**MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH**
**MOHAMED KHIDER UNIVERSITY OF BISKRA**
**Faculty of Exact Sciences**
**Computer Science Department**

**N° : ……/M2/2025**

# Dissertation

Presented to obtain the academic master's degree in Computer Science

**Option** : **Networks, Information and Communication Technologies (RTIC)**

---

# Federated Learning for Cybersecurity: Enhancing Threat Detection Across Multiple Organizations

---

**Presented by:**

- Boussaha Meriem
- Ghamri Bilal

**Supervisor :** Dr. Ilyes Naidji

**Co-Supervisor :**
Dr. Ahmed Tibermacine

**Jury :**

| | |
|---|---|
| Dr. Mokhtari Bilal | President |
| Dr. Akrour Djouher | Examiner |
| Dr. Naidji Ilyes | Supervisor |

**Academic Year : 2024-2025**

# *Acknowledgements*

First and foremost, we bow in gratitude to Allah Almighty, whose infinite mercy and boundless grace have been the foundation of every step we have taken. It is by His will that we were granted the strength to endure, the clarity to understand, and the perseverance to see this work through to completion. Without His divine guidance and blessings, none of this would have been possible.

We wish to express our deepest appreciation to our supervisor, Mr. Ilyes Naidji, whose wisdom, patience, and unwavering support have been a guiding light throughout this journey. His insightful feedback, thoughtful advice, and constant encouragement not only shaped this work but also inspired personal and academic growth. We are also sincerely grateful for his generosity in providing the essential resources and facilities that played a significant role in making this project a reality. We are truly honored to have worked under his supervision.

Finally, we reserve our deepest gratitude for our families, whose unconditional love, endless patience, and unwavering faith in us have been the cornerstone of our resilience. Their silent sacrifices, comforting words, and constant belief in our potential have given us the courage to keep going, even during the most challenging times. To them, we owe not only the completion of this thesis, but also the strength that carried us through it.

# *Dedication*

I dedicate this work with deep admiration and gratitude to my father, whose steadfast support, wisdom, and encouragement have profoundly shaped my academic and professional path. His guidance remains a beacon of inspiration for my endeavors.

To my mother, whose boundless love, resilience, and unwavering faith in my abilities have been the driving force behind this achievement. Her sacrifices and nurturing spirit are the heart of my success.

To my family, whose constant presence and uplifting support have fortified my resolve, I extend my heartfelt thanks.

With utmost respect and a commitment to reflect their enduring impact,

Bilal.

# *Dedication*

I dedicate this humble work with all the love and gratitude in my heart to the gentle memory of my grandparents, whose absence continues to echo in my life, but whose values, love, and blessings still guide me in silence. May this achievement honor their memory and the legacy they left behind.

To the memory of my father, whom I never had the chance to know, but whose absence instilled in me the strength to fight and move forward. This work is a tribute to the silent and eternal bond that unites us.

To my mother, my source of courage, tenderness, and determination. For all her sacrifices, her unconditional love, and her unwavering belief in my future, I dedicate this thesis to her with infinite gratitude.

To my family, for their presence, their support, and their constant encouragement.

To my friends, those who stood by me in moments of doubt as well as in times of joy, thank you for your sincere friendship and comforting presence.

A special sparkle of gratitude goes to my dearest Chahd, whose kindness, laughter, and endless support made even the toughest days feel lighter. You are truly one of a kind, and I'm so lucky to have you by my side.

With deepest gratitude and love,

Meriem.

# *Abstract*

The rapid evolution of digital ecosystems have intensified cyber threats, creating significant challenges for data privacy, regulatory compliance, and operational resilience in cybersecurity. This research introduces a Federated Learning (FL) framework for collaborative cybersecurity threat detection, aimed at enhancing the precision, scalability, and privacy of threat detection systems. As a decentralized machine learning paradigm, FL enables multiple entities to train a shared predictive model collaboratively without exchanging sensitive data, thereby ensuring compliance with privacy regulations and mitigating data breach risks.

This study proposes a novel FL-based approach tailored for cybersecurity, addressing a wide range of cyber threats. The framework facilitates distributed training across heterogeneous entities, leveraging local data to improve detection accuracy while maintaining data sovereignty. Through extensive empirical evaluations using real-world cybersecurity datasets, the proposed approach demonstrates superior performance in detecting threats compared to traditional centralized methods, achieving high accuracy and robustness while significantly enhancing data privacy and scalability.

The findings highlight Federated Learning as a transformative solution for privacy-preserving cybersecurity, offering a scalable and secure framework to counter dynamic cyber threats in distributed environments. This research lays the groundwork for future advancements in AI-driven, ethically compliant cybersecurity solutions.

**Keywords:** Federated Learning, Cybersecurity, Data Privacy, Intrusion Detection, Privacy-Preserving AI, Deep Learning.

# *Résumé*

L'évolution rapide des écosystèmes numériques a intensifié les cybermenaces, posant des défis majeurs en matière de confidentialité des données, de conformité réglementaire et de résilience opérationnelle en cybersécurité. Cette recherche présente un cadre basé sur l'apprentissage fédéré (Federated Learning, FL) pour la détection collaborative des menaces, visant à améliorer la précision, l'évolutivité et la confidentialité des systèmes de détection. En tant que paradigme d'apprentissage automatique décentralisé, le FL permet à plusieurs entités d'entraîner un modèle prédictif partagé sans échanger de données sensibles, garantissant ainsi le respect des réglementations sur la protection des données et réduisant les risques de violations.

Cette étude propose une approche inédite basée sur le FL, adaptée à la cybersécurité, pour contrer une vaste gamme de cybermenaces. Le cadre favorise un entraînement distribué à travers des entités hétérogènes, exploitant les données locales pour optimiser la précision de la détection tout en préservant la souveraineté des données. Des évaluations empiriques approfondies, réalisées sur des ensembles de données réels en cybersécurité, démontrent une performance supérieure dans la détection des menaces par rapport aux méthodes centralisées traditionnelles, avec une précision et une robustesse élevées, tout en renforçant la confidentialité et l'évolutivité.

Les résultats consacrent l'apprentissage fédéré comme une solution transformative pour une cybersécurité respectueuse de la confidentialité, offrant un cadre sécurisé et évolutif pour contrer les menaces dynamiques dans des environnements distribués. Cette recherche pose les bases pour de futures avancées dans les solutions de cybersécurité éthiques et basées sur l'IA.

**Mots-clés:** Apprentissage Fédéré, Cybersécurité, Confidentialité des Données, Détection de Menaces, IA Préservant la Confidentialité, Apprentissage Profond.

# Contents

# General Introduction

In this introduction, we begin by outlining the background of this thesis, focusing on the critical role of cybersecurity in modern digital ecosystems and the transformative potential of federated learning. Next, we discuss the motivations driving this work, define the specific problem addressed, and articulate the objectives. Finally, we provide an overview of the manuscript's organization to guide the reader through the research structure.

## Background

The rapid proliferation of interconnected systems and the increasing reliance on digital infrastructure have amplified the importance of cybersecurity across various sectors, including finance, healthcare, government, and critical infrastructure. As organizations and individuals become more dependent on technology, they face an evolving landscape of cyber threats, ranging from malware and phishing to advanced persistent threats (APTs) and AI-powered attacks. These threats exploit vulnerabilities in systems, networks, and human behavior, compromising the confidentiality, integrity, and availability of critical data and services.

Traditional cybersecurity approaches, such as firewalls, intrusion detection systems, and antivirus software, are increasingly inadequate against sophisticated and dynamic threats. Centralized machine learning models, which require aggregating sensitive data into a single repository for training, raise significant concerns regarding data privacy, regulatory compliance, and the risk of data breaches. In this context, federated learning (FL) emerges as a promising paradigm to address these challenges. FL enables collaborative model training across multiple decentralized entities such as edge devices, organizations, or IoT networks without sharing raw data. Instead, only model updates are exchanged, preserving data locality and enhancing privacy while enabling scalable and adaptive threat detection.

This dissertation investigates the application of federated learning to cybersecurity, with a focus on developing a privacy-preserving framework for collaborative threat detection. By leveraging decentralized data and advanced deep learning techniques, the proposed solution aims to enhance the accuracy, scalability, and robustness of intrusion detection systems while addressing privacy and compliance requirements.

## Problematic

Cybersecurity threat detection faces several critical challenges that hinder the effectiveness of traditional approaches:

1. **Evolving Threat Landscape**: Cyber threats are becoming increasingly sophisticated, with attackers leveraging advanced techniques such as AI-powered attacks,

zero-day exploits, and polymorphic malware. These dynamic threats require adaptive and real-time detection mechanisms that traditional rule-based systems struggle to provide.

2. **Data Privacy and Security**: Centralized machine learning approaches necessitate aggregating sensitive data, which increases the risk of breaches and violates privacy regulations such as GDPR. This is particularly problematic in cybersecurity, where data often includes proprietary logs, user behavior, or critical infrastructure details.

3. **Adversarial Vulnerabilities**: AI-based detection systems are susceptible to adversarial attacks, where malicious inputs are crafted to evade or mislead models. Such vulnerabilities can lead to false negatives, allowing threats to go undetected, or false positives, causing operational disruptions.

4. **Scalability and Resource Constraints**: Deploying AI models in resource-constrained environments, such as IoT devices or edge networks, requires balancing computational efficiency with detection accuracy, which is challenging in centralized architectures.

Addressing these challenges demands innovative approaches that combine advanced machine learning with privacy-preserving techniques, robust model designs, and scalable architectures tailored to the cybersecurity domain.

# Motivation and Objectives

The primary objectives of this dissertation are:

1. **To develop a federated learning framework for cybersecurity threat detection**: This involves designing and implementing a decentralized deep learning model that leverages federated learning to collaboratively train on distributed, privacy-sensitive datasets while maintaining high detection performance.

2. **To ensure data privacy and compliance**: By adopting federated learning, the framework aims to protect sensitive cybersecurity data, such as network logs and user activities, ensuring compliance with data protection regulations and reducing the risk of breaches.

3. **To enhance model robustness and fairness**: The research seeks to address issues such as data imbalance, bias in training data, and adversarial vulnerabilities by incorporating techniques like regularization, feature selection, and robust aggregation strategies.

4. **To evaluate the proposed framework against traditional methods**: Extensive experiments will compare the federated learning approach with centralized machine learning models, assessing metrics such as accuracy, precision, recall, F1-score, and computational efficiency.

5. **To provide practical insights for deployment**: The dissertation aims to offer actionable recommendations for implementing federated learning-based cybersecurity solutions in real-world scenarios, addressing challenges like communication overhead, scalability, and integration with existing security infrastructures.

This work is motivated by the need to bridge the gap between advanced AI techniques and the stringent requirements of cybersecurity, ultimately contributing to more secure, resilient, and privacy-preserving digital ecosystems.

## Manuscript Organization

The manuscript is structured into four chapters, each addressing a distinct aspect of the research:

- **Chapter 1: Cybersecurity Challenges**. This chapter introduces the foundational concepts of cybersecurity, including the CIA Triad, common and emerging threats, and traditional defense mechanisms. It highlights the role of AI in addressing modern cybersecurity challenges.
- **Chapter 2: State of the Art**. This chapter reviews related work on various cyber threats, such as DDoS, brute force, web attacks, and ransomware, providing a critical analysis of existing detection and mitigation strategies.
- **Chapter 3: Methodology of the Federated Approach**. This chapter details the design and development of the proposed federated learning-based framework, including data preprocessing, feature selection, model architecture, and federated learning algorithms.
- **Chapter 4: Implementation and Results**. This chapter presents the implementation details, experimental setup, and performance evaluation of the federated model learning.
- **General Conclusion**. The final section summarizes the key findings, discusses the implications for cybersecurity, and suggests directions for future research.

# Chapter 1

# Introduction of Cybersecurity Challenges

## 1.1 Introduction

Cyber-Physical Systems (CPS) are integrations of computation, networking, and physical processes. In CPS, embedded computers and networks monitor and control the physical processes, typically with feedback loops where physical processes affect computations and vice versa [1]. The Smart Grid is a prime example of a Cyber-Physical System applied to the energy sector [2]. It refers to an intelligent electricity network that uses digital communication technology to monitor and manage the generation, distribution, and consumption of electricity , [3], [4].

In today's digitally connected world, the importance of cybersecurity cannot be overstated. With the growing reliance on technology and the internet, individuals and organizations face continuous exposure to diverse cyber threats. This shows that cybersecurity plays a critical role in safeguarding personal information, as well as business, governmental, and organizational data [5]. As the volume and complexity of cyber threats escalate, traditional security methods are struggling to keep pace. In relation to this casuistry, Artificial Intelligence has emerged as a powerful tool that organizations and their various teams can rely on to automate tasks in the field of cybersecurity [6]. Artificial Intelligence powered systems are being used by attackers to automate and scale their attacks more effectively, while defenders have also embraced Artificial Intelligence to strengthen their cybersecurity strategies enabling faster, more accurate threat detection, prediction, and response [7]. The integration of Artificial Intelligence into cybersecurity has become essential as cyber threats grow more sophisticated and dynamic. With adversaries leveraging advanced technologies to enhance the scale and precision of their attacks, traditional defense mechanisms are no longer sufficient. AI provides a transformative approach by offering adaptive, intelligent, and real-time solutions. In this evolving landscape, adopting AI-driven cybersecurity solutions is critical to maintaining the security and resilience of modern digital systems [8].

## 1.2 Core Cybersecurity Concepts

### 1.2.1 Confidentiality, Integrity, and Availability (CIA Triad)

The CIA Triad is the cornerstone of cybersecurity, representing the primary objectives of any security strategy [5].

- **Confidentiality** ensures that only authorized entities can access information while

preventing unauthorized parties from learning anything about its content. It guarantees that sensitive information reaches the intended recipients while remaining protected from exposure to others.

- **Integrity** refers to guaranteeing that data is real, correct, and protected against unauthorized modification or alteration. It is a property that information has not been tampered with in any manner and that the information's source is legitimate.
- **Availability** is the property that allows authorized users to access and modify information in a timely and reliable manner. It ensures that critical systems and data are consistently accessible to those who are permitted to use them.

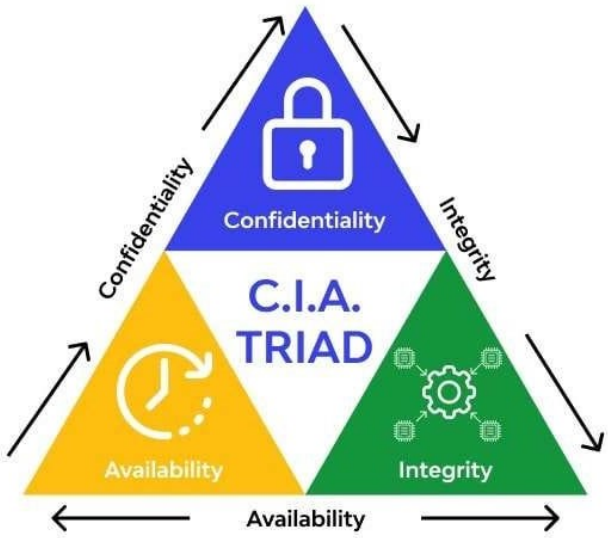


Figure 1.1: The CIA Triad

### 1.2.2 Threats, Vulnerabilities, and Attacks

- **Vulnerability** is a weakness or flaw in information systems or communication networks that could be exploited by an attacker.
- **Risk** means the probability of exposure or loss resulting from a cyber attack or data breach in a system or a network.
- **Threat** refers to any potential event or action that can cause harm to a system, network, or organization. It's the source of potential damage or disruption.

