Abou Bekr Belkaid University
Tlemcen Algeria
جامعة أبي بكر بلقايد
تلمسان الجزائر

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
FACULTY OF SCIENCES
DEPARTMENT OF COMPUTER SCIENCES

DOCTORAL THESIS

A thesis submitted for the award of the degree of

Doctor of Philosophy

In: COMPUTER SCIENCE

Speciality: Networks and Distributed Systems

by

Radjaa BENSAID

Security and Privacy Issues in Fog Computing for the Internet of Things

Thesis will be defended publicly in front of the committee composed of:

Mr Abdelkrim BENAMAR	Full Professor	University of Tlemcen	President
Mrs Nabila LABRAOUI	Full Professor	University of Tlemcen	Supervisor
Mr Ado Adamou ABBA ARI	Associate Professor	University of Meroua	Co-Supervisor
Mr Bouabdellah KECHAR	Full Professor	University of Oran 1	Examiner
Mr Sofiane BOUKLI-HACENE	Full Professor	University of Sidi Belabes	Examiner
Mr Mohammed MANA	Associate Professor	University of Tlemcen	Examiner

University Year 2023-2024

Acknowledgments

In the name of Allah, the Most Gracious, the Most Merciful.

الْحَمْدُ لِلَّهِ

I extend my deepest gratitude to the divine power that sustained me through life's challenges and enabled me to complete this dissertation. I sincerely appreciate everyone who supported me on this journey.

Special thanks go to my supervisor, Mrs. Nabila LABRAOUI, for her insightful comments, guidance, and administrative support, and to my co-supervisor, Mr. Ado Adamou Abba Ari, for his valuable advice on academic research.

I am grateful to the jury members, Mr. Abdelkrim BENAMAR, Mr. Sofiane BOUKLI-HACENE, Mr. Bouabdellah KECHAR, and Mr. Mohamed MANA, for accepting the invitation to serve and their honorary presence.

My deepest appreciation goes to Mr. Leandros MAGLARAS for his invaluable guidance and support. I also thank Mrs. Hafida SAIDI, Mrs. Sihem BENFRIHA, Mr. Joel Herve MBOUSSAM EMATI, and Mr. Ahmed Mahmoud ABDU LWAHHAB for their collaborative spirit and encouragement in our research publications.

Dedication

وَمَا تَوْفِيقِي إِلا بِاللهِ عَلَيْهِ تَوَكَّلْتُ وَإِلَيْهِ أُنِيبُ

With deepest love, I dedicate this dissertation to:
My incredible parents, whose unwavering support and sacrifices made this journey possible. Words cannot express my gratitude.
My dear sister Imane, Your strength, encouragement, and unwavering belief in me carried me through difficult times.
My loving husband, Benameur Mohamed Amine, a constant source of support and my rock throughout this entire process. I will be forever grateful.
My precious daughter, Fatima Zahraa Lina You are my strength and inspiration. The sacrifices I have made are dedicated to paving the way for your bright future, filled with joy and success.
My mother-in-law, Chahrazed, whose unwavering support has been a source of strength throughout my research.
My brother Allaa Eddin and his wife Mama, your unwavering support is a constant source of strength.
This research is dedicated to the brightest stars in my life: Nada, Mohammed, Yassin, Abd El Rahim, and Taim Allah. I hope it shows you the power of dedication and learning and inspires you to reach your dreams.
To all those who have enriched my life, I dedicate this research.

Radjaa

Abstract

In recent decades, the world has witnessed a significant increase in connectivity due to the widespread adoption of the Internet of Things (IoT), cloud, and fog computing. These technologies have revolutionized data collection, processing, and storage. Fog computing, designed to reduce latency and enable local data preprocessing, addresses some limitations of centralized cloud computing. However, it faces significant challenges in ensuring data privacy and security, being vulnerable to cyber-attacks. To fully benefit from fog computing, robust security and privacy techniques must be implemented.

In this thesis, we propose two contributions that utilize various technologies. Given the diverse privileges of users accessing fog networks, our first contribution involves the adoption of an Adaptive Neuro-Fuzzy Inference System (ANFIS) within Software-Defined Networking (SDN). This approach aims to detect and mitigate Syn flood Distributed Denial-of-Service (DDoS) attacks in fog computing networks. In our second contribution, we present a Federated Learning-Based Intrusion Detection System (IDS) approach specifically designed for IoT-enabled smart healthcare systems. This approach uses decentralized identifiers (DID) and verifiable credentials (VC) to facilitate user authentication.

Through experiments conducted on securing the fog-enabled IoT network, Our proposed solution combines IDS and SDN capabilities, using machine learning to provide robust security and preserve user privacy in fog computing environments. Experimental findings demonstrate the effectiveness of this integrated approach in balancing strong security measures and user privacy preservation.

Keywords: Security and Privacy, IoT, Fog computing, SDN, IDS, Deep Learning, Blockchain, SSI, DID, VC.

Résumé

Au fil des dernières décennies, la connectivité mondiale a connu une croissance importante grâce à l'adoption massive de l'Internet des objets (IdO), de l'informatique en nuage et de en périphérie (fog computing). Ces technologies ont révolutionné la collecte, le traitement et le stockage des données. L'informatique en périphérie, conçue pour réduire la latence et permettre le prétraitement local des données, adresse certaines limitations de l'informatique en nuage centralisée. Cependant, elle fait face à des défis importants pour garantir la confidentialité et la sécurité des données, étant vulnérable aux cyberattaques. Pour bénéficier pleinement de l'informatique en périphérie, des techniques robustes de sécurité et de confidentialité doivent être mises en œuvre.

Dans cette thèse, nous proposons deux contributions qui utilisent différentes technologies. Compte tenu des privilèges divers des utilisateurs accédant aux réseaux en brouillard, notre première contribution implique l'adoption d'un système d'inférence floue adaptatif neuro-fuzzy au sein d'un réseau défini par logiciel (SDN). Cette approche vise à détecter et à atténuer les attaques par déni de service distribué (DDoS) de type Syn flood dans les réseaux d'informatique en brouillard. Dans notre deuxième contribution, nous présentons une approche de système de détection d'intrusions (IDS) basée sur l'apprentissage fédéré spécifiquement conçue pour les systèmes de santé intelligents activés par IdO. Cette approche utilise des identifiants décentralisés (DID) et des attestations vérifiables (VC) pour faciliter l'authentification des utilisateurs.

Grâce aux expériences menées sur la sécurisation du réseau IdO activé par brouillard, notre solution proposée combine les capacités des IDS et du SDN, en utilisant l'apprentissage automatique pour fournir une sécurité robuste et préserver la confidentialité des utilisateurs dans les environnements de l'informatique en brouillard. Les résultats expérimentaux démontrent l'efficacité de cette approche intégrée dans l'équilibre entre des mesures de sécurité solides et la préservation de la confidentialité des utilisateurs.

Mots-clés: Sécurité et confidentialité, IdO, Informatique en brouillard, SDN, IDS, Apprentissage profond, Blockchain, SSI, DID, VC.

ملخَّص

في العقود الأخيرة، شهد العالم زيادة كبيرة في الاتصال بسبب الانتشار الواسع لتبني إنترنت الأشياء (IoT) والحوسبة السحابية والضبابية. هذه التقنيات أحدثت ثورة في جمع البيانات ومعالجتها وتخزينها. الحوسبة الضبابية، المصممة لتقليل التأخير وتمكين المعالجة المسبقة للبيانات محلياً، تعالج بعض القيود في الحوسبة السحابية المركزية. ومع ذلك، فإنها تواجه تحديات كبيرة في ضمان خصوصية البيانات وأمنها، كونها عرضة للهجمات الإلكترونية. للاستفادة الكاملة من الحوسبة الضبابية، يجب تنفيذ تقنيات أمن وخصوصية قوية.

في هذه الأطروحة، نقترح مساهمتين تستخدمان تقنيات متنوعة. نظرًا لتنوع امتيازات المستخدمين الذين يصلون إلى شبكات الضباب، تتضمن مساهمتنا الأولى اعتماد نظام الاستدلال العصبي الضبابي التكيفي (ANFIS) ضمن الشبكات المعرفة بالبرمجيات .(SDN) يهدف هذا النهج إلى اكتشاف والتخفيف من هجمات حرمان الخدمة الموزعة (DDoS) الناجمة عن فيضان حزم Syn في شبكات الحوسبة الضبابية. في مساهمتنا الثانية، نقدم نهجًا يعتمد على التعلم الفيدرالي لنظام كشف التسلل (IDS) المصمم خصيصًا لأنظمة الرعاية الصحية الذكية الممكّنة بإنترنت الأشياء. يستخدم هذا النهج معرفات لامركزية (DID) وشهادات موثوقة (VC) لتسهيل عملية مصادقة المستخدمين.

من خلال التجارب التي أجريناها على تأمين شبكة إنترنت الأشياء الممكّنة بالحوسبة الضبابية، يجمع الحل المقترح بين قدرات نظام كشف التسلل (IDS) والشبكات المعرفة بالبرمجيات(SDN) ، باستخدام التعلم الآلي لتوفير أمان قوي والحفاظ على خصوصية المستخدم في بيئات الحوسبة الضبابية. تظهر النتائج التجريبية فعالية هذا النهج المتكامل في تحقيق التوازن بين إجراءات الأمان القوية والحفاظ على خصوصية المستخدم.

الكلمات المفتاحية: الأمن والخصوصية، إنترنت الأشياء، الحوسبة الضبابية، الشبكات المعرفة بالبرمجيات، نظام كشف التسلل، التعلم العميق، البلوكتين، الهوية الذاتية السيادية، المعرفات اللامركزية، الشهادات الموثوقة.

List of Publications

International Journals

- Radjaa Bensaid, Nabila Labraoui, Ado Adamou Abba Ari, Leandros Maglaras, Hafida Saidi, Ahmed Mahmoud Abdu Lwahhab, Sihem Benfriha. "Toward a real-time TCP SYN Flood DDoS mitigation using Adaptive Neuro-Fuzzy classifier and SDN Assistance in Fog Computing." (2023).
  DOI: https://doi.org/10.1155/2024/6651584
  URL::://onlinelibrary.wiley.com/doi/10.1155/2024/6651584

- Sihem Benfriha, Nabila Labraoui, Radjaa Bensaid, Haythem Bany Salameh, Hafida Saidi. "FUBA: A fuzzy-based unmanned aerial vehicle behaviour analytics for trust management in flying ad-hoc networks." (2023).
  DOI: https://doi.org/10.1049/ntw2.12108
  URL:https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/ntw2.12108

- Radjaa Bensaid, Nabila Labraoui, Ado Adamou Abba Ari, Hafida Saidi, Joel Herve Mboussam Emati, and Leandros Maglaras. "Secure and Authenticated Federated Learning-based Intelligent NIDS for Smart Healthcare." (2024). Under review

International Conferences

- Radjaa Bensaid, Labraoui Nabila, and Haythem Bany Salameh. "Federated Deep Learning-based Intrusion Detection Approach for Enhancing Privacy in Fog-IoT Networks." 2023 10th International Conference on Internet of Things: Systems, Management and Security (IOTSMS). IEEE, (2023).
  DOI: 10.1109/IOTSMS59855.2023.10325826
  URL:https://ieeexplore.ieee.org/abstract/document/10325826

Contents

List of Figures

List of Tables

List of Acronyms

ACA-Py:	**Hyperledger Aries Cloud Agent Python**
ACI:	**Application Centric Infrastructure**
AI:	**Artificial Intelligence**
ANN:	**Artificial Neural Networks**
ANFIS:	**Adaptive Neuro-Fuzzy Inference System**
APIC:	**Cisco Application Policy Infrastructure Controller**
APIs:	**Application Programming Interfaces**
ATCD:	**Average Time per Credential Duration**
ATTD:	**Average Time per Transaction Duration**
CCED:	**Completed Credential Exchanges Duration**
CD:	**Connect Duration**
CIC-DDoS:	**Canadian Institute for Cybersecurity - DDoS Dataset**
CNN:	**Convolutional Neural Network**
DDoS:	**Distributed Denial of Service**
DL:	**Deep Learning**
DLT:	**Distributed Ledger Technology**
DoS:	**Denial of Service**
DT:	**Decision Tree**
DTLS:	**Datagram Transport Layer Security**
EU:	**End User**
FASA:	**Fog Adaptive Software Assistance**
FNN:	**Feedforward Neural Networks**
FL:	**Federated Learning**
FN:	**False Negatives**
FP:	**False Positives**
GAN:	**Generative Adversarial Network**
GDPR:	**General Data Protection Regulation**
gRPC:	**Remote Procedure Call**
HIDS:	**Host-based Intrusion Detection System**
HIPAA:	**Health Insurance Portability and Accountability Act**
HSMs:	**Hardware Security Modules**
HTTPS:	**Hypertext Transfer Protocol Secure**
IAM:	**Identity and Access Management**

ICD: **Issuing Credential Duration**
ICS: **Industrial Control Systems**
IDS: **Intrusion Detection System**
IID: **Independent and Identically Distributed**
IIoT: **Industrial Internet of Things**
IoMT: **Internet of Medical Things**
IoT: **Internet of Things**
IP: **Internet Protocol**
IPS: **Intrusion Prevention System**
IRC: **Internet Relay Chat**
IT: Information Technology
KNN: **K-Nearest Neighbor**
LSTM: **Long Short-Term Memory**
MITM: **Man-in-the-Middle**
ML: **Machine Learning**
MLP: **Multilayer Perceptron**
MSE: **Mean Squared Error**
NIDS: **Network-based Intrusion Detection System**
OF-Switch: **OpenFlow Switch**
ONF: **Open Networking Foundation**
ONOS: **Open Network Operating System**
P2P: **Peer-to-Peer**
RF: **Random Forest**
RFID: **Radio Frequency Identification**
ReLU: **Rectified Linear Unit**
ROC: **Receiver Operating Characteristic**
RNN: **Recurrent Neural Network**
SA-FLIDS: **Secure and Authenticated Federated Learning-based Intrusion Detection System**
SD: **Startup Duration**
SDN: **Software Defined Networking**
SDFN: **SDN-based Fog Network**
SSI: **Self-Sovereign Identity**
SVM: **Support Vector Machine**
SYN: **Synchronize**
TCP: **Transmission Control Protocol**
TLS: **Transport Layer Security**
TN: **True Negatives**
TP: **True Positives**
VC: **Verifiable Credentials**
VM: **Virtual Machine**
W3C: **World Wide Web Consortium**

General Introduction

The Internet of Things (IoT) has transformed several industries, showing a new era of interconnected devices and data-centric applications. This proliferation of IoT technologies has resulted in extensive data sets that necessitate efficient processing and management. While effective for centralized data processing, traditional cloud computing architectures face challenges such as high latency, particularly for applications requiring real-time responsiveness.

In response to these limitations, researchers have introduced the fog computing paradigm as a promising solution. Fog computing expands upon the capabilities of cloud computing by dispersing data processing tasks and relocating computing resources nearer to the network's edge, where IoT devices are located. This approach reduces latency and conserves bandwidth, making it ideal for IoT applications that require real-time data processing and decision-making.

Nevertheless, despite the advantages of fog computing, it also presents new challenges, particularly in security. IoT devices, characterized by their constrained computing resources and inherent vulnerabilities, are susceptible to exploitation by malicious actors. This susceptibility introduces risks such as unauthorized access, data breaches, and service disruptions, which can significantly impact the reliability and security of IoT applications.

Problem statement

The increasing use of Internet of Things (IoT) applications needs real-time data processing and decision-making at the network edge. While fog computing addresses cloud computing's latency issues by bringing computational resources closer to devices, the inherent limits of IoT devices (limited processing power, storage, and battery life) pose a significant challenge: How can we ensure the security of fog computing and IoT devices to guarantee the sustainability and uninterrupted functioning of IoT applications?

Thesis motivation and objectives

The widespread adoption of connected devices within the Internet of Things (IoT) revolution has left a profound impact on our daily routines, resulting in a substantial data flow that requires rapid processing. However, these IoT devices are intrinsically constrained, making them vulnerable to exploitation by malicious entities. Furthermore, deficient security and privacy protocols can endanger users by exposing sensitive data or permitting unauthorized entry into IoT devices or systems.

To confront this multifaceted challenge, there is a pressing need for a comprehensive strategy that integrates fog computing with advanced security measures to strengthen IoT networks against potential threats while simultaneously facilitating efficient data processing.

The aims of this thesis are divided into two categories: theoretical and technical.

- Theoretical objectives encompass the following points:

    1. Examination of security and privacy requirements and challenges is conducted.
    2. Description of security attacks within fog computing for IoT systems is provided.

3. An evaluation of existing security and privacy techniques is undertaken.

- Technical objectives encompass the following points:

1. A lightweight security framework is designed for integration into fog computing for IoT systems.
2. Detection and mitigation of SYN Flood DDoS attacks in fog computing are achieved through an adaptive neuro-fuzzy inference system, by leveraging software-defined networking.
3. A novel Secure and Authenticated Federated Learning-based Network Intrusion Detection System framework is developed, incorporating Blockchain for fog-IoT enabled smart healthcare systems.
4. Secure federated learning is achieved by using a blockchain-based Self-Sovereign Identity (SSI) model for client authentication.

Thesis Outline & Contributions

This thesis deals with security and privacy problems and is made up of two main parts:

1. LITERATURE REVIEW:

It is structured into three chapters, with each chapter containing an introduction to the defined problem and a thorough review of pertinent literature. The outline is as follows:

- **Chapter 1:** begins by introducing the fundamental concepts of cloud computing, fog computing, and the Internet of Things (IoT). Subsequently, it highlights the importance of security and privacy within fog computing for IoT systems.
- **Chapter 2:** Security and privacy issues are examined in detail, encompassing a wide spectrum of concerns. The chapter delves into the identification and analysis of the most significant potential attacks within fog computing for IoT systems.

- **Chapter 3:** provides an introduction to various security and privacy techniques essential for understanding the following sections of this thesis. In addition, it includes a comprehensive review of related work in the field.

2. SCIENTIFIC CONTRIBUTIONS:

We initiate the presentation of solutions and contributions to our research field, which are delineated across two chapters as follows:

- **Chapter 4:** introduces the first contribution, a novel approach (FASA) to tackle TCP-SYN Flood DDoS attacks within fog computing environments. These attacks, often launched using botnets, overwhelm servers with incomplete connection requests, hindering legitimate users. FASA utilizes an adaptive neuro-fuzzy inference system (ANFIS) to classify network traffic for real-time mitigation with the help of Software Defined Networking (SDN). The model, trained on a recent dataset (CICDDoS2019), demonstrates high accuracy and low false positives in differentiating normal and malicious packets. This translates into a secure and reliable SDN controller that safeguards the availability of fog service.

- **Chapter 5:** we introduce a proposal for a secure and Authenticated Federated Learning-based Network Intrusion Detection System in Fog-IoT-enabled Smart Healthcare Systems, known as SA-FLIDS. Our research aims to detect and mitigate cyber attacks on IoMT by leveraging the capabilities of fog computing to improve data privacy preservation and minimize communication overhead, while addressing vulnerabilities intrinsic to decentralized learning paradigms. This is achieved by leveraging a blockchain-based Self-Sovereign Identity (SSI) model for client authentication. Through performance evaluation, we demonstrate that SA-FLIDS is able to detect attacks on the Internet of Medical Things (IoMT) network while meeting requirements for privacy preservation, scalability, and sustainability.

In the concluding stage, we provide a general conclusion to the entirety of the thesis, along with insights into future work derived from this research.

Part I

LITTERATURE REVIEW

Chapter

1 From Cloud to Fog in the Age of IoT

Contents

1.1 Introduction

The proliferation of interconnected devices has led to a significant increase in data generation, posing latency and real-time analysis challenges. Cloud computing has become the essential solution to address these issues. Although cloud computing provides powerful network resources, concerns about data privacy, security, and network latency make direct attachment of various objects challenging. This has resulted in the emergence of fog computing, which extends cloud computing to the network edge. Fog computing brings computational capabilities closer to IoT devices, enabling real-time processing with low latency. Additionally, fog computing offers mobility support, location awareness, and decentralized infrastructure, with enhanced security compared to cloud computing due to its local data storage infrastructure. The primary objective of this introductory chapter is to lay the groundwork for the Literature Review section and the thesis as a whole. Beginning with fundamental concepts, we offer an overview of fog computing, cloud computing, and Internet of Things (IoT) technologies. In addition, we emphasize the architecture, advantages of fog computing and the diverse applications of the IoT. Subsequently, the chapter delves into the significance of security and privacy in fog computing for IoT. Finally, we conclude the chapter with a summary of its key points.

1.2 Background knowledge

This section presents the foundation for our research proposal by introducing the key technologies that power it: cloud computing, fog computing, and the Internet of Things (IoT). Moreover, IoT acts as a vast network of connected devices across various sectors, holding immense potential to revolutionize industries, improve efficiency, and ultimately enhance our quality of life.

In addition, fog computing plays a vital role in this process. By providing a decentralized layer of processing closer to these devices, it complements the capabilities of cloud computing. Consequently, this synergy enables the development of efficient and scalable IoT applications.

1.2.1 Cloud Computing

In recent years, cloud computing has become a powerful and efficient paradigm that has revolutionized the way computing resources are accessed. This model allows users to tap into a shared pool of configurable resources – networks, storage, and even software – on-demand via a convenient network connection. These resources can be quickly provisioned and released with minimal administrative burden, eliminating the need for initial investment in expensive equipment. Consequently, users gain the flexibility to dynamically scale their resource needs, paying only for what they use. Moreover, cloud computing empowers mobility by allowing access to these resources from anywhere, anytime, as long as there's an internet connection.

According to [1], cloud computing environments are defined by five key characteristics:

- **Self-service on-demand:** Customers can independently and automatically provision computing resources, such as server time and storage space, without needing human intervention from the cloud provider.
- **Broad network access:** Cloud computing services are available over a network and accessible via standard mechanisms by a wide range of client platforms (heterogeneous), including mobile devices, tablets, laptops, and workstations.
- **Resource pooling:** is the process of pooling computing resources to serve several users utilizing a multi-tenant architecture. This allows dynamic assignment and reassignment of resources based on consumer demand.

- **Rapid elasticity:** The capabilities available can scale rapidly both up and down (outward and inward) in response to demand. Users are given the impression that resources are boundless and can be used whenever needed.
- **Measured service:** Cloud systems provide transparency of resource usage for both providers and consumers. This is achieved through monitoring tools that enable tracking and control of resource utilization.

In the realm of cloud computing, we can discern two primary models [1]:

1. **Service model:** This model encompasses three types of services:
    - **Software as a Service (SaaS):** is the core kind of cloud computing that involves the deployment of certain programs such as enterprise resource planning, customer relationship management, Google Docs, and so on.
    - **Platform as a Service (PaaS):** In this structure, the cloud provides a platform for users to create, deploy, and manage apps. PaaS offerings include Google App Engine, IBM BlueMix, and Apache Stratos.
    - **Infrastructure as a Service (IaaS):** IaaS serves as the core cloud service, providing customers with pre-configured physical resources via a virtual interface. IaaS, in contrast to PaaS and SaaS, excludes any applications or operating systems. EC2 from Amazon, IBM SoftLayer, and Google's Compute Engine are all examples of IaaS services.
2. **Deployment model:** Cloud computing may be characterized based on its deployment model, which includes:
    - **Public Cloud:** This sort of cloud hosting provides services via a network that is accessible to the public.
    - **Private Cloud:**It is also referred to as an internal cloud because it operates within a secure environment managed by a specific organization.
    - **Hybrid Cloud:** Combining two or more cloud servers, such as private and public clouds, while maintaining them as separate entities.

- **Community Cloud:** In this paradigm, infrastructure can be shared by numerous companies from the same community that share similar issues, including security, compliance, and authority.

1.2.2 Limitations of Cloud Computing

While cloud computing provides considerable advantages, it is not exempt from drawbacks [2]. Various technical and legal factors lead certain companies to steer clear of cloud-based solutions for these reasons [3]:

- **Data security:** In recent years, there have been notable cases of extensive data breaches in iCloud storage, where the iCloud accounts of numerous celebrities were compromised, resulting in the unauthorized release of their private photos. This represents a significant drawback of the cloud, entrusting your data to online storage exposes it to potential breaches. This concern is a primary factor that deters many companies from transitioning to the cloud, despite its success.
- **Downtime:** While most cloud IT services are typically available 24/7, occasional downtime can occur, either due to planned maintenance or, as mentioned previously, limitations imposed by service providers. During such periods, services become temporarily inaccessible.
- **Limited control:** Users frequently have less control regarding their assets in the cloud system.
- **Network dependency:** Another significant drawback of cloud computing is its reliance on internet connectivity. A stable and reliable Internet connection is essential for seamless operation.
- **Latency:** Emerging applications in the IoT domain demand high real-time responsiveness. However, the traditional cloud computing model involves transmitting data to centralized data centers and waiting for a response, leading to increased system latency. For instance, high-speed autonomous vehicles

necessitate response times in milliseconds. Deviations from expected latency thresholds due to network issues can have severe consequences.

1.2.3 Fog Computing

Cloud computing has dominated the computing landscape, but a new approach called fog computing is emerging to extend its reach. Introduced by Cisco in 2012 and further refined by the OpenFog Consortium (OFC) [4], fog computing falls under the umbrella of edge computing. Edge computing aims to bring applications and services closer to the network's edge, where devices reside. This approach offloads certain tasks like processing, caching, and data aggregation from the cloud to devices at the network edge. As a consequence, Fog computing provides a more effective time to respond to applications and services [5]. Fog computing operates on a foundation of several key characteristics that define its functionality as illustrated in Figure 1.1:

- **Contextual location awareness and minimal latency:** Fog computing emerged from the idea of empowering devices at the network's edge with advanced services. This focus on the "edge" makes it perfect for applications that demand minimal latency, such as real-time gaming, high-fidelity video streaming, and augmented reality. When fog nodes are placed closer to users, data processing and response times are significantly faster compared to relying on distant cloud servers. This ability to crunch data locally translates to real-time interactions, a critical factor for applications sensitive to delays or timing constraints.

- **Geographic dispersion:** Fog computing shines compared to cloud computing when it comes to services and applications needing large-scale deployments. This is because the strength of the fog lies in its geographical distribution. Imagine high-quality streaming services for mobile vehicles – strategically placed fog nodes along highways, acting as proxies and access points, ensuring reliable delivery by processing and caching content closer to the vehicles.

- **Scalability and mobility support:** Fog computing excels at supporting mobile devices efficiently. This is because many fog applications require seamless communication with devices on the move, necessitating built-in mobility support.

- **Heterogeneity:** The power of fog computing lies in its distributed nature. Fog nodes exist in a variety of forms and sizes, distributed across a wide range of situations. However, this distribution itself creates a challenge of heterogeneity. Not only do fog nodes vary, but end-user devices themselves can have different communication protocols and capabilities. Designing systems that can effectively accommodate this heterogeneity is crucial for a successful implementation.

- **Interoperability and federation:** Smooth delivery of services such as real-time video streaming relies on collaboration between different fog providers. This necessitates two key capabilities: interoperability and federation. On one hand, fog components from various vendors need to seamlessly work together. services themselves must be federated, allowing them to function across different domains.

1.2.4 Architecture of Fog Computing

Building on a well-established three-layer foundation, the fog computing architecture adopts a hierarchical structure [5],[6], as depicted in Figure 1.2. This three-layer hierarchy consists of the following:

1. **IoT/User Layer:** This layer acts as the closest point of contact between the end-user and the physical world. Imagine a vast network of geographically distributed devices, sensors, smartphones, smart vehicles, and more. These devices form the backbone of this layer and act as the eyes and ears of fog. Their primary function is to detect and collect data on the surrounding environment, capturing characteristics of physical objects or events.

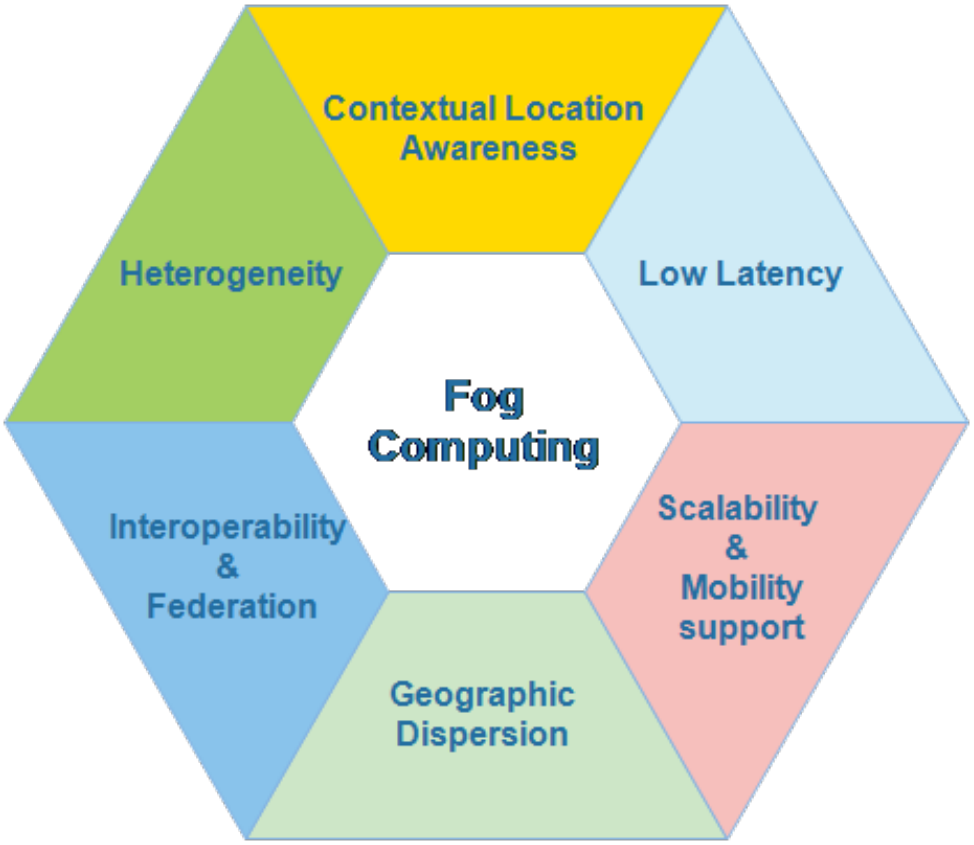


Figure 1.1. *Characteristics of Fog Computing.*

The acquired data is subsequently transferred to higher levels of the fog architecture for additional processing as well as storage.

2. **Fog Layer:** This layer, located at the network's perimeter, is made up of many fog nodes, which include routers, gateways, switches, access points, and fog-specific servers. These fog nodes are widely spread in a variety of locales, including shopping malls, bus terminals, streets, parks, and more, bridging the gap between the devices themselves and the data center. They might be fixed in permanent locations or mobile on moving carriers. End-user devices can easily communicate with fog nodes to get into resources. These nodes are capable of computing, transmitting, and temporarily storing the information gathered, allowing for smooth operations.
3. **Cloud Layer:** The cloud layer is made up of several powerful storage and server equipment that provide a variety of application support such as smart homes, connected vehicles, smart factories, and more. With its powerful computer and

storage capabilities, it allows for comprehensive computational analysis and the permanent storage of massive volumes of data.

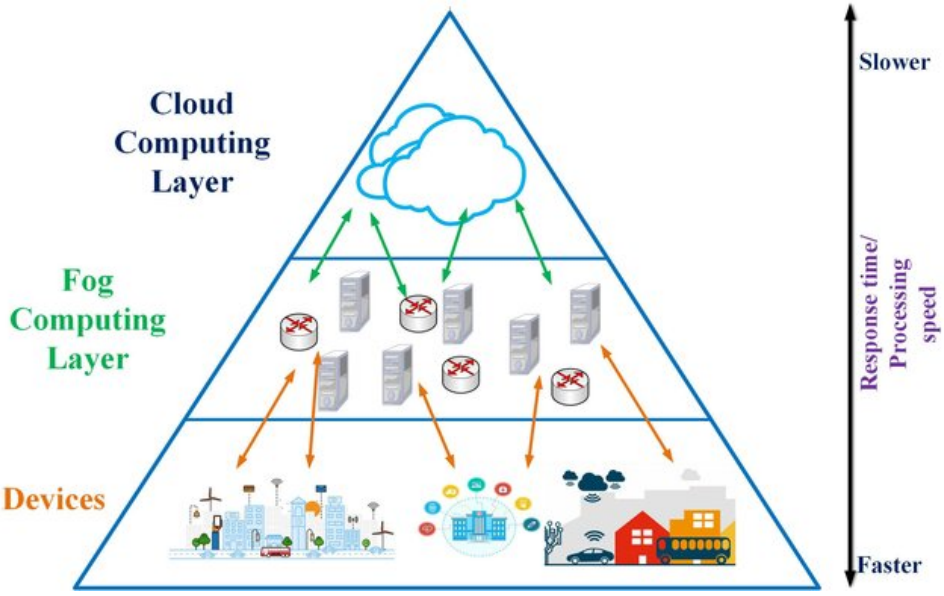


Figure 1.2. *Fog Computing Architecture [6].*

1.2.5 The Internet of Things (IoT)

is the network of sensing and actuating devices that allows information to be shared in a single framework. This enables massive sense, processing of data, and information presentation by using widespread sensors and computing power in the cloud. IoT enables creative applications by creating a shared operational perspective [7].

As presented by the Radio Frequency Identification (RFID) research group, the IoT represents the global network of interconnected objects that can be uniquely addressed via standard communication protocols. Similarly, the European IoT research project cluster defined it as a network in which things actively engage in social processes, information sharing, and interactions [8]. Objects may interact and communicate with one another, as well as their surroundings, in this network. Through or without direct human interaction, they may communicate data and information, react independently to events in the actual world, affect how procedures are carried out, start activities, and provide services [9].

Regardless of specific definitions, IoT technologies fundamentally involve three paradigms: middleware, oriented objects, and oriented semantics. The usefulness of IoT emerges in application domains where these three paradigms intersect, as this technology fusion enables solutions not possible through individual paradigms alone. Such multidisciplinary delimitation is necessary due to operationalizing IoT across interdisciplinary contexts.

1.2.5.1 IoT Applications

The broad adoption of IoT has the potential to have significant social, environmental, and economic implications. Various IoT-based ideas, including mobility, smart homes and buildings, smart grids, public safety and environmental monitoring, medical and healthcare, industrial manufacturing, agriculture and cultivation, and individual lifestyles, provide multiple benefits and are integrated into our everyday lives. We now rely significantly on these programs, realizing their importance and the benefits they offer [10]. IoT offers many applications that greatly enhance people's daily lives and activities. Figure 1.3 provides potential examples of these IoT applications [11].

- *Smart Home:* Smart Homes: This idea entails installing a network of smart gadgets (such as smart locks, baby monitors, and fire alarms) across a home that interacts locally over wireless channels. A home gateway allows you to access your home gadgets remotely.
- *Smart Healthcare:* Smart healthcare: This program facilitates the gathering, exchange, and storing of patient medical information. Medical sensors, for example, can monitor the pulse of a patient and send it to a hospital computer for diagnosis and monitoring.
- *Smart Grid:* Smart grid is a typical Internet of Things application that measures, monitors, and manages power use. It provides effective and dependable electricity management, encourages energy conservation, and lowers power grid issues and failures.

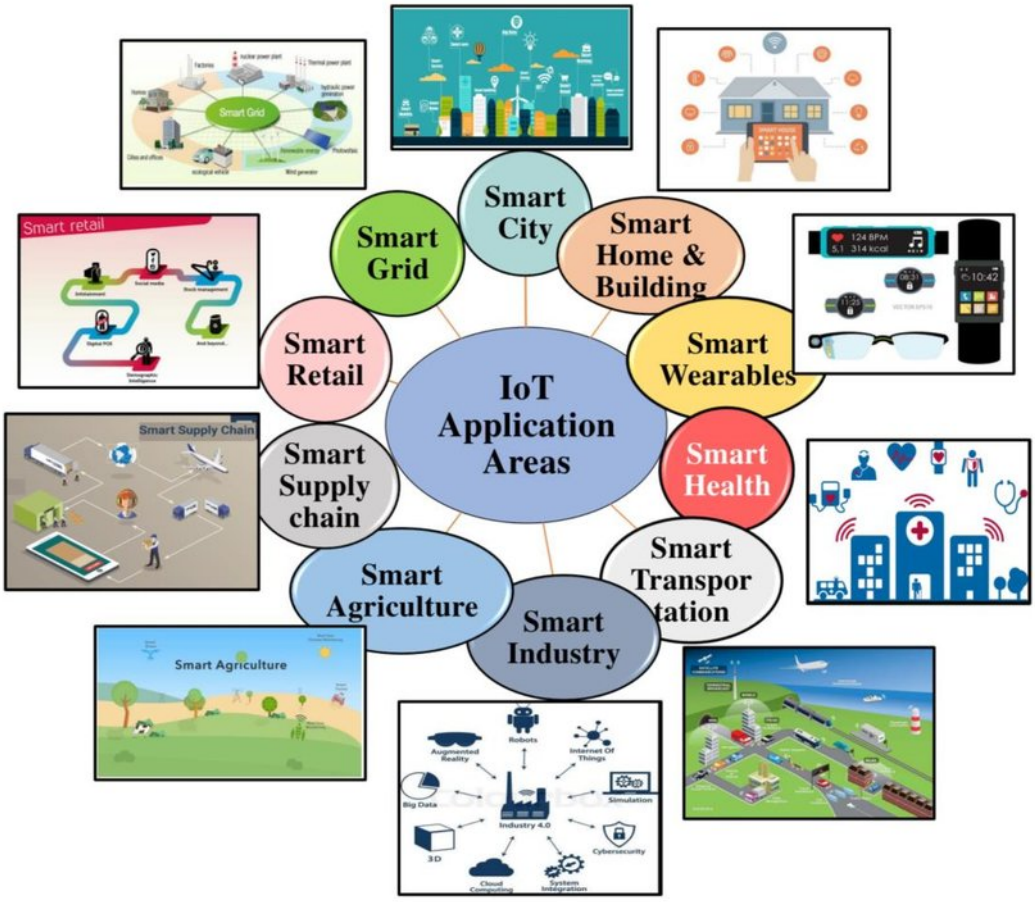


Figure 1.3. *IoT Applications [12].*

- *Smart Transportation:* Smart transportation includes a significant number of cars that may interact with one another (vehicle-to-vehicle), with infrastructure (vehicle-to-infrastructure), and with pedestrians (vehicle-to-pedestrian) via wireless connectivity. Smart cars can monitor traffic, adjust speed, and communicate data to ensure efficient and safe travel.

