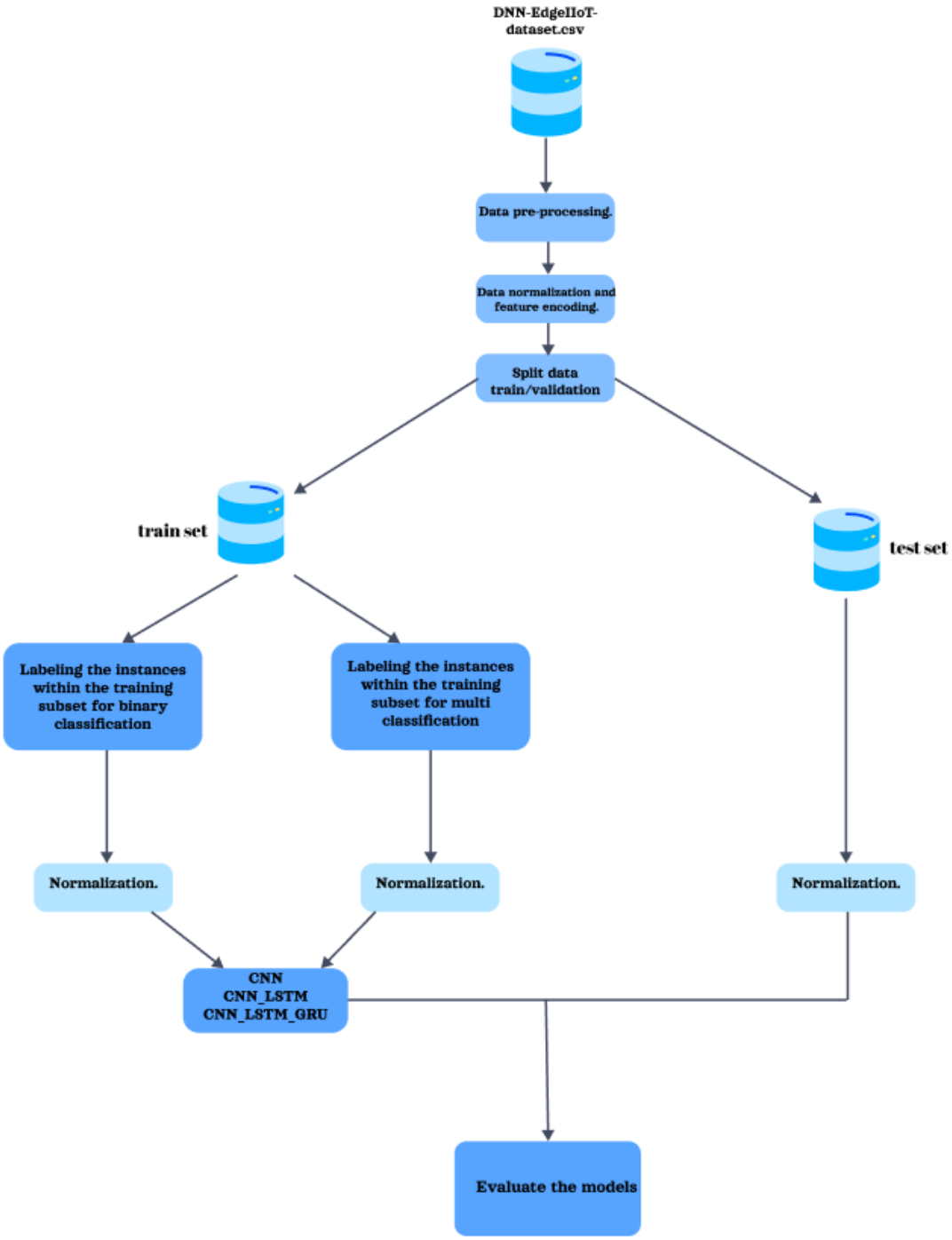


Figure 2.8: Our Proposed Methodology for DDoS Detection Using DL Methods.

2.7 Proposed Solution for Federated Based on Client Server

The Federated Learning (FL) system depicted in Figure 2.9 illustrates a decentralized approach to model training. Client devices, such as laptops and robotic arms, independently train local models using their respective data without sharing it externally. Through model aggregation on a central server, the trained parameters from each client contribute to refining a global model, which progressively improves with each iteration. This iterative refinement process ensures better generalization and performance of the global model while safeguarding data privacy, as sensitive data remains on client devices. By leveraging distributed data sources, this architecture optimizes model training while addressing critical concerns around data sensitivity and security.