These fundamental terms help define risk and shape defensive strategies in the cybersecurity landscape. A solid understanding of these elements enables analysts and systems to evaluate potential impacts and prioritize security responses [5].

### 1.2.3 Network Security, Endpoint Protection, and Access Control

- **Network Security** refers to the practice of protecting data as it travels across networks, employing tools such as encryption, intrusion detection systems, and secure communication protocols [5], [7].

- **Endpoint protection** involves securing individual devices from malicious threats, often through antivirus software, endpoint detection and response tools, and device hardening [8].


Figure 1.2: Key Components of Endpoint Protection

- **Access control** refers to the set of rules and procedures that govern who has access to a system or to physical or virtual resources. It is the process of granting users access to systems, resources, or information, as well as particular privileges often through credentials like a person's name, passwords, digital fingerprint, or a computer's serial number before granting access. These credentials can take numerous forms in physical systems, but credentials that cannot be transferred provide the best security [5].


Figure 1.3: Access Control Mechanism

### 1.2.4 Traditional Security Mechanisms

Traditional security mechanisms remain fundamental to cybersecurity infrastructures, forming the first line of defense against many threats. Firewalls serve as protective barriers, regulating traffic between trusted and untrusted networks through predefined rules. To enhance monitoring, Intrusion Detection Systems (IDS) continuously analyze network and system activities, detecting suspicious behavior or policy violations. Additionally, antivirus software identifies and removes malicious programs, helping to prevent infec-

tions and support system recovery [5], [7]. Together, these tools create a layered defense strategy that reinforces the overall security posture of digital environments.

## 1.3 Common and Emerging Cybersecurity Threats

### 1.3.1 Common Cybersecurity Threats

Cybersecurity threats encompass a diverse range of malicious activities that compromise the confidentiality, integrity, and availability of information systems [5], [9] and cyber physical systems such as smart grids [10], [11].

Among these, **Malware** (malicious software) represents one of the most pervasive threats, encompassing various forms such as viruses, worms, Trojans, ransomware, and spyware. These software programs are engineered to infiltrate and damage computer systems, exfiltrate sensitive data, or disrupt normal operations without the user's informed consent [6], [9].

**Phishing attacks** constitute another prevalent threat vector, wherein attackers impersonate legitimate entities typically via email, SMS (smishing), or fraudulent websites to deceive users into disclosing confidential information such as authentication credentials, banking details, or personal identification data. These attacks often leverage social engineering tactics and are increasingly sophisticated, making them difficult to detect [12].

A particularly destructive subclass of malware is **Ransomware**, which encrypts the victim's data and renders it inaccessible. The attacker subsequently demands a monetary ransom, often in cryptocurrency, in exchange for the decryption key. Ransomware attacks have been responsible for substantial financial losses and operational disruptions across critical infrastructure sectors [13].

**Denial-of-Service** (DoS) attacks, and their more potent distributed variant (DDoS), aim to exhaust the resources of a targeted system by flooding it with superfluous requests. This causes legitimate users to experience service outages or severe degradation in performance. Such attacks can cripple websites, cloud services, and enterprise networks [14].

**Man-in-the-Middle** (MitM) attacks involve the unauthorized interception and possible manipulation of communications between two parties. By inserting themselves into a data exchange such as an HTTPS session attackers can eavesdrop, inject malicious code, or alter messages without the knowledge of either party [5].

**Zero-day attacks** exploit previously unknown vulnerabilities in software or hardware for which no security patches or mitigations yet exist. Due to the absence of vendor awareness or defensive signatures, these attacks are particularly dangerous and can have wide-reaching implications, especially when targeting widely deployed systems [15].

**Social engineering** refers to a broad category of attacks that exploit human behavior rather than technical vulnerabilities. Through manipulation, deception, or psychological tactics, threat actors persuade users to perform actions such as downloading malware, revealing passwords, or transferring funds. This form of attack highlights the importance of user awareness and training in cybersecurity [16].

Lastly, **insider threats** originate from individuals within the organization who have legitimate access to information systems. These threats may be intentional (malicious insiders) or unintentional (negligent users), and they pose significant challenges to security management due to the trusted status of the actors involved. Detecting and mitigating

insider threats often requires the use of behavioral analytics, access control policies, and robust audit mechanisms [17].


Figure 1.4: Main Cyber Security Threats

### 1.3.2 Emerging Cybersecurity Threats

Emerging cybersecurity threats represent evolving and sophisticated attack vectors that challenge traditional defense mechanisms and demand advanced countermeasures.

Among the most significant of these threats are **Advanced Persistent Threats** (APTs), which are characterized by prolonged, stealthy cyber intrusions. These attacks are typically conducted by highly skilled adversaries, often state-sponsored or affiliated with organized cybercrime groups, aiming to establish unauthorized, long-term access to sensitive systems. APTs are strategically designed to evade detection while extracting valuable data or manipulating systems over extended periods [18].

The integration of Artificial Intelligence (AI) and Machine Learning (ML) into offensive cybersecurity strategies has given rise to **AI-powered Attacks**. Threat actors now leverage AI to conduct automated vulnerability scanning, exploit discovery, and adaptive attack planning. These capabilities enable adversaries to personalize phishing attacks, evade anomaly detection systems, and execute attacks at unprecedented speed and scale [19].

The proliferation of **Internet of Things** (IoT) devices—ranging from consumer electronics to critical infrastructure components—has introduced a wide array of exploitable vulnerabilities. Many IoT devices lack robust security architectures [20], such as firmware protections and proper encryption, rendering them susceptible to unauthorized access [21], botnet recruitment, and exploitation as entry points into broader network environments [22].

As organizations increasingly migrate to cloud computing platforms, a new landscape of risks has emerged. **Cloud Security Issues** often stem from misconfigurations, insufficient access controls, insecure APIs, and shared responsibility misunderstandings. These vulnerabilities are frequently exploited by attackers to access vast repositories of sensitive data and services, especially in hybrid or multi-cloud environments [23].

**Supply Chain Attacks** represent a particularly insidious threat, as they target indirect pathways to compromise well-defended organizations. By breaching trusted third-

party vendors or software providers, attackers can inject malicious code or backdoors into software updates and services. High-profile incidents such as the SolarWinds breach underscore the severe impact and stealth of such tactics [24].

The advent of **Deepfake and Synthetic Media Technologies** poses a unique form of cyber threat. Powered by generative AI models, these tools enable the creation of hyper-realistic yet entirely fabricated audio, video, or text content. Such synthetic media can be weaponized for social engineering campaigns, disinformation operations, identity fraud, or reputational damage—blurring the line between authentic and manipulated digital content [25].

Finally, **Quantum Computing** represents a disruptive technological advancement with profound implications for cybersecurity. While still in developmental stages, quantum computing threatens the integrity of classical cryptographic systems, particularly public-key encryption algorithms. Once quantum computers reach sufficient computational power, they may render current encryption schemes obsolete, necessitating the transition to quantum-resistant cryptography to preserve data confidentiality and integrity [26].

## 1.4 Challenges and Limitations

### 1.4.1 Evolving Threat Landscape

The cybersecurity threat landscape is continuously evolving due to the increasing sophistication and adaptability of threat actors. Cybercriminals frequently alter their techniques, tactics, and procedures (TTPs) to evade detection and countermeasures. This dynamic environment poses significant challenges for security professionals who must continually update their defenses in response to emerging threats, such as advanced persistent threats (APTs), polymorphic malware, and zero-day exploits [27]. Moreover, the proliferation of attack vectors—ranging from traditional phishing to AI-enabled intrusions—requires constant vigilance and investment in adaptive security frameworks.

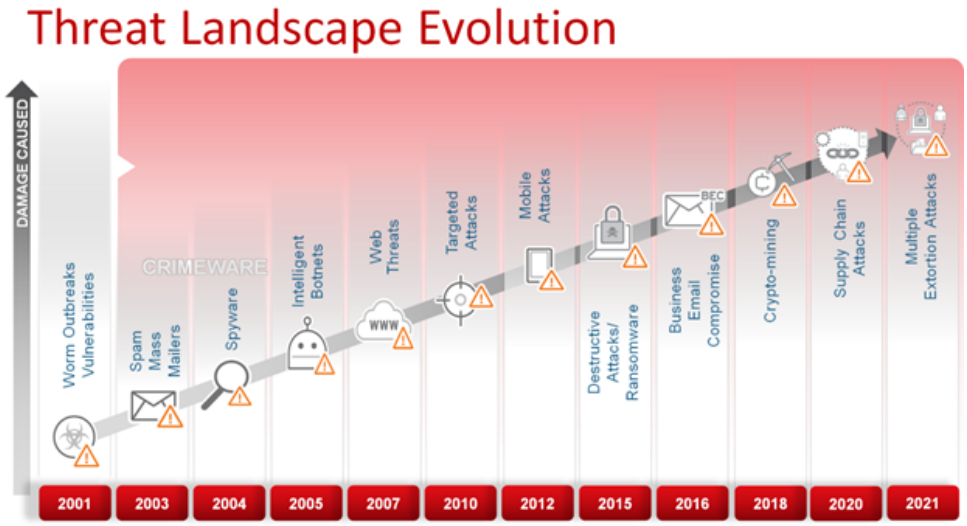


Figure 1.5: Threat Landscape Evolution

### 1.4.2 Human Factor

Despite advancements in cybersecurity technologies, human error remains one of the most exploitable vulnerabilities. Weak password practices, susceptibility to phishing, and mishandling of sensitive information often open doors to unauthorized access [28]. Studies have demonstrated that a significant portion of security breaches originate from user behavior rather than technical flaws. Consequently, enhancing cybersecurity awareness and training among users is imperative to reduce risk exposure.

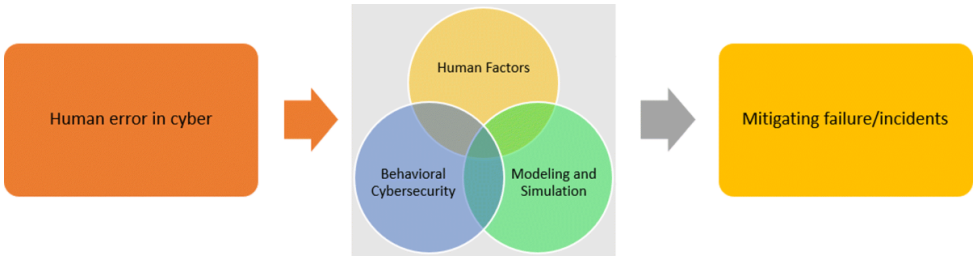


Figure 1.6: The Impact of Human Errors on Cybersecurity

### 1.4.3 Lack of Skilled Professionals

The shortage of qualified cybersecurity professionals is a critical issue globally. This skills gap hampers the ability of organizations to adequately monitor, detect, and respond to threats in real time [29]. The increasing demand for expertise in areas such as penetration testing, digital forensics, and security architecture has not been met by a proportional increase in workforce development. As a result, many organizations operate with understaffed or underqualified security teams, elevating their risk levels.

### 1.4.4 Legacy Systems and Complexity

Legacy systems, often still in use due to cost or compatibility concerns, present substantial vulnerabilities because they lack modern security features and are no longer supported with regular updates or patches. Additionally, the complexity of integrating various security tools and systems can lead to configuration errors and security gaps [30]. The broader an organization's IT infrastructure, the more difficult it becomes to ensure consistent and comprehensive security across all platforms.

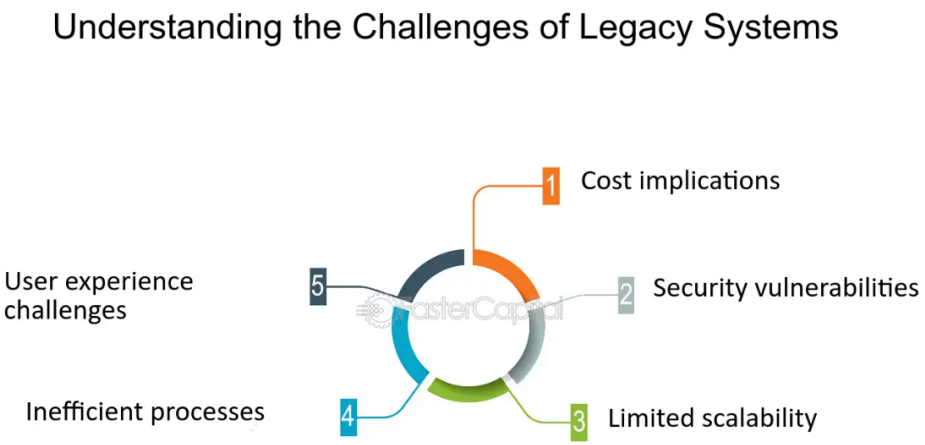


Figure 1.7: Challenges of Legacy Systems

### 1.4.5 Cost and Resource Limitations

Financial constraints significantly affect the cybersecurity readiness of small and medium-sized enterprises (SMEs). Limited budgets often result in insufficient investment in up-to-date security tools, threat detection systems, and expert personnel [31]. This economic imbalance allows well-funded attackers to exploit weaker targets that lack the resilience of larger organizations with robust defense infrastructures.

### 1.4.6 Insider Threats

Insider threats—whether intentional or accidental—pose unique challenges due to the legitimate access held by internal actors. These threats can come from disgruntled employees, negligent staff, or compromised insiders. The detection of such threats is often difficult, as they may not trigger conventional security alerts [32]. Insider attacks have been responsible for substantial data breaches and operational disruptions, underscoring the importance of behavior analytics and access control mechanisms.

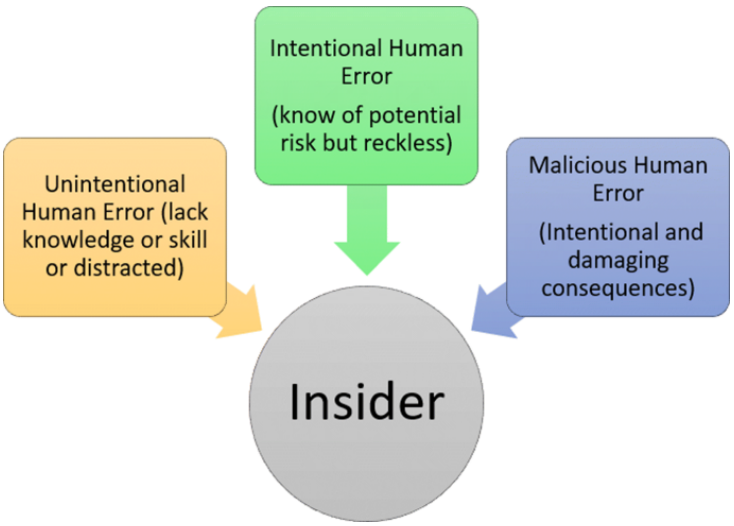


Figure 1.8: Types of insider threats categorized by nature of human error.