- *Smart Agriculture:* This application enables remote management of a variety of environmental elements comprising temperature, humidity, irrigation, soil moisture, and microclimate conditions to improve productivity and quality while minimizing costs. Sensors can also be fitted to cattle to monitor their activity and health.

- *Smart Industry (IIoT):*The Industrial (IIoT) uses machine technology to automate production operations with minimum human interaction. IIoT strives to increase

control over the manufacturing process and data management, as well as to solve challenges that will result in efficient and dependable end-products.

- *Smart Retail:* This program is for tracking merchandise in factories as well as while in transit. Sensors can be placed on retail products to track their status. Intelligent retail systems have been created to give better service to customers and attract more business.

1.3 Importance of Security and Privacy in Fog Computing for IoT

Security and privacy are paramount in fog computing for IoT, owing to several critical key reasons [5]:

1. **Data Protection:** Fog computing remains the analysis and storing of sensitive information at the network edge, closer to IoT devices. This closeness heightens the danger of data breaches, illegal access, and data tampering. Implementing strong security measures helps to ensure data integrity, confidentiality, and availability [13].
2. **Threat Landscape:** The interconnectivity of IoT devices and the increasing number of connected objects create a vast attack surface. Fog computing environments are susceptible to various security threats, including malicious software, distributed denial-of-service (DDoS) attacks, and unauthorized access operations. By prioritizing security, potential risks can be mitigated to maintain the integrity and functionality of the IoT ecosystem [14].
3. **Privacy Preservation:** IoT devices capture significant volumes of private and sensitive data, including health data, location, and user activity patterns. Privacy concerns occur when this data is transferred, processed, and stored in fog computing settings. Ensuring privacy protection mechanisms, such as

anonymization, encryption, and user permission, contributes to user trust and compliance with privacy rules [14].

4. **Trust and Reliability:** Fog computing systems rely on trust among connected devices, fog nodes, and the resources of the cloud. Establishing trust requires secure authentication, secure communication protocols, and trust management mechanisms. By ensuring the security and privacy of fog computing systems, trust can be nurtured, leading to increased reliability and confidence in the IoT ecosystem [14].
5. **Real-Time Decision Making:** Fog computing allows real-time data processing and decision-making at the network edge. Security breaches or privacy violations can disrupt these critical operations, resulting in delays, incorrect decisions, or compromised system performance. Robust security and privacy safeguards are essential to maintain the integrity and reliability of real-time IoT applications [15].
6. **Regulatory Compliance:** Fog computing necessitates compliance with privacy and data protection requirements, encompassing the General Data Protection Regulation (GDPR), the Health Insurance Portability and Accountability Act (HIPAA), and others. Following best practices in security and privacy aids organizations in fulfilling legal obligations and mitigating potential legal and financial repercussions [16].

Neglecting the importance of security and privacy in fog computing can result in severe consequences and Cyberattacks can halt critical tasks, leading to financial losses and endangering safety. By prioritizing security and privacy measures, we can release the complete potential of fog computing. This approach ensures the protection of critical data, upholds user privacy, and fosters a secure and ethically connected future.

1.4 Summary

In this chapter, we have discussed the essential ideas of cloud computing, fog computing, and IoT technology. Furthermore, we highlighted the crucial relevance

of security and privacy in fog computing for IoT.

However, the dispersed nature of fog computing creates significant security issues. To properly address these issues, the next chapter looks into the broad security and privacy problems and requirements for fog computing-enabled IoT applications.

Chapter 2

Security and Privacy Issues in Fog Computing for IoT

Contents

2.1 Introduction

This chapter delves into the critical security and privacy challenges that hinder the successful deployment of fog computing for IoT applications. We thoroughly examine the most recent research in this field, leveraging crucial ideas and principles to lay a strong groundwork for the security and privacy approaches we will present subsequently.

2.2 Security Issues in Fog Computing for IoT

Indeed, while fog computing delivers various advantages, such as reduced latency, improved bandwidth efficiency, and enhanced reliability, it also introduces its security challenges. These challenges derive from the distributed nature of fog computing, the heterogeneity of devices and systems involved, and the increased attack surface resulting from the proximity of computing resources to the edge of the network. Hence, it is imperative not to overlook examining security attacks, prerequisites, and mitigation strategies [17].

2.2.1 Security Requirements

The following subsection outlines the essential security requirements for fog-enabled IoT [18], as shown in Figure 2.1:

- *Confidentiality:* Ensure that sensitive data is only accessible to authorized users and entities throughout the fog computing network.
- *Integrity:* Ensure the accuracy and trustworthiness of the data and computations performed within the fog network, preventing unauthorized modifications or tampering.

- *Availability:* Ensure that fog resources and services are available and accessible to legitimate users, even in the face of cyberattacks or system failures.
- *Authentication:* Verify the identities of users, devices, and services that access fog resources, guaranteeing that only authorized parties are permitted to interact with the system.
- *Authorization:* Implement access control policies to determine which users and devices are authorized to perform particular actions or access specific resources within the fog network.
- *Non-repudiation:* Ensure that decisions made within the fog computing system are accountable. It is crucial to secure the vast amounts of data generated by IoT devices. These data should be encrypted before transmission to guarantee that only authorized individuals and devices can access them.
- *Privacy:* Protect the privacy of users' personal data and sensitive information processed within the fog computing environment, complying with relevant privacy regulations and standards.
- *Auditing:* Involves regularly monitoring data and other activities to ensure effective data security and preservation. This process enables individuals to determine the confidentiality of their information.

2.2.2 Security Challenges in Fog Computing for IoT

Fog computing brings cloud computing functionalities closer to the edge of the network, where IoT devices are located. This proximity to data sources improves processing efficiency and reduces response times. However, integrating fog computing with IoT devices introduces several security challenges, as shown in Figure 2.2. It is essential to address these challenges to protect sensitive information and ensure the reliable operation of these systems, we present the security challenges in fog computing for IoT: [19].

Figure 2.1. *Security Requirements.*

- **Distributed Nature:** The distributed nature of fog computing introduces complexities in managing security policies and ensuring consistent security across all fog nodes and devices.
- **Heterogeneity:** Fog networks often consist of various devices and platforms with varying security capabilities, making it difficult to enforce consistent security measures and standards.
- **Interoperability:** Ensuring interoperability and compatibility between fog devices and platforms while maintaining security can be challenging, especially in heterogeneous environments.
- **Resource Constraints:** Many fog devices, particularly IoT devices, have restricted computational capacity, memory, and energy resources, constraining the implementation of robust security mechanisms.

- **Attack Surface:** The proximity of fog nodes to the edge of the network increases the attack surface, making fog computing environments more susceptible to various cyber-attacks, including DDoS attacks, malware infections, and insider threats.
- **Dynamic Environment:** Fog computing environments are dynamic and highly adaptive, with devices joining and leaving the network frequently, posing authentication, access control, and security monitoring challenges.
- **Data Volume and Velocity:** The enormous volume and velocity of data generated and processed within fog networks can overwhelm traditional security mechanisms, which require scalable and efficient security solutions.
- **Security Management and Compliance:** Managing security policies, updates, and compliance requirements across a large number of fog devices and nodes can be complex and resource-intensive, requiring centralized management and automation.

2.3 Privacy Concerns

Ensuring the security of personal data, also known as data privacy, stands as the paramount concern within the IoT application system [19]. Therefore, the process of safeguarding personal data against illegal use, alteration, or disclosure is known as data privacy [20]. Several studies have been conducted to define and clarify the essential elements of fog, cloud, or IoT systems to address the critical concerns of data privacy.

2.3.1 Privacy Requirements

Privacy requirements are paramount considerations in fog computing, particularly in IoT environments where sensitive data is generated, processed, and transmitted across

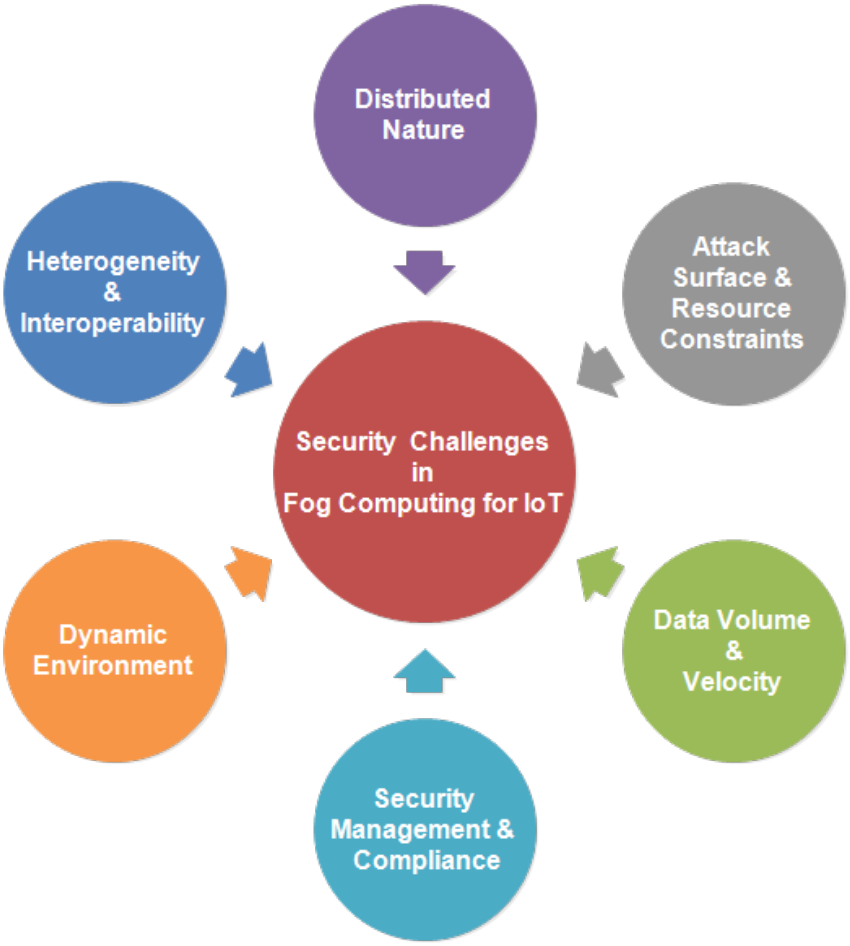


Figure 2.2. *Security Challenges in Fog Computing for IoT.*

distributed networks. Let's delve into privacy requirements specific to fog computing for IoT [21], as illustrated in Figure 2.3 :

1. **Data Minimization:** IoT devices and fog nodes should only gather and use the minimal amount of data required to achieve their goals. Avoiding needless data collection will lower the possibility of privacy breaches.
2. **User Consent and Control:** Users should have control over their data and be able to provide informed consent for its collection, processing, and sharing. Fog computing systems should implement mechanisms to obtain and manage user consent effectively.
3. **Anonymization and Pseudonymization:** To safeguard privacy, personal data should be anonymized or pseudonymized whenever feasible. This minimizes the chance of re-identification, ensuring individual data remains anonymous.

4. **Data Transparency and Accountability:** Fog computing systems need to be explicit about how they handle user data. Users have a right to know exactly how their data is gathered, processed, and shared. Furthermore, mechanisms should be implemented to hold these systems accountable for adhering to privacy regulations, empowering users with control over their data.
5. **Data Security:** Implementing powerful security standards is necessary to protect user data from unauthorized access, modification, or disclosure.
6. **Secure Communication:**Encryption and secure communication methods are required to safeguard data as it is being sent between cloud servers, fog nodes, and IoT devices.

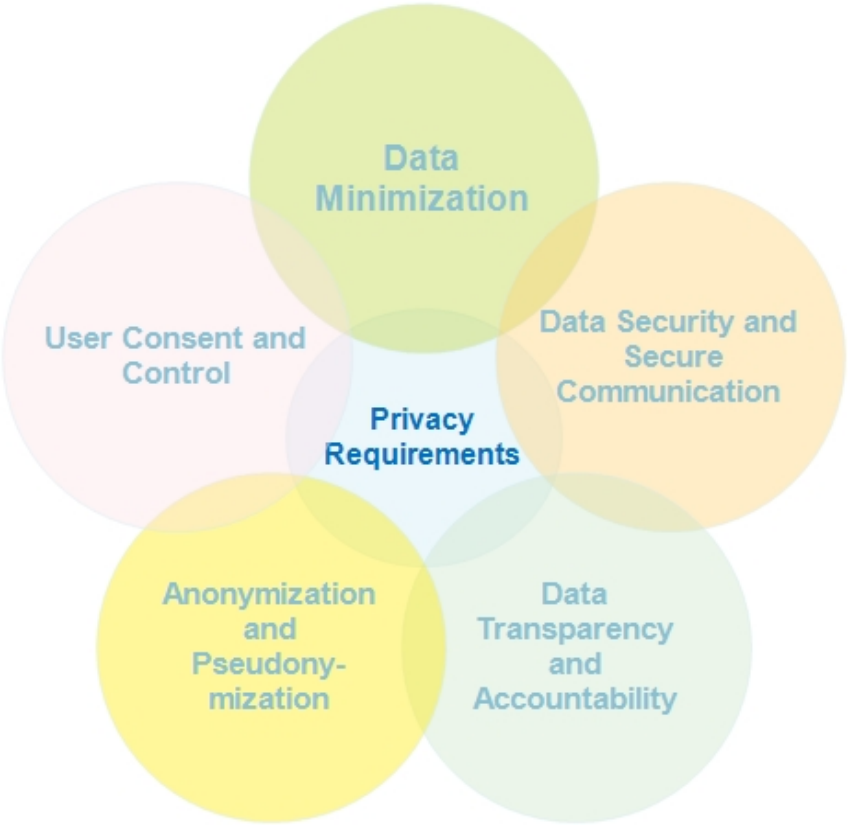


Figure 2.3. *Privacy Requirements in Fog Computing for IoT.*

2.3.2 Privacy Concerns in Fog Computing for IoT

The IoT poses a significant concern, particularly regarding the protection of personal information as it undergoes processing and storage at the network's edge,

close to end-users. The distinctive attributes of fog computing, including its decentralized structure and proximity to users, underscore the heightened importance of implementing robust privacy safeguards. The following sections detail the various aspects of privacy that are pertinent to fog computing environments [22].

- *Identity Privacy:* Within fog computing, protecting identity privacy involves meticulous management of personal identity attributes, including names, mobile numbers, addresses, and other sensitive information. These details are provided to the fog nodes to facilitate authentication and authorize access to services.
- *Data Privacy:* Data privacy concerns the risk of exposure of personal data during its transmission or storage within fog nodes. This exposure can occur due to cyberattacks, accidental losses, or data misuse. The proximity of fog nodes to end-users increases the risk, as attackers may find it easier to target these decentralized systems. Protecting data privacy requires encryption, secure communication protocols, and vigilant monitoring of data handling practices.
- *Usage Privacy:* Usage privacy pertains to safeguarding information concerning the manner and timing of users' interactions with fog services. This includes sensitive details such as users' daily routines, times when they are at home or away, and their patterns of service usage.

2.4 IoT Devices as Botnets

Resource-constrained IoT devices are susceptible to malware, potentially turning them into bots for large-scale attacks [23]. A botnet is a collection of computers linked to the internet whose security has been compromised, giving a malevolent party access to take control. Every hacked device called a "bot," is created when malicious software, commonly referred to as malware, infiltrates a computer. A botnet supervisor can coordinate the activities of these infected machines using communication channels

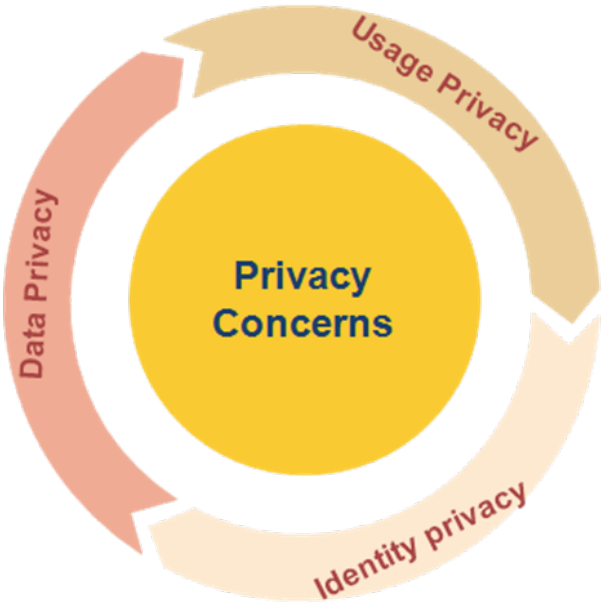


Figure 2.4. *Privacy Concerns in Fog Computing for IoT.*

created by common network protocols such as Hypertext Transfer Protocol (HTTP) and Internet Relay Chat (IRC) [24]. Distributed denial of service (DDoS) attacks, spamming, and other malicious actions are possible with these botnets. The Mirai botnet is among the most well-known instances of IoT devices becoming botnets.

2.4.1 The Mirai Botnet Attack

The Mirai botnet attack in 2017 is a prominent example of the devastating impact of botnet attacks on IoT devices [25]. Mirai built a massive botnet army by exploiting vulnerabilities in IoT devices. It achieved this by scanning the internet for devices with unprotected Telnet ports and attempting to log in using common default passwords. These compromised devices were then used to launch a large-scale DDoS attack. This attack led to widespread disruption, affecting major websites and services. The success of the Mirai botnet and its variants relied on the weak security of IoT products and technology, highlighting the critical need for improved security measures to defend against such attacks.

2.5 Security Weaknesses in IoT Devices

IoT devices, often exhibit security vulnerabilities due to various factors, including their limited computational resources and reliance on default configurations. In the following, we examine the prevalent security vulnerabilities inherent in IoT devices [26].

1. **Limited Computational Resources:**

   The limited processing power of many IoT devices hinders their ability to implement strong security features like encryption, authentication, and intrusion detection, leaving them vulnerable to attacks.Limited Memory and Storage: Constraints on memory and storage capacity may prevent IoT devices from storing security-related data securely or updating firmware to patch known vulnerabilities.

(2) **Default Configurations:**

   (a) Default or Weak Credentials: A major security concern with IoT devices is their reliance on default login credentials. These pre-programmed usernames and passwords are widely known and easily cracked by attackers, leaving devices susceptible to unauthorized access.
   (b) Default Network Settings: Default network settings, such as open Wi-Fi networks or default network protocols. This can expose them to eavesdropping, man-in-the-middle attacks, and unauthorized access.
   (c) Default Services and Ports: Many IoT devices come pre-configured with unnecessary services running or ports open by default. These extra features can be security vulnerabilities, as attackers can exploit them to attain unauthorized access to the device itself or even the entire network.

3. **Lack of Security Updates:**

   (a) Limited Patch Management: Many IoT devices suffer from infrequent or non-existent security updates from their manufacturers. This lack of

patching leaves them exposed to known vulnerabilities and exploits that attackers can readily utilize. Without timely updates, these devices remain at risk of being compromised.

(b) End-of-Life Support: Manufacturers may discontinue support for IoT devices after a certain period, leaving them without access to security patches or updates. This lack of ongoing support increases the risk of security vulnerabilities remaining unaddressed over time.

4. **Insecure Communication Protocols:**

   (a) Unencrypted Communication: Unsecured communication protocols expose sensitive data transmitted by IoT devices. This lack of encryption allows attackers to intercept, eavesdrop on, and even tamper with the data, posing a significant security risk.

   (b) Weak Authentication Mechanisms: Inadequate authentication mechanisms, such as plaintext authentication or weak cryptographic keys. This allows attackers to get illegal access as authorized users or devices, granting them unauthorized access to the entire IoT network.

5. **Physical Security Concerns:**

   (a) Physical Access: Leaving IoT devices in unsecured or public locations exposes them to physical attacks. Attackers with direct access to these devices can bypass security measures, steal them, tamper with them, or even extract sensitive information or alter their functionality entirely.

   (b) Tamper Resistance: Lack of tamper-resistant hardware or mechanisms makes IoT devices vulnerable to physical attacks, including hardware manipulation, reverse engineering, and extraction of cryptographic keys or sensitive data.

2.6 Cyber-attacks on IoT Devices

Cyber-attacks against IoT exploit vulnerabilities in devices, networks, data, and users within IoT ecosystems. These attacks pose significant risks to the integrity, availability, and confidentiality of IoT systems. Due to the growth of connected devices and their integration into multiple areas of daily existence, attacks on IoT devices have grown more frequent. Figure 2.5 present some common attacks on IoT which include [27]:

1. **Physical Attacks:**
    - Tampering: Physical manipulation of IoT devices to disrupt their functionality or extract sensitive information.
    - Theft or Vandalism: Theft or destruction of IoT devices to gain unauthorized access to data or disrupt operations.
2. **Network Attacks:**
    - Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) Attacks: Flooding IoT networks or devices with excessive traffic to overwhelm them and disrupt services.
    - Man-in-the-Middle (MITM) Attacks: malicious actors act as a hidden intermediary between an IoT device and its intended recipient. This allows them to eavesdrop on communication, potentially stealing sensitive data or even manipulating the data being transmitted.
    - Network scanning: Probing the network to identify vulnerable devices or open ports for potential exploitation.
    - Spoofing: Impersonating legitimate IoT devices or networks to gain unauthorized access or deceive users.
3. **Data Attacks:**
    - Data Breaches: Occur when attackers gain unauthorized access to sensitive data stored on the devices themselves or transmitted across IoT networks.

- Eavesdropping: Interception and monitoring of data transmissions between IoT devices and other entities.
- Replay Attacks: Capturing and replaying legitimate data transmissions to deceive IoT systems or gain unauthorized access.
- Data Manipulation: Altering information sent over IoT networks or kept on connected devices to cause harm or deceive users.
- Jamming: Disrupt wireless communication channels by emitting interference signals, making data transmission and access difficult.

4. **Software Attacks:**

- Malware Injection: Installing malicious software on IoT devices. The security of the device may then be compromised by this malware, giving hackers access to steal confidential information or interfere with essential functions.
- Firmware Vulnerabilities: Attackers can exploit weaknesses in the core software (firmware) powering IoT devices. These vulnerabilities can provide a backdoor for unauthorized access, allowing attackers to take control of the device or manipulate its functionality for malicious purposes.
- Code injection: Injecting malicious code into the software of the IoT device to exploit vulnerabilities or gain unauthorized access.
- Zero-day attacks: Exploiting previously unknown vulnerabilities in IoT devices for malicious purposes.

5. **Credential Attacks:**

- Brute Force attacks: Trying to figure out or break passwords to get into IoT networks or devices.
- Credential Theft: Stealing authentication credentials through phishing attacks or other means to gain unauthorized access to IoT systems.

6. **Social Engineering Attacks:**

- Phishing: target users who manage or interact with IoT devices. These deceptive emails or messages attempt to trick them into surrendering sensitive information, such as login credentials, or granting unauthorized access to their IoT devices or the networks they connect to.
- Social manipulation: Manipulating or deceiving users into taking actions that compromise the security of IoT devices or systems.

7. **Supply Chain Attacks:**

- Compromised Components: Introducing compromised hardware or software components into the supply chain to infiltrate IoT devices or networks.
- Tampered Firmware: Tampered Firmware: When software or firmware is altered during production or delivery, the security of connected devices is put at risk.

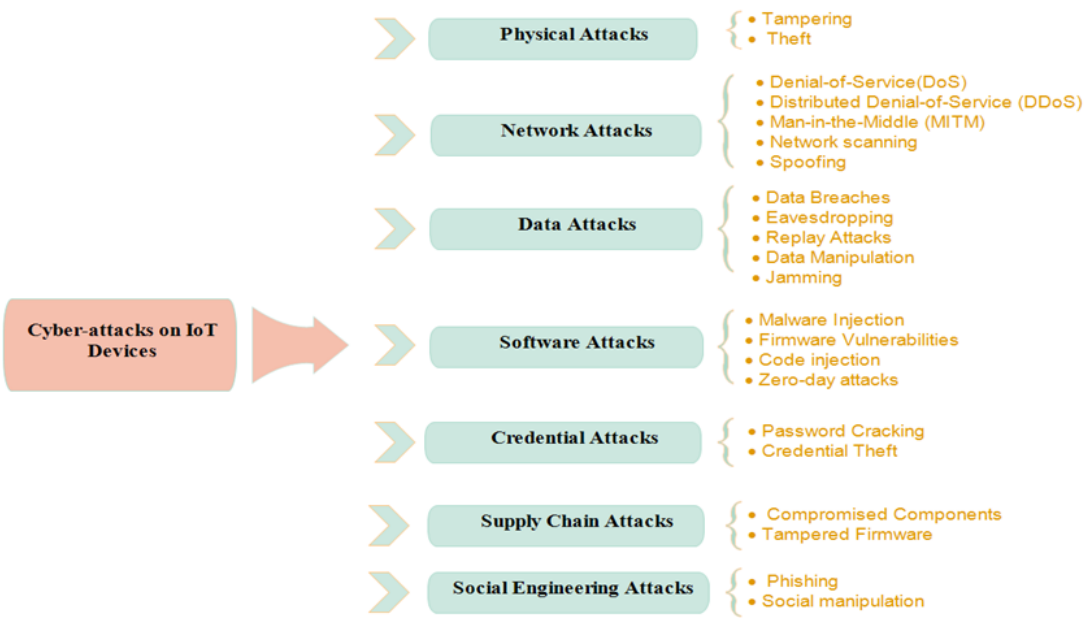


Figure 2.5. *Cyber-attacks on IoT Devices.*

2.7 Summary

In this chapter, we have investigated the security and privacy issues in Fog computing for IoT applications. We started by outlining the security needs and challenges and then proceeded to examine the privacy requirements and concerns related to the gathering of personal data. Next, we explored security vulnerabilities in IoT devices and various cyber-attacks targeting IoT devices. Ensuring robust security and privacy measures is crucial for safeguarding sensitive data, preventing unauthorized access, countering cyber-attacks, preserving user privacy, and fortifying the dependability and robustness of IoT systems. Prioritizing security and privacy is essential to maximize the benefits of fog computing in driving innovation and enabling transformative IoT solutions.

In the chapter that follows, we will analyze different strategies and methods used to improve security and maintain privacy.

Chapter

3

Advanced Techniques in Network Security and Privacy

Contents

3.1 Introduction

Fog computing in IoT applications faces many security and privacy challenges because it involves various devices and systems, operates in a distributed manner, and has an expanded attack surface due to its proximity to the network edge.

However, organizations can effectively mitigate these risks by implementing security and privacy techniques in fog computing for IoT. Robust security measures protect sensitive data, ensuring the confidentiality and integrity of communications within the fog computing environment. Ultimately, addressing security and privacy concerns enables seamless integration of fog computing into IoT ecosystems, unlocking the full potential of IoT technologies while safeguarding data and preserving user trust.

In this chapter, introductory discussions on security and privacy techniques are provided to aid in understanding the subsequent sections of this thesis. In addition, a review of relevant work-related activities is examined.

3.2 Security and Privacy-Preserving Techniques

Establishing security and privacy measures to address security and privacy challenges and strengthen the security of fog computing-enabled IoT ecosystems is imperative. Therefore, based on examining security and privacy issues outlined in the preceding chapter, this chapter explores various existing security and privacy preservation techniques. Here are some key security and privacy-preserving techniques that can be implemented for fog computing [28],[29],[30],[31],[32]:

3.2.1 Network Security

To safeguard network infrastructure from threats, various advanced techniques are used to improve security, ensure data integrity, and maintain seamless operations.

- *Software-Defined Networking (SDN):* Provides centralized control over the network, facilitating dynamic segmentation, access control, and threat detection.
- *Network segmentation:* Isolates critical systems and workloads, minimizing the impact of a breach in one segment.
- *Secure protocols:* Utilizes secure communication protocols like HTTPS and DTLS for data encryption and integrity in transit.
- *Intrusion Detection and Prevention Systems* (IDS/IPS): Monitor network traffic for suspicious activity and proactively defend against attacks.

3.2.2 Data Security

Protecting sensitive data from unauthorized access and ensuring their integrity are crucial aspects of data security, using various techniques to secure data at rest and in transit.

- *Encryption:* It encrypts data, both when it is stored and when it's being transmitted, to prevent illegal access and eavesdropping.
- *Data anonymization and pseudonymization:* Protects sensitive information while preserving some utility for analytics.
- *Secure storage:* Employs tamper-proof hardware and software solutions for secure data storage.
- *Data access control:* Enforces detailed access control policies to limit unauthorized access to data.

- *Blockchain technology:* It provides a decentralized and tamper-resistant mechanism to store and verify the integrity of data. Blockchain can guarantee the integrity and immutability of IoT device data, enhancing data security and privacy.

3.2.3 Identity and Access Management (IAM)

Managing and securing user identities and access privileges is the key to protecting sensitive resources and data within an organization.

- *Strong authentication and authorization:* It requires additional verification steps (multifactor authentication). It assigns permissions based on user roles (role-based access control) to keep user and device access secure.
- *Identity and access management systems:* Centralized systems manage user identities, credentials, and access privileges.
- *Device authentication and authorization:* Authenticates and authorizes devices before granting access to fog resources.

3.2.4 Other Techniques

Beyond network and data security, additional measures are implemented to enhance overall system integrity and proactively address potential threats.

- *Secure boot and firmware updates:* Blocks unauthorized changes to the system's core code for enhanced security.
- *Threat intelligence:*Actively collects and analyzes information on potential security threats to anticipate and prevent attacks.
- *Security monitoring and logging:* Monitors system activity and logs events for incident detection and analysis.

- *Hardware security modules:* Protect sensitive cryptographic keys for encryption and authentication.

- *Machine/deep learning methods:* can be used in threat intelligence and security monitoring. They can analyze large volumes of security-related data, identify patterns, and detect emerging threats or anomalies in real-time. This can improve incident detection and response capabilities.

3.3 Software-Defined Networking (SDN)-An overview

SDN represents a novel approach that divides the network's tasks into two entities: the control plane, which handles network operations, and the data plane, which manages data transfer. This decoupling attempts to develop a network architecture devoid of interconnected network services. The SDN architecture is structured around five primary components [33]:

1. **Controller:** Serves as the central intelligence of the SDN architecture, the SDN controller makes traffic forwarding decisions based on network policies and real-time conditions. It interfaces with the network's switches and routers to enforce these policies and configure their forwarding behavior
2. **Southbound APIs:** Southbound API: Such interfaces enable the SDN controller to connect with network devices like switches and routers. OpenFlow, NETCONF, and OVSDB are examples of commonly used Southbound APIs.
3. **Northbound APIs:** Such interfaces allow communication between the SDN controller and more advanced network management applications or orchestration systems. Northbound APIs abstract the underlying network infrastructure, providing a simplified interface for administrators to interact with the network.
4. **Network Devices:** SDN-compatible switches, routers, and other network devices implement the forwarding rules provided by the SDN controller. These devices

forward traffic according to the rules obtained from the centralized controller., rather than relying solely on locally configured routing tables.

5. **Application Layer:** SDN applications or network management software take advantage of the programmable nature of the SDN controller to implement custom network policies, optimize traffic flows, and automate network provisioning tasks. These applications can be developed by network operators, third-party vendors, or open-source communities.

SDN builds upon previous research in future network technologies, such as active networking (1990), which enables packets circulating in a network to alter network operations dynamically [34]. Active networking comprises hardware capable of routing, switching, and executing code within active packets. Unlike SDN, which separates decision-making from the data plane, active networking conducts computations within packets traversing the network. Although active networking allows real-time adjustments in the network, it faces several challenges [35]:

- No clear application.
- High cost of hardware.
- Security problem (code transported in the clear by packets).
- Interoperability problem.

3.3.1 Benifits of SDN

Decoupling the network's decision-making (control plane) from the data-carrying hardware (data plane) enables the deployment of control services on a platform with superior capabilities to conventional network devices like switches and routers. This innovative architecture offers[36]:

- Simplifying network design by minimizing the total amount of control planes.
- Flexible and programmable.

- Enable adding new services through standard interfaces such as the OpenFlow protocol (OF).
- Efficient data processing.
- Efficient deployment of new services to manage security, quality of service, etc.
- Rapid innovation.

3.3.2 Characteristics of SDN

SDN stands out from traditional networks due to its relocation of the control plane to the software layer, allowing dynamic network access and administration. Key characteristics of SDN include [37]:

1. **Centralized Management:** SDN offers a centralized view and control of the entire network, simplifying network management tasks and reducing operational overhead.
2. **Programmability:** SDN allows administrators to program network behavior through software, allowing rapid configuration changes and adaptability to changing network conditions.
3. **Flexibility and Scalability:** SDN architectures are highly flexible and scalable, allowing networks to adapt to evolving business requirements and accommodate growing traffic demands.
4. **Automation:** SDN enables automation of network provisioning, configuration, and optimization tasks, leading to improved efficiency, reliability, and agility in network operations.
5. **Cost savings:** SDN can lower network infrastructure capital and operating costs by separating the control and data planes and utilizing commodity hardware.

Figure 3.1 depicts the SDN architecture. To effectively program network equipment, it is essential that they are receptive to external directives. To facilitate this, programming interfaces, commonly referred to as APIs (Application Programming Interfaces), are

indispensable. Numerous APIs enable actions to be executed on data plane equipment. Of particular interest to us is the OpenFlow protocol, which serves as the standard for SDN.

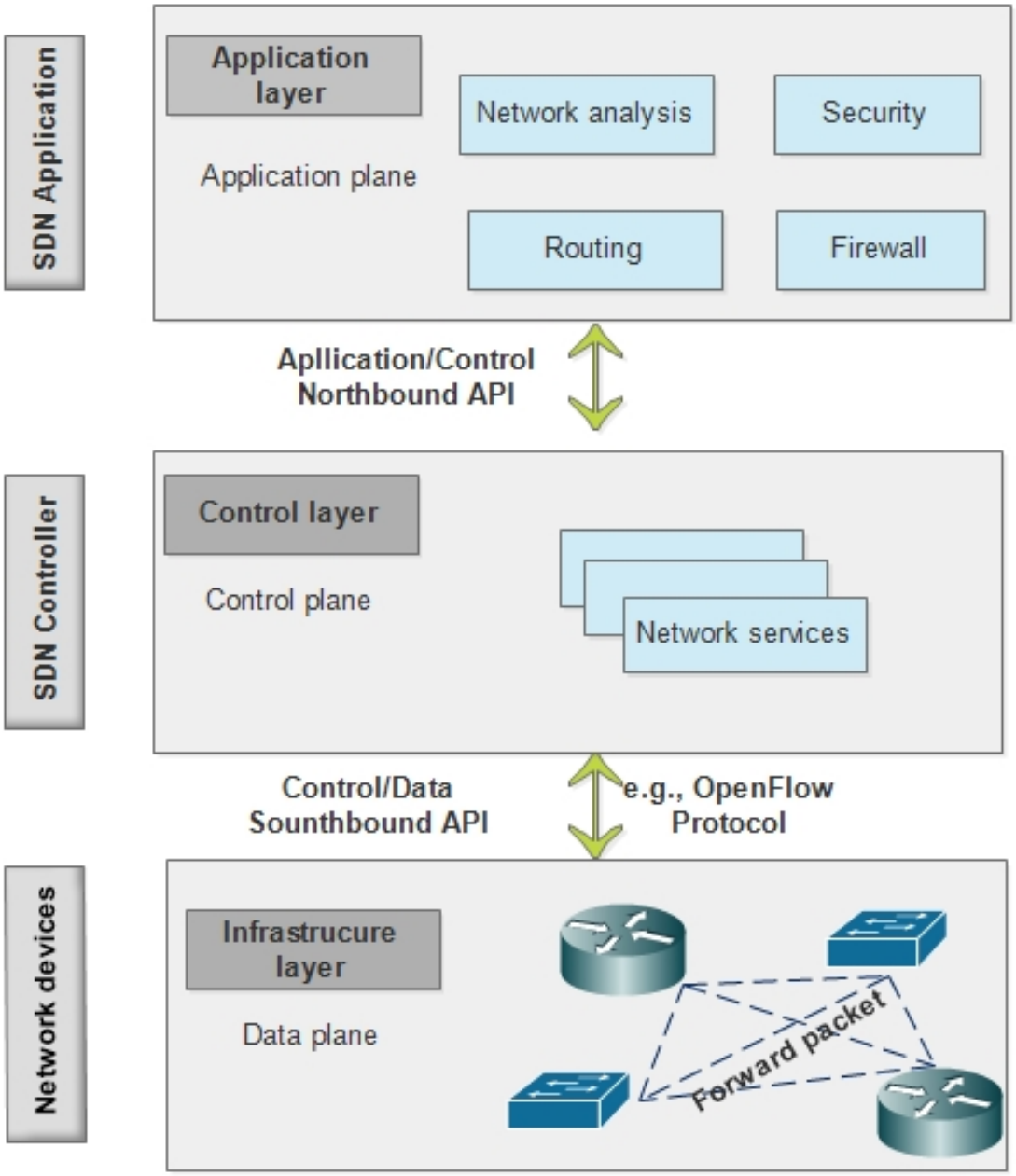


Figure 3.1. *Architecture of SDN.*

3.3.3 OpenFlow Protocol

OpenFlow is a standard protocol used in SDN to enable communication between the SDN controller and network devices such as switches and routers. The controller

uses OpenFlow to program the forwarding rules on these devices. These rules are injected into the data plane, which handles the actual forwarding of traffic [38]. Furthermore, by separating the control plane (decision-making) from the data plane (data forwarding), OpenFlow enables centralized network control. The controller makes the choices, and the network devices follow its instructions. OpenFlow is an open standard, overseen by the Open Networking Foundation (ONF). This means that any vendor's SDN controller can communicate with any OpenFlow-compatible switch or router, promoting flexibility and innovation [39]. The primary components of OpenFlow include [39]:

- Switches that adhere to the OpenFlow protocol.
- Servers equipped to execute the controller process.
- A database housing a comprehensive depiction of the network's topology.