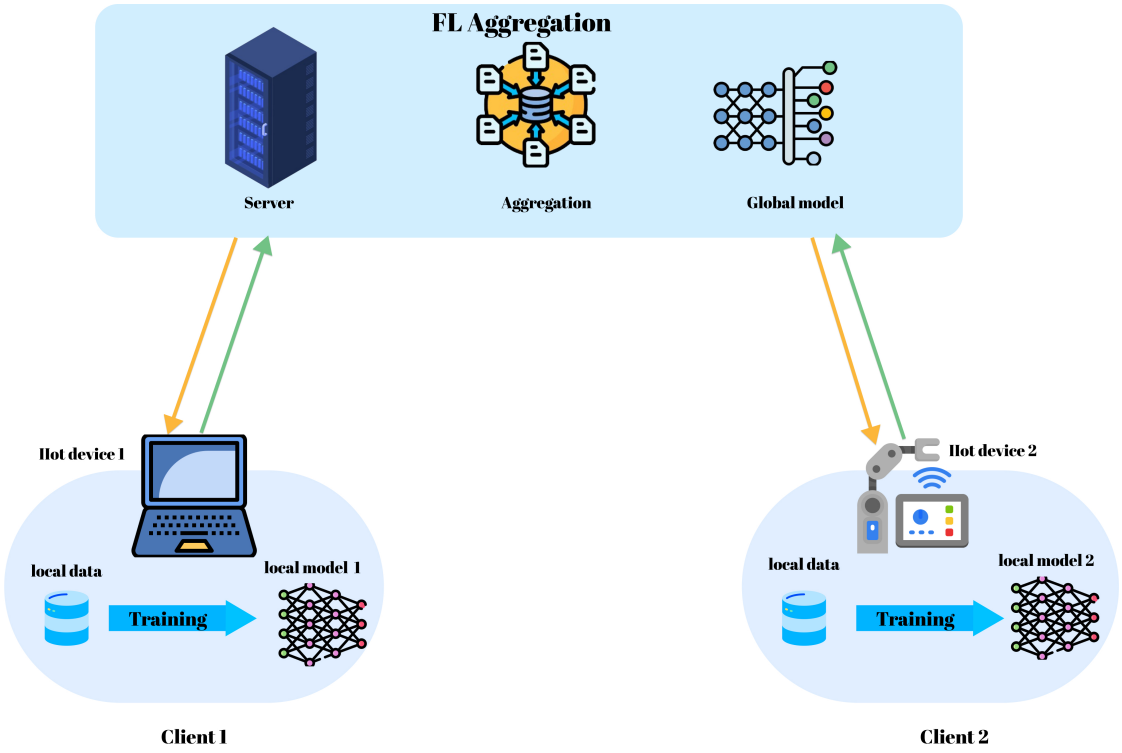


Figure 2.9: Architecture of Federated Learning Based on Client Server.

2.8 Experimental Results and Discussions

In this section, we present the results of various experiments conducted to validate the effectiveness of the proposed models. We analyze the performance metrics obtained

from these experiments to assess the model's capability in accurately classifying intrusion activities. The evaluation was carried out using the following metrics:

- **True Positive (TP):** The count of attack instances is accurately identified by the model.
- **True Negative (TN):** The count of normal instances correctly classified by the model.
- **False Positive (FP):** The count of normal instances incorrectly labeled as attacks.
- **False Negative (FN):** The count of attack instances mistakenly classified as normal.

Accuracy: Reflects the overall correctness of the model, calculated as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: Indicates the reliability of the model's attack predictions, calculated as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall: Also known as the True Positive Rate, it measures the model's ability to detect actual attacks, calculated as:

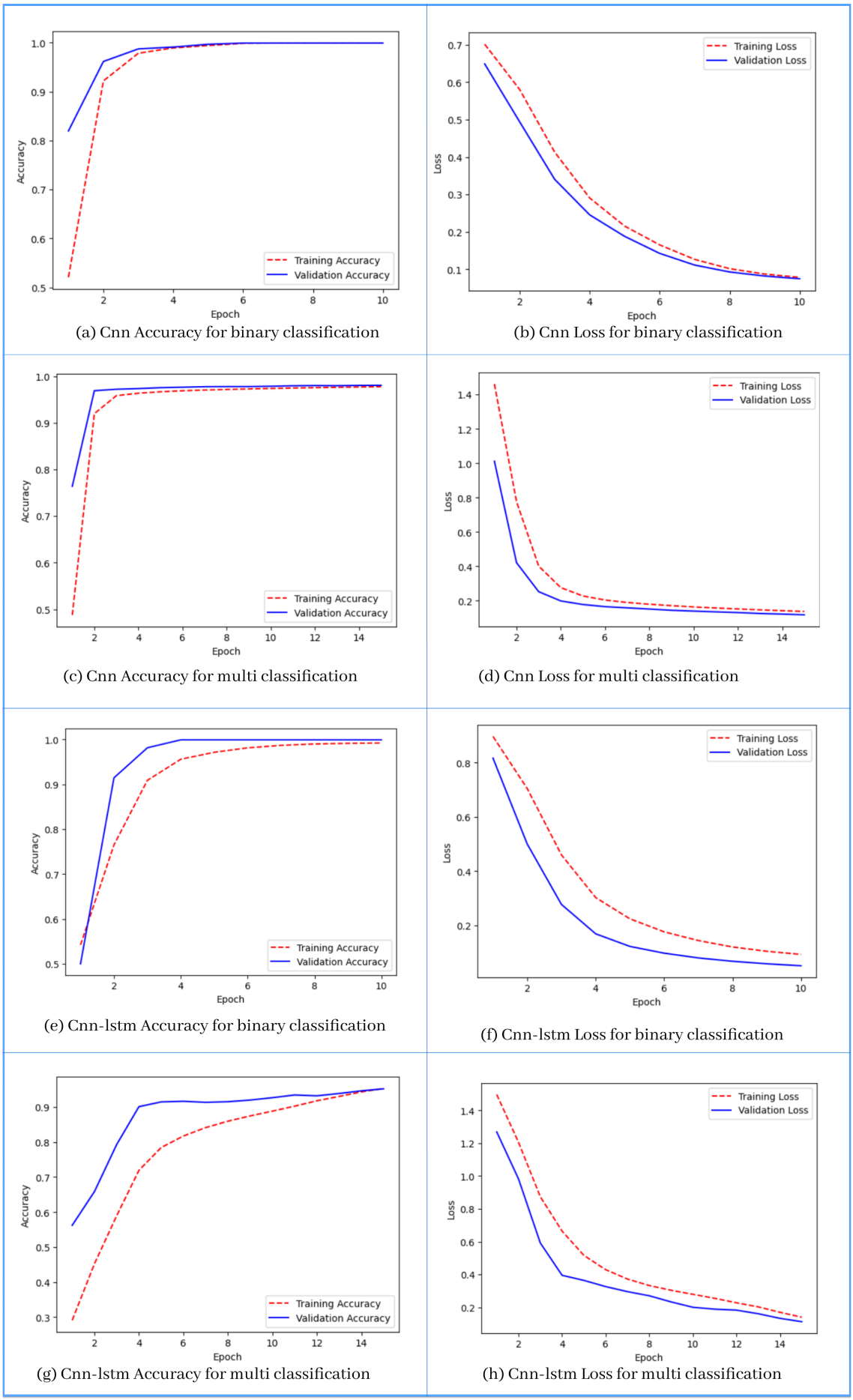
$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-score: A balanced measure that combines precision and recall, particularly useful when the class distribution is uneven, given by:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Confusion Matrix: A matrix that visualizes the performance of the classification model by displaying the TP, TN, FP, and FN counts.