### 1.4.7 Privacy and Compliance Challenges

Balancing strong cybersecurity practices with privacy protection and regulatory compliance is increasingly complex. Organizations must navigate laws such as the General Data Protection Regulation (GDPR) and sector-specific standards while maintaining effective threat mitigation strategies [30]. Misalignment between security controls and privacy obligations can result in legal penalties and reputational damage, particularly when sensitive user data is mishandled.

## 1.5 Introduction to Artificial Intelligence in Security

### 1.5.1 Overview of AI

Artificial intelligence (AI) is technology that enables computers and machines to simulate human learning, comprehension, problem solving, decision making, creativity and autonomy. In cybersecurity it refers to the application of intelligent algorithms primarily machine learning (ML), deep learning (DL), and natural language processing (NLP) to enhance digital security infrastructures [33].

**Machine learning** involves creating models by training an algorithm to make predictions or decisions based on data. It encompasses a broad range of techniques that enable computers to learn from and make inferences based on data without being explicitly programmed for specific tasks. There are many types of machine learning techniques or algorithms, including linear regression, logistic regression, decision trees, random forest, support vector machines (SVMs), k-nearest neighbor (KNN), clustering and more. Each of these approaches is suited to different kinds of problems and data. But one of the most popular types of machine learning algorithm is called a neural network. Neural networks are modeled after the human brain's structure and function. A neural network consists of interconnected layers of nodes (analogous to neurons) that work together to process and analyze complex data. Neural networks are well suited to tasks that involve identifying complex patterns and relationships in large amounts of data [34].

**Deep learning** is a subset of machine learning that uses multilayered neural networks, called deep neural networks, that more closely simulate the complex decision-making power of the human brain. Deep neural networks include an input layer, at least three but usually hundreds of hidden layers, and an output layer, unlike neural networks used in classic machine learning models, which usually have only one or two hidden layers. Because deep learning doesn't require human intervention, it enables machine learning at a tremendous scale. It is well suited to natural language processing (NLP), computer vision, and other tasks that involve the fast, accurate identification complex patterns and relationships in large amounts of data. Some form of deep learning powers most of the artificial intelligence (AI) applications in our lives today [35].

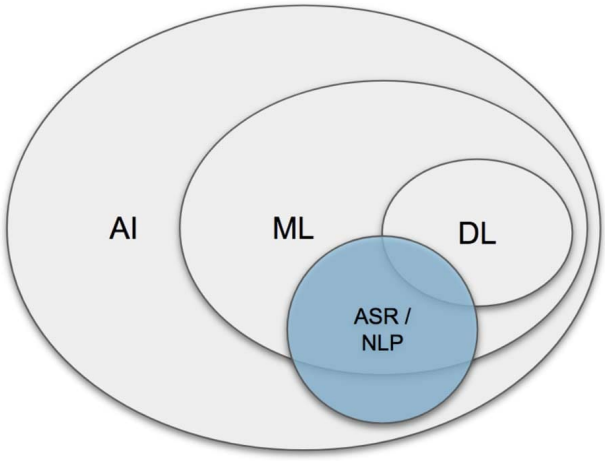


Figure 1.9: Artificial Intelligence: AI vs ML vs DL vs NLP

### 1.5.2 Why AI fits into cybersecurity

As adversaries adopt more dynamic and evasive techniques, the integration of Artificial Intelligence (AI) into cybersecurity systems has emerged as a critical advancement. AI enables intelligent, real-time analysis of vast volumes of data, allowing security frameworks to detect, analyze, and respond to threats with speed and accuracy that far exceed human capabilities. Its application introduces not only speed and efficiency but also scalability across distributed networks and adaptability through continual learning from evolving attack vectors [6].

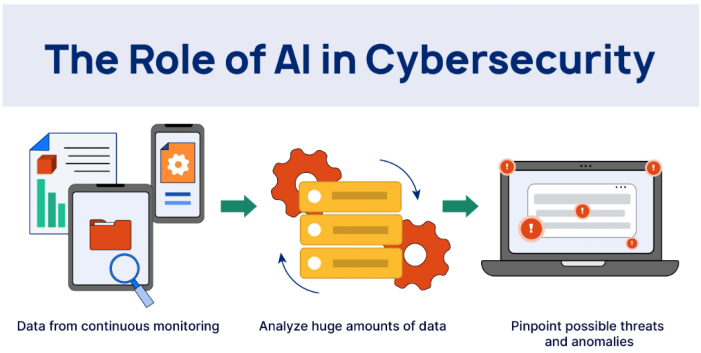


Figure 1.10: Role of Artificial Intelligence in Cybersecurity.

AI-driven security systems and its subfields such as machine learning (ML), deep learning (DL), and natural language processing (NLP) empower cybersecurity tools to identify and adapt to novel threats by recognizing complex patterns, contextualizing behaviors, and automating decision-making processes **ahmed2020machine**. Among these, machine learning plays a foundational role through three primary paradigms each with distinct applications in cybersecurity contexts.

**Supervised learning** utilizes labeled datasets to train models in recognizing predefined threat signatures. This method excels in scenarios involving well-characterized threats such as known malware variants or phishing attempts. The predictability and ac-

curacy of supervised learning make it effective for tasks such as spam filtering, signature-based intrusion detection, and malware classification.

**Unsupervised learning**, in contrast, is designed for situations where labeled data is unavailable. It identifies anomalies and clusters within datasets by detecting deviations from established norms. This approach is particularly beneficial in identifying zero-day exploits, insider threats, and previously unknown attack patterns that evade traditional detection methods. Techniques such as clustering and dimensionality reduction provide insight into complex and hidden threat behaviors [7].

**Reinforcement learning** introduces a dynamic learning framework where agents learn optimal security policies through interaction with an environment, receiving feedback in the form of rewards or penalties. In cybersecurity, this paradigm is increasingly applied in adaptive systems such as autonomous intrusion detection and prevention, dynamic honeypot configuration, and intelligent access control systems. Its strength lies in continuous optimization and real-time decision-making under uncertain or adversarial conditions [36].

Moreover, AI contributes to the automation of repetitive and time-consuming tasks such as log analysis, threat correlation, and incident triage, thus reducing analyst fatigue and improving response times. It also supports proactive risk management by enabling predictive analytics that forecast potential vulnerabilities and attack surfaces before exploitation occurs [37]. NLP is further employed to analyze unstructured threat intelligence sources—including blogs, forums, and dark web activity to extract actionable insights for threat prevention [38].

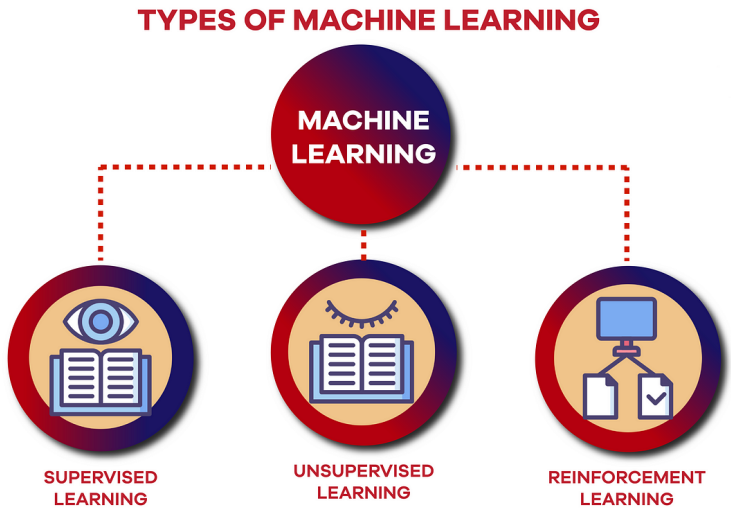


Figure 1.11: Supervised, Unsupervised and Reinforcement.

Ultimately, the integration of AI into cybersecurity represents a transformative paradigm shift from reactive, rule-based defense systems toward proactive, adaptive, and intelligent security architectures. By combining data-driven models, continuous learning mechanisms, and automated threat handling, AI equips organizations with the tools necessary to combat the increasingly complex and dynamic nature of cyber threats. As the digital landscape continues to evolve, the strategic implementation of AI in cybersecurity is not only a technological enhancement but an imperative for resilience in the face of modern cyber warfare.

## 1.6 AI Applications in Cybersecurity

### 1.6.1 Threat Detection and Prevention

AI enhances threat detection by leveraging machine learning algorithms to analyze vast datasets, identifying patterns indicative of malicious activities. This capability allows for the detection of both known threats and novel attack vectors in real-time, surpassing traditional signature-based methods. By continuously learning from new data, AI systems adapt to emerging threats, enhancing proactive defense mechanisms [6].

### 1.6.2 Behavior-Based Anomaly Detection

Through behavioral analytics, AI establishes a baseline of normal user and system behavior. Deviations from this baseline can signify potential security incidents, such as insider threats or compromised accounts. This approach is instrumental in identifying sophisticated attacks that may evade conventional detection techniques [37].

### 1.6.3 AI in Security Operations Centers (SOCs)

In Security Operations Centers, AI facilitates the automation of routine tasks, allowing security analysts to focus on complex threat investigations. AI-driven tools can prioritize alerts based on severity, correlate events across systems, and provide actionable insights, thereby enhancing the efficiency and effectiveness of security operations [6].

### 1.6.4 Automated Incident Response and Fraud Detection

AI enables rapid response to security incidents by automating containment measures, such as isolating affected systems or blocking malicious traffic. In the realm of fraud detection, AI analyzes transactional data to identify anomalies that may indicate fraudulent activities, thereby mitigating financial losses and protecting organizational assets [6].

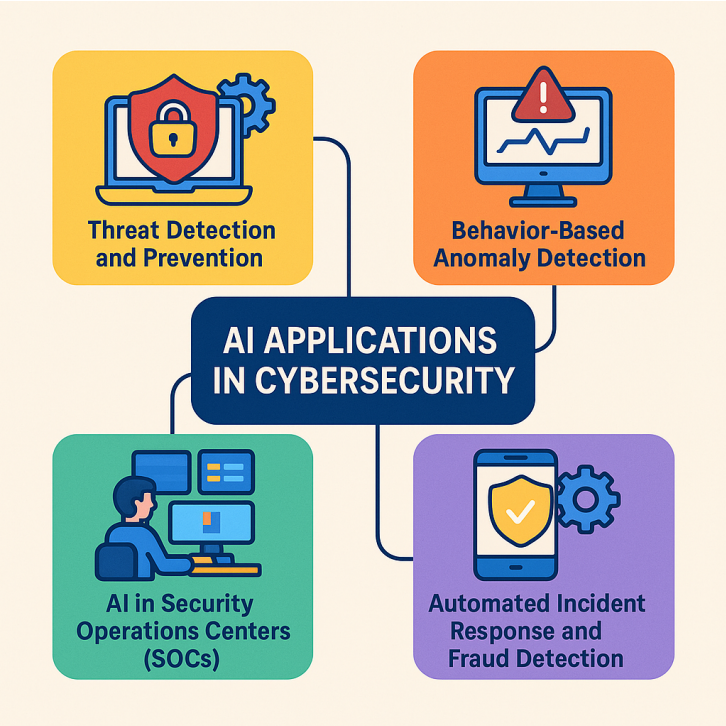


Figure 1.12: The Applications Of AI in Cybersecurity.

## 1.7 Problem Statement and objectives of the project

The proliferation of Artificial Intelligence (AI) in cybersecurity has revolutionized how threats are detected, analyzed, and mitigated. Despite its transformative potential, this integration introduces a spectrum of technical and ethical concerns, particularly around data privacy, centralized control, and operational scalability. Traditional machine learning frameworks in cybersecurity typically rely on centralized architectures where vast amounts of sensitive data must be aggregated in a single repository to facilitate model training. This not only exposes critical information to potential breaches but also conflicts with growing privacy regulations and the strategic need for decentralized infrastructures in data-sensitive environments.

In light of these constraints, this project explores federated learning as a promising paradigm to overcome the inherent limitations of centralized AI systems in cybersecurity. Federated learning enables distributed entities to collaboratively train a global model without sharing raw data, thereby preserving data sovereignty and privacy. Instead of transferring local datasets, only encrypted model updates are communicated between clients and the central server, significantly reducing the risk of data leakage. This decentralized approach aligns well with the requirements of modern cybersecurity operations, where data ownership, regulatory compliance, and cross-organizational trust boundaries must be carefully managed.

The overarching objective of this project is to design, implement, and evaluate a federated learning-based framework tailored to collaborative cyber threat detection. Specifically, the framework aims to harness the power of deep learning while mitigating the privacy risks associated with traditional centralized systems. By facilitating knowledge sharing across distributed sources without compromising data confidentiality, the proposed system aspires to enhance the accuracy, adaptability, and inclusivity of intrusion

detection models.

Moreover, the project seeks to address a set of critical technical challenges often encountered in real-world cybersecurity deployments. These include dealing with non-IID (non-independent and identically distributed) data across clients, class imbalance in threat categories, and high communication overhead inherent in federated environments. Additional emphasis will be placed on evaluating the system's robustness under adversarial conditions, where attackers may attempt to manipulate inputs or poison training updates to degrade model performance. Techniques such as model regularization, secure aggregation, and adversarial resilience strategies will be investigated to fortify the framework.

The scarcity of high-quality labeled datasets also represents a core concern, as cybersecurity data is frequently proprietary, unlabeled, or highly contextual. By enabling organizations to retain control over their data and contribute only encrypted learning signals, the proposed solution offers a viable pathway toward broader collaboration without violating confidentiality agreements or regulatory mandates.

Ultimately, this research endeavors to contribute a scalable, privacy-preserving, and resilient AI framework for modern cybersecurity environments. By merging federated learning with advanced deep learning techniques, the project aims to lay the groundwork for ethical and effective collaborative defense systems capable of responding to the evolving complexity of cyber threats.

## 1.8 Conclusion

This chapter has provided an introduction to the essential concepts of cybersecurity and the role of Artificial Intelligence (AI) in enhancing security measures. It discussed foundational elements like the CIA Triad, common cybersecurity threats, and traditional defense mechanisms, which are still critical in the evolving landscape of digital security. The integration of AI into cybersecurity introduces advanced techniques such as machine learning, deep learning, and natural language processing, offering more efficient, scalable, and adaptive solutions to counter modern cyber threats. Despite its potential, the adoption of AI in cybersecurity comes with significant challenges, including privacy risks, bias in training data, adversarial attacks, and a lack of high-quality, labeled data. Addressing these obstacles requires a careful balance of technological innovation and ethical considerations.