The primary component of switches compatible with the OpenFlow protocol is the flow table. Illustrated in Figure 3.2, each entry in this table comprises the following elements:

1. **Match Fields:** These fields correspond to packets and instruct the switching equipment (OF switch) on how to process a packet. Matching to the header field of an IP packet offers various possibilities. For instance, it's possible to search for matches with standard headers of IP packets, including Mac addresses, source IP, destination IP, source and destination ports, and more. Match Fields encompass fields from the packet header from level L1 to L4, and they are contingent upon the version of OpenFlow.
2. **Instructions**: Upon a packet matching a flow entry, a function is executed. This entails a set of actions to be performed, such as:
    - forwarding a packet to a specified OpenFlow port.
    - modifying packet headers, metadata, etc.

- Sending the packet to the controller.
- drop the packet.

3. **Timeouts:** These signify the lifetime (hard-timeout) and the idle time before expiration (idle-timeout) of OpenFlow rules. An entry persists in the flow tables of the OF switch until the timeout expires.

The primary operation of OpenFlow can be outlined as follows: when a switch receives a packet, it first checks its initial table. If the processing defined in its pipeline requires further searches, it proceeds to investigate its subsequent tables. If the switch fails to identify any rules, it discards the packet. However, it is worth noting that a default rule might exist that directs packets that do not match any OpenFlow rule to the SDN controller, rather than outright rejecting them. This process is illustrated in Figure 3.2.

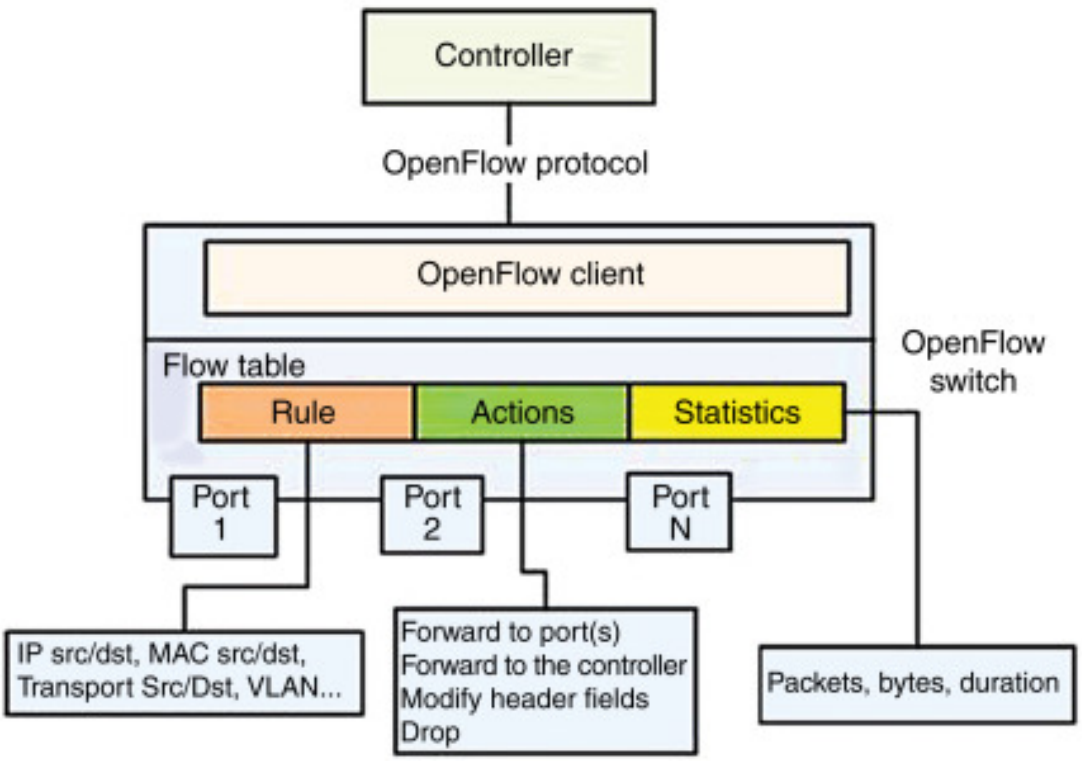


Figure 3.2. *OpenFlow Protocol [40].*

3.3.4 SDN Controller

A pivotal component of SDN is its controller, which serves as its central intelligence. The controller is an intelligent entity capable of configuring network equipment (data plane equipment) and providing a network abstraction layer. It is responsible for adding or removing flow entries from the flow table.

It allows for quick implementation of a change in the network using the OpenFlow protocol. Among the existing controllers, we can find[41]:

- **Floodlight**, created by the Open Networking Foundation (ONF), is a widely used OpenFlow controller written in Java. Its popularity spans study, education, and even real-world network deployments.
- **Ryu**, an example of an open-source SDN controller developed by NTT Laboratories. Written in Python and is meant to be lightweight, flexible, and simple to use. Ryu can configure network devices using the OpenFlow and Netconf protocols.
- **OpenDaylight**, An open-source SDN controller platform supported by a community of developers and contributors. OpenDaylight provides a modular architecture with support for various southbound and northbound protocols.
- **ONOS (Open Network Operating System)**, Another open-source SDN controller platform designed for scalability, high availability, and performance. ONOS offers support for carrier-grade SDN deployments and integration with cloud platforms.
- **Cisco Application Policy Infrastructure Controller (APIC)**, A commercial SDN controller solution from Cisco Systems, is designed for managing Cisco's Application Centric Infrastructure (ACI) and other SDN-enabled networks.

3.4 Intrusion Detection Systems (IDS)-An overview

An IDS serves as a security instrument designed to monitor network or system operations for signs of malicious behavior or policy violations. Observe network traffic or evaluate data produced by a host to identify any unauthorized intrusion attempts that breach security protocols. Such systems are the primary line of defense against cybersecurity threats prevalent in various industries providing real-time alerts upon detecting an attack. As stated above, the core principles of IDS are to monitor, detect, and respond to any unauthorized activities [42].

3.4.1 IDS-based Implementation Types

IDS fall into two categories based on how they are implemented: host/network-based IDS (HIDS), and (NIDS). Intrusion is the illegal exploitation, misuse, or manipulation of hardware and software systems, perpetrated by internal and external actors. The main goal of IDS is to protect networks or computer systems by identifying and thwarting malicious attacks targeting network systems or individual host devices.

This section will explore the various types of IDS and their respective detection methodologies. Specifically, it will explore the placement-based classification of IDS, distinguishing between HIDS and NIDS.

3.4.1.1 Host-based IDS

Typically, a HIDS acts directly on a specific computer or device, focusing primarily on internal monitoring. At the same time, its core purpose is internal surveillance; various adaptations of HIDS have emerged, some of which are capable of monitoring network activities as well. HIDS assesses the integrity of the system by scrutinizing all communication flows and promptly notifying administrators of any anomalies. This includes identifying unauthorized programs attempting to access system resources or detecting harmful alterations to the registry by a program [43].

3.4.1.2 Network-based IDS

NIDS was specifically developed to analyze the flow of data across network traffic, providing crucial protection for the infrastructure against attacks that involve the network. It detects illegal access attempts and analyzes traffic over the LAN, spanning several hosts [44]. NIDS inspects incoming and outgoing packets using detection methods to look for abnormal patterns. When an anomaly is detected, an alarm is sent to the administrator. This technology encompasses three distinct network topologies: connection directly to a port-spanning switch, use of a network tap, and inline connection. IDS technologies offer a combination of traditional IT security measures along with customized approaches that meet the unique characteristics of Industrial Control Systems (ICS) [45] .
Table 3.1 provides a concise comparison of NIDS and HIDS. This comparison highlights the main advantages and disadvantages of each type of IDS [46].

3.4.1.3 NIDS-based ML Integration

In the realm of cybersecurity, staying ahead of ever-evolving threats requires proactive measures. This subsection delves into NIDS empowered with ML for anomaly detection. The structure of a basic NIDS comprises these elements [47]:

- **Data Acquisition Sensors**: These sensors are strategically placed to collect raw data from their assigned locations. They act as the initial gathering points for the information.
- **Detection Engine**: This module compares the collected data to the provided set of rules. Once the IDS identifies a divergence from the typical status, it raises an alert.
- **Storage Module**: This component includes the IDS rule sets that the detector utilizes during the data comparison process.

Table 3.1. *Advantages and Disadvantages of NIDS and HIDS*

Type	Advantages	Disadvantages
NIDS	• Monitors entire network traffic. • Detects attacks before they reach individual hosts. • Provides centralized monitoring. • Scalable for large networks.	• Vulnerable to encrypted traffic. • May generate false positives due to network noise. • Can be bypassed by attacks on the host itself. • Limited visibility to other devices on the network.
HIDS	• Monitors individual host activities. • Provides detailed visibility into host-specific events. • Effective against insider threats. • Can detect attacks missed by NIDS.	• Relies on host resources, affecting performance. • May not detect attacks that do not affect the host. • Requires software installation on each host. • Limited to monitoring hosts within the network.

- **Response**: Upon triggering an alarm, the IDS replies with a predefined action. Depending on the level of warning, this action may involve executing a predetermined response, for example, discarding harmful packets. Alternatively, it may involve a passive reaction, like recording activities and leaving the decision to the operators.

A typical IDS architecture is depicted in Figure 3.3.

3.4.2 IDS-based Detection Method

Developers seek real-time network monitoring solutions to ensure prompt responses to intrusions or attacks. Both NIDS and HIDS can occur in three different ways

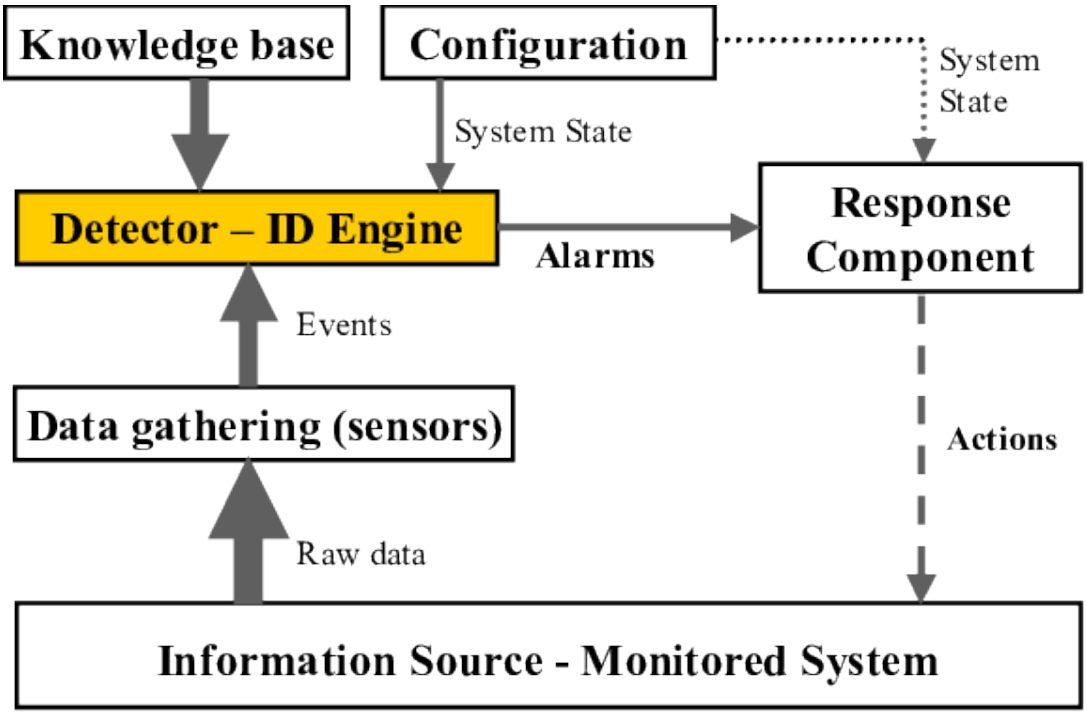


Figure 3.3. *Architecture of the Intrusion Detection System [48].*

3.4.2.1 Signature-based Detection

Signature-based IDS rely on a database of attack signatures. They identify incoming attacks by comparing network traffic patterns to these known signatures, essentially checking for a match. Signature-based methods offer a straightforward and efficient approach to network security. They function by monitoring and comparing network packets or connections against a predefined set of patterns, known as signatures [49]. This approach boasts a lower false positive rate, meaning that it rarely generates incorrect alarms. However, it has limitations. Comparing packets to large signature sets can be time-consuming, and these methods lack predictive capabilities. They can only detect anomalies predefined within the signatures, requiring frequent updates by administrators to stay effective against evolving threats.

3.4.2.2 Anomaly-based detection

Anomaly-based IDS take a different approach compared to signature-based methods. Instead of relying on predefined attack patterns, they learn the normal behavior of a

system, network, or program by creating a baseline profile [49]. Any behavior that differs considerably from the predefined baseline is marked as a possible intrusion. . This method offers several advantages:

- *Insider Threat Detection:* Anomaly-based systems can detect suspicious activity by authorized users. For example, if a user's actions deviate significantly from their typical profile (e.g., accessing unauthorized data), an alarm can be triggered, potentially uncovering insider threats.
- *Difficulty of Circumvention:* Since anomaly detection relies on customized user profiles, attackers cannot easily predict what actions will go unnoticed. This makes it harder for them to bypass the system.
- *Novel Attack Detection:* Anomaly detection can identify previously unknown attacks, as it focuses on deviations from normal behavior instead of matching specific patterns.

3.4.2.3 Hybrid-based Detection

Hybrid detection, also called specification-based detection, combines ML techniques to integrate both anomaly-based and signature-based detection methods. This approach allows for the identification of both known and novel attack patterns [50]. It is anticipated that this form of detection will soon gain prominence in the Industrial Control Systems market. When these methods are combined, the previously mentioned detection gaps, including false negatives and false positives, can be mitigated. Surveys show that the effective use of ML algorithms greatly improves the detection of cyber-attacks.

3.5 Machine/Deep Learning for Enhancing Network Security

3.5.1 Machine learning (ML)

It constitutes a branch of Artificial Intelligence (AI) committed to crafting methods capable of assimilating data and subsequently generating predictions or decisions derived from it. Inside the field of network security, ML algorithms play a critical role in boosting defenses by launching vital computations and discerning elements such as packet types within network traffic [51].

3.5.1.1 Approaches

ML algorithms can be broadly categorized into three types:

1. **Supervised algorithms,** necessitate completely labeled class data, notably for NIDS. This strategy includes splitting a dataset into training and testing subsets, with the goal of training algorithms using data with labels to build a model. The training process of supervised algorithms emphasizes their ability to determine appropriate features and classes. The data are records from the network or host sources, matched with output values labeled as "attack" or "normal" data [52]. Feature selection is employed to eliminate unnecessary features before the algorithms undergo testing to predict anomalies and relevant classes from new incoming data. The classification category encompasses commonly used supervised ML algorithms, including Support Vector Machine (SVM), Decision Tree (DT), Artificial Neural Networks (ANN), K-Nearest Neighbor (KNN), Random Forest (RF), and Naïve Bayes. Although supervised learning excels in detecting known attacks, it may not be as effective for identifying new attacks.
2. **Unsupervised algorithms,** leverage clustering algorithms to construct models from unlabeled data, enabling the differentiation of malicious inputs within

network traffic or host logs [53]. Unsupervised methods for data analysis operate without prior knowledge, utilizing statistical properties to analyze data characteristics. They bypass time-consuming training stages and focus on feature selection, utilizing clustering techniques to divide input data into clusters with optimal relevance and minimal redundancy. By establishing a threshold during analysis, these methods prioritize regular instances over anomalies within a dataset. Following clustering, each cluster is assigned a unique score, with clusters exceeding the threshold flagged as potentially malicious [54]. This approach efficiently distinguishes regular traffic, which forms larger clusters, from attacks, which manifest in smaller clusters. Unsupervised learning methods excel at detecting unknown attacks, such as zero-day attacks, without relying on labeled data. However, they may suffer from a high false-positive rate. Authors in [55] highlight the most prevalent algorithms in this domain, including fuzzy clustering, K-means clustering, and hierarchical clustering.

3. **Semi-supervised algorithms,** also called hybrid algorithms, can effectively handle partially labeled datasets, typically comprising a short subset of labeled data amidst a vast pool of unlabeled data. These algorithms often integrate elements of both unsupervised and supervised learning methodologies. The primary goal of hybrid ML algorithms is to minimize the occurrence of false negative and positive alarms. Hybrid techniques employed in IDS integrate supervised and unsupervised ML methods to enhance the system's effectiveness [56].
4. **Reinforcement algorithms** Reinforcement learning stands apart from other categories due to its distinct approach. In reinforcement learning, an agent learns through interactions with its environment. It observes the environment, selects actions, and receives rewards in response. The objective of the agent is to devise the optimal strategy to maximize cumulative rewards over time [57].

3.5.2 Deep learning (DL)

DL is a subset of ML that has acquired substantial attention and popularity because of its remarkable capacity to handle data with complex representations. DL models are fundamentally based on artificial neural networks influenced by the architecture and activity of the human brain. Such neurons include numerous layers, which allows them to acquire data with structured presentations. Each layer extracts features from incoming data and transmits them to the next layer, enabling the model to capture intricate patterns and relationships[58].

Among the primary benefits of DL is its capacity to automatically identify meaningful features from the data itself, eliminating the need for manual feature engineering. This renders DL ideal for applications like image and audio recognition, natural language processing, anomaly detection, and pattern identification in huge datasets.

3.5.2.1 Components and Architecture of DL

DL models are based on several fundamental components to achieve their remarkable capabilities. Here is a breakdown of these key elements [58]:

1. **Input Layer:** This is the entry point, receiving raw data (images, text, audio) and feeding it forward for processing.
2. **Hidden Layers:** These are the workhorses, positioned between the input and output layers. Each layer contains connected processing units called neurons that perform calculations on the data they receive.
3. **Activation Functions:** These functions add a layer of complexity by transforming the neuron outputs. Common examples include sigmoid, tanh, and ReLU functions. This non-linearity enables the model to identify complicated patterns and correlations within the data.
4. **Weights and Biases:** In a neural network, each connectivity between neurons is assigned a weight and bias. The weights control how much one neuron

affects another, while biases adjust the overall output. These parameters are continuously fine-tuned during training to improve the efficiency of the model.

5. **Output Layer:** The last layer represents the final model, which delivers predictions or classifications based on the processed data. The output layer's structure varies depending on the purpose, such as generating numerical values (regression) or assigning categories (classification).

A typical architecture of an ANN is depicted in Figure 3.4.

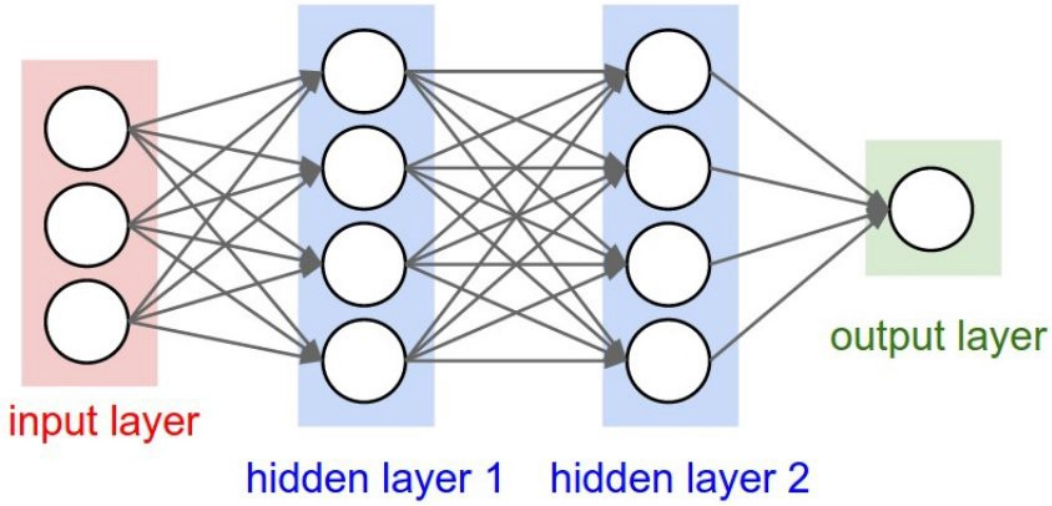


Figure 3.4. *Architecture of Artificial Neural Network [59].*

Various DL architectures have been developed [60], including:

- Long Short-Term Memory Networks (LSTM)
- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks (RNNs)
- Generative Adversarial Networks (GAN)
- Deep Reinforcement Learning
- Feedforward Neural Networks (FNN)

3.5.2.2 Neural Network Theory Content

This section presents the fundamental concepts for understanding how neural networks are trained and optimized to gain knowledge from data and make predictions [61].

1. **Feed Forward Propagation:**
    - The operation refers to the flow of data through a neural network. Information starts at the input layer, gets transformed as it passes through the hidden layers, and eventually reaches the output layer.
    - In a neural network, every neuron in a given layer gets input from the preceding layer, adjusts these inputs using weights and biases, and then processes the result with an activation function to determine its output.
    - This sequence is repeated for each layer, culminating in the final output layer, which generates a classification.
2. **Activation Functions:**
    - It adds non-linearity transformation to neural networks, allowing them to capture complex patterns in the data effectively.
    - Some widely used activation functions include the sigmoid function, tanh (hyperbolic tangent) function, ReLU (Rectified Linear Unit), and softmax function.
    - These functions introduce non-linearities that enable neural networks to approximate arbitrary functions and learn complex associations in the data.
3. **Loss Functions:**
    - Loss functions measure the discrepancy between a neural network's predicted output and the desired target output.
    - The selection of a loss function is influenced by the nature of the task (e.g., regression, classification) and the desired properties of the model.

- Some common examples include Mean Squared Error (MSE) for regression, Cross-Entropy Loss for multi-class classification, and Binary Cross-Entropy Loss for binary classification.

4. **Gradient Descent:**

   - An optimization technique is employed to adjust a neural network's parameters (weights and biases) to minimize the loss function.
   - It works by iteratively adjusting the parameters in the direction of the steepest descent of the loss function gradient.
   - The learning rate, a critical hyperparameter in gradient descent algorithms, determines the magnitude of these parameter updates.

5. **Backpropagation:**

   - Backpropagation is an algorithm utilized to calculate the gradients of the loss function relating to the parameters of a neural network.
   - It utilizes the chain rule of calculus to efficiently propagate errors backward through the network, allowing for efficient computation of parameter updates.
   - By iteratively applying backpropagation and gradient descent, neural networks are trained to decrease the loss function and increase predicting accuracy.

3.6 Self Sovereign Identity System-based Blockchain

An SSI system based on blockchain technology is a decentralized approach to managing digital identities. It gives users more control over their personal identity information, they can manipulate, share, and authenticate their identity without relying on centralized authorities. Furthermore, the SSI framework involves three primary roles: issuer, holder, and verifier. As depicted in Figure 3.5, the issuer

creates and provides credentials to the holder. The holder then securely stores these credentials and shares them with a verifier, who evaluates and verifies the credentials provided by the holder [62].

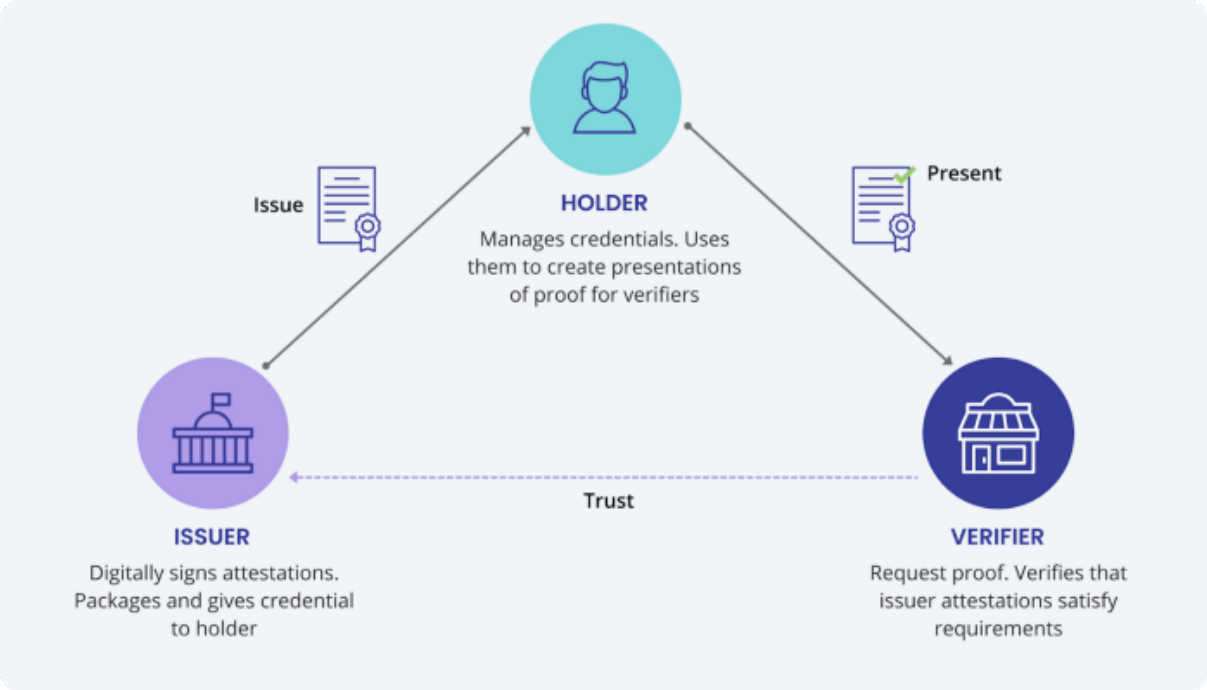


Figure 3.5. *Self-Sovereign Identity Ecosystem [63].*

3.6.1 Decentralized Identity Management

In the SSI system, individuals maintain management over their digital identities through the use of decentralized identifiers and verifiable credentials.

3.6.1.1 Decentralized Identifiers (DIDs)

DIDs are a foundational component of SSI. They are unique and generated on a blockchain network, enabling individuals to demonstrate privilege and control of their identities without the need for intermediaries. The W3C (World Wide Web Consortium) Credentials Community Group maintains the DID specification [64].

3.6.1.2 Verifiable Credentials (VCs)

VCs are digital certificates or assertions issued by trusted entities (such as governments, educational institutions, or employers) attesting to specific attributes or qualifications of an individual (e.g., age, education, employment history). These credentials are cryptographically signed and can be verified by anyone, permitting individuals to present proof of their identity or qualifications without revealing unnecessary personal information [65].

3.6.2 Blockchain Technology

Blockchain acts as the foundational infrastructure to store and manage identity-related data securely and transparently. Its decentralized nature assures that no single party has authority over the identification data, thereby enhancing security and privacy [66].

3.6.3 DID Wallets

Users interact with their digital identities through DID wallets, which are software applications that allow the product, storage, and control of DIDs and verifiable credentials. These wallets provide individuals with a user-friendly interface for managing their identity information and controlling access to it [67].

3.6.4 Selective Disclosure and Privacy

SSI systems prioritize user privacy and allow individuals to disclose only the specific information needed for a particular transaction or engagement. This selective distribution lowers the danger of identity theft and illegal access to private information [64].

3.6.5 Interoperability and Standards

To ensure widespread adoption and compatibility across different platforms and applications, SSI systems adhere to open standards and interoperability protocols, enabling seamless integration with existing identity systems and services [64].

3.7 Role of Key Technologies in Enhancing Security and Privacy

This section outlines the importance and specific roles of various advanced technologies in improving security and privacy within network systems, as demonstrated in the following table 3.2.

3.8 Related Works

Numerous researchers have presented various strategies aimed at safeguarding security and preserving data privacy within fog computing environments for IoT systems. Therefore, this section highlights a concise overview of the different security and privacy methods introduced.

3.8.1 SDN for DDoS Detection

Renyi Entropy-based Detection [68]: This method uses Renyi entropy to analyze network traffic fluctuations and identify DDoS attacks. Using a packet drop mitigation framework, it can be used in an SDN environment. However, a major limitation is the need for a precisely defined detection threshold, which can be challenging.

Multivariate Correlation Analysis [69]: This approach focuses on detecting SYN flood attacks through covariance analysis. Although it demonstrates high accuracy in

Table 3.2. *The Importance and Role of Various Technologies in Improving Security and Privacy*

Technology	Importance	Role in Security and Privacy
SDN	• Centralized network control • Enhanced flexibility and scalability	• Dynamic network segmentation • Fine-grained access control • Efficient threat detection • Rapid response to incidents
IDS	• Monitors and analyzes traffic • Detects suspicious activities	• Identifies security breaches • Real-time alerts • Proactive defense
Machine/Deep Learning	• Detects and predicts threats • Advanced Analytics	• Identifies patterns • Detects anomalies • Predicts emerging threats • Improves response times
Blockchain-based SSI	• Decentralized identity management • Enhances privacy and security	• Ensures data integrity • Secure identity verification • Selective disclosure

various network scenarios, the real-time processing demands are significant, hindering its practicality in real-time applications.

Fog-based Detection [70]: This framework utilizes fog computing to identify DDoS attacks before they reach the cloud. It uses intermediary fog servers and an effective resource allocation mechanism to handle cloud requests.

Multi-Objective Task Scheduling for Secure SDN-IoT-Fog Networks [71] Javanmardi et al. propose FUPE offers a security-focused approach to task scheduling in SDN-managed networks integrating IoT and fog computing. It leverages fuzzy logic and multi-objective particle swarm optimization (PSO) to assign tasks to fog nodes, aiming for a balance between security and efficiency. However, this approach has limitations. first, managing and interpreting the extensive rules governing FUPE's fuzzy logic system can be challenging and complex, especially for maintenance and validation. second, multi-objective optimization using PSO requires careful parameter tuning and can be computationally expensive, particularly in large networks [72].

3.8.2 SDN-based ML/DL for DDoS detection and mitigation

K-Nearest Neighbors (KNN) for Source Tracing [73]: This approach utilizes KNN-based ML to trace back the IP sources of TCP-SYN flood attacks in SDN networks. The test results showed promising identification and blocking rates for attack flows.

Source-based Mitigation with SDN and ML Classifiers [74]: This method proposes a source-based mitigation approach using ML algorithms (SVM, KNN, and Naive Bayes) within the SDN controller. However, the authors highlight the limitations of classical ML techniques in handling large data volumes and real-time applications.

Multilayer Perceptron for Low-Rate DDoS Detection [75]: This research investigates various ML algorithms for low-rate DDoS attack detection. The study identifies the Multilayer Perceptron (MLP) as the best-performed, reaching an elevated detection rate. However, classical ML techniques struggle to manage large volumes of data. Moreover, their practical application is constrained by network attack concerns in real-world scenarios. In addition, these methods require extensive learning time, rendering them unsuitable for real-time applications.

Random Neural Networks for SYN Flood Detection [76]: Both studies leverage random neural networks to classify and differentiate normal traffic from SYN flood

attacks. These approaches are suitable for resource-constrained devices like edge servers and gateways.

ANFIS-based IDS for 5G Networks [77]: This approach presents an IDS using a fuzzy inference system (ANFIS) to identify security threats on relay nodes in 5G networks. The model was tested and trained using the KDD Cup 99 datasets.

Ensemble of Neuro-Fuzzy Classifiers for DDoS Detection [78]: The authors propose a novel ensemble of fuzzy classifiers to detect DDoS attacks. The model was tested using the NSL-KDD dataset.

However, the NSL-KDD or KDD Cup 99 datasets were deemed inadequate to address the new requirement regarding DDoS attacks. This is because these datasets consist of packet traces rather than flows. As a result, implementing DDoS detection methods can pose computational challenges as the network increases in size.

3.8.3 Intrusion Detection System

Mitigating CoAP Protocol Risks[79]: This study focuses on securing the Constrained Application Protocol (CoAP), a common protocol in IoT. They propose a rule-based modular IDS framework. However, their findings favor a hybrid approach that combined rule-based and anomaly-based detection, which successfully mitigated routing attacks in CoAP networks.

Behavioral Anomaly Detection in Smart Homes [80]: This research explores the use of behavioral modeling for IDS in smart homes. Their focus lies on detecting anomalies related to nonplaying characters (NPCs) within the smart home environment. Although implementation details are not provided, the authors claim that their approach offers cost-effective and verifiable autonomous intrusion monitoring.

Securing 6LoWPAN Networks [81]: This study investigates securing 6LoWPAN (IPv6 over Low-power Wireless Personal Area Networks) networks, commonly used in IoT deployments. Their approach combines cryptography/encryption with IDS to

safeguard IoT devices. However, the limited processing power of many IoT devices can make heavy encryption challenging.

Intrusion Detection for Wireless Sensor Networks [82]: WSNs are another critical area for IoT security due to their broad applications and vulnerability to attacks. This research emphasizes the need for robust IDS in WSNs. The authors propose a rule-based IDS specifically designed for the resource constraints of WSNs. Their system successfully detects various attacks, including message delays, data alteration, and black-hole attacks.

3.8.4 Intrusion Detection System based ML/DL

Hybrid Intrusion Detection for Generic Networks [83]: This study proposed a hybrid IDS for general networks. It combined the signature-based Snort IDS with anomaly detection techniques using the Naive Bayes algorithm. The researchers evaluated various data mining techniques and found Naive Bayes effective in anomaly detection using the KDD CUP 20 dataset and the WEKA framework.

Intrusion Detection for IoT-based Disaster Warning Systems [84]: The same researchers investigated IDS specifically for IoT networks used in coal mine disaster warning systems. Their approach employed the immunity algorithm for feature selection and the Black Propagation Neural Network (BPNN) for classification. However, While achieving a high success rate, this method had limitations. The training process was lengthy (nine weeks), and BPNN was computationally expensive, making it unsuitable for resource-constrained devices like Raspberry Pi.

Analysis of IDSs for IoT [85]: This research presented a broader analysis of various IDS approaches for securing IoT networks. They categorized IDS systems into four main types: anomaly, signature, specification, and hybrid-based systems. While their study provided valuable insights into these categories, it lacked specific details regarding the anomaly detection system that utilized DL techniques.

3.8.5 SSI-based Blockchain techniques

SSIBAC for DID Management: Authors in [86] proposed the SSIBAC framework . It presents a decentralized method of identity management for access control across organizations. This system leverages blockchain technology to enable decentralized authentication while utilizing traditional access control architecture for centralized authorization.

OAuth with DID and VCs: Another approach by authors in [87] utilizes OAuth to delegate authorization server control over privilege management policies. This allows systems with restricted resources, like those in the IoT, to benefit from DID and VC technology. However, a thorough threat analysis is crucial before adopting DIDs for specific use cases, such as IoMT devices, to ensure their suitability.

Decentralized Access Control with DIDs: Authors in [88] propose a DID-based access control system that reduces the necessity for a centralized authority to grant access privileges. While offering decentralization, this approach might be less effective against hacking attempts compared to traditional centralized systems.