These metrics provide a comprehensive view of the model's performance, each highlighting different aspects of its predictive capabilities. It's crucial to consider all of them to accurately assess the model's effectiveness in detecting DDoS attacks.



(a) Cnn Accuracy for binary classification

(b) Cnn Loss for binary classification

(c) Cnn Accuracy for multi classification

(d) Cnn Loss for multi classification

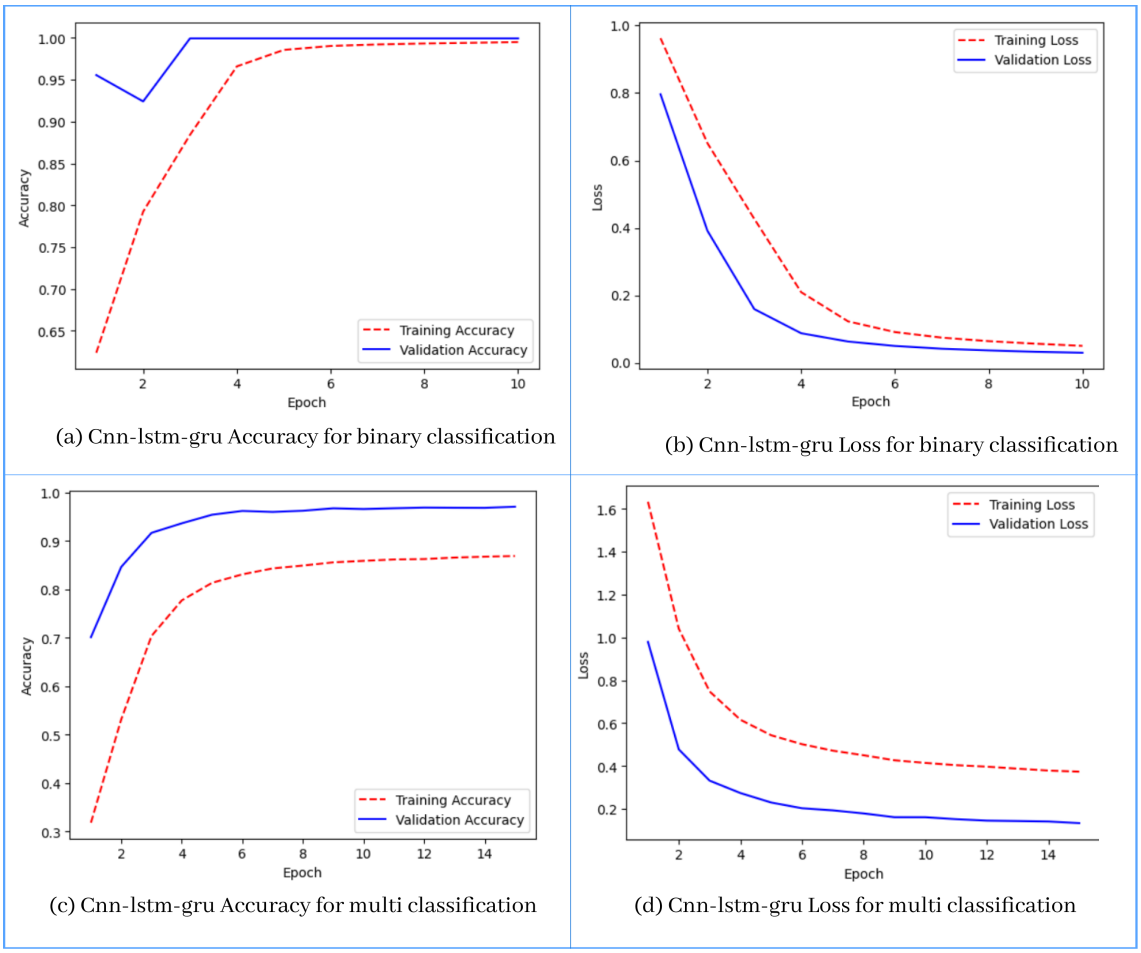
(e) Cnn-lstm Accuracy for binary classification

(f) Cnn-lstm Loss for binary classification

(g) Cnn-lstm Accuracy for multi classification

(h) Cnn-lstm Loss for multi classification

Figure 2.10: Accuracy and loss curves of the proposed models.