# Chapter 2

# State Of Art

## 2.1 Related work on Distrebuted Denial of service Attacks

Several work has been investigated in Distrebuted Denial of service attacks. This work [39] focuses on addressing the challenges faced by internet-based commercial applications such as Customer Relationship Management (CRM), Supply Chain Management (SCM), online banking, and e-commerce that rely heavily on distributed computing technologies. These applications are primary targets of large-scale DDoS attacks, which aim to deny legitimate users access to services by overwhelming servers with seemingly legitimate but fraudulent requests. In distributed environments, such servers are especially vulnerable to these attacks. Similarly, flash crowds sudden surges in legitimate user traffic triggered by popular eventscan mimic the behavior of DDoS attacks, making accurate detection even more challenging. Distinguishing between DDoS attacks and flash crowds is a complex issue, and current solutions are typically tailored to detect only one or the other. This study proposes a novel technique for accurately differentiating between various types of DDoS attacks and flash crowds, while also suggesting a prevention approach. The effectiveness of the proposed method is validated through simulations conducted using the NS-2 network simulator. While the study presents a novel classification and prevention strategy, its main limitation lies in the use of simulation-only validation (via NS-2), which may not fully reflect real-world network complexities, limiting the practical applicability of its findings.

According to the journal paper by Da Silva et al. [40] provides an extensive survey on one of the most critical and evolving threats on the Internet—Low-rate Denial of Service (LDoS) attacks. These attacks represent a stealthier and more damaging variant of traditional Denial of Service (DoS) attacks. Over the years, attackers have refined their techniques, exploiting vulnerabilities in operating systems and protocols to degrade or entirely deny services to legitimate users. While conventional High-rate DoS attacks overwhelm networks with traffic, modern LDoS attacks mimic legitimate user behavior, making them significantly harder to detect using traditional defense mechanisms. This survey not only consolidates previous research but also introduces a comprehensive taxonomy that categorizes LDoS attacks into three main groups: Quality of Service (QoS) attacks, Slow-rate attacks, and Service queue attacks, based on their operational strategies. It further presents a detailed examination of detection mechanisms and countermeasures tailored to eight specific types of LDoS attacks, emphasizing traffic throttling techniques. Additionally, the paper offers a comparative analysis of various LDoS attack tools. By

compiling and analyzing current methodologies, this work serves as a valuable resource for researchers and network administrators seeking to understand and mitigate the risks associated with LDoS attacks. Despite the comprehensive taxonomy, the key limitation of this survey is its lack of empirical evaluation of the detection and mitigation techniques, leaving their real-world effectiveness unverified.

## 2.2 Related work on Brute Force Attacks

A variety of research efforts have explored Brute Force Attacks. The present study [41] aims to develop a system that leverages graph-based clustering of log data—specifically, k-clique percolation—for the detection and analysis of SSH brute-force attacks. The methodology involves preprocessing SSH log data, constructing an IP interaction network graph, and applying clustering techniques to uncover patterns indicative of malicious behavior. In addition, the integration of a reinforcement learning model, particularly the Q-learning algorithm, enhances the system's predictive capabilities and allows it to continuously adapt to emerging attack vectors. The ultimate objective is to create a robust tool for tracking, assessing, and mitigating SSH brute-force attacks. The primary limitation of this study is its limited evaluation under real-world conditions—the proposed method is validated only on synthetic log datasets, which may not capture the full complexity or variability of live network environments.

As demonstrated in this study [42], information security practices can be significantly enhanced through the application of the CatBoost classifier within the complex field of network intrusion detection. Building on established machine learning approaches for identifying FTP and SSH brute-force attacks, this research introduces an improved CatBoost-based method in response to the increasing sophistication of such threats. A comprehensive analysis of the CSE-CIC-IDS2018 and CICIDS2017 datasets was conducted, incorporating advanced feature selection, hyperparameter tuning, and class balancing using techniques such as SMOTENC. The resulting CatBoost model achieved exceptional performance, with an accuracy rate of 99.9964% and a notably low false alarm rate. Furthermore, it demonstrated robust metrics including a consistently high F1 score, minimal loss, and an almost perfect Matthew's Correlation Coefficient (MCC), indicating strong reliability and precision. A distinguishing element of this work is the extensive use of SHAP (SHapley Additive exPlanations) values, which offer detailed insights into the model's decision-making process—highlighting the influence and contribution of individual features through metrics like Mean Absolute SHAP values and SHAP value distributions. Additionally, the model maintained an average processing speed of 5 million samples per second, illustrating its practicality for real-time security applications. Overall, the findings validate the CatBoost classifier's effectiveness in detecting sophisticated network intrusion patterns, emphasizing its relevance for advancing cyber defense strategies through improved accuracy, efficiency, and interpretability. Despite its impressive performance, the main limitation of this study is its dependence on public datasets, which may not accurately reflect evolving attack patterns or zero-day threats encountered in real-world deployments.

## 2.3 Related work on Web Attacks

Many studies have focused on Web Attacks. As we can observe in this work [43], cybersecurity has become a critical and highly influential component of system security. This research provides an in-depth analysis of SQL injection and its various types, exploring the execution of malicious code to manipulate SQL databases. Additionally, the study evaluates prevention techniques and SQL injection detection methods. A comprehensive comparative analysis of existing approaches, including Classical SQLi, Advanced SQLi, and Deep Learning methods, is presented in the results section. The paper emphasizes the prevalent cybersecurity threat of SQL injection attacks targeting databases, investigates the techniques used to carry out these attacks, examines their impact on systems, and outlines comprehensive strategies for mitigating such vulnerabilities. Although the study covers a wide range of SQL injection techniques and countermeasures, its main limitation is the lack of practical implementation or real-world validation, which reduces its utility for real-time threat detection systems.

As demonstrated in this study [44],the international student website serves as a crucial platform for a university's external communication, enabling prospective international students to gain comprehensive insights into the institution. However, such websites are frequent targets of cyberattacks, particularly Cross-Site Scripting (XSS), which remains one of the most prevalent web application vulnerabilities. This study analyzes XSS attacks and proposes several preventative measures, including input validation, output encoding, explicitly defining output encoding methods, and recognizing the limitations of blacklist-based validation. It also highlights the importance of avoiding the insertion of untrusted data in permitted contexts. Defensive techniques include HTML encoding when placing untrusted data between tags, HTML attribute encoding for attribute values, JavaScript encoding when inserting data into scripts, CSS encoding for style attribute values, URL encoding for HTML URL attributes, and the application of XSS rule engines to sanitize rich text content. These defense strategies collectively help mitigate the risks posed by XSS vulnerabilities in dynamic web environments. While the paper presents a solid theoretical foundation of XSS defense techniques, its primary shortcoming is the absence of experimental evaluation or performance benchmarks, which limits its contribution to applied cybersecurity practices.

As revealed in recent research [45], Modbus remains one of the most widely adopted industrial protocols in Supervisory Control and Data Acquisition (SCADA) systems, serving key functions such as remote device control, physical process monitoring, and data acquisition. However, its lack of built-in security features namely authentication, integrity, and confidentiality renders it highly vulnerable to cyber threats. This inherent insecurity makes industrial systems relying on Modbus attractive targets for adversaries, as exemplified by the Stuxnet incident. In this study, the vulnerabilities of the Modbus protocol are exploited through a stealthy false command injection attack that remains undetected by the SCADA operator. The proposed attack consists of two phases: (1) a pre-attack (offline) phase, during which the attacker passively captures and stores legitimate request-response pairs; and (2) an attack (online) phase, wherein the attacker injects false commands and replies with valid-looking responses from the database to mask the intrusion. If successfully executed, this type of attack could result in catastrophic consequences for SCADA systems and critical infrastructure. To counter this threat, the study also outlines potential mitigation strategies aimed at enhancing the security of Modbus-based environments. Despite its innovative stealth attack model, the study's key

limitation is the lack of implementation and testing of the proposed mitigation strategies, which makes it difficult to assess their effectiveness in real-world SCADA deployments.

## 2.4 Related work on Man in The Middle Attacks

A range of research has been undertaken regarding man-in-the-middle (MITM) attacks. This research [46] specifically focuses on the vulnerabilities of 6TiSCH networks, which are wireless sensor networks built upon Industrial Internet of Things (IIoT) devices. These networks utilize the RPL protocol to ensure energy-efficient and reliable communication. However, MITM attacks on RPL can manipulate routing information, resulting in traffic misdirection, communication blockage, and significant data loss. The study investigates the impact of such attacks on 6TiSCH networks and explores adversarial techniques targeting the RPL protocol. To counter these threats, the proposed method periodically analyzes information collected from network nodes, aiming to identify malicious activity. Given that all nodes are pre-authenticated by a central server, data packets are securely encrypted. By applying both verification procedures and statistical data analysis, the method enables effective detection of MITM attackers. Performance evaluations indicate the method's viability for enhancing security in 6TiSCH-based IIoT environments. While the proposed approach demonstrates promising results in simulation, it relies heavily on a centralized trust model, which may limit scalability and fault tolerance in decentralized or large-scale IIoT deployments.

As demonstrated in this study [47], man-in-the-middle (MITM) attacks pose a severe threat to the integrity and confidentiality of Wi-Fi networks, allowing adversaries to intercept and eavesdrop on wireless communications. These attacks are particularly dangerous due to their ability to extract sensitive information such as login credentials, credit card numbers, and other critical financial data. Despite the existence of various detection mechanisms, MITM attacks continue to occur, leading to significant security breaches. To address this ongoing challenge, the study introduces a suite of machine learning-based techniques designed to detect and accurately identify MITM activities within wireless communication environments. The proposed approach is rigorously evaluated using standard performance metrics and benchmarked against existing machine learning models, demonstrating its effectiveness in enhancing the security posture of wireless networks. Although the machine learning-based detection system achieves high accuracy, the study does not explore its performance under adversarial conditions or evaluate its real-time applicability, which limits practical deployment considerations in dynamic Wi-Fi environments.

As revealed in recent research [48], federated learning presents a promising decentralized paradigm that facilitates collaborative model training across multiple nodes while preserving data privacy by keeping raw data local. Despite these privacy advantages, the architecture remains vulnerable to critical threats, notably man-in-the-middle (MITM) attacks that can compromise model integrity during transmission. To address this challenge, the study introduces CodeNexa, a novel security framework tailored to safeguard federated learning systems against such intrusions. Departing from conventional cryptographic defenses like hash functions, digital signatures, and watermarking, CodeNexa utilizes a dynamic metric verification mechanism. This approach involves computing and securely archiving key evaluation metrics—such as accuracy, precision, recall, and AUC—at a precision level of six decimal places. These metrics are later employed during the model aggregation process to authenticate incoming updates and ensure their legiti-

macy. Empirical evaluations using the MNIST dataset highlight CodeNexa's effectiveness in identifying and discarding tampered model weights, thereby minimizing the risks associated with model poisoning and unauthorized alterations. By enabling continuous model integrity validation across distributed clients, CodeNexa offers a scalable, resilient, and adaptable defense mechanism for federated learning systems in diverse operational environments. While CodeNexa provides an innovative and scalable alternative to traditional cryptographic methods, its reliance on pre-calculated metrics may not generalize well to complex or non-standard datasets, and its computational overhead for continuous metric validation may hinder deployment in resource-constrained edge environments.

## 2.5 Related work on Denial of Service Attacks

Multiple work have been made to explore Denial of Service Attacks. this work [49] investigates the impact of denial-of-service (DoS) attacks on the observability of networked control systems (NCSs), highlighting how such attacks can compromise system functionality. It begins by demonstrating how DoS attacks affect observability, necessitating the reconstruction of the original observability matrix to account for attack characteristics. The study then establishes necessary and sufficient conditions for maintaining observability under arbitrary DoS scenarios, requiring continuous recalculation of the observability matrix. Additionally, it explores the relationship between the observability index and DoS parameters, offering criteria for assessing observability under both periodic and aperiodic attacks. The proposed approach is validated through numerical simulations, confirming its effectiveness and practical applicability. While the paper offers a rigorous theoretical framework, its application is primarily limited to linear systems and controlled environments, potentially restricting its practical scalability to complex or nonlinear real world control systems.

As demonstrated in this study [31], denial-of-service (DoS) attacks pose a substantial threat to network security, capable of disrupting services and inflicting financial damage by overwhelming systems with excessive traffic. These attacks range from basic flooding techniques to more sophisticated distributed approaches, often exploiting vulnerabilities across multiple network layers to increase their impact. Traditional detection methods, which rely on static thresholds, often fall short in identifying these evolving threats and may result in high false positive rates, leading to unnecessary alerts and inefficient resource usage. To address these limitations, this study introduces an enhanced DoS detection system leveraging Pyshark for comprehensive packet analysis and real-time traffic monitoring. The system adapts dynamically to changing traffic patterns by adjusting its thresholds and identifying attacker IP addresses. When a potential threat is detected, the system sends real-time email alerts containing key information such as the attacker's IP and packet count, enabling swift response. This dynamic and responsive approach significantly bolsters network defenses against DoS attacks by ensuring timely detection and intervention. Despite its practical design, the system lacks a comprehensive evaluation against advanced evasion techniques and does not provide a formal analysis of scalability or computational overhead in high-throughput environments.

As revealed in recent research [28], HTTP low and slow DoS attacks represent a subtle yet highly disruptive threat, wherein attackers deliberately send HTTP requests at an extremely slow pace to exhaust server resources. Targeting specific applications or server components, these attacks exploit the behavior of thread-based web servers such as Apache and IIS, causing server threads to remain occupied and ultimately denying access to

legitimate users. Unlike traditional DoS attacks that rely on high volumes of traffic, low and slow attacks are stealthy in nature and evade detection by conventional network-layer defense tools. Recognizing this limitation, the study proposes a novel detection method based on Long Short-Term Memory (LSTM) deep learning architecture. By training on the CIC DoS dataset alongside a synthetically generated dataset, the model demonstrated exceptional effectiveness in identifying these stealthy threats, achieving a detection accuracy of 99%. This approach provides a robust and adaptive solution to the growing challenge of HTTP low and slow DoS attacks. Although the LSTM model performs well in detection accuracy, the study does not address real-time performance, false positive handling, or generalizability across different server architectures, limiting insight into its deployment feasibility in production systems.

## 2.6 Related work on Phishing Attacks

Numerous academic works have been made to explore Phishing Attacks, This effort [50] addresses the growing threat of phishing, a highly effective form of cybercrime wherein attackers deceive individuals to steal sensitive data. Recognized as one of the most prevalent online scams today, phishing leads to substantial losses involving personal information, identity theft, and organizational and governmental security breaches. A common vector for these attacks is the use of phishing websites that mimic legitimate platforms. Attackers craft these deceptive pages and disseminate their links through social media, messaging applications, or spam, exploiting user trust. The escalating number of victims is largely attributed to the limitations of existing security technologies. Prior research has largely focused on isolated phishing techniques and defense mechanisms, often neglecting the complete phishing lifecycle. To bridge this gap, this study introduces a comprehensive model of phishing that encapsulates all critical dimensions—including attack stages, attacker profiles, targets, channels, and tactics. By presenting the full anatomy of phishing attempts, this model enhances public understanding of the duration and progression of such campaigns, laying the groundwork for more effective countermeasures. Given the anonymity of cyberspace and the lack of stringent regulations, phishing attacks continue to thrive. Current detection systems have demonstrated limited effectiveness, underscoring the need for smarter approaches. In response, this work proposes an LSTM-based artificial intelligence detection system, which has shown promising results in both accuracy and performance, offering a more intelligent and adaptive solution for combating phishing threats. While the proposed model contributes valuable insights into the phishing lifecycle and presents a robust detection technique, the study does not delve deeply into real-world deployment scenarios or provide comparative benchmarks with existing anti-phishing solutions, limiting the assessment of its practical applicability.