DID-based Attribute-Based Access Control (ABAC): a DID-based ABAC system is presented in [89] to address privacy concerns associated with traditional ABAC. This system is implemented on a power transaction platform.

3.9 Summary

This chapter has examined security and privacy techniques in fog computing within the IoT domain. In subsequent chapters, we will present our two contributions that address security and privacy concerns in fog computing for IoT applications.

Part II

CONTRIBUTIONS

Chapter

4

Toward a real-time TCP SYN Flood DDoS Mitigation using an ANFIS Classifier and SDN Assistance in Fog Computing

Contents

4.1 Introduction

This chapter introduces our initial contribution to enhancing fog computing security for the IoT. To confront the challenges posed by various attacks against fog servers, including Distributed Denial of Service (DDoS) attacks that significantly impact the reliability and availability of fog services, we propose a real-time TCP SYN Flood DDoS mitigation method.

This approach leverages the benefits of Software Defined Networking (SDN) and the Adaptive Neuro-Fuzzy Inference System (ANFIS) to bolster network security within fog computing. The primary advantage of this proposal is its ability to detect and effectively counteract SYN Flood DDoS attacks in real-time. Hence, the objectives of this chapter are as follows:

1. We introduce a novel model FASA, a **F**og computing-based approach to mitigate SYN Flood DDoS attacks using an **A**daptive Neuro-Fuzzy Inference System (ANFIS) with **S**oftware Defined Networking (SDN) **A**ssistance.
2. We implement the ANFIS model to autonomously train fog servers and differentiate between normal and malicious packets.
3. The ANFIS model is integrated into the SDN controller and deployed on fog servers using a dataset from the SDN environment. Its primary objective is to allow access to benign packets while rejecting malicious ones, ensuring the creation of a secure and reliable SDN controller that maintains fog service availability.
4. Our proposed evaluation method utilizes the recently released dataset CIC-DDoS2019 and the SDN dataset. To enhance overall performance, it undergoes experimental analysis concerning data availability and algorithm efficiency.

The structure of this chapter is as follows: Section 2 offers background information and formally defines the FASA network architecture. In Section 3, we present our

proposed models, followed by the proposed models in Section 4. Experimentation and evaluation results are reported in Section 5. Finally, we conclude the chapter.

4.1.1 Motivation

The explosion of Internet of Things (IoT) devices has dramatically impacted our daily lives, generating a vast amount of data that demands real-time processing. While cloud computing offers a centralized solution, its high latency due to the mainframe architecture hinders its effectiveness.

Fog computing emerges as a promising alternative, acting as a distributed paradigm that brings cloud services closer to the network edge. This approach offers low latency and high bandwidth, ideal for supporting various IoT applications. However, fog computing introduces new security challenges. DDoS attacks, for instance, can severely affect the reliability and availability of fog services.

To address these challenges, robust security measures are crucial. Implementing classification and mitigation techniques can significantly improve security. Ultimately, ensuring strong privacy and security is essential for fog computing to fully support the ever-growing world of IoT applications.

4.2 Background knowledge

This section provides the necessary context for our proposed model. Initially, we offer an overview of DDoS attacks and various detection methods. Next, we introduce the Adaptive Network-based Fuzzy Inference System (ANFIS) detection algorithm. Finally, we discuss Software Defined Networking (SDN) technology.

4.2.1 DDoS Attacks and Fog Computing

The DDoS attack represents an advanced form of DoS attack, distinguished by its potential to operate in a "distributed" manner. Unlike other attacks, its primary objective is to cause harm to a target for personal motives, financial gain, or to gain notoriety. This attack, which prioritizes availability, seeks to render the victim system inaccessible to authorized users [90]. It involves the utilization of a vast number of compromised and widely dispersed devices, referred to as bots or zombie devices, infected with malicious malware or under the control of an attacker [91]. Consequently, the attacker centrally manages and orchestrates these machines to execute an assault on the target machine [92].

4.2.1.1 Types of DDoS Attacks on Fog Computing

Within the realm of fog computing, DDoS attacks come in many forms, all targeting the functionality and accessibility of network services[93] (see Figure 4.1 for a visual representation).

a. **Application-Bug Level DDoS**

   here, DDoS attacks, such as HTTP POST and HTTP PRAGMA, exhaust the application system, leading to its failure or temporary shutdown.

b. **Infrastructural Level DDoS**

   DDoS attacks here aim to deplete network bandwidth, buffers, CPU, and storage, thereby obstructing legitimate users from accessing them. Hence, the sole requirement for this attack is the victim's IP address. It can be divided into two types: direct and reflector attacks.

   - *Direct Attack*

     This attack leverages compromised devices, or "bots," to flood the target with malicious requests. This overwhelms the system's resources, bandwidth, and services, ultimately blocking access for legitimate users.

The attack can target different network layers (network layer) or specific applications (application layer).

- Network Layer DDoS: This type of attack inundates the network infrastructure with a massive amount of data packets using various protocols. Common culprits include:

  TCP SYN Flood: Exploits the three-way handshake process to tie up server resources without completing connections.

  UDP Flood: Sends a constant stream of User Datagram Protocol packets, overwhelming the target with raw data.

  ICMP Flood: Floods the network with ICMP messages (like "ping"), consuming bandwidth and hindering legitimate traffic.

- Application Layer DDoS: This attack targets specific applications or services running on a system. A popular method involves sending a massive amount of HTTP requests, aiming to exhaust the server's resources and crash the application. The difficulty in detecting these attacks, often mimicking legitimate user traffic, makes them a significant security concern.

- *Reflector Attack*

  In this attack, the attacker masks their identity by spoofing the IP address. They then target a large number of vulnerable "reflector" servers, tricking them into sending amplified responses back to the intended victim. This creates a flood of unwanted traffic that overwhelms the target's resources.

4.2.1.2 DDoS Defense Mechanisms

This section explores different defense mechanisms employed for detecting and mitigating DDoS attacks to enhance the security of fog computing [93]. Fog computing provides computing capabilities through fog nodes located closer to the edge, which imposes a significant burden on network management. To tackle this challenge,

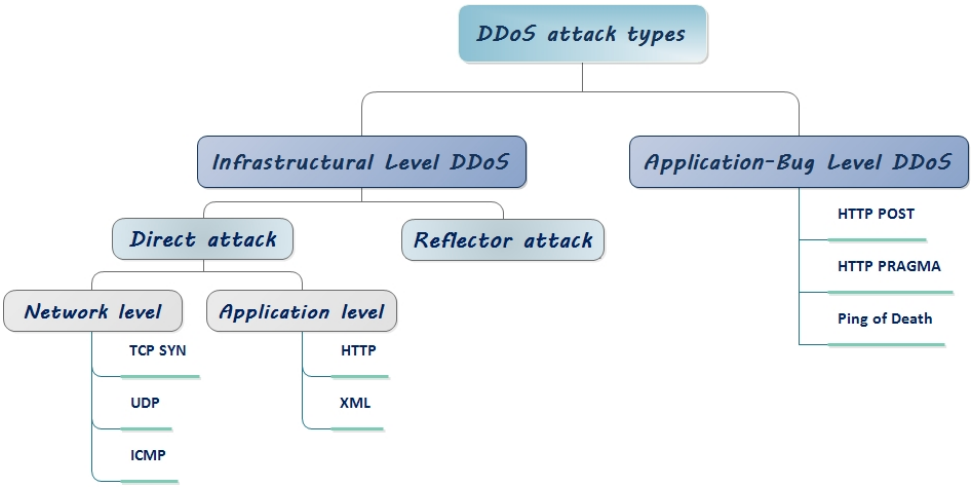


Figure 4.1. *Types of DDoS Attack in Fog Computing.*

the implementation of SDN technology can ensure the security of fog computing in various aspects:

- *Monitoring the network:* Continuous monitoring of the network allows for the timely detection and rejection of any suspicious data attempting to disrupt services. By conducting this monitoring at fog nodes, legitimate users can access services without difficulty.
- *Priority-based and isolated traffic:* This involves prioritizing legitimate network traffic over illegitimate traffic, utilizing shared resources like CPU or I/O. SDN can effectively reject harmful traffic by segregating it through VLAN ID/tag.
- *Access control mechanism for resources in the network:* Implementing an efficient access control system is essential for preventing DDoS attacks.
- *Shared network:* The shared network poses a significant security risk since it is accessible to anyone, emphasizing the importance of robust security measures.

4.2.1.3 TCP SYN Flood Attack

The SYN flood attack, a well-known DoS attack targeting TCP servers since 1996 [94], exploits a vulnerability in the three-way handshake process that initiates TCP connections [95]. By leveraging this weakness in the protocol, attackers flood the server

with a surge of incomplete connection requests (SYN packets). This overwhelms the server's resources, preventing it from responding to legitimate requests and effectively blocking service for normal users.

While modern operating systems and equipment often incorporate countermeasures against SYN flood attacks, they remain a significant threat.

4.2.2 The Adaptive Network-based Fuzzy Inference System (ANFIS) detection algorithm

ANFIS offers a unique approach to decision-making by blending the strengths of fuzzy logic and neural networks [96]. Fuzzy systems excel at handling imprecise data, but their rule development can be challenging. This is where neural networks come in ANFIS leverages neural learning to automatically adjust fuzzy system parameters, simplifying the development process [97]. This combination makes ANFIS a powerful tool for various applications, including control systems, data analysis, and decision support [98]. ANFIS leverages two sets of parameters to connect its fuzzy rules: premise parameters define the conditions for each rule, and consequent parameters determine the output. This structure allows ANFIS to learn and adapt over time, unlike traditional fuzzy systems with manually defined rules. The core of ANFIS lies in its five-layered architecture as shown in Figure 4.2. Square nodes within these layers represent the adjustable parameters that ANFIS utilizes for learning. Circular nodes, on the other hand, perform specific calculations on the data flowing through the network. In our case, the ANFIS model employs two input variables (represented by y, treated as non-linear) and generates a single output (f). Notably, each input is described by two linguistic terms, enabling ANFIS to handle complex relationships within the data: A_1 and A_2 for the variable x, and B_1 and B_2 for the variable y, respectively.

The structure of Sugeno fuzzy models is based on *IF-THEN* rules, illustrated as follow: [98]:

- **Rule 1:** If x is $A_1 \wedge y$ is B_1, then $f_1 = p_1x + q_1y + r_1$
- **Rule 2:** If x is $A_2 \wedge y$ is B_2, then $f_2 = p_2x + q_2y + r_2$

Where p_i, q_i, and r_i i=1,2, represent the adjustable parameters of the conclusion part (trained during the process).

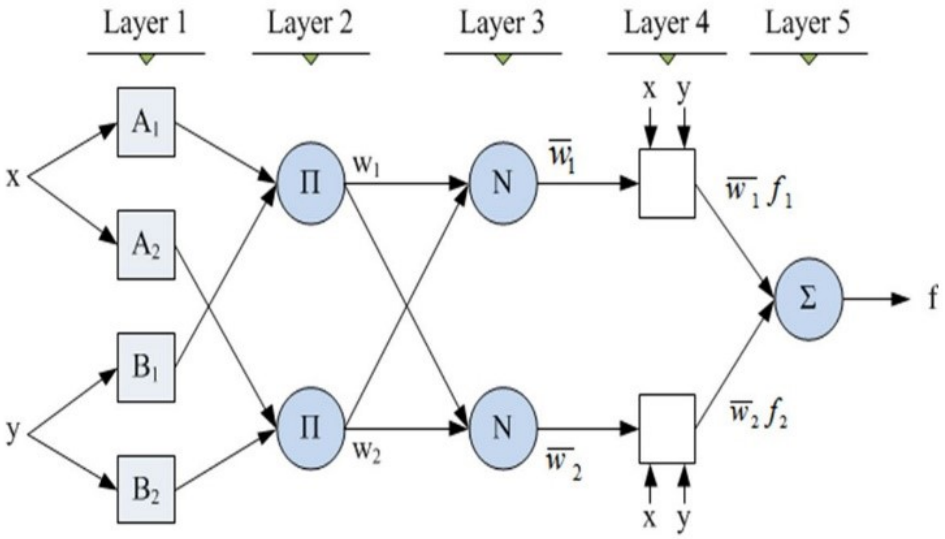


Figure 4.2. *ANFIS Architecture Layers.*

Layer 1: O1, represents the membership function μ of a fuzzy set $A_i (or B_i)$.

$$O_{1,i} = \mu_{A_i}(x) \qquad i = 1, 2$$

$$O_{1,i} = \mu_{B_{i-2}}(y) \qquad i = 3, 4$$

Gaussian membership functions are widely preferred for defining fuzzy sets due to their smoothness and straightforward mathematical representation. These bell-shaped curves offer a key advantage: they are continuous (smooth) and have non-zero values at all points. This characteristic allows for a more adequate representation of fuzzy concepts compared to functions with abrupt transitions.

In this study, we leverage the Gaussian membership function, commonly used to model uncertainty in real-world measurements. As represented by Equation (4.1), the function is defined by two key parameters:

- **c (mean):** This parameter determines the center of the Gaussian curve, signifying the point where the membership degree is maximum.
- σ **(standard deviation):** This parameter controls the width of the curve, influencing the spread of membership values.

A larger σ results in a wider curve, indicating a higher degree of uncertainty. Here, it's important to note that c and σ are considered premise parameters, meaning they are non-linear and contribute to the conditions (premises) of the fuzzy rules.

$$\mu_A(x) = ae^{\frac{-(x-c)^2}{(2\sigma)^2}} \tag{4.1}$$

Layer 2: often referred to as the rule layer, plays a crucial role in combining the membership degrees generated in Layer 1. Each node (denoted by w) in this layer calculates the product $\prod$ of all the incoming signals (membership degrees) from Layer 1. This product essentially reflects the degree to which all the conditions (premises) of a fuzzy rule are satisfied by the input data, it is determined using the equation(4.2).

$$O_(2,i) = w_i = \mu_A(x) \cdot \mu_B(x) \qquad i = 1,2 \tag{4.2}$$

The output of each node displays the firing strength of a rule. The function employed by nodes in this layer can be any other fuzzy AND T-norm operator, such as min. **Layer 3:** In the normalization layer, each node corresponds to a specific rule and calculates the proportion of the firing strength of that rule to the total firing strength of all rules. This process is illustrated in equation (4.3).

$$O_{3,i} = \bar{w_i} = \frac{w_i}{w_1 + w_2} \tag{4.3}$$

The results produced by this layer are termed normalized firing strengths. **Layer 4:** In the defuzzification layer, parameters are denoted as consequent parameters. Each node possesses a function where $\bar{w_i}$ represents a normalized firing strength from layer 3, and p_i,q_i,r_i constitute the set of linear node parameters, defined as consequent parameters

of this node. Additionally, f_i signifies the output of the rule, as depicted in equation (4.4).:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i) \tag{4.4}$$

Layer 5: In this layer, a single node sums up all incoming signals to calculate the overall output, as illustrated in equation (5).:

$$O_{5,i} = overalloutput = \sum_{i=0}^{n} \bar{w}_i f_i = \frac{\sum w_i f_i}{\sum w_i} \tag{4.5}$$

Unlike traditional fuzzy systems with predefined rules, ANFIS can learn and adapt to new data. This adaptation is achieved through a hybrid learning algorithm that combines two powerful techniques: gradient descent and least squares [99].

Forward Pass (Optimizing Consequent Parameters):

- During this pass, the least squares method is used to adjust the parameters that define the consequent part of the fuzzy rules (often referred to as consequent parameters).
- Essentially, the system processes the input data through the first three layers, and the least squares method determines the optimal values for these consequent parameters to minimize the error between the desired output and the actual output generated so far.

Backward Pass (Optimizing Premise Parameters):

- In this stage, the ADAM optimization algorithm [100] comes into play. This method is used to update the parameters that define the conditions (premises) of the fuzzy rules (often called premise parameters).
- Error signals are propagated backward through the network, and the ADAM algorithm adjusts the premise parameters to minimize the overall error.

This hybrid approach offers several advantages:

- **Faster Convergence:** Compared to the traditional backpropagation method, the hybrid approach reduces the search space for optimal parameters, leading to faster convergence during training [101].
- **Effectiveness:** Studies have shown that this hybrid learning algorithm is highly effective in training ANFIS models [102].

4.2.3 Software-Defined Networking (SDN)

SDN, a network paradigm, empowers users to directly manage network resources through orchestration, control, and utilization of software applications [103]. Additionally, SDN divides the control and data planes, commonly enhancing network efficiency. While the data plane forwards packets between locations, the control plane determines their propagation through the network. Further details on SDN are discussed in Chapter 3, Section 3.3.

4.2.4 System Model

The objective of this research is to construct a distributed FASA framework for alleviating SYN flood attacks within the network environment by identifying and intercepting attacks near their sources. Employing fog computing alongside SDN facilitates swifter and more precise attack detection utilizing the ANFIS model. By deploying compute power near the operational process and distributing the workload within the system via a FASA mitigation scheme, fog computing is well-suited for countering SYN flood attacks. In this section, we begin by delineating SYN flood DDoS attacks in fog computing, followed by an exploration of the FASA network architecture.

4.2.5 SYN Flood DDoS Attack

As depicted in Figure 4.3, during a standard TCP three-way handshake initiation, the End User (EU) sends a SYN packet to the fog server. Subsequently, the fog server responds with an SYN/ACK packet. The EU is then expected to transmit an ACK packet back to the fog server. Once all these steps are successfully executed, the connection is established [104]. However, a significant drawback of TCP connections is the inability to maintain half-open connections. The fog server finds itself in a half-open connection state as it awaits the EU's reply to acknowledge the three-way handshake.

Moreover, IoT devices are constrained by limited computation, storage capacity, and short battery life, rendering them susceptible to compromise, damage, or hijacking. Consequently, exploiting these limitations, an attacker may infiltrate IoT devices and employ them as botnets to generate and dispatch excessive SYN request packets with a falsified source IP address to fog servers. Consequently, the ACK packet fails to reach the fog server, which remains in the open port state, awaiting the ACK packet. Additionally, the SYN/ACK packets are routed to the falsified host, and the three-way handshake procedure remains incomplete. Furthermore, the connection registration persists in the connection delay buffer until the timeout occurs, obstructing legitimate users from accessing the services [105].

4.2.6 FASA Network Architecture

To address the concerns surrounding SYN flood DDoS attacks in network systems effectively, attack prevention must be integrated into fog computing based on SDN. Indeed, this paper introduces a novel distributed fog defensive system for SYN flood DDoS attacks utilizing ANFIS and SDN Assistance (FASA). The FASA architecture comprises three layers: the cloud layer, the SDN-based fog (SDFN) Layer, and the things layer, as illustrated in Figure 4.4.

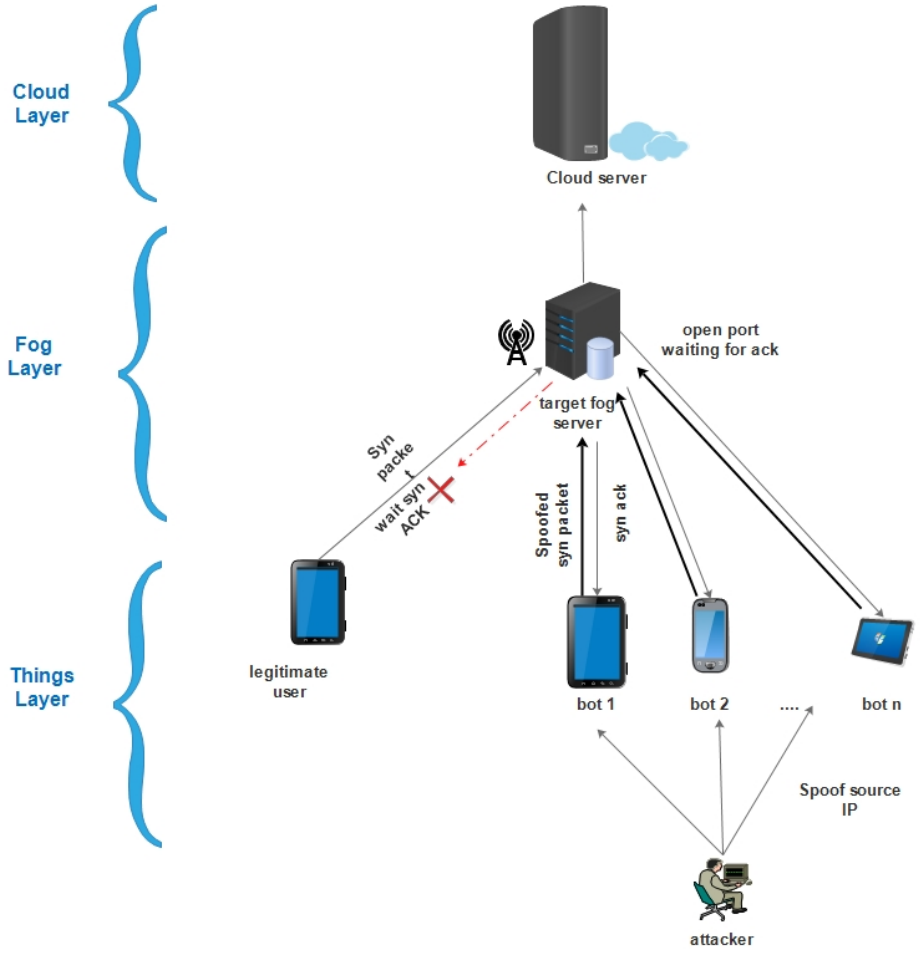


Figure 4.3. *SYN Flood DDoS Attack in Fog Computing.*

a. **Cloud Layer**: as a computing paradigm, the cloud represents an approach to managing a collection of configurable computing resources. It offers flexible, on-demand services and allows system access from any location and at any time. This enables users to utilize resources according to their specific requirements. Cloud technology provides notable features such as immediate adaptability and measurable services [106].

   To automate and seamlessly integrate cloud applications provisioning with the network, SDN (Software-Defined Networking) and cloud technology can be combined. In the FASA system, cloud computing refers to the application plane, which encompasses a variety of useful applications. These applications communicate with the controller to abstract a logically centralized controller, enabling coordinated decision-making.

b. **SDN-based Fog Network layer (SDFN)**: The SDFN layer integrates the fog computing and SDN paradigms to effectively detect and counter DDoS attacks. Recent advances in SDN have created opportunities to introduce intelligence within networks. SDN offers various advantages, such as logically centralized control, software-based traffic analysis, a comprehensive network view, and flexible updates to forwarding rules. These benefits contribute to the enhancement and facilitation of machine learning applications [106]. Consequently, the SDFN layer introduces novel approaches to address DDoS attacks in fog computing environments using SDN. This layer comprises two sub-layers, namely SDFN-server and SDFN-node, which work together to achieve this objective.

   - **SDFN-server:**This specific sub-layer represents the control plane that is implemented on fog servers. It incorporates an intelligent ANFIS (Adaptive Neuro-Fuzzy Inference System) classifier into the control network to make decisions regarding the classification of traffic flows. Subsequently, policies are adjusted based on these decisions.

     Furthermore, the SDFN server establishes communication with the cloud layer (application) through the northbound interface. It also interacts with the SDFN-node layer through the southbound interface.

   - **SDFN-node:** This particular sub-layer pertains to the data plane comprised of physical equipment within the network, such as switches and routers. Its primary function is to facilitate the forwarding of network traffic to their respective destinations. This forwarding process is achieved through the utilization of the OpenFlow protocol.

c. **Things layer:** The role of this layer is to perform the tasks of sensing, collecting, and transmitting data from wireless sensors and end-users to the fog computing infrastructure. The packets that are transmitted through this layer can be classified as either benign or malicious based on their content.

To provide a clearer explanation of the framework for identifying and defending against SYN flood DDoS attacks, the following assumptions have been made:

- The susceptibility of the SDN-based Fog Network server (SDFN-server) to compromise is acknowledged.
- The DDoS attacks specifically target the SDFN servers using TCP SYN flood techniques.
- The integrity of the SDN controller and the switch remains intact, without any compromise.
- The possibility of IoT devices being vulnerable to hacking is considered.

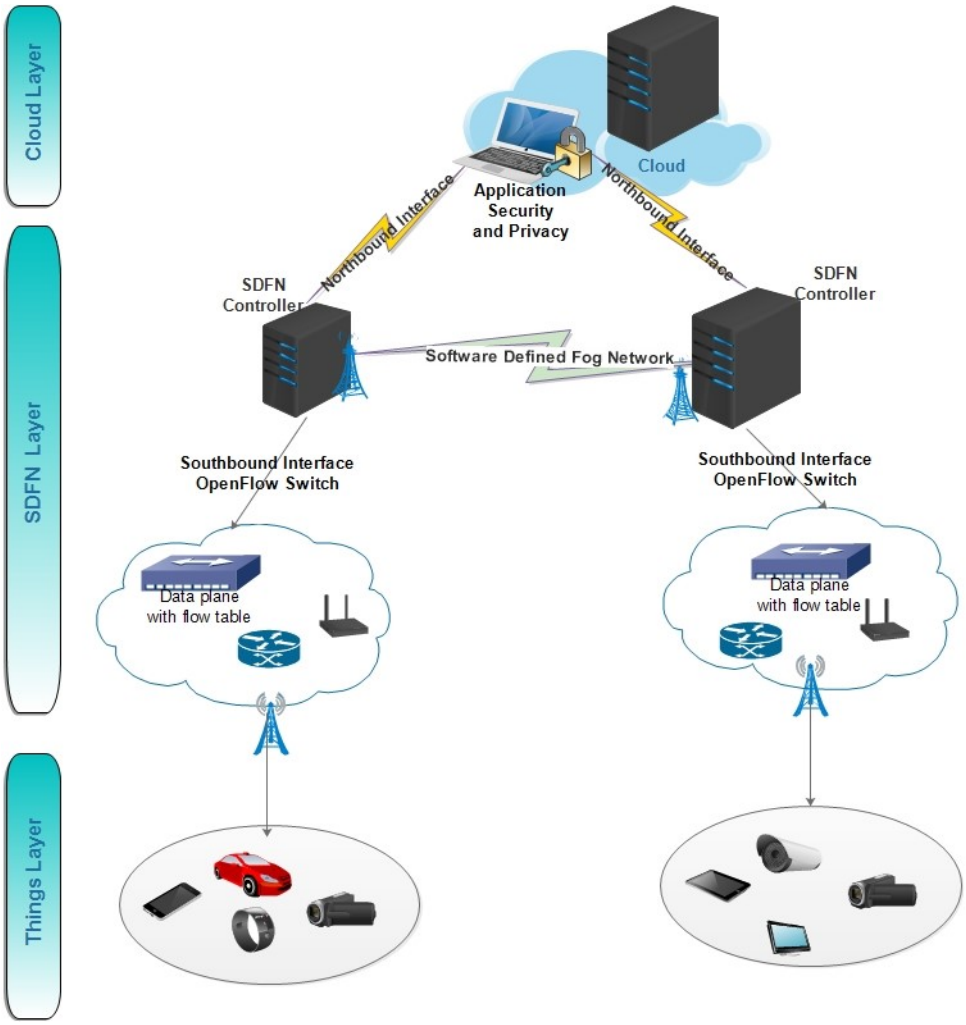


Figure 4.4. *Network Model: SDN-based Fog Network (SDFN).*

4.3 Proposed FASA Framework

The occurrence of SYN flood DDoS attacks can rapidly disrupt a network, and their swift execution makes them challenging to detect. Consequently, it is crucial to have effective measures in place for detecting and mitigating these attacks. In fog computing, there is a need for a detection approach that can filter and block malicious requests before they can negatively impact fog services. To address this, our FASA framework has been developed, enabling real-time identification and immediate mitigation of SYN flood attacks within fog computing. The framework's functionality is illustrated in Figure 4.5.

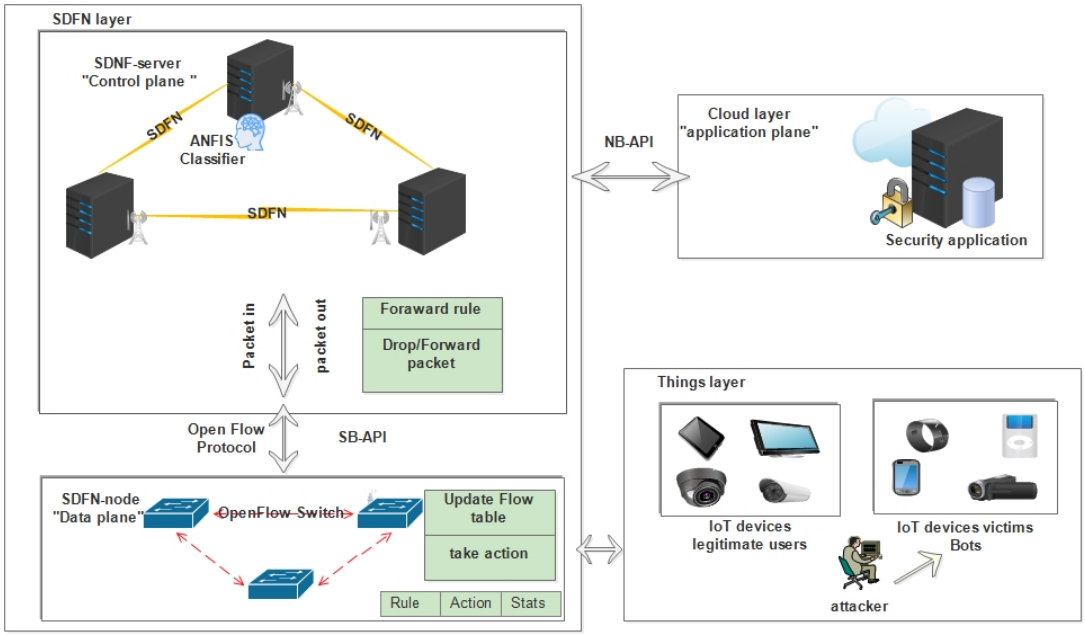


Figure 4.5. *The Proposed FASA Framework.*

4.3.1 The Detection Process

The FASA framework utilizes the ANFIS model and SDN network to ensure the availability of services in the fog network. To achieve the objectives of recognition and detection, a fog layer is established between the cloud layer and the Things layer. This fog layer incorporates recognition techniques that are capable of handling and processing malicious traffic. Additionally, an SDN controller is deployed on the fog

layer to control packets originating from various system nodes, thereby enhancing security and network management.

Moreover, the SDFN-server undergoes prior training using ANFIS algorithms and is tested with two distinct datasets, namely CIC-DDoS2019 and the SDN dataset. Following a successful data preprocessing step, the most significant features are extracted. These features are then divided into training data and testing data to enable self-training of the SDFN server for SYN flood attack identification. Once trained, the ANFIS model can determine the legitimacy of incoming packets, and the controller makes decisions based on this information. The flowchart in Figure 4.6 illustrates this process.

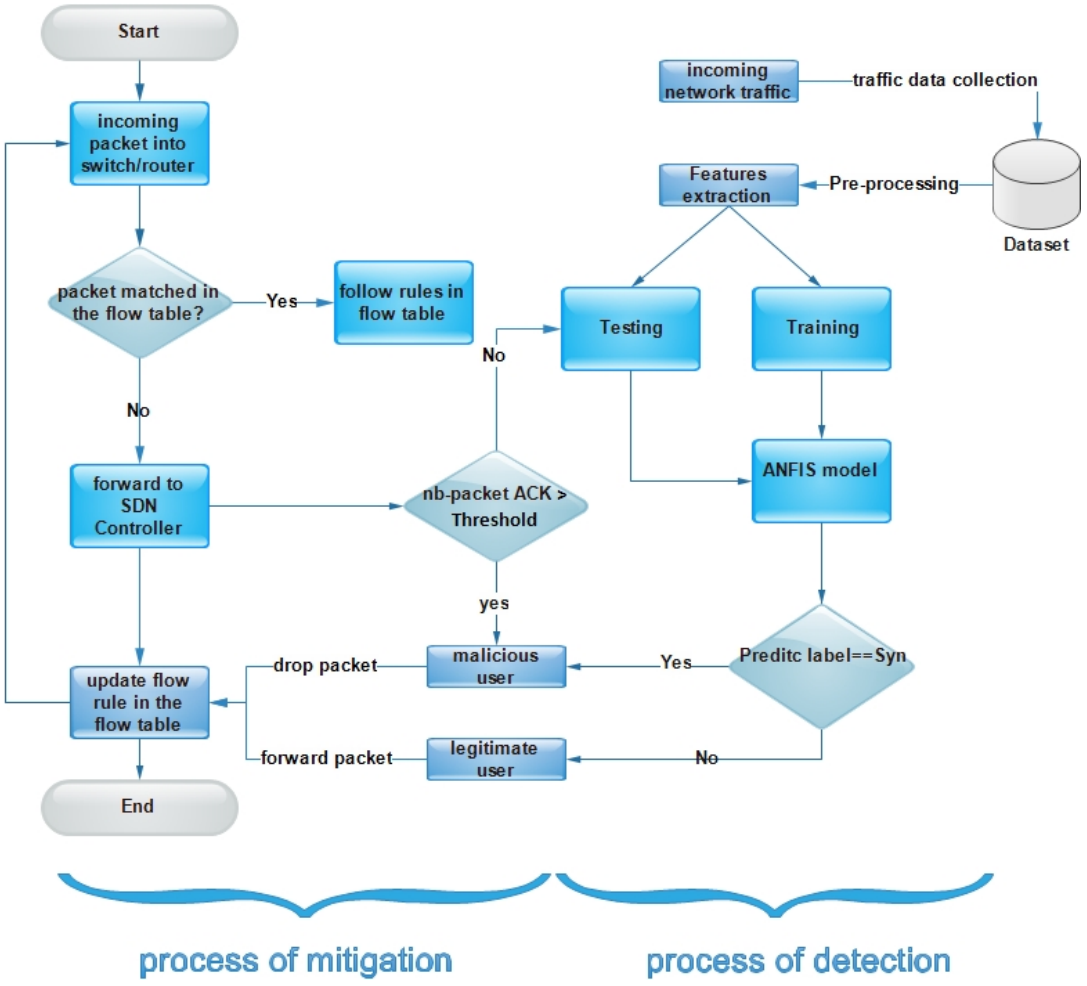


Figure 4.6. *Flowchart of the Proposed Model FASA.*

4.3.2 The Mitigation Process

SDN simplifies the implementation of complex mitigation models by providing a straightforward approach. When an OpenFlow switch receives a packet, it compares it to the matching rule in its flow table. Based on the match result, the switch either forwards the packet to the destination according to the rule or requests assistance from the controller if no suitable rule is found. This request is initiated through the SB-API by the OpenFlow agent within the switch, as depicted in Figure 4.6 of the flowchart.

In the context of attack identification, a threshold value can be determined to assess the maximum serving capacity based on available computational resources. If the number of service requests exceeds this limit, a packet is considered malicious [107]. Conversely, if the number of requests falls below the threshold capacity, the packet undergoes prediction using the ANFIS classifier within the fog server. Thus, the real-time mitigation phase is triggered when the ANFIS model detects a SYN flood packet. The purpose of this phase is to execute defensive measures that limit the damage caused by the exploit.

To initiate the mitigation process, the packet traverses the OpenFlow protocol, which executes the updated rule in the flow table if the packet is determined to be from a legitimate user, granting access. However, if the packet is identified as malicious, the controller identifies the most frequently occurring source MAC address with different source IP addresses. This information is used to determine the infected port number. By correlating the identified MAC address with the corresponding port on the switch, the controller identifies the specific port through which the attack traffic enters the network. To prevent further damage, the controller instructs the OpenFlow switches to drop all packets originating from the host associated with the identified MAC address. Additionally, the switch is directed to block traffic on the specific port linked to the infected host, effectively disabling any communication through that port. Furthermore, the controller updates the flow table of the switch to modify the rules

related to receiving or forwarding packets to the identified port. This ensures that any packets destined for that port are dropped or redirected, mitigating the attack.

As a result, TCP SYN flooding attacks can be identified and prevented by promptly blocking the switch port directly connected to the attacker's host.

Algorithm 1 FASA framework process

```
Input: incoming packet of traffic flow to the switch
Output: response with flow classification and decision
if packet matched in the flow table then
    Apply the rule in the flow table;
else
    Forward packet to SDFN-server;
    Apply ANFIS classifier;
    if flow classified as malicious packet then
        Retrieve the MAC address of the attacker;
        Update rule table in flow table with a malicious user;
        Make a decision:
        Drop the packets with this source MAC address;
        Block the infected switch port;
    else
        Update rule table in SDN with the legitimate user;
        Make decision: Forward the packet to the destination;
    end if
end if
```

4.4 Experiments and Results

4.4.1 Experimental Setup

In this section, we will explore the different tools employed to construct the experimental setup for detecting SYN flood attacks in simulated SDN and fog computing environments. The network traffic was captured and analyzed in real-time using Wireshark [108].

The entire experiment was conducted on a Windows 10 operating system, utilizing an Intel i3 processor and 8GB of RAM. To emulate the network behavior, the SDN Mininet network emulator [109] was utilized, along with the Ryu controller [110].