(a) Cnn-lstm-gru Accuracy for binary classification

(b) Cnn-lstm-gru Loss for binary classification

(c) Cnn-lstm-gru Accuracy for multi classification

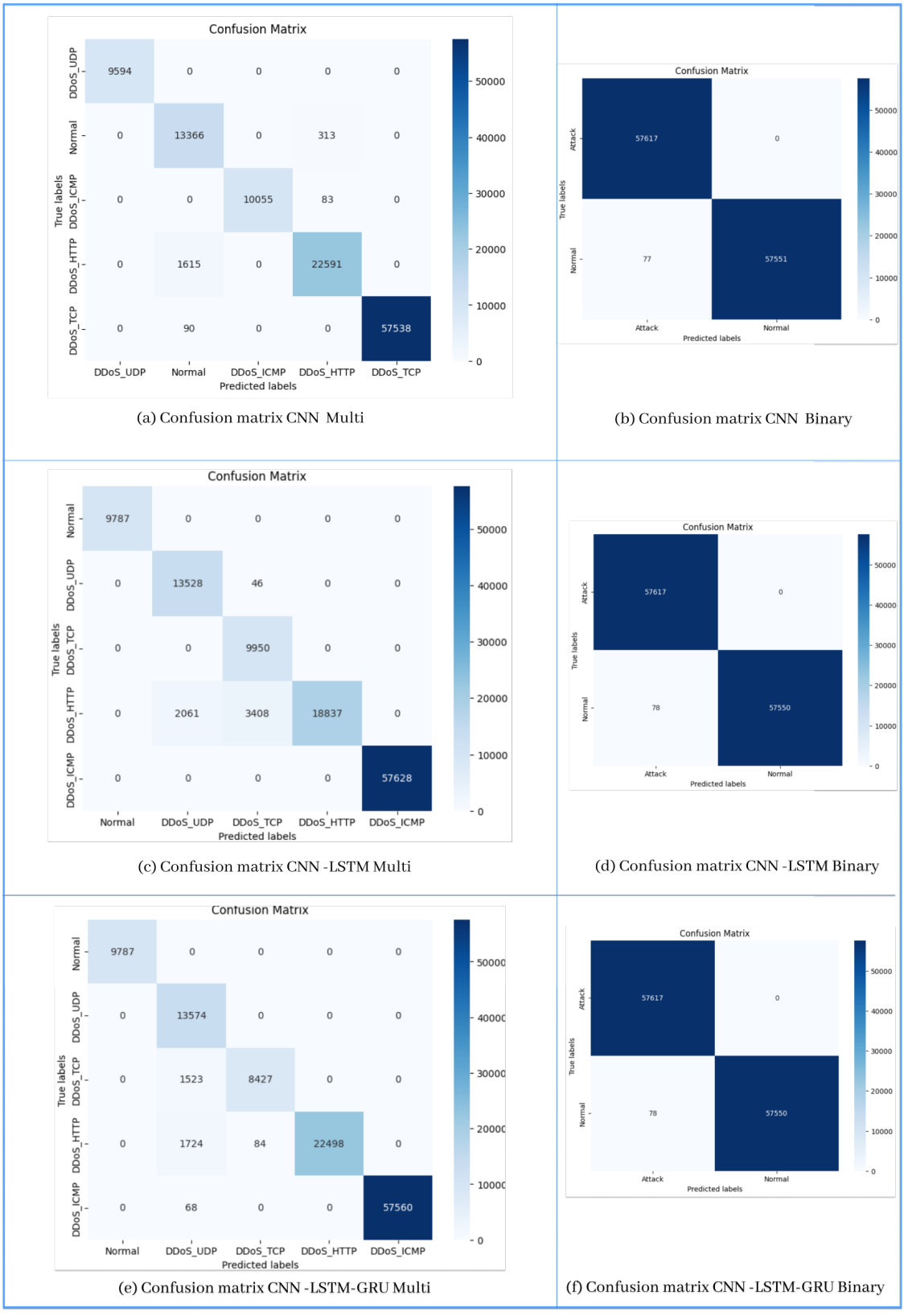
(d) Cnn-lstm-gru Loss for multi classification

Figure 2.11: Accuracy and loss curves of the cnn-lstm-gru models.

As depicted in Figure 2.10 and Figure 2.11 the accuracy for both the training and validation phases steadily increases from the beginning to the end of the training process, eventually converging to 1. Meanwhile, the loss graphs consistently decrease, reaching a minimum value approaching 0. These trends indicate that the model continues to enhance its training and learn more effectively with each epoch.