As discussed in this work [51], phishing remains one of the most pervasive forms of cybercrime, leveraging deceptive techniques to extract personal and sensitive information from unsuspecting users. In the evolving digital landscape, where technological advancements facilitate seamless connectivity and data exchange, the threat of malicious activity continues to grow. Traditional phishing detection methods often fall short in addressing the adaptive and sophisticated strategies employed by attackers. To enhance detection capabilities, this study introduces a novel approach utilizing a hybrid Convolutional Long Short-Term Memory (ConvLSTM) neural network, specifically designed to identify phishing in text-based communications. Central to this approach is the integration of the ConvLSTM architecture within a Federated Learning (FL) framework, which ensures pri-

vacy preservation and accommodates decentralized data across multiple organizations or entities. This design not only safeguards user data but also enables collaborative model training without compromising confidentiality. Experimental results demonstrate that the federated ConvLSTM model surpasses conventional detection techniques in terms of accuracy, precision, and recall. Its ability to generalize across diverse datasets makes it a dependable and scalable solution for phishing detection in an era where data protection and high detection efficacy are paramount. Although the model demonstrates high effectiveness and respects privacy constraints, the work does not address computational costs or the communication overhead introduced by federated learning, which could be limiting factors in large-scale or resource-constrained environments.

## 2.7 Related work on Infiltration Attacks

Several researchers have looked into Infiltration Attacks. this research work [52] addresses the escalating severity of cybersecurity threats, highlighting the growing prevalence of persistent and systematic attacks. Attackers often exploit vulnerabilities in network systems, launching targeted and multi-phase attacks. In response, security personnel must possess effective strategies to capture, analyze, and trace attack information in real-time, minimizing the impact on services. The paper reviews existing multi-stage infiltration attack trapping techniques, including methods based on attack chains, event chains, behavior chains, and honeypot technology. It then proposes a novel approach for trapping multi-stage infiltration attacks, aiming to enhance detection and response capabilities. While the work offers a solid theoretical foundation and classification of infiltration trapping techniques, it lacks experimental validation or simulation-based evaluation, making it difficult to assess the real-world effectiveness and scalability of the proposed framework.

As we can observe in this work [53], the rapid computerization of vehicles has led to the increased interconnectivity of automotive networks with external systems, making automotive security a critical concern. The Controller Area Network (CAN) bus, the most commonly used internal network in vehicles, is facing rising vulnerabilities. With the advancement of networked and autonomous driving technologies, the previously isolated nature of automotive internal control networks is diminishing, and existing security standards are no longer sufficient according to modern benchmarks. This paper provides an overview of current defense strategies against CAN bus infiltrations, highlights the limitations of these strategies, and discusses the challenges associated with enhancing security in automotive embedded systems. Although the paper offers a valuable overview of automotive network vulnerabilities, it does not propose novel detection or mitigation techniques, nor does it experimentally compare existing methods under standardized conditions—limiting its contribution to a mainly analytical level.

## 2.8 Related work on Ransomeware Attacks

Various works have examined Ransomeware Attacks, The present study [54] explores the Internet of Things (IoT), a transformative technology widely adopted across domains such as agriculture, smart economies, homes, and healthcare. IoT systems consist of interconnected devices and diverse sensors that communicate via the internet, forming multi-layered architectures typically comprising perception, network, and application layers. Despite their growing ubiquity and practical utility, these smart devices often lack

robust security measures, rendering them vulnerable to a broad spectrum of cyberattacks. This review provides a comprehensive overview of various threats targeting IoT application layer protocols, with particular emphasis on ransomware and its different forms. By detailing these security challenges, the study underscores the critical need for enhanced protection mechanisms within IoT ecosystems. Although the paper provides a valuable taxonomy and threat landscape overview, its analysis remains high-level, and it does not empirically evaluate proposed protection strategies or recommend concrete mitigation tools, limiting its practical applicability in real-world IoT deployments.

As highlighted in the research conducted by previous scholars [55], ransomware poses a severe and disruptive threat by rendering users incapable of accessing or interacting with their system data. Once ransomware infiltrates a system, it encrypts both utility and system files, effectively halting all user operations. To mitigate such attacks, the implementation of honeypot-based defenses has been proposed. Honeypots act as decoy systems designed to lure malicious entities, capturing and analyzing unauthorized activities within the network. When a suspicious or anonymous packet attempts to infiltrate the system, the honeypot records its behavior. In the case of ransomware activity, where the malware initiates data encryption, the honeypot can detect such malicious behavior. Upon identifying unauthorized encryption, the system is promptly isolated from the network to prevent further spread. This paper analyzes various honeypot techniques and introduces the concept of a specialized honeypot, referred to as the s-honeypot, aimed at enhancing the system's resilience and providing proactive protection against ransomware attacks. The honeypot-based approach demonstrates innovation in proactive detection; however, the study does not extensively address the limitations of false positives or the overhead costs associated with real-time monitoring, which are crucial considerations for deployment in dynamic and large-scale networks.

## 2.9 Related work on Botnet Activity

Several researchers have looked into Botnet Activity. This analysis [56] presents a comprehensive comparative evaluation of various machine learning techniques for detecting and predicting malicious activities associated with IoT botnets, addressing the escalating security concerns driven by the proliferation of Internet of Things (IoT) devices. Utilizing the real-world CICIoT2023 Dataset—which captures diverse device interactions and communication patterns—this study systematically investigates the effectiveness of classifiers such as support vector machines (SVM), k-nearest neighbours (k-NN), Naive Bayes, random forest (RF), logistic regression (LR), and decision trees (DT). A suite of performance metrics including accuracy, precision, recall, F1-score, ROC curve, and confusion matrix is employed to assess the detection capabilities of each model. Additionally, the analysis explores the trade-offs between computational complexity and detection performance, facilitating the identification of the most suitable algorithms for various IoT security contexts. This analysis ultimately supports the development of more resilient and adaptive IoT botnet detection strategies, offering valuable insights for researchers, practitioners, and industry experts dedicated to fortifying IoT environments against evolving cyber threats. While the research effectively benchmarks different machine learning models, it does not integrate real-time detection or adaptive learning mechanisms, which are crucial for countering continuously evolving botnet behaviors. Furthermore, the analysis lacks exploration of ensemble or hybrid models, which could potentially yield superior performance by leveraging the strengths of individual classifiers.

As illustrated in the aforementioned study [57], the rapid expansion of devices connected to the Internet has fueled the growth of the Internet of Things (IoT), yet many of these devices remain inherently insecure, making them vulnerable to a wide range of cyberattacks. In particular, IoT systems have increasingly become targets of intelligent and behaviorally diverse botnet activities, including distributed denial of service (DDoS) attacks, phishing, and spamming. While such botnets have posed serious risks to Internet infrastructure for years, effective network forensic techniques capable of accurately identifying and tracing sophisticated botnet behavior remain limited. Recent efforts have applied classification algorithms, such as decision trees, to model and detect these attacks; however, these approaches often suffer from high error rates in tracking botnet traces. This has driven the development of advanced classification-based network forensic techniques, leveraging network flow identification in conjunction with refined decision tree algorithms like C4.5. Experimental results indicate that this combined approach significantly enhances the accuracy of botnet detection, classification, and tracing within infected IoT networks. Although the study presents a promising improvement over basic classification methods, its evaluation lacks comparative performance data against more modern deep learning or ensemble techniques, which are increasingly relevant in cybersecurity applications. Additionally, the scalability of the proposed solution in large-scale IoT deployments is not fully addressed, leaving its practical implementation feasibility uncertain.

## 2.10 Related work on Port Scanning

Numerous academic works have been dedicated to Port Scanning. This study [58] emphasizes the importance of cybersecurity as a collection of techniques aimed at safeguarding the confidentiality, integrity, and availability of computer data against various threats. It highlights that a scanning attack is not a standalone technique but a two-phase process, where the initial scanning phase involves identifying vulnerabilities in communication channels, followed by the actual attack. Given that ports serve as critical attack surfaces—facilitating data ingress and egress—port scanning plays a pivotal role in locating open ports on networked systems that may be exploited. While numerous prior solutions have focused on detecting slow port scanning attacks, these methods predominantly rely on static time intervals. In contrast, the approach proposed in this paper enables detection not only within fixed time frames but also in scenarios where scanning rates vary gradually. The proposed technique is also capable of analyzing live data streams, enhancing its practical applicability. Additionally, packet-based analysis is utilized to identify various types of port scanning attacks, and the method demonstrates high detection accuracy. Furthermore, the study introduces a classification mechanism that distinguishes between single and parallel port scans based on the number of attempts, effectively differentiating fast scans from slow ones. While the study offers a significant improvement over conventional static-interval detection schemes, it lacks integration with adaptive or learning-based models that could further refine detection under real-world conditions. Additionally, the evaluation appears to be limited to predefined attack patterns, which may not fully reflect the unpredictability and sophistication of modern port scanning tactics in the wild. The scalability and performance under high network load conditions are also not thoroughly explored.

As revealed in recent research [59], port scanning remains a significant and persistent threat within modern communication networks. Often employed as a reconnaissance tool

prior to launching cyberattacks, port scans can also disrupt application performance and reduce overall throughput. This study introduces a novel architecture utilizing sequential neural networks (NNs) to classify network packets, segment TCP datagrams, identify TCP packet types, and detect port scan activities. By leveraging the adaptive learning capabilities of sequential models, the system decomposes the complex detection process into manageable components. Post-classification, an analytical phase is applied to identify scanning behavior. The research demonstrates that neural networks can achieve recognition rates exceeding 99% for both general packet classification and detailed TCP packet analysis. Moreover, empirical evaluation using real NMAP-generated pcap files validates the model's effectiveness in accurately detecting open ports and scan attempts, while maintaining a low false positive rate. Despite its strong performance metrics, the model's dependency on labeled training data may limit its generalizability across diverse network environments. Furthermore, the study does not extensively address the impact of adversarial manipulation or evasion techniques, which are increasingly employed to bypass machine learning-based defenses. The system's computational demands and real-time deployment feasibility, particularly on resource-constrained edge devices, are also left unexplored.

# Chapter 3

# Methodology of the Federated Learning Approach

## 3.1 Introduction

The growing reliance on network-connected systems has significantly increased the surface for potential cyberattacks. Traditional machine learning approaches for intrusion detection often rely on centralized data collection, which raises serious concerns regarding user privacy, data security, and regulatory compliance. Moreover, centralized solutions face scalability limitations and are unsuitable for heterogeneous environments such as edge devices and IoT networks.

Federated learning emerges as a promising paradigm to address these limitations by enabling collaborative model training without exposing raw data. This decentralized approach ensures data sovereignty and privacy while maintaining high detection performance. By integrating deep learning techniques within a federated architecture, it becomes possible to develop intelligent and adaptive intrusion detection systems capable of learning from diverse, distributed, and privacy-sensitive datasets.

## 3.2 Investigated Security Threats

The rapid expansion of digital infrastructure has introduced a wide range of cybersecurity threats that compromise the confidentiality, integrity, and availability of networked systems. In order to develop robust intrusion detection mechanisms, it is essential to understand the nature and behavior of these threats. This section provides an in-depth overview of the main categories of attacks investigated in this study.Each threat type is examined with respect to its operational method, impact on network resources, and potential indicators that can be leveraged for detection and mitigation.

### 3.2.1 Brute Force Attacks

- **FTP Brute Force:** A File Transfer Protocol (FTP) brute force attack is a method where an attacker systematically attempts a large number of username-password combinations to gain unauthorized access to an FTP server. The attack leverages automated tools to guess valid credentials by exploiting weak authentication mechanisms. Once access is granted, attackers can exfiltrate data, modify files, or use the server as a launch point for further attacks [60].

- **SSH Brute Force:** Secure Shell (SSH) brute force attacks involve repeated login attempts using various combinations of usernames and passwords on SSH-enabled systems. Although SSH is cryptographically secure, brute force attempts can succeed when weak or default credentials are used. These attacks often target misconfigured or poorly secured systems and are a common precursor to lateral movement within a network [60].

### 3.2.2 Denial of Service (DoS) Attacks

- **HTTP Denial of Service (DoS) :** An HTTP DoS attack targets web servers by sending an overwhelming number of HTTP requests. These requests are often legitimate-looking, making detection and mitigation difficult. The aim is to exhaust server resources, leading to service unavailability for legitimate users [61].

- **Slowloris :** Slowloris is a type of application layer DoS attack that keeps many connections to the target web server open and holds them open as long as possible. It does this by sending partial HTTP requests and then periodically sending subsequent HTTP headers, preventing the server from closing these connections. This consumes server resources and can eventually exhaust the maximum concurrent connection limit [61].

- **Slow HTTP Post:** This attack sends a legitimate HTTP POST request in a slow manner by deliberately sending the body of the request in small fragments. The target server, expecting the full content, allocates resources and waits indefinitely, leading to denial of service when executed at scale [61].

- **GoldenEye :** GoldenEye is a tool-based HTTP DoS attack that sends large volumes of HTTP GET or POST requests to a server with randomized headers. The aim is to exhaust the server's resources, and unlike Slowloris, GoldenEye uses multithreading to amplify the attack's intensity [62].

### 3.2.3 Web Attacks

- **Web Attack – Brute Force:** This category includes automated attacks against web application login interfaces, attempting multiple password combinations to gain unauthorized access. Attackers exploit predictable user behavior and weak passwords [60].

- **Web Attack – SQL Injection:** SQL Injection (SQLi) involves inserting or manipulating SQL queries via web input fields. By injecting malicious SQL code, attackers can bypass authentication, access or modify data, and execute administrative operations on the database server [63].

- **Web Attack – Cross-Site Scripting (XSS):** XSS is a client-side code injection attack where malicious scripts are injected into otherwise benign and trusted websites. These scripts execute in the victim's browser, enabling session hijacking, data theft, or redirection to malicious websites [64].

### 3.2.4 Infiltration Attacks

Infiltration attacks focus on bypassing perimeter defenses to introduce malware or gain unauthorized access to internal network resources. These attacks are often conducted through phishing emails, malicious file downloads, or exploitation of known vulnerabilities, enabling attackers to persist within the target system and exfiltrate sensitive data [64].

### 3.2.5 Port Scanning

- **Port Scan – Service Scan:** Service scans identify active services running on open ports by sending crafted packets and analyzing the responses. These scans provide information about operating systems, software versions, and potential vulnerabilities, often serving as a reconnaissance phase for targeted attacks [65].
- **Port Scan – SYN Scan:** SYN scanning, also known as half-open scanning, involves sending TCP SYN packets to various ports and analyzing the responses (SYN-ACK or RST). It is a stealthy method for identifying open ports because it does not complete the TCP handshake, making it less likely to be logged by intrusion detection systems [65].