Ryu is an open-source platform that offers transparency and flexibility, allowing for customization and extension of functionalities. Its Python-based architecture promotes accessibility and simplifies development, enabling swift implementation of SDN applications. Moreover, Ryu supports multiple protocols, including OpenFlow, ensuring seamless communication with diverse network devices. Its compatibility with various networking technologies and hardware makes it well-suited for heterogeneous infrastructures, making it an ideal choice for this research [111].
To train and test our ANFIS model, we employed the Python programming language along with the Keras [112] and TensorFlow [113] libraries for deep learning. To prevent overfitting, we utilized the stratified K-Fold cross-validation technique [114] in the ANFIS algorithm. By employing this method, each fold maintains a class distribution that is identical to the original dataset, resulting in a more accurate and reliable model assessment.

For binary classification, we utilized the Binary Cross entropy loss function, which is a commonly used loss function. The default learning rate in Keras was set to 0.001. Additionally, we selected the Adam optimizer [100], which is an adaptive algorithm for optimizing learning rates in neural network models.

Furthermore, to evaluate the performance and efficiency of the FASA system, we conducted the study using two different scenarios. This allowed us to analyze the system's effectiveness under varied conditions.

- *Scenario 1*: The performance of the FASA system is assessed by implementing it within the SDN environment.
- *Scenario 2*: The performance of the FASA system is evaluated using the publicly available dataset CIC-DDoS 2019 [115]

4.4.2 Experimental Analysis

In the following subsection, we will present the details of each test scenario and present the results of the studies.

a. **Scenario 1** For our experiment, we utilized the Mininet network emulator [109] to create virtual network topologies comprising of controllers, hosts, links, and switches. To run Mininet and Ryu controllers [110], we employed two virtual machines running the Linux operating system. The Ryu controller, based on a Python program, supports various network management protocols, including OpenFlow switches. We specifically chose the Ryu controller for our SDN networking environment due to its simplicity in deployment, scalability, and straightforward architecture. Ryu controllers are widely used in SDN networking, particularly for lightweight traffic communication and control. Moreover, the Ryu controller provides a routing link to OpenFlow switches, ensuring that the network topology can carry out data analysis.

   To emulate our network structure, we implemented a linear topology within Mininet. This topology consists of 8 switches connected to the Ryu controller, with each switch linked to 8 hosts. In total, there are 64 hosts connected to the OpenFlow virtual switches, as illustrated in Figure 4.7.

   The IP address assigned to the Ryu controller is 192.168.162.133. Similarly, each host is assigned a unique IP address. For example, the IP address of Host1 is "10.0.0.1/24", and its corresponding MAC address is 00:00:00:00:00:01, converted from hexadecimal to an integer.

   In scenario 1, the overall workflow includes three main processes: data generation and collection, detection, and mitigation. These processes are implemented using the Mininet virtual machine (VM) and the Ryu controller VM, both of which are based on the Python programming language.

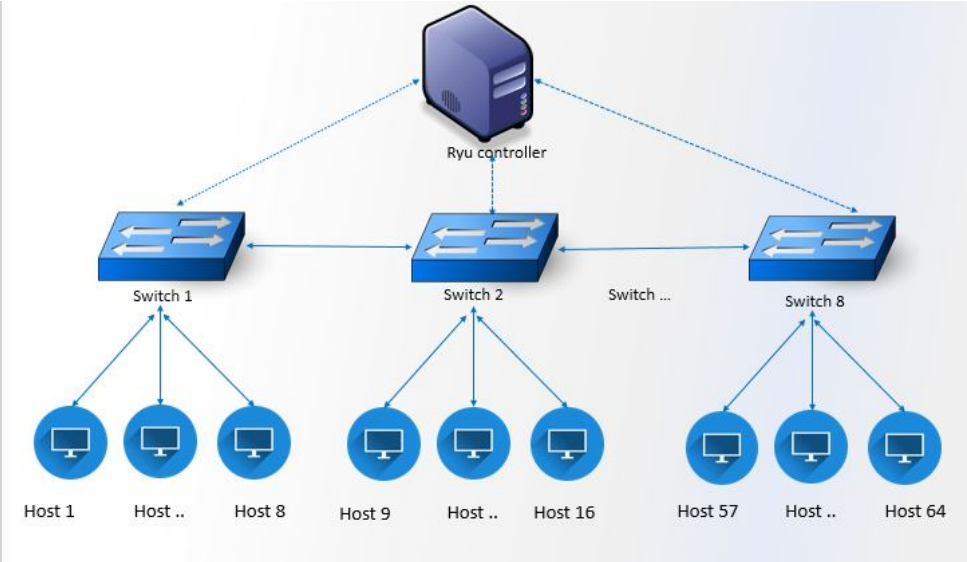


Figure 4.7. *Emulated SDN Network on Scenario 1.*

Table 4.1. *The Collected Dataset using SDN*

SDN dataset	All samples	BENIGN	SYN
Train/test	737156	816156	1553311

- ***Data generation and collection process:*** The SDN dataset is generated using both the Mininet emulator and Ryu controller. We collect normal traffic data by utilizing the "iperf" command while designating one host (Host1) as a Simple HTTP Server listening on port 80. Additionally, we gather SYN flood traffic data using the Hping3 tool, which generates packets with random IP addresses. Hping3 is an open-source TCP/IP protocol tool written in the TCL language. It allows programmers to create scripts for handling and analyzing TCP/IP packets within a limited timeframe.

  MAC addresses play a crucial role in mitigating SYN flood attacks, as layer 2 switches forward incoming traffic based on MAC addresses. They also aid in identifying the infected source port. Furthermore, the layer 4 switch relies on the source and destination ports, which are vital in the flow table. The flow table includes several features such as datapath ID, source IP, source MAC, destination IP, destination MAC, IP protocol, ICMP code, ICMP type, packet counts, and flags. Detailed information about the collected dataset is provided in Table 4.1.

- ***The detection process:*** The detection process involves several steps. Firstly, the collected data (as shown in Table 4.1) undergoes pre-processing. Subsequently, the dataset is divided into a training set, which constitutes 80% of the data, and a testing set, which comprises 20% of the data. The goal is to utilize the ANFIS algorithm with cross-validation to train the dataset and achieve a 100% accuracy, while also mitigating the risk of overfitting.

  Once the Ryu controller receives packets in various forms of regular and attack traffic, it collects the corresponding features and assigns their values to the predicted dataset. The detection module, powered by the ANFIS algorithm, then examines each flow entry to identify potential anomalies or attacks.

- ***The mitigation process:*** The mitigation process involves addressing DDoS attacks, which are challenging to mitigate due to IP spoofing. Simply blocking the suspected attacker's IP address is ineffective in mitigating such attacks. To achieve our objective of obtaining a list of edge switches directly connected to each host, we will store the MAC address, port number, and switch ID for each host in a Python dictionary. This dictionary serves as a data structure from which we can retrieve the necessary parameters for creating mitigation rules. During the mitigation process, each flow entry undergoes the detection process to determine if it is a normal packet or a malicious packet. The flow entry is then forwarded to the Ryu controller, which makes decisions based on the prediction results. If the predicted value of the flow entry is 1, indicating an SYN flood attack where the attacker transmits both the real source MAC address and a random false source IP, the higher MAC address repeated with different IPs in each flow entry indicates that the attacker is the host associated with that MAC address. In this case, we use the assigned MAC address to obtain the port number and switch ID from the dictionary. The Ryu controller responds by enforcing a rule that rejects all packets originating from the attacker,

and this rule is sent to the affected switch. The switch, upon receiving the rule, blocks the specific port directly connected to the attacker's host. By implementing this rule, the switch effectively prevents any communication from the attacker's host through that particular port, thus mitigating the impact of the attack.

Both the hard timeout and idle timeout are important parameters that need to be adjusted during the mitigation process:

- Idle timeout refers to the duration within which a flow rule will be deleted if no match occurs with incoming packets.
- Hard timeout specifies the automatic deletion of a flow rule after the hard timeout period has expired since the rule was created.

Table 4.2. *Experiment Parameters*

Parameters	**Value**
Traffic generation tool	iperf, Hping3
Simulation time	140 sec
Bandwidth	100 Mbits/sec
Data collection interval	5 sec
Server	host 1

In the event of an attack, the Ryu controller applies blocking measures on the OpenFlow (OF) switch. The blocking is configured with an idle time of 0 seconds and a hard time of 300 seconds, set with a high priority of 1000 for our model. This means that the switch will continue to block the source port for 300 seconds without notifying the controller.

On the other hand, if the detection result is 0, indicating normal traffic, the idle time is set to 200 seconds for each flow entry, and a fixed priority of 10 is assigned. If no matching occurs within this period, the flow rule will be removed after 200 seconds. The hard time is set to 400 seconds, after which all flow entries will be deleted.

During the experiment, real-time flow traffic was captured using Wireshark. Figure 4.8 displays the plot of packets per second versus time, representing

traffic flow. Additionally, Table 4.2 presents the parameters used in the experiment.

At the start of the experiment, normal traffic is sent out at time 0 seconds. Subsequently, a SYN flood attack is initiated, and at around time 60, the packet rate reaches a threshold value close to 700 packets per second. The ANFIS detection module identifies the attack and the mitigation module is in play. The controller employs appropriate flow rules to mitigate the attack by dropping packets, blocking the source ports involved in the attack, and instructing the switches to update the flow table accordingly. The attack is successfully mitigated in less than 5 seconds, resulting in a significant drop in the packet rate. The graph demonstrates the continuous flow of normal traffic without any disruptions until the end of the 140-second experiment. This period is crucial, as it showcases the controller's ability to effectively handle incoming packets.

Figure 4.9 illustrates that during the attack, there was a decrease in bandwidth consumption, dropping to as low as 90 Mbits/sec. However, the bandwidth quickly recovered to its pre-attack state and remained relatively stable at around 100 Mbits/sec. This demonstrates the effectiveness of our model in mitigating the impact of the attack and restoring normal network performance.

b. **Scenario 2** In the second scenario, we assess the effectiveness of our proposed model in identifying TCP SYN flood DDoS attacks. To evaluate the model, we utilize the CIC-DDoS dataset developed by Sharafaldin et al. (2019) [115]. This dataset is specifically designed for detecting DDoS attacks and classifying different attack types. It is presented in a CSV format and contains a combination of benign traffic and various popular DDoS attacks that occurred in 2019. The dataset was collected over two days and represents real-world network traffic captured in PCAP files. Additionally, the dataset includes labeled traffic flows

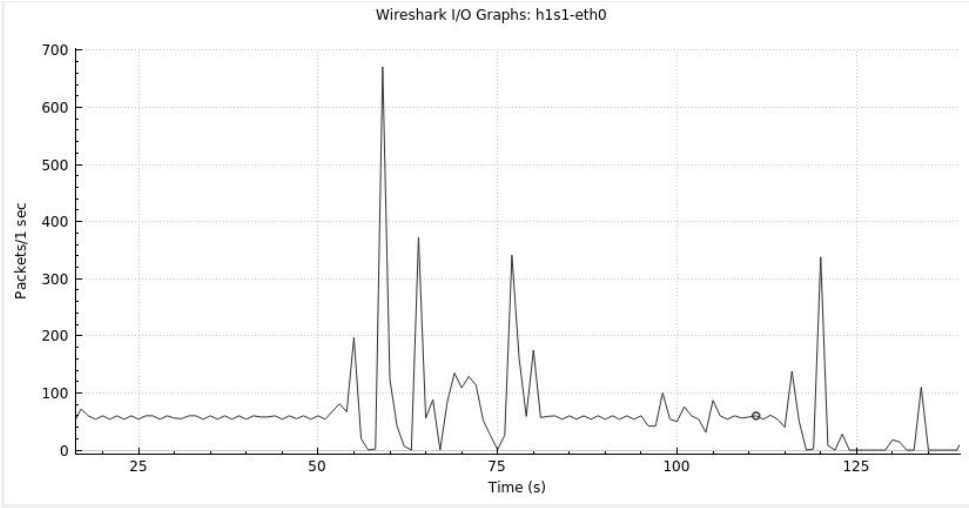


Figure 4.8. *Real-time Detection and Mitigation of Syn Flood Attack.*

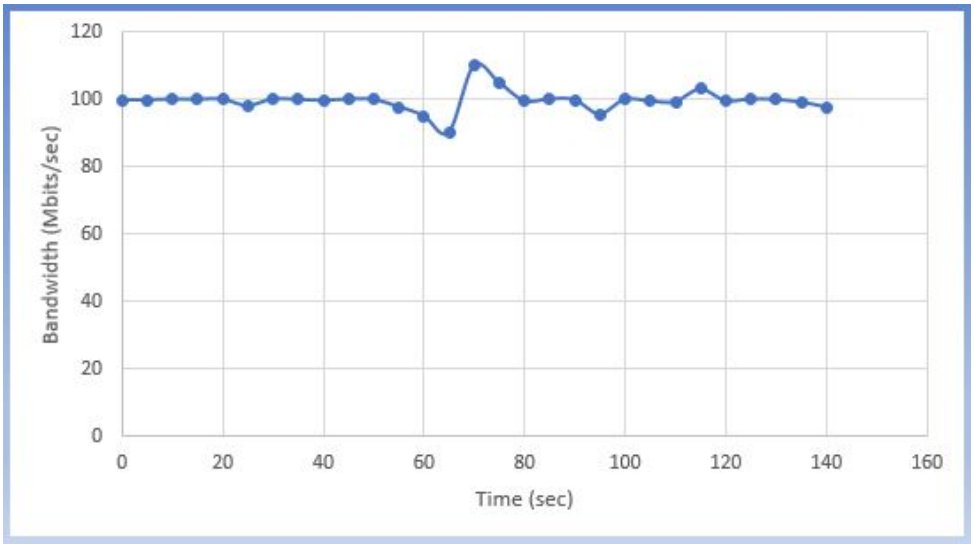


Figure 4.9. *Bandwidth Usage.*

obtained through network traffic analysis using CICFlowMeter-V3. Originally, the dataset consisted of 88 features.

For this particular scenario, we focus on the SYN flood dataset, as outlined in 4.3. The TCP SYN flood attack belongs to the exploitation category of DDoS attacks and takes advantage of vulnerabilities in TCP connection protocols. The dataset covers two days, with each day featuring a different attack category and a broad range of imbalanced class distribution.

- ***Resampling data:*** Data resampling is performed to address the issue of imbalanced classification in both the training and testing datasets. The

Table 4.3. *SYN Flood CIC-DDoS 2019 Dataset*

All New dataset	All samples	BEGNIN	SYN
Training day	1582681	392	1582289
Testing day	4320541	35790	4284751

minority class, labeled as "BENIGN," has a small number of samples, which can adversely affect the model's learning and decision-making capabilities. Moreover, it can lead to overfitting in our model.

To mitigate this problem, we construct a new dataset that consists of samples from both the training and testing datasets. Specifically, we include all samples labeled as "BENIGN," which accounts for 20% of the total dataset. Additionally, we include 80% of the samples labeled as "SYN." The resulting dataset, presented in Table 4.4, aims to balance the representation of the classes and alleviate the challenges associated with imbalanced classification.

Table 4.4. *The New Balanced SYN Flood Dataset*

New dataset	All samples	BEGNIN	SYN
Train/test	180910	36182	144728

- ***Data Pre-processing:*** Data preprocessing plays a crucial role in the analysis of our dataset, which consists of 88 features. In this section, we will discuss the techniques employed to cleanse and prepare the data for utilization in our proposed ANFIS algorithms. The objective is to eliminate undesirable attributes and make necessary adjustments to ensure the data is suitable for training. By implementing a data preprocessing step, as depicted in Figure 4.10, we aim to enhance the reliability of the training process, leading to a more accurate model.

  ✓ Initially, we eliminated features from the dataset that possessed a unique value across the entire dataset and did not have any impact on the training phase ('Bwd PSH Flags',' Fwd URG Flags', ' Bwd URG Flags', 'FIN Flag Count',' Fwd Avg Bytes/Bulk', Fwd Avg

Packets/Bulk', ' Fwd Avg Bulk Rate', 'Bwd Avg Bytes/Bulk', ' PSH Flag Count',' ECE Flag Count',' Bwd Avg Packets/Bulk', 'Bwd Avg Bulk Rate').

✓ In the flow data extracted from the Syn CSV file, certain values of 'Init Win bytes forward' and 'Init Win bytes backward' were erroneously set to -1. However, initiating a byte window of size -1 is illogical and not feasible. This issue was attributed to a software problem with CICFlowmeter. To ensure the smooth progression of the training phase, it is necessary to either set these values to 0 or remove them from the dataset. Some values of 'Init Win bytes forward' and 'Init Win bytes backward' of flow data from the Syn CSV file were set to -1.

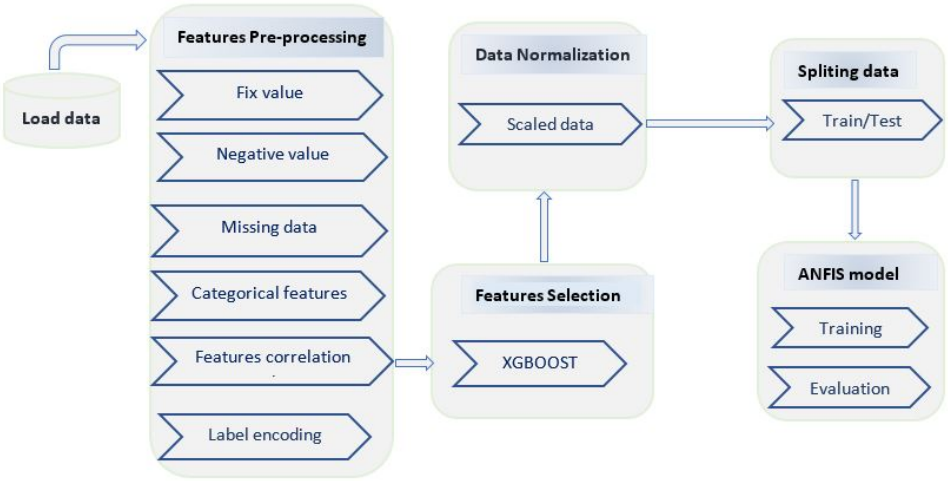


Figure 4.10. *Data Pre-processing.*

✓ Dealing with missing data had a disruptive effect on the training of the model. To address this issue, the lines in the dataset that contained 'infinity' and 'NaN' values in the 'Flow Bytes/s' and 'Flow Packets/s' features were removed.

✓ We eliminated categorical features from the dataset that have the potential to vary across different networks ('Source Port', 'Destination Port', 'Source IP', 'Destination IP', 'Flow ID', 'SimillarHTTP', 'Unnamed: 0', 'Timestamp').

- ✓ To accurately identify significant features, we removed columns from the dataset that exhibited a correlation coefficient higher than 0.8(' Total Backward Packets', ' Total Length of Bwd Packets', ' Fwd Packet Length Std', ' Bwd Packet Length Min', ' Bwd Packet Length Mean', ' Bwd Packet Length Std', ' Flow IAT Mean', ' Flow IAT Std', ' Flow IAT Max', 'Fwd IAT Total', ' Fwd IAT Mean', ' Fwd IAT Std', ' Fwd IAT Max', ' Fwd IAT Min', ' Bwd IAT Std', ' Bwd IAT Max', ' Fwd Header Length', ' Bwd Header Length', ' Max Packet Length', ' Packet Length Mean', ' Packet Length Std', ' Packet Length Variance', ' RST Flag Count', ' Average Packet Size', ' Avg Fwd Segment Size', ' Avg Bwd Segment Size',' Fwd Header Length.1', 'Subflow Fwd Packets', ' Subflow Fwd Bytes', ' Subflow Bwd Packets', ' Subflow Bwd Bytes', ' Active Max', ' Active Min', 'Idle Mean', ' Idle Max', ' Idle Min').
- ✓ To facilitate the detection and classification of DDoS attacks, the dataset is divided into two distinct classes. The "BENIGN" label is encoded as "0," while the "Syn" label is encoded as "1" in the dataset specifically designed for detecting SYN flood DDoS attacks on network traffic.
- ✓ Feature selection is employed to identify important data features and reduce the amount of data needed for detection purposes. In this process, we utilize the XGBoost technique, which assigns an importance score to each feature based on its impact on critical decision-making using boosted decision trees [116]. Subsequently, based on the assigned feature importance ratings, we remove features that are deemed to have negligible significance. ' Protocol', ' Flow Duration',' Total Fwd Packets', ' Fwd Packet Length Max', 'Bwd Packet Length Max', ' Flow IAT Mean', ' Flow IAT Min', 'Bwd IAT Total', ' Bwd IAT Mean', ' Bwd IAT Min', 'Fwd PSH Flags',' Fwd Packets/s', ' Bwd Packets/s', ' Min Packet Length',' SYN Flag Count',' CWE Flag Count', ' Down/Up Ratio',' Init Win bytes backward', ' act data pkt fwd', 'Active Mean',

' Active Std', ' Idle Std', and choose nine ideal feature subsets, as presented in Table 4.5.

Table 4.5. *Features Selected with XGBoost*

Feature Name	Description
Total Length of Fwd Packets	Overall size of packet in the forward direction.
Fwd Packet Length Mean	Mean size of the packet in the forward direction.
ACK Flag Count	Number of packets with ACK.
URG Flag Count	Number of packets with URG.
Init Win bytes forward	Number of bytes sent in the initial window in the forward direction.
min seg size forward	The observed minimum segment size in the forward direction.
Inbound	The direction in which traffic moves between networks.
Label	Type of packets for classification.

- To ensure consistency, we normalize the data by scaling all features to a range of 0 to 1. As mentioned earlier, the dataset is split into training and testing data. Cross-validation is employed during the training phase to prevent overfitting.
- Finally, we evaluate the performance of the ANFIS model by making predictions on unseen data. The following section presents a detailed analysis of the model's performance and the obtained results.

4.4.3 Performance Metrics

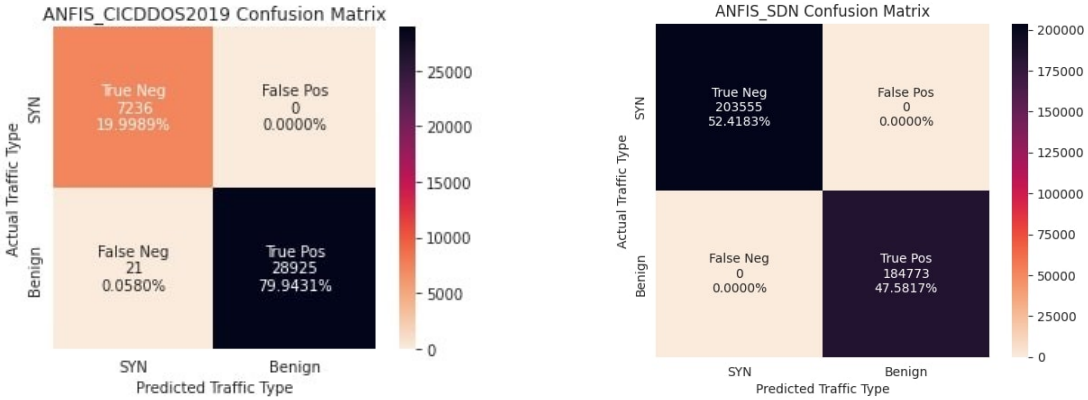
Accurately evaluating models relies on utilizing appropriate performance metrics. In this section, we examine the following performance metrics to evaluate the FASA framework:

- *True Negatives (TN)*: This refers to the accurate identification of normal flow data as such.
- *True Positives (TP)*:This represents the correct identification of malicious flow data.

- *False Positives (FP)*: This indicates the misclassification of normal flow data as malicious traffic.

- *False Negatives (FN)*: This signifies the misclassification of malicious flow data as normal flow data.

Furthermore, we present the confusion matrix to describe the classification performance of our model. The confusion matrix summarizes the correct and false predictions achieved through our proposed approach, as illustrated in Figure 4.11. Accurately distinguishing the Benign class within our model is of paramount



(a) *Using CIC-DDoS dataset.* **(b)** *Using SDN dataset.*

Figure 4.11. *Confusion Matrix of the ANFIS Model.*

importance, as high false positive rates can lead to unnecessary complexity and unwarranted alerts. Our primary objective is to minimize the false positive rate. As a result, our framework achieves a false positive rate of 0% in both the CIC-DDoS2019 and SDN datasets. Additionally, it achieves a false negative rate of 0.058% in the CIC-DDoS2019 dataset and 0% in the SDN dataset.

To assess the model's performance in distinguishing between false positives and true positives, we utilize the Receiver Operating Characteristic (ROC) curve. This curve illustrates the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR). The area under the ROC curve (AUC) serves as a measure of the model's ability to differentiate between the two classes. As depicted in Figure 4.12, our model achieves an AUC of 99.96% using the CIC-DDoS2019 dataset and 100% using

the SDN dataset. These high AUC values indicate that our proposed model effectively separates positive from negative classes.

To ensure the model's generalizability and mitigate overfitting, we employ established techniques like k-fold cross-validation. Additionally, meticulous selection and optimization of impactful traffic features enhance the model's proficiency in distinguishing between normal and attack behaviors. The fusion of fuzzy logic and neural learning components proves effective in capturing complex traffic patterns. Moreover, training on diverse attack data distributions further enhances the model's robustness. To conduct a comprehensive comparative evaluation of our proposed

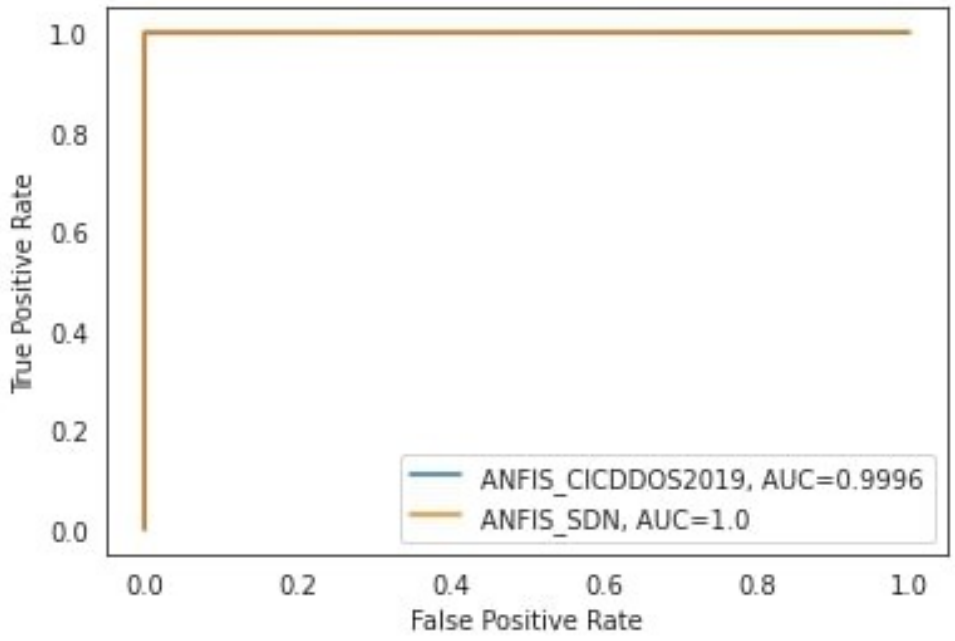


Figure 4.12. *ROC Cuvre of ANFIS Model.*

model, we have employed various measures such as accuracy, precision, recall, and F-score. These metrics are commonly used in SYN flood DDoS detection systems and provide valuable insights into the performance of our model. Let's delve into the description of these metrics:

1. Accuracy measures the overall correctness of the model's predictions by calculating the ratio of correctly classified instances to the total number of

instances. It is calculated as follows:

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{4.6}$$

2. Precision quantifies the model's ability to identify positive instances out of the predicted ones correctly. It is calculated by dividing the number of true positives by the sum of true positives and false positives, it is computed as follows:

$$precision = \frac{tp}{tp + fp} \tag{4.7}$$

3. The false positive rate is determined by calculating the ratio of negative samples that were incorrectly classified as positive using the following formula:

$$fp - rate = \frac{fp}{fp + tn} \tag{4.8}$$

4. The recall Recall, also known as sensitivity or true positive rate, measures the model's ability to correctly identify positive instances out of the total actual positive instances. It is calculated by dividing the number of true positives by the sum of true positives and false negatives., It is determined using the equation:

$$recall = tp - rate = \frac{tp}{tp + fn} \tag{4.9}$$

5. The F-score is a combined measure that takes into account both precision and recall. It provides a balanced assessment of the model's performance by considering the harmonic mean of precision and recall:

$$f1 - score = \frac{2 \times precision \times recall}{(precision + recall)} \tag{4.10}$$

4.4.4 Evaluation Results

To validate our system, we conducted a comparison between the FASA framework and the FUPE method [117], along with other DDoS attack detection systems that

were applied to SDN using the CIC-DDoS 2019 dataset. The comparison results are presented in Table 4.6.

Table 4.6. *The Evaluated Metrics were used to Compare the Results of ANFIS with other Methods.*

Method	Accuracy	precision	Recall	F1-score
ANFIS SDN	100	100	100	100
ANFIS CIC-DDoS2019	99.95	100	99.94	99.95
FUPE [117]	98.2	96.08	98	N/A
CNN [118]	95.4	93.3	92.4	92.8
GAN [119]	94.38	94.08	97.89	95.94
MLP [120]	95.01	95.46	94.51	94.98

The first approach, FUPE [117], utilizes a fuzzy-based multi-objective particle swarm optimization approach to address security-aware task scheduling in IoT-fog networks. The second approach is based on Convolutional Neural Network (CNN) [118], which is a cost-effective supervised classifier designed to detect suspicious events in a data center. Another approach involves the use of Generative Adversarial Network (GAN) [119] for identifying DDoS threats in SDN environments. Lastly, the Multi-layer Perceptron (MLP) [120] is employed to detect and prevent Low Rate-DDoS attacks in SDN settings.

Figure 4.13 provides a comprehensive analysis of the metric findings from the comparative approaches, illustrating the performance of each method in a comparative manner.

Figure 4.13 provides a comprehensive analysis of the performance of various methods, including our FASA framework, in identifying SYN flood DDoS attacks. Notably, our model using the SDN dataset demonstrates superior performance compared to all previous techniques, achieving 100% accuracy, precision, recall, and F1-score. These results closely resemble the outcomes obtained using the CIC-DDoS2019 dataset.

The accuracy of each learning algorithm is also evaluated. Among the classifiers, the ANFIS algorithm achieves the highest accuracy rating of 99.95%, followed by the

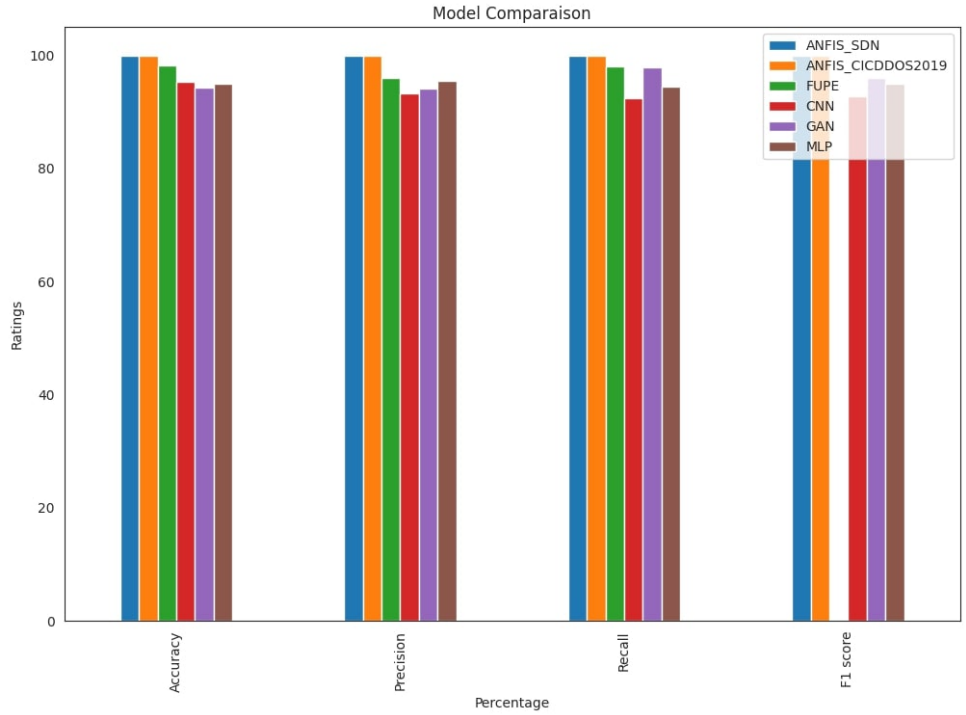


Figure 4.13. *Evaluation Metrics for Comparative Methods.*

FUPE approach with 98.2% and the CNN algorithm with 94.83%. The MLP and GAN classifiers attain accuracies of 95.4% and 95.01%, respectively.

Figure 4.13 further illustrates the precision of each algorithm in distinguishing between legal and malicious traffic. The ANFIS algorithm achieves a precision of 100%, followed by FUPE with a precision of 96.08%, and the MLP with a precision of 95.46%. The GAN and CNN algorithms achieve precisions of 94.08% and 93.3%, respectively.

Additionally, the recall values of all methods employed in the performance evaluation are depicted. The ANFIS algorithm exhibits a recall value of 99.94%, followed by FUPE with 98%, while GAN achieves a recall rating of 97.89%. In comparison, the CNN achieves the lowest recall value of 92.4%, while the MLP achieves a recall of 94.51%.

Furthermore, Figure 14 presents the F1 scores of the classification methods. The ANFIS algorithm obtains the highest F1-score of 99.95%. On the other hand, GAN, MLP, and CNN achieve F1-scores of 95.94%, 94.98%, and 92.8%, respectively. The F1 score for the FUPE approach is not mentioned.

In conclusion, our FASA framework outperforms the other evaluated approaches, demonstrating its effectiveness in identifying SYN flood DDoS attacks. The promising test results highlight the efficacy of our approach and its potential for enhancing network security. .

4.5 Summary

In this chapter, we proposed a mitigation framework called FASA, which combines Fog computing, Adaptive Neuro-Fuzzy Inference System (ANFIS), and Software Defined Networking (SDN) to combat SYN Flood DDoS attacks. By integrating SDN and the fog environment with the ANFIS machine learning algorithm, our framework introduces intelligence into the SDN controller, making it more suitable, efficient, and secure against SYN flood attacks.

We conducted training and evaluation of our framework using the recently released CIC-DDoS2019 dataset, which includes a comprehensive collection of SYN flood DDoS attacks. The performance assessment revealed that our proposed model achieves high detection accuracy and low rates of false positives and false negatives. This is a significant achievement, as our framework outperforms well-known machine learning algorithms in terms of precision, recall, and F-score, offering superior evaluation metrics.

Overall, our findings highlight the effectiveness and robustness of the FASA framework in mitigating SYN Flood DDoS attacks, providing a promising solution for enhancing network security.

Chapter

5

Intelligent Intrusion Detection through Federated Learning in Fog-IoT Enabled Smart Healthcare Systems

Contents

5.1 Introduction

In this chapter, we present SA-FLIDS, a Secure and Authenticated Federated Learning-based Network Intrusion Detection System for Fog-IoT-enabled Smart Healthcare Systems. SA-FLIDS ensures a high level of security for IoMT devices through secure federated learning, preserving privacy. We focus on detecting and mitigating cyber attacks on IoMT devices using fog computing, while also addressing vulnerabilities in decentralized learning. Client authentication is achieved through a blockchain-based Self-Sovereign Identity model. Performance evaluation demonstrates SA-FLIDS' effectiveness in identifying and mitigating attacks while meeting privacy, scalability, and sustainability requirements.

The structure of this chapter is as follows: Section 2 offers a background on the subject matter. In Section 3, we provide a formal definition of cyber attacks and risk in IoMT-enabled smart healthcare systems, and we present the goals and design of the SA-FLIDS system. Section 4 introduces our proposed SA-FLIDS model. The security and privacy analysis is outlined in Section 5. In Section 6, we report and discuss the results of our evaluations. The chapter concludes with a final section summarizing the key findings.

5.1.1 Motivation

Smart healthcare systems, which leverage advancements in technology like IoT and cloud computing [121], have the potential to revolutionize healthcare by improving services, reducing costs, and enabling more accurate diagnoses and treatments [122] [123]. However, the integration of various technologies and medical devices also exposes these systems to cyber-attacks, posing risks to patient safety and the availability of healthcare services [124].

To protect smart healthcare systems from cyber threats, it is crucial to deploy a Network-based Intrusion Detection System (NIDS). Traditional IDSs, however, have

limitations in terms of false positives, scalability, and efficiency. Therefore, scalable and effective cybersecurity measures are urgently needed to safeguard IoT networks and medical devices in smart healthcare services.

In the field of IoMT-based healthcare security, Machine Learning (ML) approaches, particularly Federated Learning (FL), have gained traction for detecting anomalies and malicious activities. FL allows mobile devices to collaboratively train a shared model while preserving data privacy. However, FL is susceptible to adversarial attacks such as poisoning and Sybil attacks, which can compromise the accuracy and convergence of the model.