The confusion matrix illustrated in Figure 2.12 shows the performance of the proposed models by displaying the true labels against the predicted labels.

The multi-classification report of CNN, CNN-LSTM, and CNN-LSTM-GRU models is shown in Figure 2.13.



(a) Confusion matrix CNN Multi

(b) Confusion matrix CNN Binary

(c) Confusion matrix CNN -LSTM Multi

(d) Confusion matrix CNN -LSTM Binary

(e) Confusion matrix CNN -LSTM-GRU Multi

(f) Confusion matrix CNN -LSTM-GRU Binary

Figure 2.12: Confusion matrix of the proposed models.

	precision	recall	f1-score	support
DDoS_HTTP	1.00	1.00	1.00	9660
DDoS_ICMP	0.89	0.97	0.93	13628
DDoS_TCP	1.00	0.99	0.99	10011
DDoS_UDP	0.98	0.93	0.95	24318
Normal	1.00	1.00	1.00	57628
accuracy			0.98	115245
macro avg	0.97	0.98	0.97	115245
weighted avg	0.98	0.98	0.98	115245

(a) Multi classification report CNN model

	precision	recall	f1-score	support
DDoS_HTTP	1.00	1.00	1.00	9787
DDoS_ICMP	0.87	1.00	0.93	13574
DDoS_TCP	0.74	1.00	0.85	9950
DDoS_UDP	1.00	0.77	0.87	24306
Normal	1.00	1.00	1.00	57628
accuracy			0.95	115245
macro avg	0.92	0.95	0.93	115245
weighted avg	0.96	0.95	0.95	115245

(b) Multi classification report CNN-LSTM model

	precision	recall	f1-score	support
DDoS_HTTP	1.00	1.00	1.00	9787
DDoS_ICMP	0.80	1.00	0.89	13574
DDoS_TCP	0.99	0.85	0.91	9950
DDoS_UDP	1.00	0.93	0.96	24306
Normal	1.00	1.00	1.00	57628
accuracy			0.97	115245
macro avg	0.96	0.95	0.95	115245
weighted avg	0.98	0.97	0.97	115245

(c) Multi classification report CNN-LSTM-GRU model

Figure 2.13: Multi classification report of the proposed models.

Model	Accuracy	Precision	Recall	F1 Score	classes
CNN	0.999331	0.999332	0.999331	0.999331	2
CNN-LSTM	0.999323	0.999324	0.999323	0.999323	2
CNN-LSTM-GRU	0.999323	0.999324	0.999323	0.999323	2
CNN	0.980918	0.981924	0.980918	0.981085	5
CNN-LSTM	0.952145	0.962180	0.952145	0.951985	5
CNN-LSTM-GRU	0.970506	0.976029	0.970506	0.971224	5

Table 2.5: The results of the proposed models for DDoS detection.

The binary classification performance across all models (CNN, CNN-LSTM, and CNN-LSTM-GRU) demonstrates exceptional effectiveness, with accuracy, precision, recall, and F1 scores uniformly around 0.9993. This indicates that these models are nearly flawless in detecting attacks, with negligible differences observable only at the fourth decimal place. Such high and consistent metrics suggest that the models are robust and well-trained and that the data for both classes is likely more straightforward to classify accurately, contributing to the uniformly high performance observed.

In multi-class classification, the CNN model stands out with the highest precision (0.981924) and F1 score (0.981085), indicating superior performance compared to the other models. The CNN-LSTM model shows lower performance, with accuracy, precision, recall, and F1 scores ranging from 0.9521 to 0.9622, suggesting difficulties with more complex data. The CNN-LSTM-GRU model performs better than the CNN-LSTM but not as well as the CNN, with an accuracy of 0.970506 and precision of 0.976029.

The CNN model is the best choice for federated learning in this context due to its high performance in both binary and multi-class classification and its simpler architecture, making it ideal for efficient federated DDoS detection. Flower, a federated learning framework from a University of Oxford research project, supports various machine learning frameworks like PyTorch, TensorFlow, and NumPy. Flower's focus on customization, extensibility, framework-agnosticism, and understandability ensures seamless integration of the CNN model, optimizing its deployment in federated DDoS detection scenarios.

Simulation Setup for Federated Learning with Flower:

A structured Python environment is essential for Flower simulations. We use conda to create and manage this environment, ensuring all necessary packages and dependencies are installed. Below are the steps and components we follow:

1. Setting Up the Environment: We use conda to install necessary packages like PyTorch.
2. Configuration Management: We utilize Hydra for easy experiment configuration management.
3. Dataset Preparation: We partition datasets to simulate client distributions.

```
trainloaders, validationloaders, testloader, num_features =
   prepare_dataset(cfg.num_clients, cfg.batch_size)
    print(len(trainloaders), len(trainloaders[0].dataset))
```

4. Model: The model we are going to use is a simple Convolutional Neural Network. This step is the same in classical training
5. Federated Learning: To utilize the Flower framework, we need to establish a few essential classes. For conducting the simulation, the following components are required: **Flower Client Class:**This class should implement three specific methods.

```
class FlowerClient(fl.client.NumPyClient):
    def __init__(self, trainloader, valloader, num_features,
       num_classes) -> None:
        super().__init__()
        self.trainloader = trainloader
        self.valloader = valloader
        self.model = Net(num_features, num_classes)
        self.device = torch.device("cuda:0" if
           torch.cuda.is_available() else "cpu")
        self.model.to(self.device)
```