### 3.2.6 Distributed Denial of Service (DDoS)

DDoS attacks are large-scale versions of DoS attacks that leverage multiple compromised devices to flood a target with traffic or requests, exhausting its resources and rendering it inaccessible. Attack vectors may include volumetric attacks, protocol attacks, and application-layer attacks. These attacks are coordinated through botnets or malware-infected hosts [66].

### 3.2.7 Botnet Traffic

Botnet traffic originates from networks of compromised devices controlled by a central entity, often used for coordinated malicious activities including spam campaigns, DDoS attacks, and credential stuffing. Botnets can communicate via command-and-control (C2) servers or peer-to-peer structures to receive instructions and update payloads [67].

### 3.2.8 Distributed Reflection Denial of Service (DRDoS) Attacks

DRDoS attacks exploit the amplification effect of misconfigured servers (e.g., DNS, NTP, SSDP) to reflect traffic to a target. The attacker sends small forged requests to these servers with the victim's IP address, prompting large responses that overwhelm the victim. This results in effective bandwidth amplification with minimal resource use from the attacker [66].

### 3.2.9 Heartbleed Attack

Heartbleed is a critical vulnerability in the OpenSSL cryptographic library (CVE-2014-0160). It allows attackers to read arbitrary memory from servers using vulnerable versions of OpenSSL. Exploiting this flaw, attackers can access sensitive information such as private keys, user credentials, and session cookies, compromising the confidentiality and integrity of secure communications [68].

## 3.3 Data Presentation

The development of effective intrusion detection systems (IDS) requires access to realistic, high-quality datasets that accurately represent modern network traffic patterns and diverse cyber threats. For this research, the CIC-IDS-2017 dataset, developed by the Canadian Institute for Cybersecurity (CIC), is utilized. It offers a comprehensive, labeled collection of network traffic data simulating real-world usage combined with a wide spectrum of attack scenarios. This dataset has become a benchmark for cybersecurity research due to its scale, diversity, and relevance [69].

### 3.3.1 Dataset Overview

The CIC-IDS-2017 dataset was generated over a seven-day period in a controlled environment that mimicked realistic enterprise-level network behavior. It includes normal user activities such as web browsing, email communication, VoIP, video streaming, file transfers, and chat applications. These activities were designed using behavior profiles based on real-world data collected from institutions like ISCX and Statistics Canada. Simultaneously, multiple types of cyberattacks were injected at specified times and from designated IP addresses to ensure accurate labeling and ground-truth tracking [70].

The dataset contains approximately 2.8 million network flows, each described by 80+ features, and labeled as either BENIGN or one of fourteen distinct attack types. The attacks were grouped into categories such as brute force, denial of service (DoS), distributed denial of service (DDoS), infiltration, botnet activity, and web-based exploits.

### 3.3.2 Data Collection Process

The network traffic was captured using the CICFlowMeter tool, which extracts bidirectional NetFlow-style data from packet captures (PCAP files). This tool records both header and statistical flow features while preserving timestamps and flow direction. The traffic was generated using a testbed composed of virtual machines and physical servers running a variety of services (e.g., FTP, SSH, HTTP, HTTPS, MySQL) and client-side interactions to simulate genuine enterprise activity.

Attack traffic was generated using known tools such as Hulk, GoldenEye, Slowloris, Xerxes, Metasploit, and Nmap, providing a wide spectrum of signature- and behavior-based attack profiles. This ensures that the dataset captures not only volumetric attacks but also low-and-slow stealthy techniques [71].

### 3.3.3 Feature Composition

Each record in the dataset corresponds to a unique bidirectional flow and is represented by over 80 numerical and categorical features, derived from packet-level statistics. These features can be categorized into the following groups:

- **Basic Connection Features:** Flow duration, total forward and backward packets, source and destination IPs, ports, and protocol.
- **Content Features:** Number of packets with FIN, SYN, RST, PSH, ACK, URG, and CWE flags; header length; and packet content characteristics.

- **Time-Based Features:** Flow inter-arrival times, active and idle durations, and packet timing statistics in both directions.
- **Statistical Features:** Minimum, maximum, mean, and standard deviation of packet lengths and inter-arrival times, along with flow byte/packet rates.
- **Direction-Based Features:** Flow features calculated separately for forward and backward streams to capture asymmetrical behavior typical of certain attacks.

These features are essential for identifying temporal and behavioral patterns in network traffic, which are vital for the accurate detection of both known and zero-day attacks [70], [71].

### 3.3.4 Labeling and Data Organization

The dataset is labeled based on synchronized schedules between traffic generation and attack execution. Each attack was launched during specific time windows, and the IP addresses of the attacker and victim machines were logged to ensure precise labeling. The final output is organized into daily CSV files—one per day of traffic collection—with each row corresponding to one network flow. This structure supports both time-series analysis and classification tasks.

The dataset also includes metadata and documentation, allowing researchers to associate each attack instance with the corresponding tool, method, and expected behavior, enhancing reproducibility and interpretability [69].

### 3.3.5 Statistical Overview

The full dataset includes more than 2.8 million records across seven days of capture. Some days focus primarily on normal traffic, while others are attack-heavy. The largest volume of attack data corresponds to DoS Hulk, which alone comprises more than 1.5 million entries, contributing to the dataset's known class imbalance. On the other hand, attacks such as Infiltration and Heartbleed are sparsely represented with only a few hundred samples each, making them valuable for testing model robustness against minority classes [71].

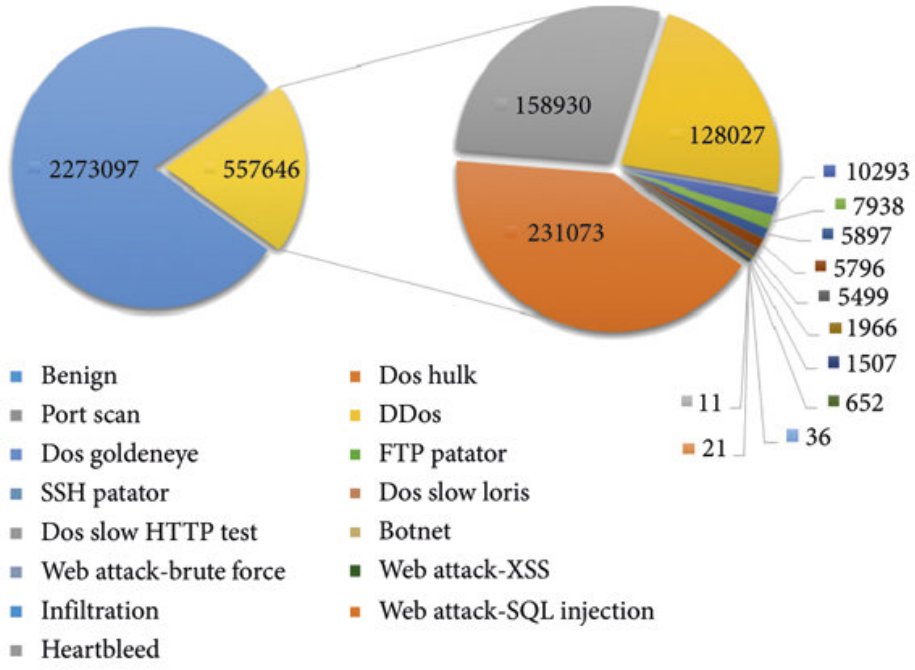


Figure 3.1: Distribution of Labels in the Dataset.

## 3.4 Data Preprocessing

Robust data preprocessing is essential to ensure the quality, integrity, and usability of the dataset prior to applying any machine learning techniques. In this work, the CIC-IDS2017 dataset was preprocessed in a multi-stage pipeline designed to clean, normalize, structure, and prepare the data for both centralized and federated learning scenarios. This section describes the applied preprocessing steps in detail.

### 3.4.1 Initial Dataset Import and Inspection

The raw CSV dataset was ingested using the `pandas` library [72]. The first operation involved stripping all column names of leading and trailing whitespaces, which are known to cause inconsistencies in programmatic access and analysis. An inspection of the column names was also performed to verify the presence of essential attributes such as the `Label` column, which denotes the ground truth class for each record.

### 3.4.2 Handling Missing and Infinite Values

Real-world datasets, especially those collected from network traffic environments, are prone to missing and corrupted values. To handle these issues:

- All instances of positive and negative infinity were first replaced with `NaN` placeholders.
- Numerical columns with missing values were then imputed using mean imputation, ensuring no null entries remained in the feature space.

This step preserved the size of the dataset while mitigating the impact of incomplete samples, thus improving generalization and model stability.

### 3.4.3 Non-Numerical Feature Elimination

To ensure compatibility with neural network architectures, all non-numeric features were removed, with the exception of the `Label` column. Specifically, features of type `object` or categorical string values were discarded, as they could not be directly processed by standard numerical models without prior encoding. Additionally, if a `Timestamp` column was detected, it was dropped due to its redundancy in temporal inference and unsuitability for statistical learning without further transformation.

### 3.4.4 Label Transformation and Integrity Verification

The `Label` column, which indicates the ground truth class for each record, was first inspected to ensure its presence and consistency across all instances. A binary transformation was then applied to standardize the classification task. The CIC-IDS2017 dataset contains multiple string-based labels representing distinct types of attacks and benign traffic. For binary classification:

- All attack-related labels were unified and encoded as `1`.
- The label `"BENIGN"` was encoded as `0`.

This transformation enabled the use of binary cross-entropy loss and facilitated compatibility with sigmoid-activated output layers in neural architectures. All labels were cast to the `int64` type to ensure compatibility with TensorFlow operations [73]. Finally, the dataset was split into an input feature matrix (`X`) and a binary target vector (`y`) to prepare for supervised learning.

### 3.4.5 Dimensional Consistency and Validation

Following the cleaning phase, a critical dimensionality check was performed to confirm that the feature matrix contained exactly 20 columns. This verification step was necessary to ensure the dataset conformed to the expected model input specification. Any deviations triggered a hard stop in the pipeline, preserving the integrity of the preprocessing workflow.

### 3.4.6 Feature Scaling and Normalization

To standardize the scale of feature values and enhance the convergence behavior of neural models, all input features were normalized using a Min-Max normalization strategy. This technique scales each feature individually to the range $[0, 1]$, defined as:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

This scaling procedure reduces bias toward features with larger magnitudes and ensures numerical stability during gradient-based optimization.

### 3.4.7 Train-Test Stratified Partitioning

In preparation for model training and validation, the dataset was split into training and testing subsets with an 80/20 ratio. Stratified sampling was employed to maintain the original class distribution across both partitions. This stratification prevents class imbalance from biasing the evaluation metrics and ensures fair generalization assessments.

### 3.4.8 Data Type Conversion and Final Checks

To guarantee computational efficiency and hardware compatibility particularly in GPU environments all feature vectors were converted to the `float32` data type. This conversion reduced memory overhead and improved matrix multiplication speed in the training process. Final integrity checks confirmed the successful completion of all preprocessing operations before the data was dispatched to the modeling stage.

This preprocessing pipeline ensures that the dataset is clean, balanced, and optimized for high-performance learning in both centralized and federated environments.

## 3.5 Feature Selection

Feature selection plays a critical role in intrusion detection systems (IDS), particularly when working with high-dimensional datasets such as CIC-IDS2017. A well-designed selection process not only improves computational efficiency and model interpretability but also enhances generalization by reducing noise and minimizing overfitting. In this study, we adopted a univariate filter-based approach using the `SelectKBest` method from the `scikit-learn` library [74].

### 3.5.1 Motivation and Rationale

The CIC-IDS2017 dataset contains a large number of numerical features, many of which are correlated, redundant, or irrelevant for intrusion classification. Feeding such features into a neural model without dimensionality reduction can lead to increased training time, poor convergence, and degraded accuracy. To address this, we evaluated several selection strategies and chose `SelectKBest` due to its simplicity, scalability, and ability to work independently of the underlying machine learning model.

Unlike wrapper or embedded methods, which are computationally expensive and model-dependent, filter methods like `SelectKBest` apply statistical techniques to evaluate each feature independently of the classifier. This allows for a fast, interpretable, and easily reproducible selection mechanism that is ideal for preprocessing in both centralized and federated learning settings [75].

### 3.5.2 Algorithmic Overview of SelectKBest

The `SelectKBest` method operates by ranking features according to a specified scoring function and selecting the top $k$ features with the highest scores. Formally, for a feature set $\{x_1, x_2, ..., x_n\}$ and a target vector $y$, the algorithm proceeds as follows:

1. Compute a score $s_i = f(x_i, y)$ for each feature $x_i$ using a scoring function $f$.

2. Sort the features based on their scores in descending order.

3. Retain the top $k$ features with the highest scores.

In our implementation, the scoring function $f$ is given by `mutual_info_classif`, which computes the mutual information between each feature and the target variable [74].

### 3.5.3 Mutual Information as a Scoring Function

Mutual Information (MI) quantifies the amount of shared information between two random variables. For a feature $X$ and a class label $Y$, MI is defined as:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left( \frac{p(x,y)}{p(x)p(y)} \right)$$

Unlike correlation-based methods, mutual information can capture both linear and non-linear dependencies. This makes it highly suitable for network traffic data, which often exhibit complex relationships between input features and attack types. The `mutual_info_classif` function estimates MI using non-parametric entropy estimators, making it robust to noise and applicable to both continuous and discrete features [76].

### 3.5.4 Application and Outcome

The dataset was first normalized using Min-Max scaling [74], and the transformed features were then passed to `SelectKBest` with $k = 20$ as the desired number of features. The resulting selection included the most informative features across various traffic flow metrics, packet-level statistics, and window sizes. The selected features are:

- Destination Port
- Flow Duration
- Total Length of Fwd Packets
- Fwd Packet Length Max
- Fwd Packet Length Mean
- Bwd Packet Length Max
- Flow Bytes/s
- Flow Packets/s
- Flow IAT Mean
- Flow IAT Max
- Fwd IAT Mean
- Fwd IAT Std
- Bwd IAT Max

- Fwd Header Length
- Bwd Packets/s
- Max Packet Length
- Packet Length Mean
- Packet Length Variance
- Init_Win_bytes_forward
- Init_Win_bytes_backward

These features span both flow-based and statistical descriptors, offering a diverse and high-informative representation of network behavior. Their selection significantly reduced the feature space while retaining the essential discriminative characteristics needed for accurate binary classification.

### 3.5.5 Justification of Method Selection

`SelectKBest` with mutual information was chosen over other feature selection methods due to its:

- **Model-independence:** It does not rely on a specific machine learning algorithm, ensuring that the feature selection step remains reusable and modular.
- **Non-linear sensitivity:** MI captures complex dependencies that simpler techniques such as correlation or variance thresholding cannot detect.
- **Computational efficiency:** Its runtime is linear in the number of features, making it scalable to high-dimensional network traffic data.
- **Reproducibility:** The deterministic nature of `SelectKBest` ensures that the same input always leads to the same selected features.