To address these challenges, the integration of blockchain technology with FL can provide a secure framework. Blockchain offers advantages in terms of immutability, transparency, decentralization, and data security. By incorporating blockchain and Self-Sovereign Identity (SSI) technologies, a novel framework called SA-FLIDS (Secure and Authenticated FL-based NIDS) can be developed to protect smart healthcare networks. SA-FLIDS analyzes incoming traffic, detects and mitigates cyber-attacks on IoMT devices, and ensures privacy-preserving with secure FL. It employs an identity management and device authentication scheme to prevent Sybil attacks and ensure that only trusted nodes contribute to the training process. Participants' privacy is safeguarded as SA-FLIDS does not share users' private data with the centralized training module.

5.2 Background Knowledge

This section provides an overview of relevant concepts, including intrusion detection systems (IDS), Federated Learning (FL), and Self-Sovereign Identity (SSI) techniques based on Blockchain. These concepts are essential for understanding and implementing our proposed model.

5.2.1 Intrusion Detection System

An IDS is a security tool that monitors network activities to identify unauthorized actions or policy violations [125]. It logs information about intrusions, issues alerts, and takes necessary actions to mitigate or correct unauthorized access. IDS checks incoming network traffic and safeguards authorized users from suspicious activity that might compromise an information system's availability, confidentiality, or integrity [126].

In the context of fog-IoT systems in smart healthcare, a distributed detection system based on anomaly-based IDS is considered. This system helps identify abnormal behavior, unauthorized access attempts, or suspicious network activities. It utilizes predefined rules or ML algorithms to detect potential threats and generates alerts for further investigation [127].

5.2.2 Federated Learning (FL)

In traditional ML for IoT applications, usually, every IoT device uploads its data to cloud servers. , creating a shared model used across devices [128]. However, this centralized learning approach faces challenges related to privacy, latency, bandwidth, and connectivity. To overcome these limitations, FL was developed as an innovative ML technique. FL allows the training of a model using decentralized data sources without the need to exchange the data itself [129], as shown in Figure 5.1. By adopting FL, the privacy and security of local data samples can be preserved since they are no longer shared with external entities [130]. Additionally, FL reduces requirements for latency, power, and storage because data transmission is minimized due to the absence of a central entity storing all the data [131].

There are three prevalent types of FL [132]:

1. **Horizontal FL** This approach is used when datasets share the same feature space but differ in the sampling space. In other words, the data collected by different devices share common attributes but represent different instances or samples.
2. **Vertical FL:** This type of FL is implemented when datasets have different feature spaces but the same sample space. It means that the data collected from different devices have different attributes or features, but they correspond to the same instances or samples.
3. **Federated Transfer Learning:** This approach is employed when datasets have both different feature space and sampling space. In this case, the data collected from various devices not only have different attributes but also represent different instances or samples.

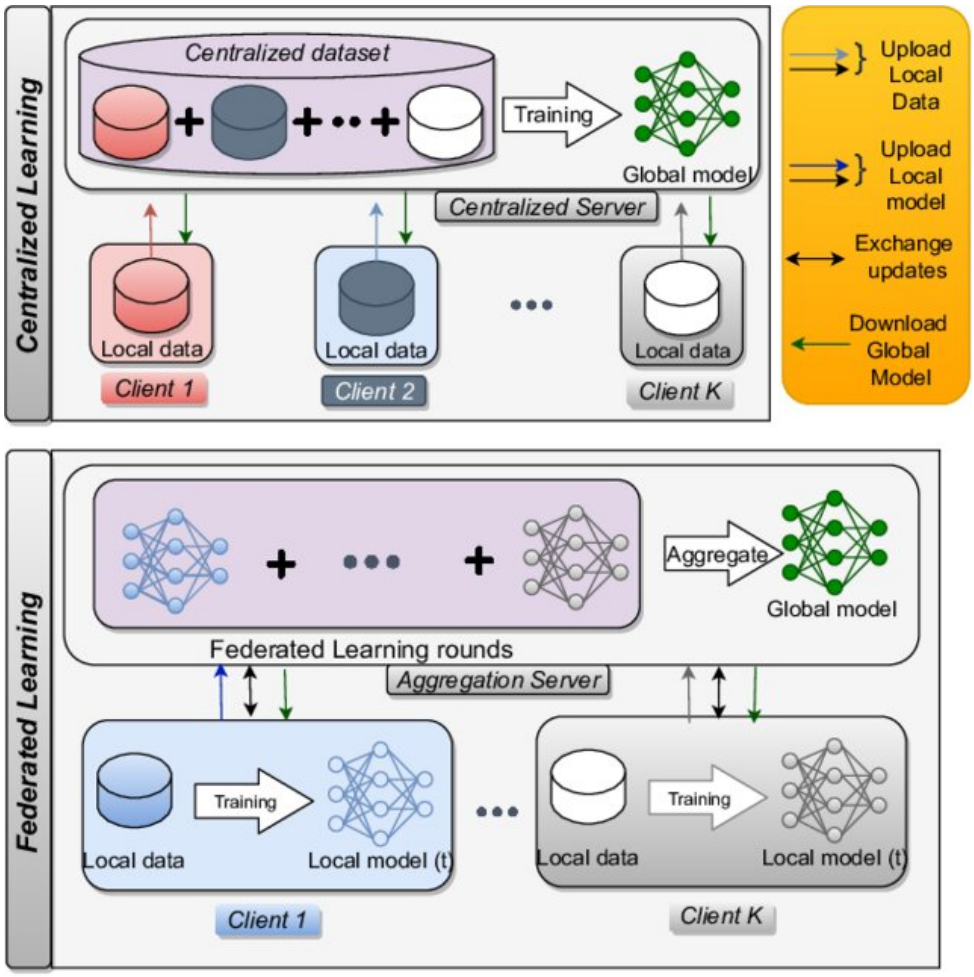


Figure 5.1. *Federated Learning and Centralized Learning [133].*

5.2.3 Federated learning for IoT Intrusion Detection

FL has been widely adopted in Internet of Things (IoT) networks due to its ability to enhance privacy and security while minimizing data transmission [134]. In the context of IDS, this approach facilitates the creation of more sophisticated ML models by utilizing diverse data samples from multiple sources, all while maintaining network users' privacy during the learning process [130].

In this approach, the current model is retrieved and updated locally On IoT devices using the relevant data. These locally trained models are then transmitted back to a central server for aggregation, such as averaging weights, resulting in an improved single global model that is returned to the IoT devices. The successful deployment of FL relies on the effective distribution of data, which presents both practical and technical challenges.

5.2.4 Blockchain-based SSI

Now, turning our attention to SSI based on Blockchain, this section delves into the use of Blockchain technology for creating secure and decentralized identity management systems.

5.2.4.1 Blockchain

Blockchain technology has gained prominence due to its properties of decentralization, immutability, and persistence within distributed peer-to-peer networks. Blockchain provides a secure, transparent, and decentralized system through its core elements like blocks, transaction records, consensus mechanisms, smart contracts, mining, immutable ledger, cryptographic keys, and hash functions [135].

5.2.4.2 Self-Sovereign Identity (SSI)

It represents a decentralized paradigm for digital identity management. The primary goal of SSI is to empower individuals with ownership and control over their digital credentials, enabling them to establish trusted relationships and access information while proving their identity [135]. SSI is built upon two fundamental standards: Decentralized Identifier (DID) and Verifiable Credential (VC). These standards provide the necessary infrastructure and protocols to ensure secure, privacy-preserving, and user-centric digital identity management [135]. More details about blockchain-based SSI are discussed in Chapter 3, Section 3.6

5.3 Cybersecurity Challenges and Risks in Smart Healthcare Systems Enabled by IoMT

In smart healthcare systems enabled by the IoMT, ensuring reliable and secure communication is of utmost importance due to its direct impact on patient well-being. However, the unique characteristics of medical IoT devices, such as limited resources (e.g., poor battery life and limited memory), make them susceptible to various hacking attempts. These compromised devices can be hijacked and incorporated into botnets by malicious actors. Consequently, several types of cyberattacks can be carried out using compromised IoMT devices. Here are some common types of attacks that may be carried out on compromised IoMT devices:

1. **Denial-of-Service (DoS) and Distributed DoS (DDoS) Attacks:** These attacks are designed to disrupt the availability of IoMT operations. In DoS attacks, a single botnet is employed to flood IoMT devices, while DDoS attacks involve multiple botnets working together [23].
2. **Information Gathering Attacks:** Attackers gather comprehensive information about targeted IoMT devices. This involves reconnaissance activities such as scanning attacks, which serve as preparatory steps for subsequent attacks [136].

3. **Exploiting Web-Based Vulnerabilities:** Attackers specifically target web services running on IoMT devices. Web-based attacks encompass various methods, including injection, hijacking, poisoning, spoofing, and DoS, to compromise the security of web services [136].
4. **Communication Spoofing Attacks:** suspicious actors use these attacks to masquerade as victim systems, gaining unauthorized access to network traffic. The primary objectives include system access, data theft, and malware dissemination [137].
5. **Brute-Force Attacks:** These attacks exploit weak passwords by systematically trying every possible combination of characters. This makes strong, complex passwords essential for protecting your accounts [138].
6. **Mirai, an IoT Threat:** a well-known example of a widespread DDoS attack specifically designed to target IoMT devices [139].

Therefore, if a malicious IoMT device compromises a fog server, it opens the door to unidentified individuals accessing confidential patient data and electronic health records (EHRs). This breach of data privacy and security poses a significant risk to patients. The compromised data often includes highly personal and private information, like credit card details, health conditions, and other sensitive data, leaving patients vulnerable to various threats.

Furthermore, the unavailability of fog servers disrupts crucial healthcare services, including real-time monitoring of patient's vital signs. This compromise in tracking and monitoring essential health indicators in real-time creates potential risks and challenges in delivering timely and appropriate care, as depicted in Figure 5.2.

In critical situations, such as emergencies, the heightened risk to patients' lives underscores the severity of the potential consequences. It highlights the urgent need to establish robust cybersecurity measures and implement comprehensive security protocols to safeguard patient data. These procedures are crucial for maintaining the sustainability of critical healthcare services and effectively mitigating risks.

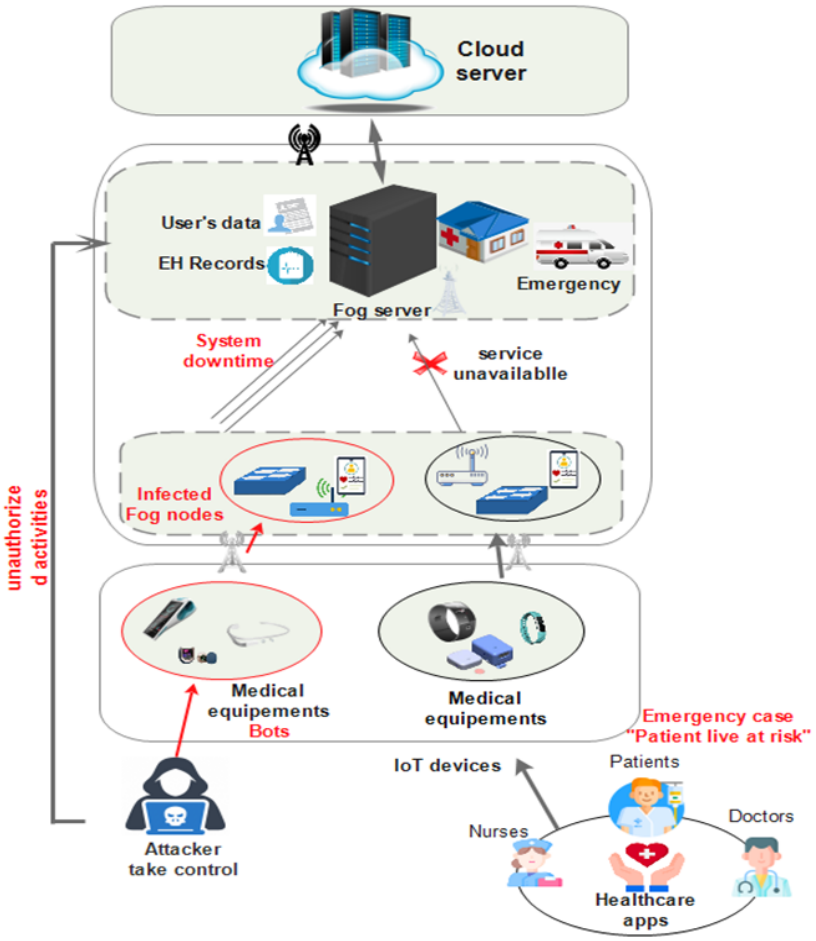


Figure 5.2. *Cyber-attacks in Smart Healthcare.*

5.4 SA-FLIDS System and Design Goal

In this section, we delve into the architecture and threat model of the SA-FLIDS system, as well as discuss the design goals of the system.

5.4.1 SA-FLIDS System Architecture

The SA-FLIDS system introduces a novel approach: a FL-based IDS for Smart Healthcare that leverages Blockchain Technology, secure communication protocols, DID, and VC. The primary objective of the SA-FLIDS system is to analyze network traffic, detect and mitigate cyber-attacks targeting IoMT devices, and improve the overall security of smart healthcare systems. The proposed architecture of our model

consists of three layers: the Cloud layer, the Fog layer, and the IoMT device layer, as illustrated in Figure 5.3.

1. **Cloud Layer:** The Cloud layer serves as the foundational infrastructure that facilitates the delivery of scalable and flexible services, accessible from any location [140]. Within this layer, fog nodes and fog servers can dynamically allocate resources according to the requirements of the FL process.
2. **Fog Network Layer:** The Fog Network Layer is where the intelligent IDS based on FL and blockchain technology is deployed. This layer is responsible for overseeing the network and making decisions related to traffic flow classification. It can be implemented at each hospital or clinic. The Fog Network Layer consists of two distinct sub-layers:

    - **Fog Server:** The Fog Server serves as the central server responsible for initiating and establishing a shared global model architecture among the participating fog nodes. Its primary role is to add verified model weights to the blockchain, guaranteeing that local model updates are securely and transparently aggregated and shared across the network using the Transport Layer Security (TLS) protocol [141]. Furthermore, the deployment of blockchain technology enables identity management, ensuring that only authenticated devices are allowed to participate in the FL process.
    - **Fog Nodes:** Fog Nodes are distributed entities comprising physical components such as mobile devices, gateways, routers, and switches. In the context of FL, they serve as clients and perform local model training to safeguard sensitive medical data. The closeness of fog nodes to the fog server and the IoMT layer, together with their increased processing capacity, memory, and connection as compared to individual IoMT devices, makes them ideal for local model training. Each fog node is uniquely identified by a DID, which is registered on the blockchain. DIDs play a vital role in various functions such as signing documents or transactions, establishing secure

and persistent communication channels, and facilitating the exchange of encrypted private messages [140].

3. **IoMT Layer:** is responsible for sensing, collecting, encrypting, and uploading medical data to the fog nodes for secure local model training. This layer handles the transmission of various types of data, including benign network traffic as well as potential cyber-attack classification.

   Additionally, each IoMT device is assigned a unique DID that is registered on the blockchain. The DID plays a crucial role in the authentication process, ensuring the secure and authorized access of IoMT devices within the system.

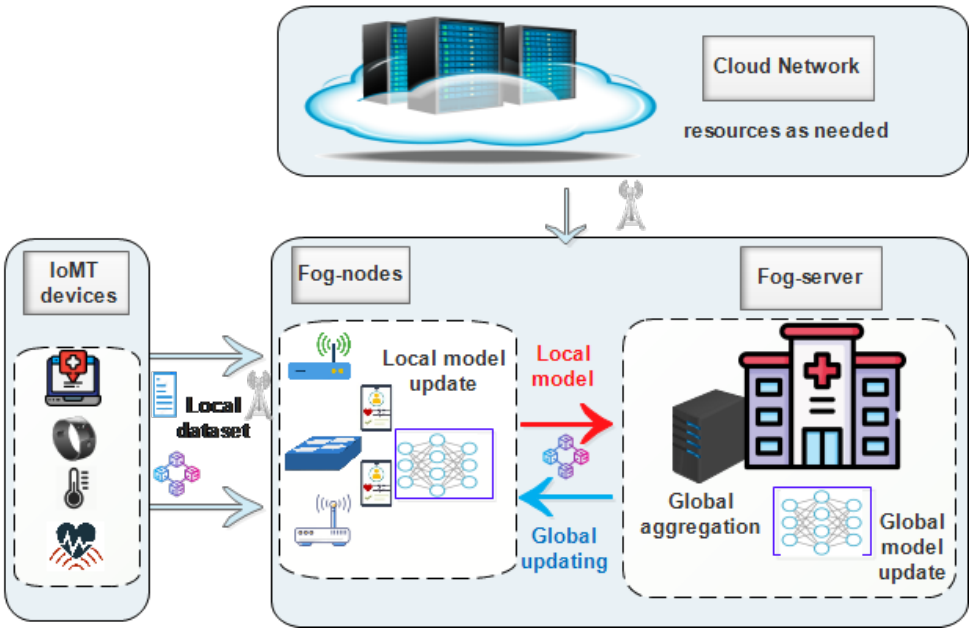


Figure 5.3. *The Proposed Model.*

5.4.2 Challenges and Design Goal

The SA-FLIDS system presents a groundbreaking approach to combating cyber-attacks targeting IoMT devices in smart healthcare systems. It introduces a secure and authenticated FL-based IDS for Smart Healthcare, leveraging Blockchain technology. The prior goal of this system is to analyze network traffic, detect and mitigate potential attacks, and uphold privacy in IoMT-based smart healthcare applications deployed on distributed fog networks. However, it is important to acknowledge that FL systems are

susceptible to adversarial attacks, which pose significant challenges to their security and effectiveness.

1. **Sybil Attack:** This is a type of attack where a malicious actor creates a large number of fake identities to damage the integrity of the FL process [141]. These fabricated identities can manipulate collaborative model training by injecting biased or misleading information into the aggregated model updates.
2. **Eavesdropping Attack:** An eavesdropping attack involves the unauthorized interception and monitoring of communication between participants in the FL process. An attacker who successfully eavesdrops on the communication can gain access to sensitive information, such as model updates or raw data, potentially resulting in privacy breaches [142].
3. **Data Poisoning Attack:** in FL is a malicious attempt to influence the collaborative learning process by injecting adversarial data into the training set of participating devices [142]. The objective of this attack is to compromise the integrity and performance of the global model.

To address the aforementioned challenges, the SA-FLIDS system focuses on the following design goals:

1. **Ensuring the security of federated learning:** The system intends to use strong security mechanisms to preserve the confidentiality and authenticity of the FL process as well as confidentiality. This includes implementing mechanisms to detect and mitigate adversarial attacks, such as Sybil attacks and data poisoning attacks.
2. **Authenticating participating IoMT devices:** The system enforces strict authentication protocols to ensure that only authenticated and authorized IoMT devices can participate in the FL process. This helps avoid inappropriate access and ensures the system's integrity.
3. **Securing communication channels:** The system prioritizes secure communication between IoMT devices, fog nodes, and fog servers. It

employs encryption techniques and protocols to prevent data tampering and eavesdropping during communication between nodes, thereby safeguarding the privacy of sensitive information.

4. **Mitigating the impact of outliers and malicious participants:** The system aims to minimize the influence of outliers and malicious participants during the computation of the global model. By reducing the impact of adversarial participants engaging in poisoning attacks and introducing noisy data, the integrity and accuracy of the global model can be preserved.

5.5 Proposed SA-FLIDS System

In this section, we provide a comprehensive overview of our proposed scheme, SA-FLIDS, designed to address the critical aspects of security and privacy preservation within a smart healthcare system.

5.5.1 Identification and Authentication in SA-FLIDS

The SA-FLIDS model incorporates a robust identification and authentication mechanism, leveraging Blockchain-based DIDs and VCs. This mechanism enhances the participant identification process within the federated learning framework, ensuring secure and authenticated communication among IoMT devices, fog nodes, and a central server.

To participate in the SA-FLIDS system, each fog node, fog server, and IoMT device generates its own unique DID and registers it within the system, as depicted in Figure 5.5. The DIDs are securely stored in the blockchain, serving as a basis for the subsequent authentication process using VCs. This approach guarantees secure and tamper-resistant identification, maintaining data integrity, and preventing unauthorized alterations.

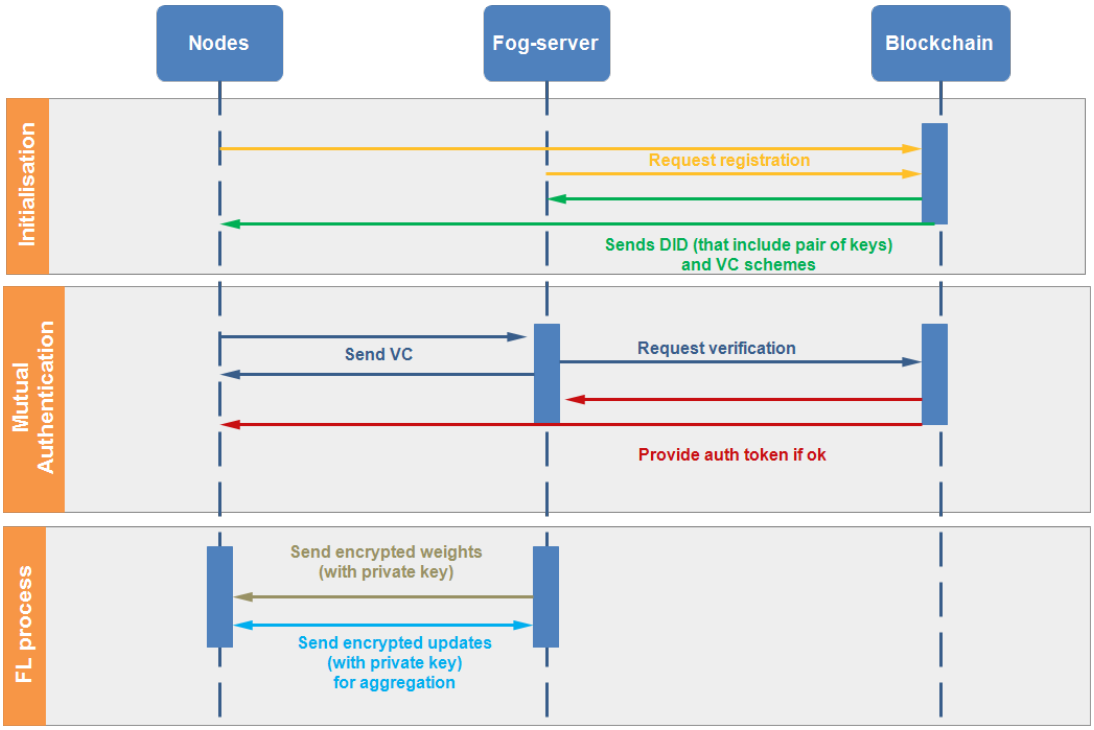


Figure 5.4. *Sequence Diagram of our SA-FLIDS Model.*

Furthermore, each node generates VCs containing essential information such as node name, manufacturing identifier, and commissioning date, based on a predefined schema. These credentials are cryptographically verified by the server against the blockchain as demonstrated in Figure 5.4, ensuring the authenticity and legitimacy of the participating nodes. Nodes that fail the verification process are excluded from the FL process. In addition to robust identification and authentication, the SA-FLIDS system prioritizes secure communication channels. To achieve this, we have implemented the efficient Remote Procedure Call (gRPC) framework for seamless communication between nodes. Furthermore, TLS is enforced to establish end-to-end encryption, bolstering the security and privacy of the communication channels.

5.5.2 Federated Learning Process in SA-FLIDS

Upon successful authentication and verification, the SA-FLIDS system initiates the FL process, which involves the cooperation and collaboration of fog nodes under the supervision of the fog server.

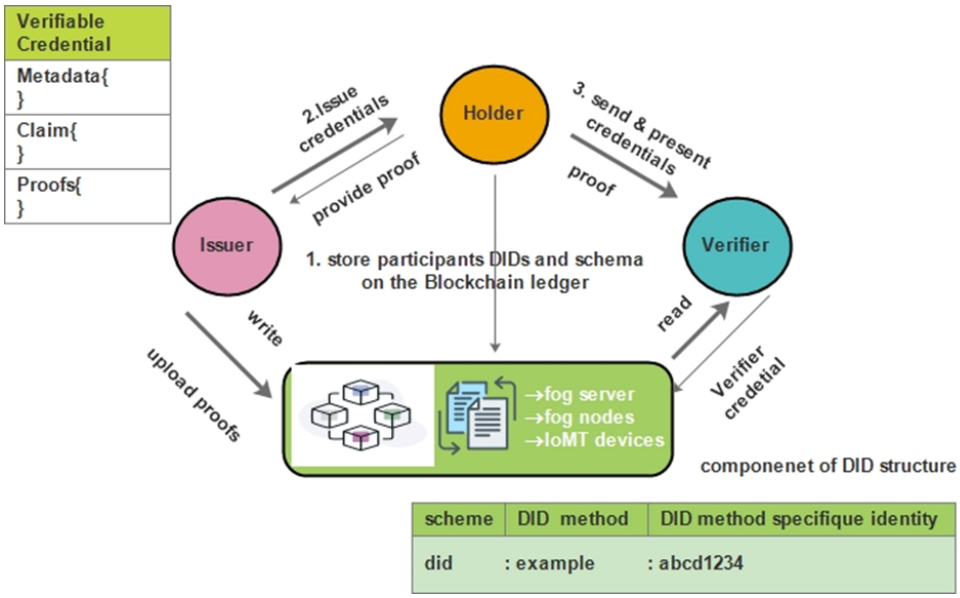


Figure 5.5. *Trust Triangle for VC.*

The fog server establishes a distributed global model architecture across the participating fog nodes. In each training round, every fog node updates its local model by training on the data collected from IoMT devices within its proximity. These local model updates are then transmitted back to the central fog server.

The central fog server collects and aggregates updated models from all fog nodes using a robust aggregation technique, such as trimmed mean aggregation. This aggregation process combines model updates to generate an updated global model. The fog server subsequently distributes the updated global model back to all the fog nodes.

This iterative process continues for each training round until the final global model is obtained, which is then ready for use. It is important to note that all communications during this process occur securely. The gRPC framework is used for efficient communication between nodes, while the TLS protocol is implemented to ensure end-to-end encryption in communication channels, enhancing the confidentiality and integrity of the exchanged data.

The final global model obtained from the FL process is then utilized for the detection and classification of traffic patterns, distinguishing between normal and

potentially malicious patterns. Algorithm 2 provides a demonstration of the FL process within SA-FLIDS.

Algorithm 2 Federated Learning

Initialize global model θ_0
for $t = 1$ to T **do**
    for each fog node i **do**
        Collect local data D_i from IoMT devices
        Train local model θ_i^t on D_i
        Send θ_i^t to server
    end for
    Aggregate local models: $\{\theta_1^t, \theta_2^t, ..., \theta_N^t\}$
    $\theta_{t+1} = \text{TrimmedMean}(\{\theta_1^t, \theta_2^t, ..., \theta_N^t\})$
    for each fog node i **do**
        Distribute θ_{t+1} to fog node i
    end for
end for
Output: Final global model θ_T

5.5.3 Secured FL Integration with the IDS

The SA-FLIDS model incorporates a robust IDS that integrates with the FL framework to provide enhanced detection and mitigation capabilities.

5.5.3.1 Detection and Mitigation model

The IDS, implemented within the fog server, is responsible for not only identifying potential intrusions but also taking immediate and appropriate actions to prevent or minimize the impact of such attacks. To accomplish this, the secured FL model, trained through a robust process, is integrated into the IDS, enriching the fog server's architecture.

The detection and mitigation process involves continuous monitoring of incoming traffic by the fog server. Fresh traffic data is fed into the FL model, which has been previously trained, to differentiate between normal and potentially malicious traffic patterns. If the FL model predicts the incoming traffic as normal, access is granted,

showcasing the model's ability to make real-time decisions based on learned patterns of normal behavior.

However, if the FL model identifies incoming traffic as indicative of an intrusion or attack, it triggers an alarm in the system monitoring, promptly alerting the system to the potential threat. The IDS responds in real time, taking proactive measures to block and drop malicious packets, thus mitigating the impact of the attack.

This seamless integration of the FL model with the IDS enhances The capacity of the system to identify and promptly respond to possible intruders. The IDS leverages the real-time predictions from the FL model, enabling swift and proactive measures to protect the smart healthcare system. Figure 5.7 illustrates the flow of this process.

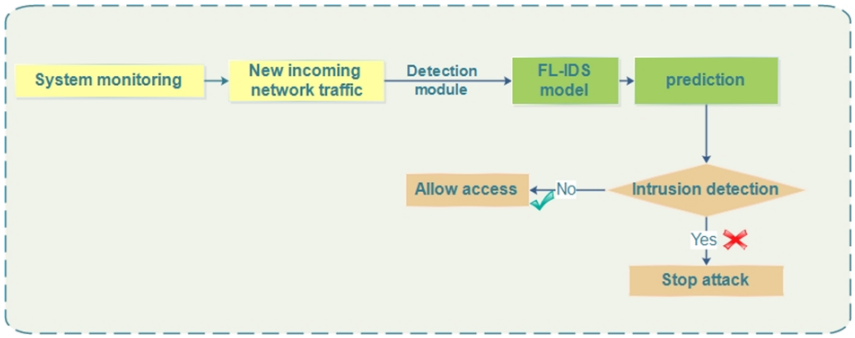


Figure 5.6. *SA-FLIDS Detection and Mitigation Scheme.*

5.5.3.2 Trimmed-Mean Aggregation Method

The Trimmed-Mean aggregation method is a robust approach used to address the influence of outliers and malicious participants during the computation of the global model in FL. By integrating this method with the classification algorithms employed for training, it improves the accuracy and reliability of the global model.

The Trimmed-Mean applies a trimming process, where a certain percentage of extreme values (outliers) are removed from the set of local models. This helps mitigate the impact of outliers and improves the robustness of the aggregation process. The influence function of the Trimmed-Mean is derived coordinate-wise, assuming the

independence of the coordinates. This involves utilizing the quantile function to filter out the influence of outliers.

The Trimmed-Mean aggregation formula calculates the weighted average of the remaining models in the trimmed set. The weight assigned to each local model represents its contribution to the global model. The formula is:

$$\text{Trimmed-Mean} = \frac{\sum_i (w_i \cdot m_i)}{\sum_i w_i} \tag{5.1}$$

where:

w_i represents the weight assigned to the $i-th$ local model.

m_i represents the $i-th$ local model in the trimmed set M_t.

Through the utilization of the Trimmed-Mean aggregation method, the accuracy and reliability of the global model are enhanced, particularly in the presence of outliers and malicious participants. This contributes to improved performance in the FL process.

5.6 Experiments and Results

This study explores the potential of SA-FLIDS in detecting intrusion in IoMT networks. As a result, this decentralized approach could be crucial for securing healthcare applications. In this section, we present the experimental setup along with the evaluation metrics and results.

5.6.1 Experimental Setup

In this section, we delve into the experimental setup for both FL and blockchain-based SSI.

5.6.1.1 Experimental setup for federate learning

Table 5.1 presents the parameters used in federated deep learning. In our study, we conducted experiments deploying our model with client sets denoted as K, where K = 10. We employed the Independent and Identically Distributed (IID) approach, ensuring that the data distribution across the dataset matches the distribution of data for each client. Furthermore, To mitigate overfitting, the following techniques were used:

- **Stratified K-fold Cross-Validation:** Set k=5 to split the data into five subsets, evaluating the model's performance.
- **L2 Regularization:** Applied with a factor of 0.01 to the Dense layers, adding a penalty to the loss function based on the weights' magnitude, simplifying the model.
- **Early Stopping:** With the patience of 3, training stops if the validation loss does not improve, monitoring the validation set's performance.

Table 5.1. *Federated Deep Learning Classifier Parameter Setting*

	Parameter	Value
Federated Deep Learning classifier	Local epoch	10
	Global epoch	4
	Batch size	128
	Hidden layer	2
	Hidden nodes	128,64
	Activation function	Relu
	Regularization	L2
	Classification function	Sigmoid/ softmax
	Optimizer, learning rate	Adam, 0.001
	Loss function	Binary_crossentropy Categorical_crossentropy

5.6.1.2 Experimental setup for blockchain-based SSI

In the simulation environment, client authentication within the blockchain-based SSI system was implemented using the DIDs and VCs model, leveraging Hyperledger Indy [143] and Hyperledger Aries.[144] Hyperledger Indy, tailored for Identity management, provided the distributed ledger implementation, while Hyperledger Aries facilitated connectivity between Indy and front-end applications.

The setup involved a device running the Ubuntu 18.04 LTS operating system. Several essential software components were installed, including Von-Network for the blockchain ledger, Docker for containerization, and Docker-Compose for managing multi-container Docker applications as shown in Figure 5.7.

Containerization was utilized to host the fog server, fog node, and Aca-py Agent, all integrated into Docker containers to enable blockchain communication. The Aca-py Agent is based on the Hyperledger Aries Cloud Agent Python (ACA-Py) library and offers core functionalities such as secure storage management and message exchange.

To ensure secure communication among nodes, the gRPC protocol was employed, guaranteeing confidentiality and integrity.

In terms of networking configuration, the deployment comprised 10 Docker containers, each representing a device communicating with the fog server. Every container hosted von-network nodes, while separate containers were assigned to each agent, ensuring individualized functionalities within the system

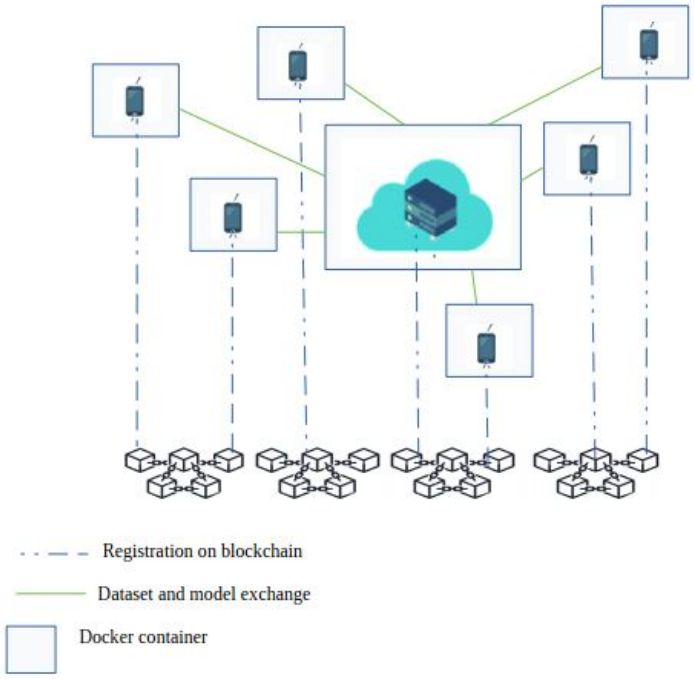


Figure 5.7. *Blockchain Network.*

5.6.1.3 Datasets description

Datasets play an essential role in both training and assessing IDSs within IoT networks. The choice of suitable datasets tailored to particular tasks holds significant importance, particularly in evaluating the efficacy of FL approaches for IoT networks. In our experiment, we incorporated two recent datasets specifically designed to mimic real-world conditions for IDSs: CICIoT2023 [145], made available in 2023, and the Edge-IIoTset dataset, released in 2022.

1. ***CICIoT2023 dataset:*** is a [145] novel and extensive IoT attack dataset to foster the development of security analytics applications in real IoT operations. To accomplish this, 33 attacks are executed in an IoT topology composed of 105 devices, and all attacks are executed by malicious IoT devices targeting other IoT devices. We analyzed a dataset containing 47 features (not including label and sublabel) based on 2,366,956 samples extracted from the first 10 CSV files provided by the Canadian Institute.
2. ***Edge-IIoTset dataset:*** It's tailored specifically for IIoT and IoT applications [143], providing an authentic test environment closely resembling real-world IoT/IIoT settings. Within this environment, we conducted simulations of genuine cyberattacks to collect datasets comprising both legitimate and malicious network traffic. This dataset includes data generated by various IoT devices, spanning from heart rate sensors to flame sensors, temperature, and humidity sensors. The testbed is structured into seven interconnected layers. We utilized the Selected dataset for ML and DL/DNN-EdgeIIoT-dataset CSV file[143], which contains 61 features and 2,219,201 samples, encompassing both normal traffic and 14 distinct attacks in the IoT and IIoT environments.

5.6.1.4 Preprocessing

The datasets undergo several preprocessing steps to ensure their suitability for analysis as demonstrated in Figure 5.8. After cleaning the data we first, address imbalanced data by implementing synthetic minority over-sampling and under-sampling techniques to enhance predictive performance, particularly for minority classes. Secondly, data transformation is conducted using the StandardScaler for standardization, adjusting data to have a mean of 0 and a standard deviation of 1. Additionally, feature importance analysis is performed using insights from random forest and XGBoost experiments. Finally, the processed dataset is split into an 80% training set and a 20% testing set, ensuring no duplication between the two, contributing to refining the dataset for subsequent analysis and modeling tasks. In

the case of the CICIoT dataset, we opt to eliminate the Brute and Web attack labels due to their limited number of samples, which could potentially skew the analysis and compromise the reliability of the results. The detailed features selected and the attacks used are outlined in Table 5.2 provided below.