Client Creation Function: A function responsible for instantiating Flower clients.

```
def generate_client_fn(trainloaders, valloaders, num_features,
   num_classes):
    def client_fn(cid: str):
        return FlowerClient(
            trainloader=trainloaders[int(cid)],
            valloader=valloaders[int(cid)],
            num_features=num_features,
            num_classes=num_classes,
        ).to_client()
    return client_fn
```

Federation Strategy: After each round of federated learning, it is necessary to collect model weights from clients. We will use the default and most widely used strategy, known as the uniform average (FedAvg). Before implementing the FedAvg strategy, we need to define a function that defines how the evaluation metrics are calculated.

```
# Define your strategy with reduced fraction_fit and
   fraction_evaluate
strategy = fl.server.strategy.FedAvg(
    fraction_fit=0.1,
    min_fit_clients=cfg.num_clients_per_round_fit,
    fraction_evaluate=0.1,
    min_evaluate_clients=cfg.num_clients_per_round_eval,
    min_available_clients=cfg.num_clients,
    on_fit_config_fn=get_on_fit_config(cfg.config_fit),
```

```
    evaluate_fn=get_evaluate_fn(num_features, cfg.num_classes,
       testloader),
)
```

Training: In the 'start-simulation' function, we will specify the training details. We plan to conduct the training for x number of rounds, and the agents will be executed sequentially. The final parameter is set in the following code snippet to improve execution speed with the Ray library used to run the simulation.

```
# Start Simulation
history = fl.simulation.start_simulation(
    client_fn=client_fn,
    num_clients=cfg.num_clients,
    config=fl.server.ServerConfig(num_rounds=cfg.num_rounds),
    strategy=strategy,
    client_resources={"num_cpus": 2, "num_gpus": 0.0},
)
```

6. Saving Results: We store simulation outcomes for future reference.

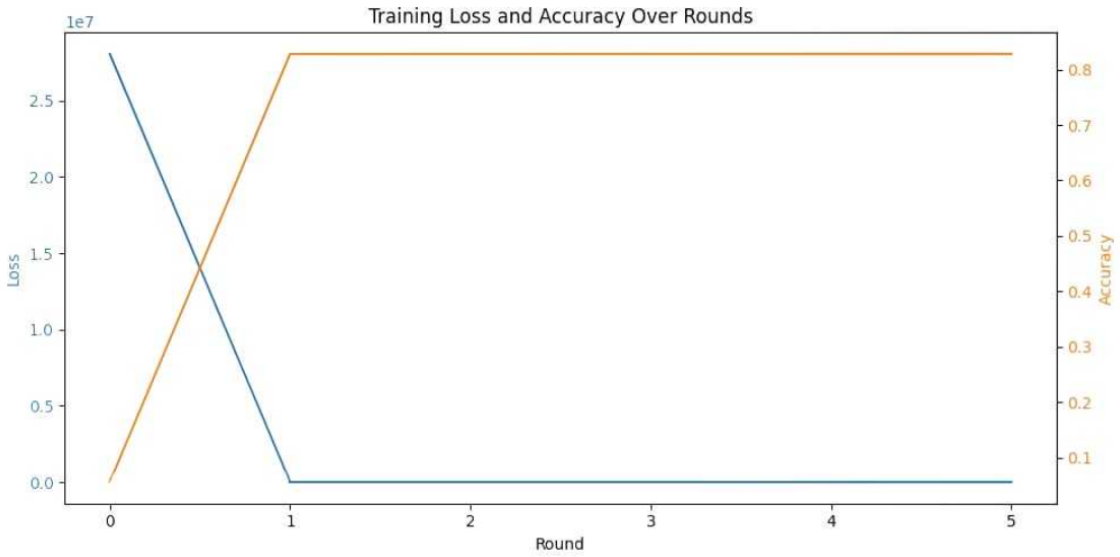


Figure 2.14: CNN in Federated Learning Performance.

The Federated Learning (FL) experiment employed a CNN model, undergoing multiple training rounds to evaluate loss and accuracy. Initially, the model performed poorly with high loss (2.7) and low accuracy (0.05). Subsequent rounds saw a notable improvement, notably in the second round where loss sharply decreased (0.1) alongside a substantial accuracy increase (82%). While subsequent rounds exhibited a gradual decrease in loss and continued accuracy enhancement, the rate slowed, reflecting diminishing returns. Overall, FL proved effective in enhancing the model's predictive capabilities, demonstrating its

ability to maintain high accuracy while minimizing loss across iterative training rounds. This iterative process typifies convergence towards an optimal solution in training.

2.9 Conclusion

In conclusion, the integration of the CNN model with Flower framework in a structured simulation setup presents a powerful approach for federated DDoS detection. The CNN model's strong performance and simplicity make it well-suited for federated learning scenarios, while Flower's adaptability streamlines its integration and optimization. This combination not only enhances network security against DDoS attacks but also improves resource utilization and scalability in federated environments.