This approach is particularly advantageous in intrusion detection, where the dimensionality of features may fluctuate, but real-time detection performance must remain stable, fast, and interpretable.

## 3.6 Federated Learning-Based Approach

### 3.6.1 Introduction

In the realm of cybersecurity threat detection, data privacy is paramount. Traditional centralized machine learning approaches necessitate aggregating data from multiple sources into a single repository, posing significant privacy risks. Federated Learning (FL) emerges as a solution, enabling the development of robust models without compromising data confidentiality. By allowing individual clients to train models locally and share only model updates, FL ensures that sensitive data remains on-premise, aligning with privacy regulations and reducing the risk of data breaches [77].

### 3.6.2 Federated Learning Process

Federated Learning (FL) is an iterative and decentralized machine learning paradigm that enables multiple clients to collaboratively train a shared model under the orchestration of a central server while keeping the raw training data localized on edge devices. This approach enhances data privacy, reduces communication costs, and supports learning in scenarios with data heterogeneity and distribution constraints. The overall process comprises several fundamental stages [77], [78]:

1. **Initialization:** The FL process begins with the initialization of a global model by the central server. This model is either trained on a small public dataset, randomly initialized, or pre-trained on a relevant task. The initialized model parameters are then broadcasted to a selected subset of participating clients. This step sets the foundation for the collaborative training process.

2. **Client Selection and Configuration:** Due to resource limitations and communication overhead, not all clients may participate in every training round. The server selects a fraction of available clients, often based on criteria such as availability, computational capability, or network reliability. The selected clients receive the current global model along with training hyperparameters.

3. **Local Training:** Each selected client performs model training using its own local dataset for a predefined number of epochs or steps. This decentralized training phase ensures that sensitive data remains on the device, aligning with privacy-preserving goals. During this phase, clients compute updates to the model parameters (e.g., gradients or full weight deltas) using stochastic gradient descent (SGD) or other optimization algorithms.

4. **Model Update Transmission:** Upon completing local training, clients transmit their model updates to the central server. These updates typically include the difference between the locally trained model and the initial global model or the computed gradients. The exchanged information is usually encrypted or anonymized to further protect client data privacy and integrity.

5. **Aggregation:** The server collects the updates from the participating clients and aggregates them to generate a new global model. The most common aggregation strategy is Federated Averaging (FedAvg), where the server computes a weighted average of the client updates, with weights proportional to the size of each client's local dataset. More advanced strategies may incorporate trust levels, historical performance, or robust aggregation techniques to handle stragglers or malicious updates.

6. **Model Redistribution and Convergence Monitoring:** The updated global model is redistributed to clients for the next round of local training. This iterative process continues for several rounds until a convergence criterion is met, such as minimal changes in model accuracy, loss stabilization, or reaching a predefined number of communication rounds. Throughout the process, optional centralized or federated evaluation may be performed to monitor model performance on a validation dataset.

7. **Final Model Deployment:** Once convergence is achieved, the final global model can be deployed for inference across clients or in a central location. Depending on the application, the model may also be fine-tuned or calibrated further to address any remaining domain-specific challenges.

This iterative FL process ensures that knowledge is collectively distilled from decentralized and potentially non-IID (independent and identically distributed) data sources without directly accessing private information, thereby addressing key challenges in modern machine learning applications involving privacy, scalability, and data ownership.

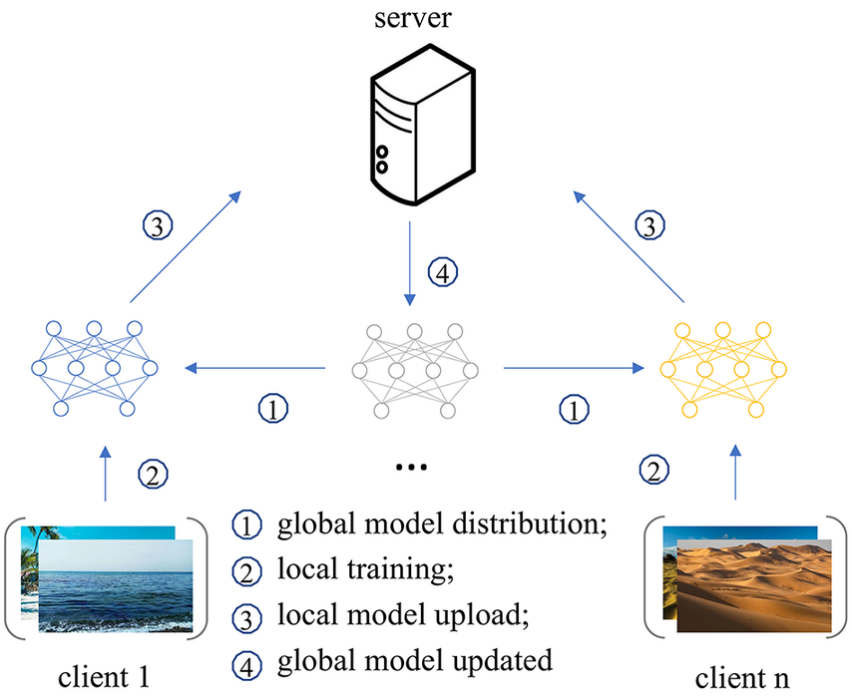


Figure 3.2: The framework of Federated Approach.

### 3.6.3 Federated Learning Architectures

Federated Learning (FL) can be systematically categorized according to the distribution of data among participants and the underlying system topology that governs communication and model aggregation. These architectural classifications directly impact the design choices, communication efficiency, privacy guarantees, and applicability of FL across various domains.

Data Distribution Paradigms:

Data heterogeneity is one of the defining characteristics of FL environments [79]. Based on how data is distributed across participating entities, FL can be categorized into the following paradigms:

- **Horizontal Federated Learning (HFL):** Also referred to as sample-based FL, this architecture is suitable when different clients share the same feature space but possess different data instances. For example, multiple hospitals located in different regions may collect similar types of patient health data (e.g., same features like age,

blood pressure, or diagnosis categories), but the patient populations differ. In such settings, model updates benefit from diverse but semantically similar datasets.

- **Vertical Federated Learning (VFL):** Also known as feature-based FL, this paradigm applies when different entities hold data with distinct feature spaces but on the same set of data instances. An example scenario would be a bank and an e-commerce company that serve the same customer base but collect different types of information. VFL typically requires secure entity alignment (e.g., using privacy-preserving record linkage) and encrypted computation methods such as homomorphic encryption or secure multiparty computation.
- **Federated Transfer Learning (FTL):** FTL is designed for situations where clients differ both in terms of feature space and data instances. It leverages transfer learning techniques to facilitate knowledge sharing across heterogeneous datasets. FTL is particularly useful when overlapping data is minimal or nonexistent, and it requires sophisticated mapping functions or embedding strategies to align learned representations across domains.

System Topologies:

The design of communication and coordination mechanisms in FL significantly influences its scalability, fault tolerance, and privacy robustness [80]. Based on the organizational structure of the participating entities and servers, FL can be categorized into the following system topologies:

- **Centralized Federated Learning:** This is the most widely adopted architecture, wherein a central server orchestrates the entire training process. The server is responsible for client selection, model distribution, update aggregation, and convergence monitoring. Despite its simplicity and efficiency, centralized FL introduces a potential single point of failure and may become a performance bottleneck in large-scale systems.
- **Decentralized Federated Learning:** In this architecture, there is no central coordinating server. Instead, clients communicate directly in a peer-to-peer manner, often structured using overlay networks or blockchain-based infrastructures. Each client exchanges model updates with its peers, and consensus protocols or gossip algorithms are employed for aggregation. While decentralized FL enhances robustness and reduces reliance on a central entity, it presents challenges in synchronization, model consistency, and communication overhead.
- **Hierarchical Federated Learning:** This topology introduces an intermediate layer—typically composed of edge servers or regional aggregators—between clients and the central server. Clients send updates to their respective edge nodes, which perform partial aggregation before forwarding the results to a global server. Hierarchical FL is particularly well-suited for large-scale or geographically distributed networks, such as industrial IoT systems, where edge servers reduce latency, balance loads, and enable local adaptation.

These architectural variants allow FL to be flexibly adapted to a wide range of real-world scenarios, addressing diverse data distributions, infrastructure limitations, and privacy requirements. A clear understanding of these paradigms is essential when designing federated systems tailored to application-specific constraints and goals.

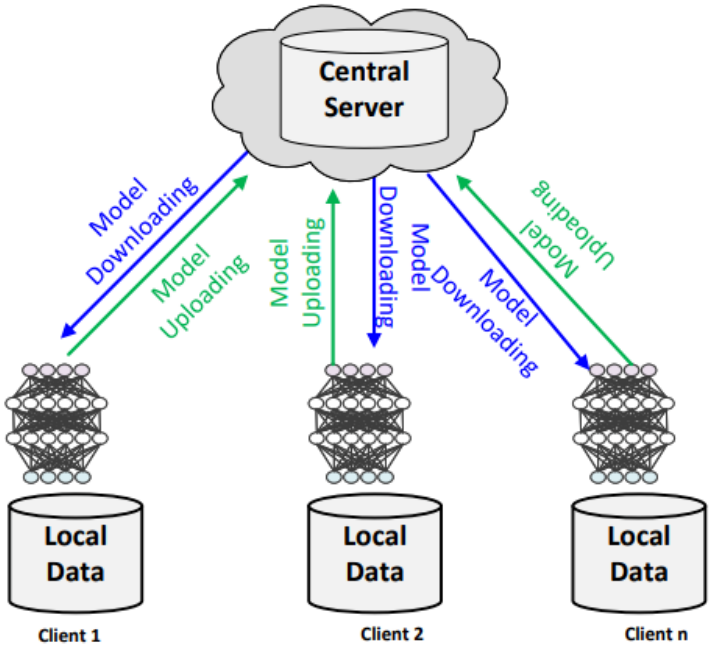


Figure 3.3: Federated Learning Architecture.

### 3.6.4 Federated Learning Algorithms

#### 3.6.4.1 Federated Averaging (FedAvg)

FedAvg is the foundational algorithm in FL, where each client performs local training and the server averages the model updates. Mathematically, the global model updates [77] at round $t$ is:

$$w^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^{(t+1)}$$

where $w_k^{(t+1)}$ is the model from client $k$, $n_k$ is the number of data points at client $k$, and $n = \sum_{k=1}^{K} n_k$.

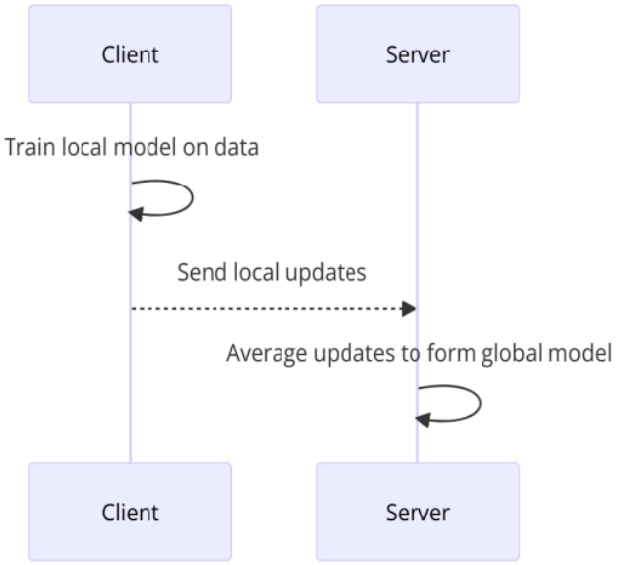


Figure 3.4: FedAvg Algorithm.

#### 3.6.4.2 Federated Proximal (FedProx)

FedProx extends FedAvg by adding a proximal term to the local objective, addressing issues of data heterogeneity [80]:

$$\min_w f_k(w) + \frac{\mu}{2}\|w - w^{(t)}\|^2$$

where $f_k(w)$ is the local loss function, $w^{(t)}$ is the global model at round $t$, and $\mu$ is a regularization parameter.

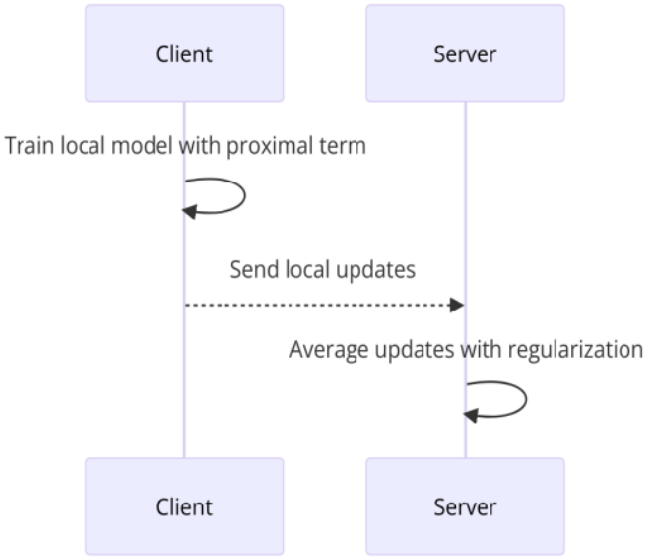


Figure 3.5: FedProx Algorithm.

#### 3.6.4.3 Other Variants

- **FedNova:** Normalizes updates to address client heterogeneity [81].

- **SCAFFOLD:** Uses control variates to correct client drift [82].
- **FedOpt:** Applies advanced optimization techniques like Adam or Yogi at the server level [83].

### 3.6.5 Proposed Model

#### 3.6.5.1 Model Overview

The proposed model is a deep feedforward neural network designed for binary classification tasks in cybersecurity threat detection. It consists of multiple dense layers with regularization mechanisms and normalization techniques to enhance generalization and learning stability. This model is referred to as a **Deep Regularized Feedforward Network (DRFN)**. The architecture is optimized specifically for tabular data extracted from the CIC-IDS2017 dataset, featuring 20 numerical input features [84].