Table 5.2. *Datasets Description for Experimental Evaluation.*

	CICIoT2023	**Edge-IIoTset**
Features selected	'flow_duration', 'Header_Length', 'Protocol Type', 'Rate', 'Srate', 'syn_count','urg_count', 'rst_count', 'Tot sum', 'Min', 'Max', 'AVG','Tot size', 'IAT', 'Magnitude', 'Variance'	'http.content_length', 'http.request.method', 'http.referer', 'http.request.version', 'tcp.ack', 'tcp.ack_raw', 'tcp.checksum', 'tcp.flags', 'tcp.len', 'tcp.seq', 'udp.time_delta', 'dns.qry.name.len', 'mqtt.conack.flags', 'mqtt.protoname', 'mqtt.topic'
Label	'Benign','DDoS', 'DoS', 'Mirai', 'Recon','Spoofing'	'DoS/DDoS', 'Information gathering', 'Injection', 'Malware', 'Man in the middle', 'Normal'

5.6.2 Evaluation Metrics

In this section, we introduce the metrics employed in our experiments to evaluate both FL and SSI-based DID.

5.6.2.1 Metrics used for federated learning evaluation

When evaluating the performance of intrusion detection using federated deep learning, we employed the most widely adopted metrics. These evaluation metrics are comprehensively described in Chapter 4 (Section 4.4.3), and they include True Negatives (TN), True Positives (TP), False Positives (FP), False Negatives (FN), accuracy, precision, recall, and F1-score. Utilizing these standard metrics allowed us

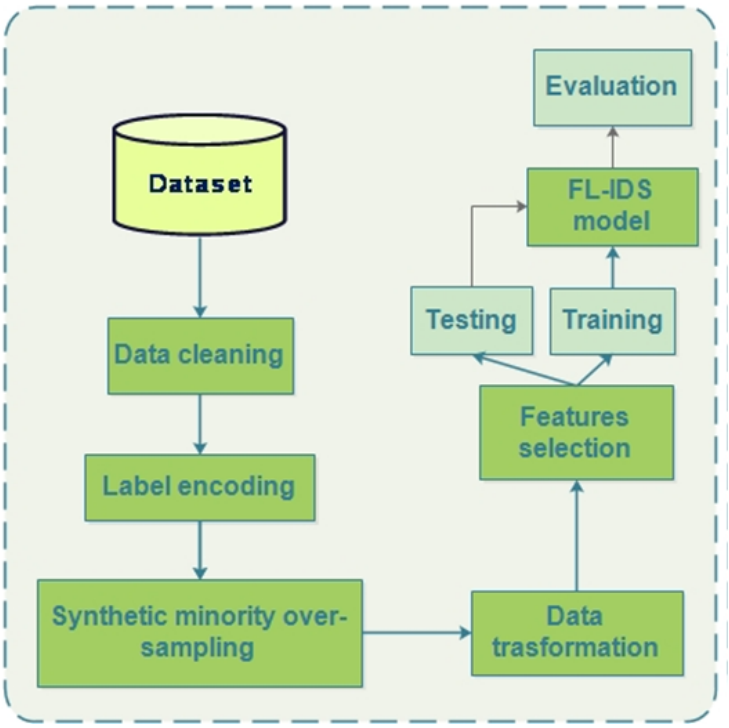


Figure 5.8. *Data Preprocessing*

to conduct a systematic comparative analysis of our proposed model's performance against other relevant approaches in the field.

5.6.2.2 Metrics used for Blockchain-based SSI Evaluation

To compute the metrics outlined below, we utilize the following formulas:

1. **Startup Duration (SD)**: The duration for the system to initiate.
2. **Connect Duration (CD)**: The time required for the system to establish connections between nodes and the Fog server.
3. **Publish Duration (PD)**: The duration for the system to publish schema credentials and related settings.
4. **Issuing Credential Duration (ICD)**: The time taken for the system to issue credentials.
5. **Completed Credential Exchanges Duration (CCED)**: The total time needed for all credential exchanges to conclude.

6. **Average Time per Credential Duration (ATCD)**: The average time taken to issue a single credential.

$$\text{ATCD} = \frac{ICD}{N_{\text{credentials}}} \tag{5.2}$$

Where:

ICD is the Issuing Credential Duration.

$N_{\text{credentials}}$ is the total number of credentials issued.

7. **Average Time per Transaction Duration (ATTD)**: The average time taken per transaction.

$$\text{ATTD} = \frac{\sum_{i=1}^{N_{\text{transactions}}} T_i}{N_{\text{transactions}}} \tag{5.3}$$

Where:

$N_{\text{transactions}}$ is the total number of transactions.

T_i is the duration of each transaction i.

5.6.3 Evaluation Results

We utilize federated deep learning-based NIDS models to detect cyber-attacks in IoMT environments, specifically focusing on the networks of healthcare applications. Our training incorporates the most recent datasets for IDS, including CICIoT2023 and the Edge-IIoTset dataset. We conduct experiments employing binary and multi-class classification techniques for each dataset.

5.6.3.1 Binary classification

In this subsection, we present the evaluation results for binary classification scenarios using FL. In addition, we provide an evaluation of blockchain-based SSI.

1. **Federated learning Evaluation Results** We employed 150,000 samples for both benign and attack instances in both datasets, ensuring a balanced dataset for a comprehensive and meaningful comparison. Remarkably, our model demonstrates impeccable performance, achieving perfect scores of 100% across all metrics for the Edge-IIoT dataset. In contrast, the results for the CICIoT2023 dataset remain highly promising, with an accuracy of 99.09%, indicating a low error rate in classifying both benign and malicious traffic. Furthermore, achieving a perfect precision of 100%, along with a recall of 98% and an F1-score of 99%, underscores the robust overall performance of the model, as shown in Figure 5.9. The classification performance of our model is depicted through the confusion

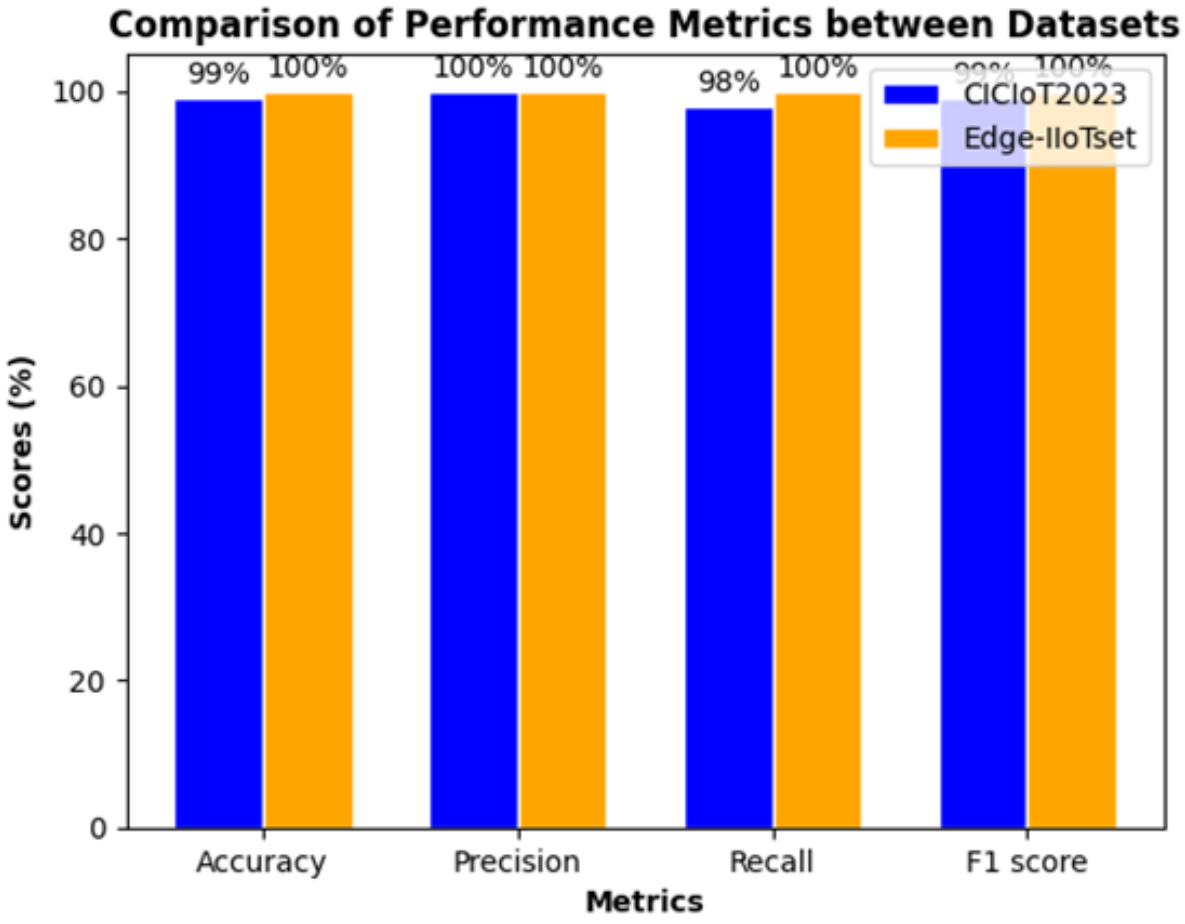
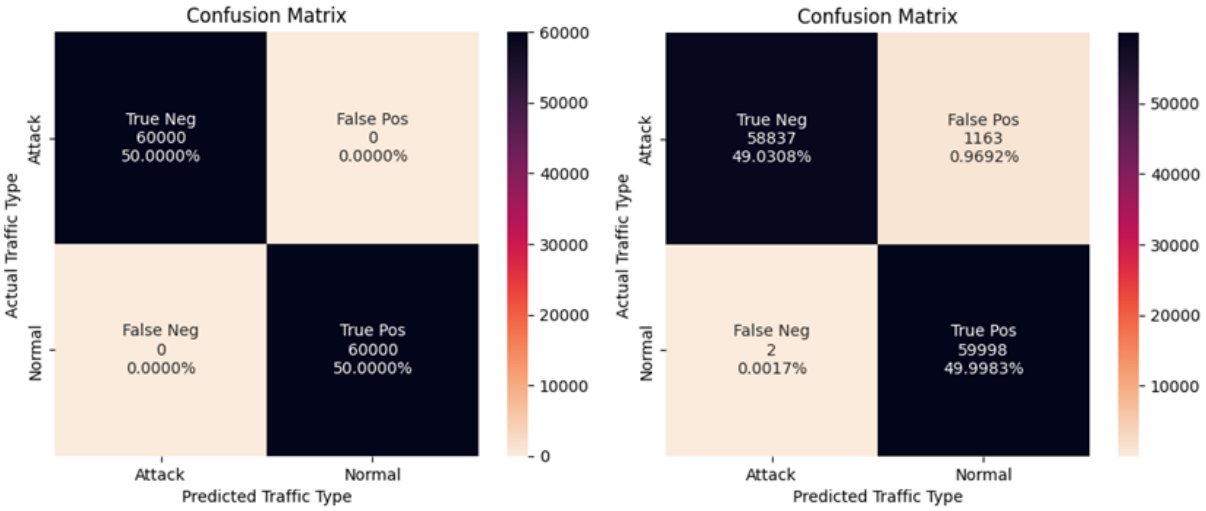


Figure 5.9. *Evaluation of Performance in Binary classification.*

matrix presented in Figure 5.10, providing a concise summary of the model's accurate and erroneous predictions The primary goal is to minimize both false positive and false negative rates, ensuring precise classification outcomes. Our

proposed model effectively achieves this objective, exhibiting false positive and negative rates of 0% in the Edge-IIoTset dataset. For the CICIoT2023 dataset, we observe a negligible false negative rate of 0.0017%, alongside false positive rates of 0. 96%, which confirms the accuracy and efficiency of the model in mitigating classification errors.



(a) *EdgeIIoTset dataset.* **(b)** *CICIoT2023 dataset.*

Figure 5.10. *Confusion Matrix in Binary Classification.*

2. **SSI-based DID Evaluation Results** As illustrated in Figure 5.11 below both datasets exhibit similar startup durations (SD), with EdgeIIoTset demonstrating a marginally quicker performance by 0.01 seconds. Furthermore, the EdgeIIoTset dataset shows a shorter connect duration (CD) of 0.02 seconds compared to the CICIoT2023 dataset. Both datasets share the same publish duration (PD) of 9.15 seconds. However, in terms of issuing credential duration (ICD), the EdgeIIoTset dataset outperforms the CICIoT2023 dataset by 0.87 seconds. Regarding completed credential exchange duration (CCED), the EdgeIIoTset dataset exhibits a reduction of 8.9 seconds compared to the CICIoT2023 dataset. Additionally, the average time per credential duration (ATCD) is shorter for the 'EdgeIIoTset' dataset in comparison to the 'CICIoT2023' dataset. Finally, the average time per transaction duration (ATTD) is marginally higher for the 'EdgeIIoTset' dataset when contrasted with the 'CICIoT2023' dataset.

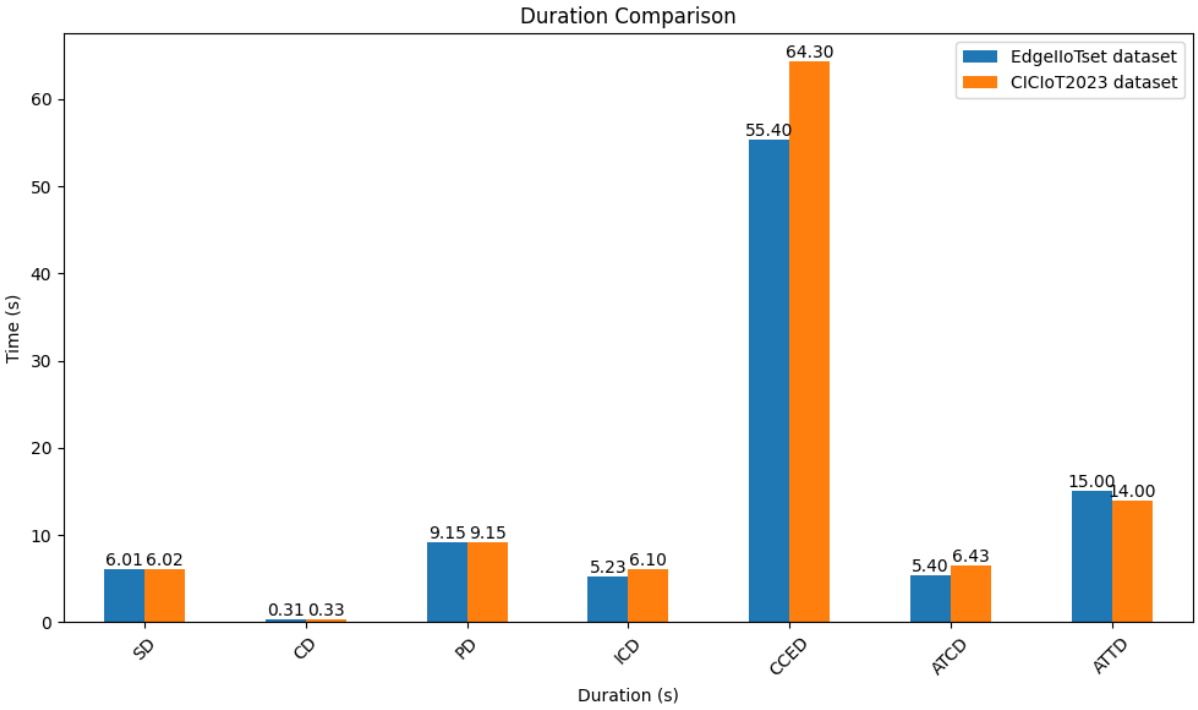


Figure 5.11. *Comparative Performance Analysis of Blockchain-based SSI for Binary Classification.*

5.6.3.2 Multiclass classification

In this subsection, we present the evaluation results for both the CICIoT2023 and EdgeIIoTset datasets in multiclass classification scenarios using FL. Additionally, we provide an evaluation of SSI-based DID.

1. **Federated learning Evaluation Results** The evaluation of both the CICIoT2023 and Edge-IIoTset datasets reveals strong performance across diverse classes, as demonstrated in Table 5.3. Within the CICIoT2023 dataset, each class achieves good performance in most metrics. Notably, the DDoS, DoS, and Mirai attack classes in the CICIoT2023 dataset exhibit great classification capabilities, demonstrating perfect performance across all metrics with complete precision, recall, and F1-scores of 100%. Furthermore, the Spoofing, Benign, and Information Gathering classes show good precision with 87%, 83%, and 81%, respectively. However, their recall and F1 scores vary. The Benign class achieves a high recall of 85% and an F1-score of 84%. The Information Gathering class

presents a relatively good recall of 87% and a corresponding F1-score of 84%. The Spoofing class achieves a moderate recall at 79% and an F1-score of 83%.

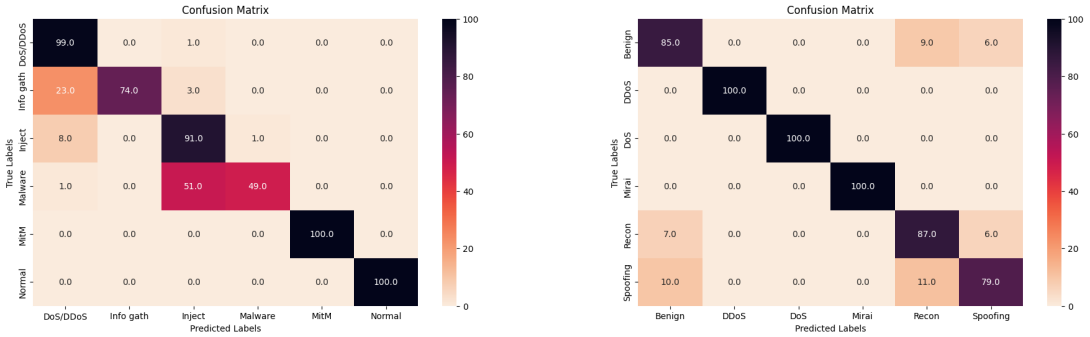
Transitioning to the Edge-IIoTset dataset, Man in the Middle, and Normal classes also exhibit high performance, achieving perfect precision, recall, and F1-scores of 100%. followed by the DoS/DDoS class with high precision 86% and very high recall 99%, resulting in a strong F1-score 92%. Furthermore, other classes show more variability. The Information Gathering class has an excellent precision of 99%, but its recall is lower at 74%, resulting in an F1-score of 84%. The Injection class shows a precision of 75% and a high recall of 91%, leading to an F1-score of 82%. The Malware Attacks class, despite its high precision of 96%, suffers from a low recall of 49%, resulting in a lower F1-score of 65%.

Table 5.3. *Classification Report.*

Dataset	Class	Precision	Recall	F1-score
CICIoT2023	Benign	83%	85%	84%
	DDoS	100%	100%	100%
	DoS	100%	100%	100%
	Mirai	100%	100%	100%
	Information gathering	81%	87%	84%
	Spoofing	87%	79%	83%
Edge-IIoTset	DoS/DDoS	86%	99%	92%
	Information gathering	99%	74%	84%
	Injection	75%	91%	82%
	Malware attacks	96%	49%	65%
	Man in the Middle	100%	100%	100%
	Normal	100%	100%	100%

The confusion matrix depicted in Figure 5.12 provides valuable insights into the prediction frequency for each class compared to their actual occurrences. Within the CICIoT dataset, remarkable performance was observed for classes such as DoS, DDoS, and Mirai, with the model accurately predicting all samples, achieving a flawless prediction rate of 100%. Additionally, the classification of Reconnaissance exhibited high accuracy, with 87% of samples correctly classified followed by Begnin traffic with 85%. However, in subsequent classes such

as Spoofing, accuracy decreased, with only 79% respectively, of the samples accurately classified. Likewise, within the Edge-IIoTset dataset, classes like DoS/DDoS, Man in the Middle, and Normal traffic demonstrated robust performance, as the model accurately predicts all samples, resulting in a 99% and 100% prediction rate. The classification of the Injection class followed suit, with 91% of samples correctly classified. Nevertheless, as we delve into subsequent classes such as Information Gathering and Malware, declined, with only 74% and 49%, respectively, of samples accurately classified.



(a) *EdgeIIoTset dataset.* **(b)** *CICIoT2023 dataset.*

Figure 5.12. *Confusion Matrix in Multiclass Classification.*

2. **SSI-based DID Evaluation Results** As illustrated in Figure 5.13 both datasets exhibit similar startup durations (SD), with EdgeIIoTset demonstrating a slight advantage of 0.08 seconds. The CICIoT2023 dataset shows a shorter connect duration (CD) by 0.01 seconds compared to EdgeIIoTset. Additionally, both datasets share the same publish duration (PD) of 9.15 seconds. However, the CICIoT2023 dataset boasts a shorter issuing credential duration (ICD) of 0.29 seconds compared to EdgeIIoTset. Regarding the completion of the credential exchange duration (CCED), the CICIoT2023 dataset surpasses EdgeIIoTset by 5 seconds. Furthermore, the CICIoT2023 dataset demonstrates a shorter average time per credential duration (ATCD) by 0.36 seconds compared to EdgeIIoTset. Lastly, the CICIoT2023 dataset shows a shorter average time per transaction duration (ATTD) by 3 seconds compared to the EdgeIIoTset.

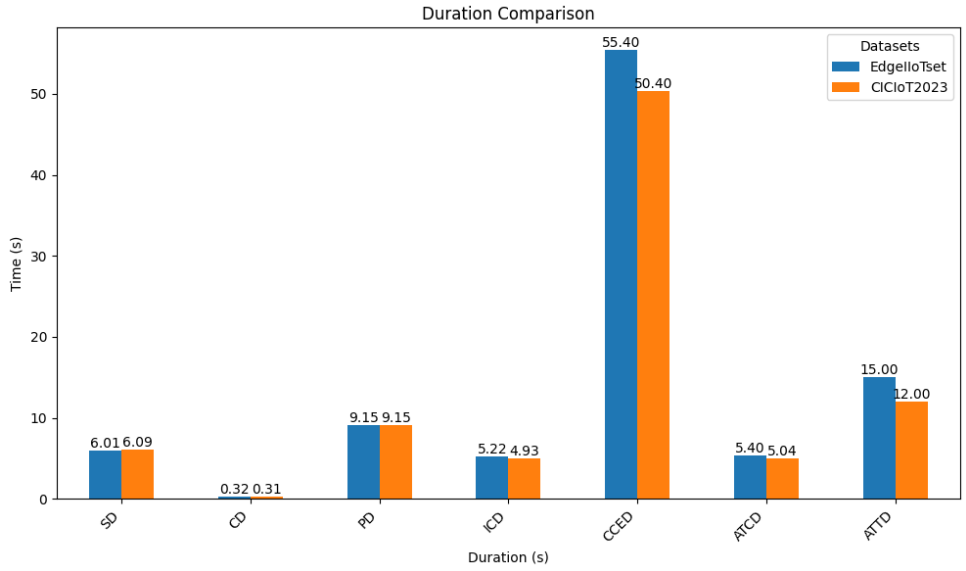


Figure 5.13. *Comparative Performance Analysis of Blockchain-based SSI for Multiclass Classification.*

5.7 Security and Privacy Analyses

Our study focuses on identifying potential threats targeting smart healthcare systems facilitated by IoMT networks. The objective is to mitigate the risks associated with unauthorized access to sensitive patient data via malicious IoMT devices. Consequently, this section conducts a comprehensive examination of the privacy and security capabilities of the SA-FLIDS model. It is crucial to recognize that lacking adequate security measures may deter individuals from engaging in healthcare applications, thereby impeding the success of these technological advancements. Our SA-FLIDS framework tackles these concerns by implementing robust security protocols, instilling user confidence, and fostering widespread adoption and sustainability of healthcare technologies. Furthermore, the analysis procedure is underpinned by a theoretical exploration of SA-FLIDS's resilience against potential threats outlined in the adversary model (Section 5.2).

Table 5.4. *A Comprehensive Comparison between existing Works and our Model.*

Models	FL	IDS	Environment	Security Techniques of FL			
				Against adversarial	secure communication	Blockchain	SSI Authentication
Schneble and Thamilarasu 2019 [146]	✓	✓	Medical CPS	X	X	X	X
Chatterjee and Hanawal 2021 [147]	✓	✓	IoT Network	X	X	X	X
Man, Zeng, and Yang 2021 [148]	✓	✓	IoT Network	X	X	X	X
Rey, S'anchez, and Celdr'an 2022 [149]	✓	X	IoT Network	✓	X	✓	X
Ruzafa-Alcazar, Fernandez-Saura, and Marmol-Campos 2021 [150]	✓	✓	Industrial IoT Network	✓	✓	X	X
Zhao, Chen, and Wu D. 2019 [151]	✓	X	General purpose	X	X	X	X
Friha, Ferrag, and Shu 2022 [152]	✓	✓	Agriculture IoT	X	✓	✓	X
Ashraf et al. 2022 [153]	✓	✓	Healthcare IoT Network	✓	X	✓	X
Preuveneers et al. 2018 [154]	✓	✓	Industrial IoT Network	✓	X	✓	X
Lakhan et al. 2022 [155]	✓	X	Healthcare IoT Network	X	X	✓	X
OUR MODEL	✓	✓	Healthcare IoT Network	✓	✓	✓	✓

Table 5.4 presents an extensive comparative analysis between existing systems and our proposed model. This analysis scrutinizes the security provisions offered by the SA-FLIDS model in contrast to other FL-based IDS. Moreover, we juxtapose the SA-FLIDS system within the broader scope of security considerations in FL. Notably, the SA-FLIDS model elevates this paradigm by integrating additional layers of security, including user authentication and secure communication channels throughout the FL process, facilitated by a dependable aggregation technique. Furthermore, our comprehensive approach enhances the safeguarding of sensitive data and distinguishes itself as the sole scheme integrating SSI for user authentication in the FL process. This feature fortifies the system's security stance by ensuring authorized access and deterring unauthorized participation.

Conclusion and Future Directions

The rapid advancements in technology have greatly improved people's lives by simplifying tasks and resolving complex issues. One such popular technology is the Internet of Things (IoT), which has gained widespread adoption and enables real-time data collection while establishing seamless connections between devices and users. To effectively utilize the potential of IoT, it is crucial to select appropriate architectures for data storage, processing, and analytics, such as fog computing. However, it is important to acknowledge that certain drawbacks accompany these technologies, including concerns regarding privacy and susceptibility to various attacks and threats.

Therefore, this thesis focuses on the theoretical objectives of conducting a comprehensive examination of the security and privacy concerns, threats, and existing solutions and techniques in fog computing for IoT. Additionally, the technical objective is to develop defensive systems that can safeguard the network and data from cyber attacks. In order to preserve user privacy, this research aims to explore the application of emerging technologies such as Software-Defined Networking (SDN), Network Intrusion Detection Systems (NIDS), Adaptive Neuro-Fuzzy Inference Systems (ANFIS), federated learning, blockchain, and the concept of Self-Sovereign Identity (SSI).

To offer an in-depth analysis of the proposed methodologies, this thesis presents two main contributions.

The first contribution of this thesis introduces an innovative approach called FASA, designed to address TCP-SYN Flood DDoS attacks specifically within fog computing environments. These attacks, often orchestrated through botnets, overwhelm servers by inundating them with incomplete connection requests, causing disruptions for legitimate users. FASA leverages the power of an adaptive neuro-fuzzy inference system (ANFIS) to accurately classify network traffic in real-time, coupled with the capabilities of Software Defined Networking (SDN) for efficient mitigation. By training the model on a recent dataset (CICDDoS2019), FASA demonstrates exceptional accuracy and minimal false positives in distinguishing normal and malicious packets. Consequently, this framework ensures the security and reliability of the SDN controller, thereby safeguarding the availability of fog services.

The second contribution of this thesis presents a proposal called SA-FLIDS, which stands for Secure and Authenticated Federated Learning-based Network Intrusion Detection System in Fog-IoT-enabled Smart Healthcare Systems. This research focuses on identifying and mitigating cyber attacks on the Internet of Medical Things (IoMT) by utilizing the potential of fog computing. The aim is to enhance data privacy preservation, minimize communication overhead, and address vulnerabilities inherent in decentralized learning approaches. To achieve this, SA-FLIDS incorporates a blockchain-based Self-Sovereign Identity (SSI) model for client authentication. Through comprehensive performance evaluations, our study demonstrates that SA-FLIDS effectively detects attacks within the IoMT network while ensuring privacy preservation, scalability, and sustainability requirements are met.

In the future, our focus on mitigating security and privacy concerns within the domain of fog computing for IoT extends beyond the aspects covered in the present thesis. This section presents an overview of two prospective projects that we aim to pursue in the future.

1. *Integration of Solutions in IoT Ecosystem Projects:* Our first prospective project involves collaborating with IoT ecosystem projects to integrate our

developed frameworks, such as FASA and SA-FLIDS, into their infrastructure. By partnering with industry stakeholders and deploying our solutions in operational environments, we can evaluate their performance, scalability, and effectiveness under real-world conditions. This real-world deployment will provide valuable insights into the practical challenges and opportunities for enhancing security and privacy in IoT ecosystems.

2. *Addressing specific application domains:* Consider tailoring our future work towards a specific application domain of fog-IoT, such as smart cities, industrial automation, or autonomous vehicles. Analyze the unique security and privacy challenges in these domains and propose solutions building upon our existing contributions.

Publications

Back

Research Article | Open Access

Toward a Real-Time TCP SYN Flood DDoS Mitigation Using Adaptive Neuro-Fuzzy Classifier and SDN Assistance in Fog Computing

Radjaa Bensaid, Nabila Labraoui, Ado Adamou Abba Ari, Leandros Maglaras, Hafida Saidi, Ahmed Mahmoud Abdu Lwahhab, Sihem Benfriha

First published: 23 February 2024
https://doi.org/10.1155/2024/6651584

Academic Editor: Andrea Michienzi

Abstract

The growth of the Internet of Things (IoT) has recently impacted our daily lives in many ways. As a result, a massive volume of data are generated and need to be processed in a short period of time. Therefore, a combination of computing models such as cloud computing is necessary. The main disadvantage of the cloud platform is its high latency due to the centralized mainframe. Fortunately, a distributed paradigm known as fog computing has emerged to overcome this problem, offering cloud services with low latency and high-access bandwidth to support many IoT application scenarios. However, attacks against fog servers can take many forms, such as distributed denial of service (DDoS) attacks that severely affect the reliability and availability of fog services. To address these challenges, we propose mitigation of fog computing-based SYN Flood DDoS attacks using an adaptive neuro-fuzzy inference system (ANFIS) and software defined networking (SDN) assistance (FASA). The simulation results show that the FASA system outperforms other algorithms in terms of accuracy, precision, recall, and *F*1-score. This shows how crucial our system is for detecting and mitigating TCP-SYN floods and DDoS attacks.

1. Introduction

The growing number of connected objects, from millions to billions in various fields, is leading to an explosion in the amount of data. These huge volumes of data cause a lack of latency and make real-time analysis complex and difficult. To solve these issues, the deployment of computing models such as cloud and fog computing is crucial [1]. Technologies of cloud computing enable an extremely powerful computer resource over the network. Nevertheless, due to several concerns about data privacy and security, attaching more diverse types of objects immediately to the cloud is extremely difficult, as are network

Received: 1 March 2023 | Revised: 7 September 2023 | Accepted: 31 October 2023

DOI: 10.1049/ntw2.12108

IET Networks

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

FUBA: A fuzzy-based unmanned aerial vehicle behaviour analytics for trust management in flying ad-hoc networks

Sihem Benfriha[1] | **Nabila Labraoui**[2] | **Radjaa Bensaid**[1] | **Haythem Bany Salameh**[3,4,5] | **Hafida Saidi**[1]

[1]STIC Laboratory, University of Abou Bekr Belkaid, Chetouane, Tlemcen, Algeria

[2]LRIT Laboratory, University of Abou Bekr Belkaid, Chetouane, Tlemcen, Algeria

[3]Artificial Intelligence Research Center, Al Ain University, Al Ain, UAE

[4]Telecommunication Engineering Department, Yarmouk University, Irbid, Jordan

[5]College of Engineering, Staffordshire University, Stoke-in City, UK

Correspondence

Sihem Benfriha.

Email: benfriha.sihem@univ-tlemcen.dz

Abstract

Flying Ad-Hoc Network (FANET) is a promising ad hoc networking paradigm that can offer new added value services in military and civilian applications. Typically, it incorporates a group of Unmanned Aerial Vehicles (UAVs), known as drones that collaborate and cooperate to accomplish several missions without human intervention. However, UAV communications are prone to various attacks and detecting malicious nodes is essential for efficient FANET operation. Trust management is an effective method that plays a significant role in the prediction and recognition of intrusions in FANETs. Specifically, evaluating node behaviour remains an important issue in this domain. For this purpose, the authors suggest using fuzzy logic, one of the most commonly used methods for trust computation, which classifies nodes based on multiple criteria to handle complex environments. In addition, the Received Signal Strength Indication (RSSI) is an important parameter that can be used in fuzzy logic to evaluate a drone's behaviour. However, in outdoor flying networks, the RSSI can be seriously influenced by the humidity of the air, which can dramatically impact the accuracy of the trust results. FUBA, a fuzzy-based UAV behaviour analytics is presented for trust management in FANETs. By considering humidity as a new parameter, FUBA can identify insider threats and increase the overall network's trustworthiness under bad weather conditions. It is capable of performing well in outdoor flying networks. The simulation results indicate that the proposed model significantly outperforms FNDN and UNION in terms of the average end-to-end delay and the false positive ratio.

KEYWORDS

computer network security, fuzzy logic, mobile ad hoc networks

1 | INTRODUCTION

Our world has changed and is still evolving due to rapidly developing technology in sensors, communications, and networking over the past few decades [1]. Unmanned Aerial Vehicles (UAVs) have been proposed for a multitude of applications in both military and civilian domains, encompassing ad hoc networks, search and rescue missions, electronic operations in hostile zones, ground target identification and tracking, automated forest fire surveillance, wind energy generation [2], and a host of other possibilities. Furthermore, flying ad hoc networks (FANETs), a revolutionary concept, comprise a group of UAVs that cooperate to perform some crucial missions [3]. However, many cyberattacks against UAVs have emerged since 2007 [4], and their impact can be dangerous with divesting effects. Therefore, it is essential to protect FANETs from insider and outsider attacks. In FANETs, drones can leave and rejoin the network anytime, creating an opportunity for attackers to compromise a node and impersonate a legitimate one, leading to insider attacks. Insiders use their trusted access to carry out illicit actions. As a result, they are undetectable by external network security

2023 10th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)

2023 10th International Conference on Internet of Things: Systems, Management and Security (IOTSMS) | 979-8-3503-9389-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/IOTSMS59855.2023.10325826

Federated Deep Learning-based Intrusion Detection Approach for Enhancing Privacy in Fog-IoT Networks

Bensaid Radjaa[1], Labraoui Nabila[2], Haythem Bany Salameh[3]
[1]STIC Laboratory, Abou Bekr Belkaid University, Tlemcen, Algeria
[2]LRI laboratory, Abou Bekr Belkaid University, Tlemcen, Algeria
[3]College of Engineering, Al Ain University, Al Ain, UAE

***Abstract*—The Internet of Things (IoT) revolution has led to a proliferation of connected devices. However, these IoT devices face inherent limitations, such as limited computing power, storage capacity, and battery life. This makes them susceptible to misuse and exploitation. Attackers exploit these vulnerabilities to compromise IoT devices and create botnets that threaten fog-IoT networks. Therefore, developing effective cyber-attack detection mechanisms such as Machine Learning (ML) based Intrusion Detection Systems (IDSs) becomes crucial, which is imperative to safeguard fog-IoT infrastructures. However, conventional ML approaches often require centralized data storage on a single server or in the cloud, leading to concerns regarding data confidentiality, communication overhead, and energy consumption. This paper addresses this issue by leveraging IDS-based anomaly detection to prevent cyber attacks on IoT networks. Specifically, we propose using Federated Deep Learning (FDL) across a fog-based IDS architecture that utilizes the Lost Short-Term Memory (LSTM) model and the Bot-IoT dataset. Our solution adopts a local learning approach, allowing devices to acquire knowledge from others by sharing only model updates without exposing their data. By adopting the FDL approach, the detection model demonstrates a comparable (slightly improved) performance compared to existing centralized deep learning while ensuring data privacy-preserving.**

***Index Terms*—Fog-IoT network, IDS-based anomaly detection, Federated Deep Learning.**

I. Introduction

The rapid growth of wireless communication technology has accelerated the massive deployment of the Internet of Things (IoT), resulting in a vast number of interconnected physical devices and sensors forming IoT applications [1]. This network enables seamless information sharing with cloud or fog computing systems over the Internet, improving data exchange and connectivity [2]. Fog computing involves a diverse range of interconnected intelligent objects capable of sensing, collecting, and processing data for tasks such as analysis, control, monitoring, and real-time decision-making [3]. However, integrating Fog computing with IoT raises security and privacy concerns due to IoT device resource limitations within the fog network. Consequently, these vulnerabilities expose the network to attacks, including hacking techniques that compromise sensitive data or disrupt fog services [4]. Implementing robust security measures becomes imperative to address these security concerns. In this regard, Intrusion Detection Systems (IDS) play a crucial role in mitigating these risks by actively identifying and stopping potential attacks, thereby enhancing the overall security of fog computing environments [5]. Moreover, there are two primary types of IDS: signature-based and anomaly-based. Signature-based IDSs rely on predefined attack patterns or signatures. However, a main limitation of these methods is their inability to detect zero-day attacks, which exploit new or previously unknown vulnerabilities [6]. On the other hand, anomaly-based methods leverage Machine Learning (ML) techniques, including Deep Learning (DL), to create a model by analyzing the usual behavior and characteristics of the system, identifying deviations as abnormalities [7]. However, conventional ML approaches often require centralized data storage on a single server or in the cloud, leading to concerns regarding data confidentiality, communication overhead, and energy consumption [8]. Federated Learning (FL) is a promising technique to address these limitations, enabling knowledge exchange while preserving privacy and reducing expenses [9]. By empowering devices to train a unified model collaboratively while keeping training data on the device, FL separates machine learning capabilities from the need for centralized data storage [10]. As a result, this technique can significantly mitigate privacy and security risks. This paper proposes a federated deep learning-based Intrusion Detection System (FDL-IDS) for Fog-IoT Network (FIN), facilitating collaboration across edge devices to securely exchange data and deliver robust attack detection in IoT-based smart city applications. Furthermore, the proposed detection system detects anomalies by classifying the network traffic as benign or malicious. It attempts to achieve high accuracy, low false positives, and low communication costs while remaining flexible and scalable for IoT environments. The main contributions of this work are as follows.