General Conclusion

Synthesis

In response to the escalating threat of cyber attacks, our research focused on developing a DDoS attack detection system leveraging Deep Learning techniques. We utilized the "Edge-IIoTset dataset" and employed various preprocessing methods to enhance data accessibility and accuracy. Exploring Deep Learning models like CNN, CNN-LSTM, and CNN-LSTM-GRU for binary and multi-class classification tasks, we encountered challenges in parameter selection, particularly in choosing activation functions. Despite these hurdles, we opted for the CNN model for federated learning due to its simplicity and promising performance. Through meticulous data preprocessing, model development, and parameter tuning, we achieved encouraging results, showcasing the effectiveness of our approach in DDoS attack detection. These findings highlight the potential for further advancements in federated learning for cybersecurity applications.

Perspectives

Moving forward, there are several promising avenues for future work based on our research findings. Firstly, conducting a detailed exploratory analysis of PCAP files using various network traffic flow generators can offer valuable insights into attack patterns in IIoT traffic, enhancing our system's detection capabilities. Additionally, exploring unsupervised deep learning methods such as autoencoders and generative adversarial networks for anomaly detection in network traffic can complement existing supervised approaches and uncover novel attacks. Integration with other security measures like intrusion detection systems and firewall technologies can bolster our defense mechanism against cyber threats.

Moreover, employing a variety of deep learning models, including deep neural net-

works (DNNs) and recurrent neural networks (RNNs), can significantly enhance the robustness and accuracy of our detection system by capturing complex temporal patterns and intricate features in network traffic data. Developing mechanisms for real-time monitoring and response to detected DDoS attacks, along with evaluating our system's effectiveness in real-world IIoT environments, will be critical steps in advancing cybersecurity. By pursuing these avenues, we aim to further strengthen our DDoS attack detection system and contribute to the broader enhancement of cybersecurity in IIoT environments.

Bibliography

[1] trendmicro. https://www.trendmicro.com/vinfo/us/security/definition/industrial-internet-of-things-iiot.

[2] kaspersky, September 13, 2023. https://www.kaspersky.com/about/press-releases/2023_attacks-on-industrial-sector-hit-record-in-second-quarter-of-2023.

[3] Aishah Abdullah, Reem Hamad, Mada Abdulrahman, Hanan Moala, and Salim Elkhediri. Cybersecurity: a review of internet of things (iot) security issues, challenges and techniques. In *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, pages 1–6. IEEE, 2019.

[4] L Aberle. A comprehensive guide to enterprise iot project success. *IoT Agenda*, 1, 2015.

[5] Nasr Abosata, Saba Al-Rubaye, Gokhan Inalhan, and Christos Emmanouilidis. Internet of things for system integrity: A comprehensive survey on security, attacks and countermeasures for industrial applications. *Sensors*, page 2, 2021.

[6] Bandar Alotaibi. A survey on industrial internet of things security: Requirements, attacks, ai-based solutions, and edge computing opportunities. *Sensors (Basel, Switzerland)*, page 7470, Aug 2023.

[7] Asmaa Shaker Ashoor and Sharad Gore. Importance of intrusion detection system (ids). *International Journal of Scientific and Engineering Research*, 2(1):1–4, 2011.

[8] Zeinab Bakhshi, Ali Balador, and Jawad Mustafa. Industrial iot security threats and concerns by considering cisco and microsoft iot reference models. In *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 173–178. IEEE, 2018.

[9] A. Bhandari, A.L. Sangal, and K. Kumar. Characterizing flash events and distributed denial-of-service attacks: An empirical investigation. *Security and Communication Networks*, 9:2222–2239, 2016.

[10] M. Burhan, R. Rehman, B. Khan, and B.-S. Kim. Iot elements, layered architectures and security issues: A comprehensive survey. *Sensors*, page 2796, Aug 2018.

[11] Muhammad Burhan, Rana Asif Rehman, Bilal Khan, and Byung-Seo Kim. Iot elements, layered architectures and security issues: A comprehensive survey. *Sensors*, 18(9), 2018.

[12] Jeff Desjardins. visualcapitalist, January 9, 2018. `https://www.visualcapitalist.com/timeline-industrial-internet-things/`.

[13] G. Falco, C. Caldera, and H. Shrobe. Iiot cybersecurity risk modeling for scada systems. *IEEE Internet of Things Journal*, pages 4486–4495, 2018.

[14] Mohamed Amine Ferrag and et al. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, 50:102419, 2020.

[15] Mohamed Amine Ferrag, Othmane Friha, Djallel Hamouda, Leandros Maglaras, and Helge Janicke. Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning. *IEEE Access*, 10:40281–40306, 2022.

[16] Mohamed Amine Ferrag, Oualid Friha, Leandros Maglaras, Helge Janicke, and Lei Shu. Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis. *IEEE Access*, 9:138509–138542, 2021.

[17] Francisco Javier Folgado, David Calderón, Isaías González, and Antonio José Calderón. Review of industry 4.0 from the perspective of automation and supervision systems: Definitions, architectures and recent trends. *Electronics*, 13(4), 2024.

[18] GeeksforGeeks. Intrusion detection system (ids). `https://www.geeksforgeeks.org/intrusion-detection-system-ids/`.

[19] Sandeep Gurung, Mirnal Kanti Ghose, and Aroj Subedi. Deep learning approach on network intrusion detection system using nsl-kdd dataset. *International Journal of Computer Network and Information Security (IJCNIS)*, 11(3):8–14, 2019.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[21] IBM. Deep learning. `https://www.ibm.com/topics/deep-learning`.

[22] Max Ingham, Jims Marchang, and Deepayan Bhowmik. Iot security vulnerabilities and predictive signal jamming attack analysis in lorawan. *IET information security*, pages 368–379, 2020.

[23] Pls Jayalaxmi, Rahul Saha, Gulshan Kumar, Neeraj Kumar, and Tai-Hoon Kim. A taxonomy of security issues in industrial internet-of-things: Scoping review for existing solutions, future implications, and research challenges. *IEEE Access*, pages 25344–25359, 2021.

[24] Evgeny Kalinin, Danila Belyakov, Dmitry Bragin, and Anton Konev. Iot security mechanisms in the example of ble. *Computers*, 10(12):162, 2021.

[25] T Kamaleshwar, R Lakshminarayanan, Yuvaraja Teekaraman, Ramya Kuppusamy, and Arun Radhakrishnan. Self-adaptive framework for rectification and detection of black hole and wormhole attacks in 6lowpan. *Wireless Communications and Mobile Computing*, pages 1–8, 2021.

[26] Kanimozhi and T Prem Jacob. Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset cse-cic-ids2018 using cloud computing. In *2019 International Conference on Communication and Signal Processing (ICCSP)*, pages 0033–0036. IEEE, 2019.

[27] Ruhul Amin Khalil et al. Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications. *IEEE Internet of Things Journal*, 8(14):11016–11040, 2021.

[28] S.S. Kolahi, K. Treseangrat, and B. Sarrafpour. Analysis of udp ddos flood cyber attack and defense mechanisms on web server with linux ubuntu 13. In *Proceedings of the International Conference on Communications, Signal Processing, and their Applications (ICCSPA'15)*, Sharjah, United Arab Emirates.

[29] F. Laghrissi, S. Douzi, and K. et al. Douzi. Intrusion detection systems using long short-term memory (lstm). *J Big Data*, 8:65, 2021.

[30] J. Li, L. Lyu, X. Liu, X. Zhang, and X. Lyu. Fleam: A federated learning empowered architecture to mitigate ddos in industrial iot. *IEEE Transactions on Industrial Informatics*, 18:4059–4068, 2021.

[31] R. Mohammadi, R. Javidan, and M. Conti. Slicots: An sdn-based lightweight countermeasure for tcp syn flooding attacks. *IEEE Transactions on Network and Service Management*, 14:487–497, 2017.

[32] Saeid Nahavandi. Industry 5.0—a human-centric solution. *Sustainability*, page 4371, 2019.

[33] Dinh C. Nguyen, Ming Ding, Pubudu N. Pathirana, Aruna Seneviratne, Jun Li, Dusit Niyato, and H. Vincent Poor. Federated learning for industrial internet of things in future industries. *IEEE Wireless Communications*, 28(6):192–199, 2021.

[34] Yash Shah and Shamik Sengupta. A survey on classification of cyber-attacks on iot and iiot devices. In *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 0406–0413. IEEE, 2020.

[35] Karanpreet Singh, Paramvir Singh, and Krishan Saluja. A systematic review of ip traceback schemes for denial of service attacks. *Computers Security*, 56, 07 2015.

[36] Konstantinos Tsiknas, Dimitrios Taketzis, Konstantinos Demertzis, and Charalabos Skianis. Cyber threats to industrial iot: A survey on attacks and countermeasures. *IoT*, pages 163–186.

[37] Analytics Vidhya. A brief overview of recurrent neural networks (rnn). `https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/`, 2022.

[38] Yanqi Zhao, Yiming Liu, Aikui Tian, Yong Yu, and Xiaojiang Du. Blockchain based privacy-preserving software updates with proof-of-delivery for internet of things. *Journal of Parallel and Distributed Computing*, pages 141–149, 2019.

[39] Lu Zhou, Kuo-Hui Yeh, Gerhard Hancke, Zhe Liu, and Chunhua Su. Security and privacy for the industrial internet of things: An overview of approaches to safeguarding endpoints. *IEEE Signal Processing Magazine*, pages 76–87, 2018.

Acronyms

IFA Fundamental Computing and its Applications

RSD Networking and Distributed Systems

AI Artificial Intelligence

NTIC New Technologies of Information and Communication

IoT Internet of Things

DDoS Distributed Denial of Service

ICS Industrial Control Systems

IIoT Industrial Internet of Things

CNN Convolutional Neural Network

LSTM Long Short-Term Memory

GRU Gated Recurrent Unit

IT Information Technology

OT Operational Technology

SCADA Supervisory Control and Data Acquisition

PLC Programmable Logic Controller

MITM Man-in-the-Middle

IIRA Industrial Internet Reference Architecture

UDP User Datagram Protocol

TCP Transmission Control Protocol

HTTP Hypertext Transfer Protocol

ICMP Internet Control Message Protocol

FL Federated Learning

SMOTE Synthetic Minority Over-sampling Technique

FLWR Flower

CI Critical Infrastructure

IDS Intrusion Detection System

ML Machine Learning

DL Deep Learning