- This paper proposes a fog-IoT intrusion detection system (IDS) model to mitigate attacks on fog-IoT systems. A lightweight detection model addresses the challenges of limited memory and computational resource constraints on edge devices. Additionally, long-short-term memory (LSTM) networks are chosen due to their ability to handle diverse datasets and distributions without restrictions

979-8-3503-9389-7/23/$31.00 ©2023 IEEE

Bibliography

[1] P. Mell, T. Grance, *et al.*, "The nist definition of cloud computing," 2011.

[2] M. M. Islam, S. Morshed, and P. Goswami, "Cloud computing: A survey on its limitations and potential solutions," *International Journal of Computer Science Issues (IJCSI)*, vol. 10, no. 4, p. 159, 2013.

[3] A. AlTwaijiry, "Cloud computing present limitations and future trends," *ScienceOpen Preprints*, 2021.

[4] C. Arivazhagan. and V. Natarajan., "A survey on fog computing paradigms, challenges and opportunities in iot," in *2020 International Conference on Communication and Signal Processing (ICCSP)*, 2020, pp. 0385–0389. DOI: 10.1109/ICCSP48568.2020.9182229.

[5] P. Habibi, M. Farhoudi, S. Kazemian, S. Khorsandi, and A. Leon-Garcia, "Fog computing: A comprehensive architectural survey," *IEEE Access*, vol. 8, pp. 69 105–69 133, 2020. DOI: 10.1109/ACCESS.2020.2983253.

[6] H. Rafique, M. A. Shah, S. U. Islam, T. Maqsood, S. Khan, and C. Maple, "A novel bio-inspired hybrid algorithm (nbiha) for efficient resource management in fog computing," *IEEE Access*, vol. 7, pp. 115 760–115 773, 2019.

[7] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[8] E. Borgia, "The internet of things vision: Key features, applications and open issues," *Computer Communications*, vol. 54, pp. 1–31, 2014.

[9] D. Uckelmann, M. Harrison, and F. Michahelles, "An architectural approach towards the future internet of things," in *Architecting the internet of things*, Springer, 2011, pp. 1–24.

[10] O. Vermesan and P. Friess, *Internet of things: converging technologies for smart environments and integrated ecosystems*. River publishers, 2013.

[11] R. Chataut, A. Phoummalayvane, and R. Akl, "Unleashing the power of iot: A comprehensive review of iot applications and future prospects in healthcare, agriculture, smart homes, smart cities, and industry 4.0," *Sensors*, vol. 23, no. 16, p. 7194, 2023.

[12] A. Tabassum and W. Lebda, "Security framework for iot devices against cyber-attacks," *arXiv preprint arXiv:1912.01712*, 2019.

[13] A. Bhagat, S. Mittal, U. Faiz, and D. K. Sharma, "Data security and privacy functions in fog data analytics," *Fog Data Analytics for IoT Applications: Next Generation Process Model with State of the Art Technologies*, pp. 355–385, 2020.

[14] A. A.-N. Patwary, A. Fu, R. K. Naha, *et al.*, "Authentication, access control, privacy, threats and trust management towards securing fog computing environments: A review," *arXiv preprint arXiv:2003.00395*, 2020.

[15] M. Aazam, S. Zeadally, and K. A. Harras, "Deploying fog computing in industrial internet of things and industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4674–4682, 2018.

[16] J. Shahid, R. Ahmad, A. K. Kiani, T. Ahmad, S. Saeed, and A. M. Almuhaideb, "Data protection and privacy of the internet of healthcare things (iohts)," *Applied Sciences*, vol. 12, no. 4, p. 1927, 2022.

[17] F. Bi, X. Miao, W. Chen, W. Fang, W. Zhang, and D. Wang, "An overview on the applications and security issues of fog computing," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, IEEE, 2018, pp. 1792–1797.

[18] A. Ali, M. Ahmed, M. Imran, and H. A. Khattak, "Security and privacy issues in fog computing," *Fog Computing: Theory and Practice*, pp. 105–137, 2020.

[19] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog computing for internet of things applications: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 601–628, 2017.

[20] M. Binjubeir, A. A. Ahmed, M. A. B. Ismail, A. S. Sadiq, and M. K. Khan, "Comprehensive survey on big data privacy protection," *IEEE Access*, vol. 8, pp. 20 067–20 079, 2019.

[21] C. Thapa and S. Camtepe, "Precision health data: Requirements, challenges and existing techniques for data security and privacy," *Computers in biology and medicine*, vol. 129, p. 104 130, 2021.

[22] Y. I. Alzoubi, V. H. Osmanaj, A. Jaradat, and A. Al-Ahmad, "Fog computing security and privacy for the internet of thing applications: State-of-the-art," *Security and Privacy*, vol. 4, no. 2, e145, 2021.

[23] R. Bensaid, N. Labraoui, A. A. Abba Ari, *et al.*, "Toward a real-time tcp syn flood ddos mitigation using adaptive neuro-fuzzy classifier and sdn assistance in fog computing," *Security and Communication Networks*, vol. 2024, 2023.

[24] A. H. Arif and N. E. Hasan, "Boteliminator: Detecting botmaster & botnet using ip traceback mechanism," Ph.D. dissertation, Department of Computer Science and Engineering (CSE), Islamic University of …, 2013.

[25] C. Kelly, N. Pitropakis, S. McKeown, and C. Lambrinoudakis, "Testing and hardening iot devices against the mirai botnet," in *2020 International conference*

on cyber security and protection of digital services (cyber security), IEEE, 2020, pp. 1–8.

[26] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "Iot: Internet of threats? a survey of practical security vulnerabilities in real iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, 2019.

[27] Y. Shah and S. 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)*, IEEE, 2020, pp. 0406–0413.

[28] M. Farooq, M. H. Khan, and R. A. Khan, "Implementation of network security for intrusion detection & prevention system in iot networks: Challenges & approach," *Int. J. Advanced Networking and Applications*, vol. 15, no. 05, pp. 6109–6113, 2023.

[29] M. R. Ahmed, S. Shatabda, A. M. Islam, M. T. I. Robin, *et al.*, "Intrusion detection system in software-defined networks using machine learning and deep learning techniques–a comprehensive survey," *Authorea Preprints*, 2023.

[30] S. M. Pournaghi, M. Bayat, and Y. Farjami, "Medsba: A novel and secure scheme to share medical data based on blockchain technology and attribute-based encryption," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 4613–4641, 2020.

[31] A. B. Pratomo, S. Mokodenseho, and A. M. Aziz, "Data encryption and anonymization techniques for enhanced information system security and privacy," *West Science Information System and Technology*, vol. 1, no. 01, pp. 1–9, 2023.

[32] I. A. Mohammed, "Intelligent authentication for identity and access management: A review paper," *International Journal of Managment, IT and Engineering (IJMIE)*, vol. 3, no. 1, pp. 696–705, 2013.

[33] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[34] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in sdn-openflow networks," *Computer Networks*, vol. 71, pp. 1–30, 2014.

[35] A. Hakiri, A. Gokhale, P. Berthou, D. C. Schmidt, and T. Gayraud, "Software-defined networking: Challenges and research opportunities for future internet," *Computer Networks*, vol. 75, pp. 453–471, 2014.

[36] V. Thirupathi, C. Sandeep, N. Kumar, and P. P. Kumar, "A comprehensive review on sdn architecture, applications and major benifits of sdn," *International Journal of Advanced Science and Technology*, vol. 28, no. 20, pp. 607–614, 2019.

[37] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software defined networking: State of the art and research challenges," *Computer Networks*, vol. 72, pp. 74–98, 2014.

[38] M. Alsaeedi, M. M. Mohamad, and A. A. Al-Roubaiey, "Toward adaptive and scalable openflow-sdn flow control: A survey," *IEEE Access*, vol. 7, pp. 107 346–107 379, 2019.

[39] M. K. Jaiswal, "Introduction to openflow," in *Innovations in Software-Defined Networking and Network Functions Virtualization*, IGI Global, 2018, pp. 52–71.

[40] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications surveys & tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[41] M. M. Elmoslemany, A. S. T. Eldien, and M. M. Selim, "Performance analysis in software defined network controllers," in *2020 15th International Conference on Computer Engineering and Systems (ICCES)*, IEEE, 2020, pp. 1–6.

[42] A. A. Ghorbani, W. Lu, and M. Tavallaee, *Network intrusion detection and prevention: concepts and techniques*. Springer Science & Business Media, 2009, vol. 47.

[43] I. Martins, J. S. Resende, P. R. Sousa, S. Silva, L. Antunes, and J. Gama, "Host-based ids: A review and open issues of an anomaly detection system in iot," *Future Generation Computer Systems*, vol. 133, pp. 95–113, 2022.

[44] R. Samrin and D. Vasumathi, "Review on anomaly based network intrusion detection system," in *2017 international conference on electrical, electronics, communication, computer, and optimization techniques (ICEECCOT)*, IEEE, 2017, pp. 141–147.

[45] J. Sajani and D. S. Manikandan, "Analysing and monitoring of network ids using intrusion detection," *International Journal of Computer Engineering & Technology*, vol. 8, no. 3, pp. 20–27, 2017.

[46] R. Alshamy and M. Ghurab, "A review of big data in network intrusion detection system: Challenges, approaches, datasets, and tools," *Journal of Computer Sciences and Engineering*, vol. 8, no. 7, pp. 62–74, 2020.

[47] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions," *Electronics*, vol. 9, no. 7, p. 1177, 2020.

[48] A. Lazarevic, V. Kumar, and J. Srivastava, "Intrusion detection: A survey," *Managing cyber threats: Issues, approaches, and challenges*, pp. 19–78, 2005.

[49] A. Gangwar and S. Sahu, "A survey on anomaly and signature based intrusion detection system (ids)," *International Journal of Engineering Research and Applications*, vol. 4, no. 4, pp. 67–72, 2014.

[50] M. A. Alsoufi, S. Razak, M. M. Siraj, *et al.*, "Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review," *Applied sciences*, vol. 11, no. 18, p. 8383, 2021.

[51] P. Casas, P. Fiadino, and A. D'Alconzo, "Machine-learning based approaches for anomaly detection and classification in cellular networks.," in *TMA*, 2016, pp. 1–8.

[52] Z. Yang, X. Liu, T. Li, *et al.*, "A systematic literature review of methods and datasets for anomaly-based network intrusion detection," *Computers & Security*, vol. 116, p. 102 675, 2022.

[53] M. Usama, J. Qadir, A. Raza, *et al.*, "Unsupervised machine learning for networking: Techniques, applications and research challenges," *IEEE access*, vol. 7, pp. 65 579–65 615, 2019.

[54] G. Pu, L. Wang, J. Shen, and F. Dong, "A hybrid unsupervised clustering-based anomaly detection method," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 146–153, 2020.

[55] A. Mohamed, M. K. Najafabadi, Y. B. Wah, E. A. K. Zaman, and R. Maskat, "The state of the art and taxonomy of big data analytics: View from new big data framework," *Artificial Intelligence Review*, vol. 53, pp. 989–1037, 2020.

[56] A. A. Megantara and T. Ahmad, "A hybrid machine learning method for increasing the performance of network intrusion detection systems," *Journal of Big Data*, vol. 8, no. 1, p. 142, 2021.

[57] M. A. Wiering and M. Van Otterlo, "Reinforcement learning," *Adaptation, learning, and optimization*, vol. 12, no. 3, p. 729, 2012.

[58] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[59] AIML, *What is the basic architecture of an artificial neural network (ann)?* https://aiml.com/what-is-the-basic-architecture-of-an-artificial-neural-network-ann/, 2024.

[60] M. Z. Alom, T. M. Taha, C. Yakopcic, *et al.*, "A state-of-the-art survey on deep learning theory and architectures," *electronics*, vol. 8, no. 3, p. 292, 2019.

[61] N. Ketkar and J. Moolayil, "Feed-forward neural networks," *Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch*, pp. 93–131, 2021.

[62] N. Naik and P. Jenkins, "Sovrin network for decentralized digital identity: Analysing a self-sovereign identity system based on distributed ledger technology," in *2021 IEEE International Symposium on Systems Engineering (ISSE)*, IEEE, 2021, pp. 1–7.

[63] Affinidi, *What are verifiable credentials?* https://affinidi.medium.com/what-are-verifiable-credentials-79f1846a7b9, 2024.

[64] World Wide Web Consortium. "World Wide Web Consortium recommendation: Decentralized identifiers (dids) v1.0." (2022), [Online]. Available: https://www.w3.org/press-releases/2022/did-rec/ (visited on 04/05/2024).

[65] A. Grech, I. Sood, and L. Ariño, "Blockchain, self-sovereign identity and digital credentials: Promise versus praxis in education," *Frontiers in Blockchain*, vol. 4, p. 616 779, 2021.

[66] J. B. Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. T. Moreno, and A. Skarmeta, "Privacy-preserving solutions for blockchain: Review and challenges," *IEEE Access*, vol. 7, pp. 164 908–164 940, 2019.

[67] J. Sedlmeir, R. Smethurst, A. Rieger, and G. Fridgen, "Digital identities and verifiable credentials," *Business & Information Systems Engineering*, vol. 63, no. 5, pp. 603–613, 2021.

[68] A. Ahalawat, K. S. Babu, A. K. Turuk, and S. Patel, "A low-rate ddos detection and mitigation for sdn using renyi entropy with packet drop," *Journal of Information Security and Applications*, vol. 68, p. 103 212, 2022.

[69] S. Jin and D. S. Yeung, "A covariance analysis model for ddos attack detection," in *2004 IEEE international conference on communications (IEEE Cat. No. 04CH37577)*, IEEE, vol. 4, 2004, pp. 1882–1886.

[70] K. Bhushan *et al.*, "Ddos attack defense framework for cloud using fog computing," in *2017 2nd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT)*, IEEE, 2017, pp. 534–538.

[71] S. Javanmardi, M. Shojafar, R. Mohammadi, A. Nazari, V. Persico, and A. Pescapè, "Fupe: A security driven task scheduling approach for sdn-based iot–fog networks," *Journal of information security and applications*, vol. 60, p. 102 853, 2021.

[72] S. Javanmardi, M. Shojafar, R. Mohammadi, M. Alazab, and A. M. Caruso, "An sdn perspective iot-fog security: A survey," *Computer Networks*, vol. 229, p. 109 732, 2023.

[73] N. N. Tuan, P. H. Hung, N. D. Nghia, N. Van Tho, T. V. Phan, and N. H. Thanh, "A robust tcp-syn flood mitigation scheme using machine learning based on sdn," in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, IEEE, 2019, pp. 363–368.

[74] R. Priyadarshini, R. Kumar Barik, and H. Dubey, "Fog-sdn: A light mitigation scheme for ddos attack in fog computing framework," *International Journal of Communication Systems*, vol. 33, no. 9, e4389, 2020.

[75] J. A. Perez-Diaz, I. A. Valdovinos, K.-K. R. Choo, and D. Zhu, "A flexible sdn-based architecture for identifying and mitigating low-rate ddos attacks using machine learning," *IEEE Access*, vol. 8, pp. 155 859–155 872, 2020.

[76] O. Brun, Y. Yin, E. Gelenbe, Y. M. Kadioglu, J. Augusto-Gonzalez, and M. Ramos, "Deep learning with dense random neural networks for detecting attacks against iot-connected home environments," in *Security in Computer and Information Sciences: First International ISCIS Security Workshop 2018, Euro-CYBERSEC 2018, London, UK, February 26-27, 2018, Revised Selected Papers 1*, Springer International Publishing, 2018, pp. 79–89.

[77] R. Devi, R. K. Jha, A. Gupta, S. Jain, and P. Kumar, "Implementation of intrusion detection system using adaptive neuro-fuzzy inference system for 5g wireless communication network," *AEU-International Journal of Electronics and Communications*, vol. 74, pp. 94–106, 2017.

[78] A. S. Boroujerdi and S. Ayat, "A robust ensemble of neuro-fuzzy classifiers for ddos attack detection," in *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*, IEEE, 2013, pp. 484–487.

[79] J. Krimmling and S. Peter, "Integration and evaluation of intrusion detection for coap in smart city applications," in *2014 IEEE Conference on Communications and Network Security*, IEEE, 2014, pp. 73–78.

[80] B. Arrington, L. Barnett, R. Rufus, and A. Esterline, "Behavioral modeling intrusion detection system (bmids) using internet of things (iot) behavior-based anomaly detection via immunity-inspired algorithms," in *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, IEEE, 2016, pp. 1–6.

[81] A. Le, J. Loo, A. Lasebae, M. Aiash, and Y. Luo, "6lowpan: A study on qos security threats and countermeasures using intrusion detection system approach," *International Journal of Communication Systems*, vol. 25, no. 9, pp. 1189–1212, 2012.

[82] A. P. R. Da Silva, M. H. Martins, B. P. Rocha, A. A. Loureiro, L. B. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, 2005, pp. 16–23.

[83] M. Kansra and P. D. Chadha, "Cluster based detection of attack ids using data mining," *International Journal of Engineering Development and Research*, 2016.

[84] K. Zhao and L. Ge, "A survey on the internet of things security," in *2013 Ninth international conference on computational intelligence and security*, IEEE, 2013, pp. 663–667.

[85] S. Hajiheidari, K. Wakil, M. Badri, and N. J. Navimipour, "Intrusion detection systems in the internet of things: A comprehensive investigation," *Computer Networks*, vol. 160, pp. 165–191, 2019.

[86] R. Belchior *et al.*, "Ssibac: Self-sovereign identity based access control," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, 2020, pp. 1935–1943.

[87] D. Lagutin *et al.*, "Enabling decentralised identifiers and verifiable credentials for constrained iot devices using oauth-based delegation," in *Proceedings of the Workshop on Decentralized IoT Systems and Security (DISS 2019), in Conjunction with the NDSS Symposium, San Diego, CA, USA*, vol. 24, 2019.

[88] E. Jung, "A decentralized access control model for iot with did," in *IT Convergence and Security*, Springer, 2021, pp. 141–148.

[89] B. Kim *et al.*, "Attribute-based access control (abac) with decentralized identifier in the blockchain-based energy transaction platform," in *2021 International Conference on Information Networking (ICOIN)*, IEEE, 2021, pp. 845–848.

[90] O. Osanaiye, K.-K. R. Choo, and M. Dlodlo, "Distributed denial of service (ddos) resilience in cloud: Review and conceptual cloud ddos mitigation framework," *Journal of Network and Computer Applications*, vol. 67, pp. 147–165, 2016.

[91] S. Khattak, N. R. Ramay, K. R. Khan, A. A. Syed, and S. A. Khayam, "A taxonomy of botnet behavior, detection, and defense," *IEEE communications surveys & tutorials*, vol. 16, no. 2, pp. 898–924, 2013.

[92] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, "Botnet in ddos attacks: Trends and challenges," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2242–2270, 2015.

[93] B. Paharia and K. Bhushan, "A comprehensive review of distributed denial of service (ddos) attacks in fog computing environment," *Handbook of Computer Networks and Cyber Security: Principles and Paradigms*, pp. 493–524, 2020.

[94] C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram, and D. Zamboni, "Analysis of a denial of service attack on tcp," in *Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No. 97CB36097)*, IEEE, 1997, pp. 208–223.

[95] R. Santos, D. Souza, W. Santo, A. Ribeiro, and E. Moreno, "Machine learning algorithms to detect ddos attacks in sdn," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 16, e5402, 2020.

[96] J.-S. Jang, "Anfis: Adaptive-network-based fuzzy inference system," *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[97] S. Benfriha and N. Labraoui, "Insiders detection in the uncertain iod using fuzzy logic," in *2022 International Arab Conference on Information Technology (ACIT)*, IEEE, 2022, pp. 1–6.

[98] N. Walia, H. Singh, and A. Sharma, "Anfis: Adaptive neuro-fuzzy inference system-a survey," *International Journal of Computer Applications*, vol. 123, no. 13, 2015.

[99] S. Ghosh, S. Biswas, D. Sarkar, and P. P. Sarkar, "A novel neuro-fuzzy classification technique for data mining," *Egyptian Informatics Journal*, vol. 15, no. 3, pp. 129–147, 2014.

[100] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[101] R. Devi, R. K. Jha, A. Gupta, S. Jain, and P. Kumar, "Implementation of intrusion detection system using adaptive neuro-fuzzy inference system for 5g wireless communication network," *AEU-International Journal of Electronics and Communications, 74, 94-106.*, 2017.

[102] V. Bureva, "Generalized net model of information security activities in the automated information systems," in *Advances and New Developments in Fuzzy Logic and Technology: Selected Papers from IWIFSGN'2019–The Eighteenth*

International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets held on October 24-25, 2019 in Warsaw, Poland, Springer, 2021, pp. 280–288.

[103] A. Aguado, M. Davis, S. Peng, *et al.*, "Dynamic virtual network reconfiguration over sdn orchestrated multitechnology optical transport domains," *Journal of Lightwave Technology*, vol. 34, no. 8, pp. 1933–1938, 2016.

[104] S. Sathyadevan, K. Achuthan, R. Doss, and L. Pan, "Protean authentication scheme–a time-bound dynamic keygen authentication technique for iot edge nodes in outdoor deployments," *IEEE access*, vol. 7, pp. 92 419–92 435, 2019.

[105] J.-P. A. Yaacoub, O. Salman, H. N. Noura, N. Kaaniche, A. Chehab, and M. Malli, "Cyber-physical systems security: Limitations, issues and future trends," *Microprocessors and microsystems*, vol. 77, p. 103 201, 2020.

[106] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (sdn) and distributed denial of service (ddos) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE communications surveys & tutorials*, vol. 18, no. 1, pp. 602–622, 2015.

[107] R. Vishwakarma and A. K. Jain, "A survey of ddos attacking techniques and defence mechanisms in the iot network," *Telecommunication systems*, vol. 73, no. 1, pp. 3–25, 2020.

[108] A. Nath, *Packet Analysis with Wireshark*. Packt Publishing Ltd, 2015.

[109] *M. team, mininet overview*, http://mininet.org/overview/, 2023.

[110] *Ryu sdn framework*, https://ryu-sdn.org/l, 2023.

[111] S. Bhardwaj and S. N. Panda, "Performance evaluation using ryu sdn controller in software-defined networking environment," *Wireless Personal Communications*, vol. 122, no. 1, pp. 701–723, 2022.

[112] *Keras.io*, https://keras.io, 2023.

[113] M. Abadi, "Tensorflow: Learning functions at scale," in *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming*, 2016, pp. 1–1.

[114] S. Prusty, S. Patnaik, and S. K. Dash, "Skcv: Stratified k-fold cross-validation on ml classifiers for predicting cervical cancer," *Frontiers in Nanotechnology*, vol. 4, p. 972 421, 2022.

[115] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*, IEEE, 2019, pp. 1–8.

[116] Y. Xia, C. Liu, Y. Li, and N. Liu, "A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring," *Expert systems with applications*, vol. 78, pp. 225–241, 2017.

[117] S. Javanmardi, M. Shojafar, R. Mohammadi, A. Nazari, V. Persico, and A. Pescapè, "Fupe: A security driven task scheduling approach for sdn-based iot–fog networks," *Journal of information security and applications*, vol. 60, p. 102 853, 2021.

[118] M. V. de Assis, L. F. Carvalho, J. J. Rodrigues, J. Lloret, and M. L. Proença Jr, "Near real-time security system applied to sdn environments in iot networks using convolutional neural network," *Computers & Electrical Engineering*, vol. 86, p. 106 738, 2020.

[119] M. P. Novaes, L. F. Carvalho, J. Lloret, and M. L. Proença Jr, "Adversarial deep learning approach detection and defense against ddos attacks in sdn environments," *Future Generation Computer Systems*, vol. 125, pp. 156–167, 2021.

[120] J. A. Perez-Diaz, I. A. Valdovinos, K.-K. R. Choo, and D. Zhu, "A flexible sdn-based architecture for identifying and mitigating low-rate ddos attacks using machine learning," *IEEE Access*, vol. 8, pp. 155 859–155 872, 2020.

[121] D. Lee and S. N. Yoon, "Application of artificial intelligence-based technologies in the healthcare industry: Opportunities and challenges," *International Journal of Environmental Research and Public Health*, vol. 18, no. 1, p. 271, 2021.

[122] A. Kumari, S. Tanwar, S. Tyagi, *et al.*, "Fog computing for healthcare 4.0 environment: Opportunities and challenges," *Computers Electrical Engineering*, vol. 72, pp. 1–13, 2018.

[123] M. E. Mondajar, R. Avtar, H. L. B. Diaz, *et al.*, "Digitalization to achieve sustainable development goals: Steps towards a smart green planet," *Science of The Total Environment*, vol. 794, p. 148 539, 2021.

[124] A. Djenna and D. E. Saïdouni, "Cyber attacks classification in iot-based-healthcare infrastructure," in *2018 2nd Cyber Security in Networking Conference (CSNet)*, IEEE, 2018, pp. 1–4.

[125] N. Chakraborty, "Intrusion detection system and intrusion prevention system: A comparative study," *International Journal of Computing and Business Research (IJCBR)*, vol. 4, no. 2, pp. 1–8, 2013.

[126] A. Khraisat, I. Gondal, P. Vamplew, *et al.*, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.

[127] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for iot security based on learning techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019.

[128] S. AbdulRahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5476–5497, 2020.

[129] C. Zhang, Y. Xie, H. Bai, *et al.*, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106 775, 2021.

[130] M. SARHAN, S. LAYEGHY, N. MOUSTAFA, *et al.*, "Cyber threat intelligence sharing scheme based on federated learning for network intrusion detection," *Journal of Network and Systems Management*, vol. 31, no. 1, p. 3, 2023.

[131] M. Aledhari, M. A. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140 699–140 725, 2020.

[132] O. A. Wahab, A. Mourad, H. Otrok, and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1342–1397, 2021.

[133] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40 281–40 306, 2022.

[134] R.Bensaid., . N.Labraoui, and H. B. Salameh, "Federated deep learning-based intrusion detection approach for enhancing privacy in fog-iot networks," in *2023 10th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, IEEE, 2023, pp. 156–160.

[135] H. Saidi, N. Labraoui, A. A. A. Ari, L. A. Maglaras, and J. H. M. Emati, "Dsmac: Privacy-aware decentralized self-management of data access control based on blockchain for health data," *IEEE Access*, vol. 10, pp. 101 011–101 028, 2022.

[136] M. Jensen, N. Gruschka, and R. Herkenhöner, "A survey of attacks on web services," *Comput. Sci.-Res. Dev.*, vol. 24, pp. 185–197, 2009.

[137] J. R. van der Merwe, X. Zubizarreta, I. Lukčin, A. Rügamer, and W. Felber, "Classification of spoofing attack types," in *2018 European Navigation Conference (ENC)*, 2018, pp. 91–99.

[138] D. Stiawan, M. Idris, R. F. Malik, S. Nurmaini, N. Alsharif, and R. Budiarto, "Investigating brute force attack patterns in iot network," *J. Electr. Comput. Eng.*, vol. 2019, p. 4 568 368, 2019.

[139] J. Gamblin, *Mirai botnet*, https://github.com/jgamblin/Mirai-Source-Code, 2017.

[140] S. Figueroa-Lorenzo, J. A. Benito, and S. Arrizabalaga, "Modbus access control system based on ssi over hyperledger fabric blockchain," *Sensors*, vol. 21, no. 16, p. 2021, 2021.

[141] A. Grafberger, M. Chadha, A. Jindal, J. Gu, and M. Gerndt, "Fedless: Secure and scalable federated learning using serverless computing," in *2021 IEEE International Conference on Big Data (Big Data)*, IEEE, 2021, pp. 164–173.

[142] Z. Lian, C. Zhang, K. Nan, and C. Su, "Spoil: Sybil-based untargeted data poisoning attacks in federated learning," in *International Conference on Network and System Security*, Springer Nature Switzerland, Aug. 2023, pp. 235–248.

[143] A. Banerjee, B. Dutta, T. Mandal, R. Chakraborty, and R. Mondal, "Blockchain in iot and beyond: Case studies on interoperability and privacy," in *Blockchain based Internet of Things*, Springer, 2022, pp. 113–138.

[144] T. Manoj, K. Makkithaya, and V. Narendra, "A blockchain based decentralized identifiers for entity authentication in electronic health records," *Cogent Eng*, vol. 9, no. 1, p. 2 035 134, 2022.

[145] E. C. P. Neto, S. Dadkhah, R. Ferreira, *et al.*, "Ciciot2023: A real-time dataset and benchmark forlarge-scale attacks in iot environment," 2023.

[146] W. Schneble and G. Thamilarasu, "Attack detection using federated learning in medical cyberphysical systems," in *28th International Conference on Computer Communications and Networks (ICCCN)*, IEEE, 2019, pp. 1–8.

[147] S. Chatterjee and M. K. Hanawal, "Federated learning for intrusion detection in iot security: A hybrid ensemble approach," *ArXiv*, 2021. eprint: 2106.15349.

[148] D. Man, F. Zeng, W. Yang, *et al.*, "Intelligent intrusion detection based on federated learning for edge-assisted internet of things," *Secur. Commun.Networks*, vol. 2021, 9361348:1–9361348:11, 2021.

[149] V. Rey, P. M. S'anchez, A. H. Celdr'an, *et al.*, "Federated learning for malware detection in iot devices," *Comput. Networks*, vol. 204, p. 108 693, 2022.

[150] P. Ruzafa-Alcazar, P. Fernandez-Saura, E. Marmol-Campos, *et al.*, "Intrusion detection based on privacy-preserving federated learning for the industrial iot," *IEEE Transactions on Industrial Informatics*, 2021.

[151] Y. Zhao, J. Chen, D. Wu, *et al.*, "Multi-task network anomaly detection using federated learning," in *Proceedings of the Tenth International Symposium on Information and Communication Technology*, 2019.

[152] O. Friha, M. A. Ferrag, L. Shu, *et al.*, "Felids: Federated learning-based intrusion detection system for agricultural internet of things," *Journal of Parallel and Distributed Computing*, vol. 165, pp. 17–31, 2022.

[153] E. Ashraf, N. F. Areed, H. Salem, E. H. Abdelhay, and A. Farouk, "Fidchain: Federated intrusion detection system for blockchain-enabled iot healthcare applications," *Healthcare*, vol. 10, no. 6, p. 1110, 2022.

[154] D. Preuveneers, V. Rimmer, I. Tsingenopoulos, J. Spooren, W. Joosen, and E. Ilie-Zudor, "Chained anomaly detection models for federated learning: An intrusion detection case study," *Applied Sciences*, vol. 8, no. 12, p. 2663, 2018.

[155] A. Lakhan, M. A. Mohammed, J. Nedoma, *et al.*, "Federated-learning based privacy preservation and fraud-enabled blockchain iomt system for healthcare," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 2, pp. 664–672, 2022.

Abstract
The thesis explores the profound increase in connectivity facilitated by IoT, cloud, and fog computing, which have revolutionized data management. However, ensuring privacy and security has become a paramount challenge. To address this, the thesis provides a comprehensive overview of fog computing, IoT, and smart healthcare, focusing particularly on privacy and security concerns. It introduces two significant contributions: firstly, the incorporation of an Adaptive Neuro-Fuzzy Inference System within Software-Defined Networking to detect and mitigate Syn flood DDoS attacks in fog networks, and secondly, the proposal of a Federated Learning-Based Intrusion Detection System tailored for IoT-enabled smart healthcare systems, utilizing decentralized identifiers and verifiable credentials for user authentication. Through experimental validation on securing fog-enabled IoT networks, the proposed solutions demonstrate a high level of security while preserving user privacy. By harnessing the machine learning capabilities of both IDS and SDN, the solutions effectively protect network traffic in fog computing environments for IoT applications, emphasizing robust security measures alongside user privacy preservation.
Keywords: Security and Privacy, IoT, Fog computing, SDN, IDS, Deep Learning, Blockchain, SSI, DID, VC.
Résumé
La thèse explore la profonde augmentation de la connectivité facilitée par l'Internet des Objets (IdO), le cloud et le fog computing, qui ont révolutionné la gestion des données. Cependant, garantir la confidentialité et la sécurité est devenu un défi primordial. Pour y remédier, la thèse offre un aperçu complet du fog computing, de l IdO et des soins de santé intelligents, en mettant particulièrement l'accent sur les préoccupations en matière de confidentialité et de sécurité. Elle introduit deux contributions significatives : premièrement, l'incorporation d'un Système d'Inférence Neuro-Fuzzy Adaptatif dans le Réseau Défini par Logiciel(SDN) pour détecter et atténuer les attaques DDoS par Syn flood dans les réseaux de fog, et deuxièmement, la proposition d'un Système de Détection d'Intrusion basé sur l'Apprentissage Fédéré adapté aux systèmes de soins de santé intelligents activés par l'IoT, utilisant des identifiants décentralisés et des justificatifs vérifiables pour l'authentification des utilisateurs. À travers une validation expérimentale sur la sécurisation des réseaux IdO activés par le fog, les solutions proposées démontrent un haut niveau de sécurité tout en préservant la confidentialité des utilisateurs. En exploitant les capacités d'apprentissage automatique à la fois des IDS et du SDN, les solutions protègent efficacement le trafic réseau dans les environnements de fog computing pour les applications IdO, mettant l'accent sur des mesures de sécurité robustes en même temps que la préservation de la confidentialité des utilisateurs.
Mots-clés : Sécurité et confidentialité, IdO, Informatique en brouillard, SDN, IDS, Apprentissage profond, Blockchain, SSI, DID, VC.

الملخص
تتناول الأطروحة الزيادة الكبيرة في التوصيل التي تُيَسرها تقنيات الإنترنت الأشياء (IoT) والحوسبة السحابية وحوسبة الضباب، والتي غيّرت إدارة البيانات. ومع ذلك، أصبح ضمان الخصوصية والأمان تحدٍ أساسيًا. لمعالجة هذه المشكلة، تقدم الرسالة نظرة شاملة على حوسبة الضباب والإنترنت المشترك والرعاية الصحية الذكية، مع التركيز بشكل خاص على المخاوف المتعلقة بالخصوصية والأمان. تقدم الرسالة مساهمتين هامتين: أولاً، دمج نظام الاستنتاج العصبي الضبابي المتكيف داخل شبكة الجرد الناجمة عن البرمجيات لاكتشاف وتخفيف هجمات توزيع الخدمة المنكوبة بفيض (Syn flood) في شبكات الضباب، وثانياً، اقتراح نظام الكشف عن التسلل المستند إلى التعلم الاتحادي المُضبّط لنظم الرعاية الصحية الذكية المُمكّنة بواسطة الإنترنت، باستخدام معرفات مُتمركزة ومستندات قابلة للتحقق لمصادقة المستخدمين. من خلال التحقق التجريبي من تأمين شبكات (IoT)المُمكّنة بواسطة الضباب، تُظهر الحلول المقترحة مستوى عالٍ من الأمان مع الحفاظ على خصوصية المستخدمين. من خلال استغلال قدرات التعلم الآلي في كل من نظم الكشف عن التسلل والجرد الناجم عن البرمجيات، تحمي الحلول بشكل فعال حركة المرور عبر الشبكة في بيئات حوسبة الضباب لتطبيقات(IoT) ، مع التركيز على إجراءات الأمان القوية إلى جانب الحفاظ على خصوصية المستخدمين.
الكلمات المفتاحية: أمان وخصوصية، إنترنت الأشياء، الحوسبة الضبابية، شبكة الجرد الناجمة عن البرمجيات، نظام كشف التسلل، التعلم العميق، سلسلة الكتل، هوية ذاتية-سيادية، معرف متمركز، وثائق قابلة للتحقق.