#### 3.6.5.2 Architectural Design

The architecture of the DRFN is composed of four main hidden layers with ReLU activations, each followed by batch normalization and dropout layers. The number of neurons decreases progressively, forming a pyramid-shaped architecture, which encourages hierarchical feature learning and prevents overfitting in deeper layers. The output layer uses a sigmoid activation function to produce a binary class prediction. The model is trained with the AdamW optimizer, which integrates adaptive learning rates with weight decay for improved generalization. The complete architecture is summarized below:

- **Input Layer:** 20 neurons (number of features)
- **Hidden Layer 1:** Dense(64) + ReLU + BatchNorm + Dropout(0.5)
- **Hidden Layer 2:** Dense(32) + ReLU + BatchNorm + Dropout(0.4)
- **Hidden Layer 3:** Dense(16) + ReLU + BatchNorm + Dropout(0.3)
- **Output Layer:** Dense(1) + Sigmoid

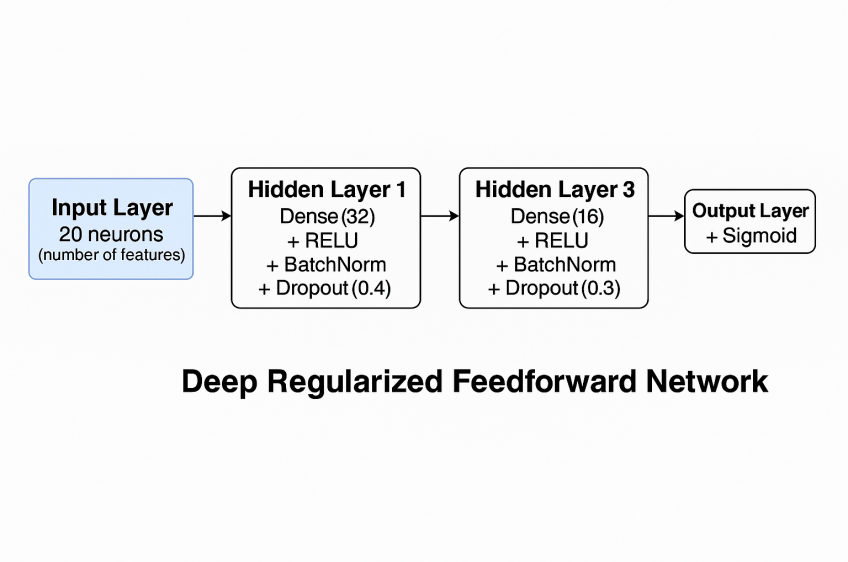


Figure 3.6: A schematic representation of the Deep Regularized Feedforward Network.

#### 3.6.5.3 Model Optimization and Stability

The choice of **AdamW** as the optimization algorithm addresses several shortcomings of the standard Adam optimizer [85], particularly in preventing overfitting due to uncontrolled parameter magnitudes. Combined with dropout regularization and $\ell_2$ penalties, the model remains stable and generalizes well even on imbalanced datasets. Batch normalization accelerates convergence [86] and ensures consistent activation distributions during training. Gradient clipping is applied to prevent exploding gradients in deep configurations, ensuring robustness across various federated learning rounds.

#### 3.6.5.4 Comparative Analysis with Other Neural Models

- **Comparison with MLP (Standard Feedforward Network)**

  While traditional MLPs consist of dense layers without regularization or normalization, the DRFN integrates dropout, batch normalization, and weight decay, addressing overfitting and vanishing gradients. Empirical tests show that standard MLPs are prone to overfitting on tabular data without sophisticated regularization mechanisms, which this architecture successfully overcomes.

- **Comparison with LSTM and GRU**

  Although LSTM and GRU architectures are powerful for sequence modeling [87], [88], their applicability to static, tabular network traffic data is limited. Recurrent architectures such as LSTM and GRU tend to overfit and exhibit increased computational complexity when applied to non-sequential data. Moreover, the absence of temporal dependencies in the input features makes such recurrent layers unnecessarily complex and inefficient for the target task.

- **Comparison with CNN-based Architectures**

  CNNs are effective in capturing local spatial patterns [89] and are predominantly used in image or sequential signal processing. When applied to tabular data, CNNs

offer no inherent advantage and often underperform due to the lack of spatial correlations between features. In contrast, the DRFN is tailored for capturing global feature interactions through fully connected layers, making it more suitable for network-based cybersecurity applications.

- **Comparison with Hybrid or Attention-Based Models**

  Hybrid models or those incorporating attention mechanisms [90] often achieve state-of-the-art performance but introduce substantial architectural complexity, computational overhead, and training instability in federated settings. The DRFN strikes a balance between performance and simplicity, ensuring compatibility with limited-resource clients in federated environments while maintaining high classification accuracy.

| Model | Architecture Type | Best Suited For | Suitability for Tabular Data | Overfitting Tendency | Computational Cost | Generalization Ability | Remarks |
|---|---|---|---|---|---|---|---|
| CNN | Convolutional Neural Network | Images, spatial signals | Poor | Medium | High | Medium | Ineffective for tabular data due to lack of spatial correlation |
| LSTM | Recurrent Neural Network | Sequential, time-series data | Low | High | Very High | Low to Medium | Overkill for static inputs; sensitive to noise |
| GRU | Gated Recurrent Unit | Sequential, time-series data | Low | High | High | Low to Medium | Simplified LSTM, but still suboptimal for tabular features |
| MLP | Standard Feedforward Neural Network | General purpose | Good (with tuning) | High | Low to Medium | Medium | Easy to implement, lacks inherent regularization |
| Hybrid/Attention Models | Transformer or multi-branch models | Multimodal or complex tasks | Moderate | Medium | Very High | High (but unstable) | High accuracy but complex, unstable in federated settings |
| DRFN (Proposed) | Deep Regularized Feedforward Net | Tabular binary classification | Excellent | Low | Medium | High | Optimized with dropout, batch normalization, and AdamW; ideal for federated and tabular scenarios |

Figure 3.7: Comparative Table of Neural Models for Cybersecurity Threat Detection.

#### 3.6.5.5 Why DRFN Outperforms Simpler Models

The DRFN's superior performance lies in its integration of best practices from deep learning research—namely, depth with width reduction, dropout regularization, batch normalization, weight decay, and a robust optimizer. These techniques jointly enhance its capacity to learn non-linear feature interactions without sacrificing generalization. Empirical results confirm that the DRFN consistently achieves higher metrics (accuracy, F1-score, AUC-ROC) compared to simpler MLPs and even LSTMs/GRUs when applied to structured, tabular cybersecurity data.

In conclusion, the DRFN architecture offers a robust, generalizable, and computationally efficient solution for detecting cybersecurity threats in tabular network traffic datasets. It combines the strengths of deep learning regularization techniques with architectural simplicity and is well-suited for federated deployment, outperforming standard MLPs and sequence-oriented models like LSTM or GRU in this specific domain.

### 3.6.6 Global Architecture

This schema summarizes the global architecture of the federated learning system using the CIC-IDS-2017 dataset, involving Data Preprocessing, 5 Clients, and a Server. The system iterates over 15 rounds, employing FedAvg or FedProx strategies.

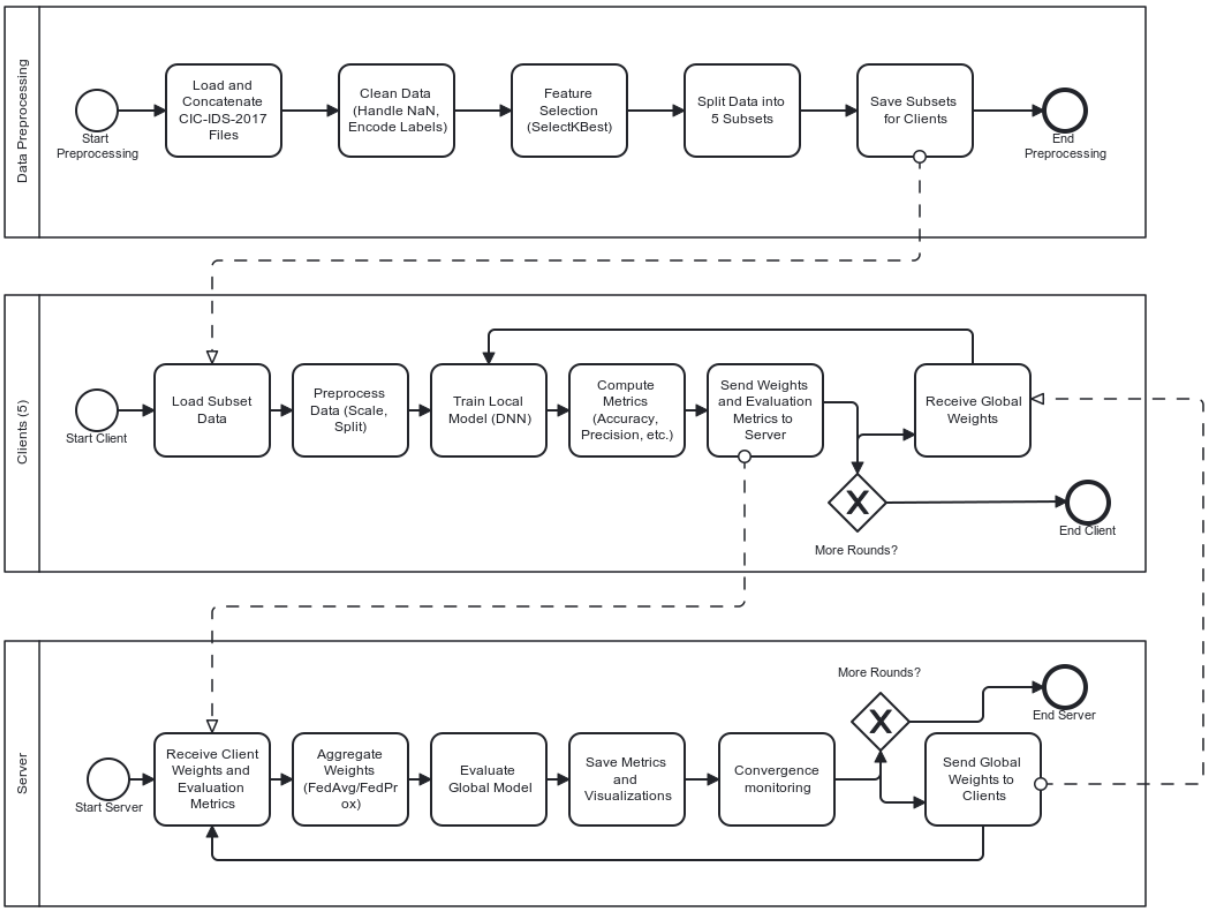


Figure 3.8: Global Architecture Schema.

### 3.6.7 Challenges and Solutions in Federated Learning

Federated Learning (FL) presents a paradigm shift in distributed machine learning by enabling collaborative model training without the need to centralize raw data. However, the deployment of FL in real-world scenarios faces several critical challenges, which must be addressed to ensure efficient, robust, and secure learning outcomes.

- **Data Heterogeneity:** One of the foremost challenges in FL is the presence of non-independent and identically distributed (non-IID) data across participating clients. Unlike traditional centralized learning where data distribution is typically assumed to be uniform, clients in FL often possess data that vary significantly in size, feature distribution, and label proportions due to differing local environments and user behaviors. This heterogeneity can cause model updates to diverge during aggregation, leading to slower convergence rates and degraded global model performance. To address these issues, advanced optimization algorithms have been proposed. For instance, *FedProx* introduces a proximal term to the local objective function, which

constrains local updates to remain close to the global model, thereby reducing divergence caused by heterogeneous data distributions. Similarly, *SCAFFOLD* utilizes control variates to correct client drift by estimating the update direction more accurately, which mitigates the bias introduced by non-IID data. These methods enhance the stability and generalizability of the federated model in heterogeneous environments [91].

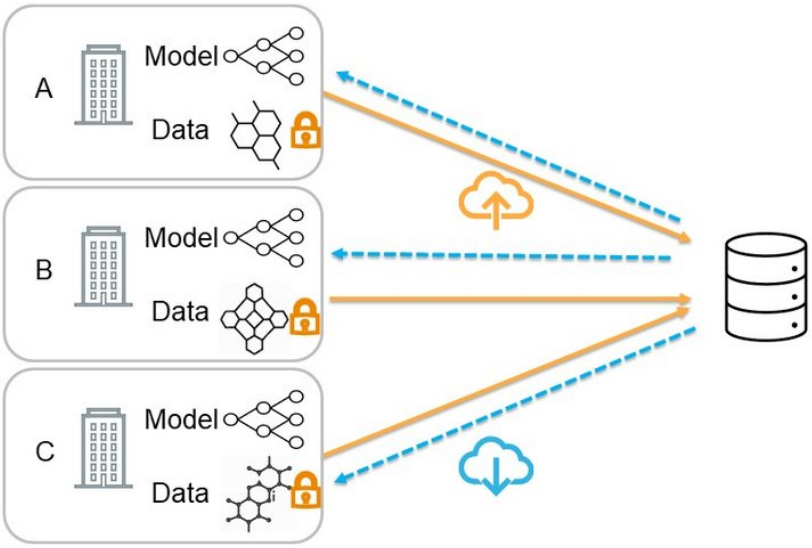


Figure 3.9: Data Heterogeneity.

- **Communication Efficiency:** Another significant bottleneck in FL arises from the high communication overhead involved in frequent model parameter exchanges between the central server and numerous distributed clients. As FL often operates over bandwidth-constrained networks, continuous transmission of large model updates can result in latency and increased operational costs. To improve communication efficiency, various techniques have been developed. Model compression approaches, such as quantization, sparsification, and pruning, reduce the size of the transmitted updates without severely impacting model accuracy. For example, gradient quantization lowers the precision of model parameters, thereby decreasing communication load. Moreover, asynchronous update schemes allow clients to communicate with the server independently and at different time intervals, which alleviates synchronization delays and supports scalability in heterogeneous network conditions. Additionally, periodic aggregation strategies that limit communication rounds by allowing multiple local updates before synchronization can significantly reduce communication frequency[92].

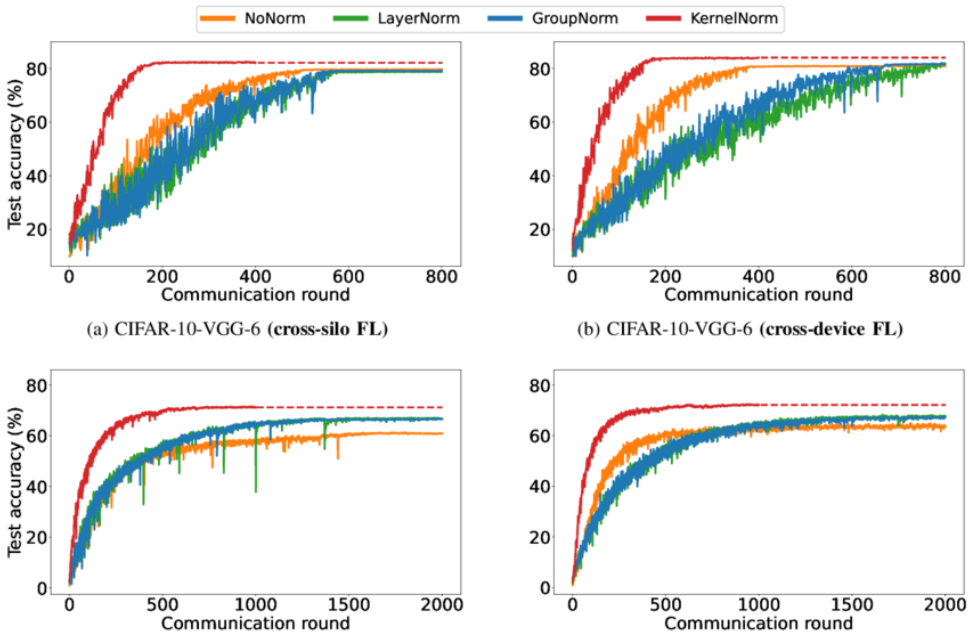


Figure 3.10: Federated Learning Communication Efficiency.

- **Privacy and Security:** Although FL inherently enhances data privacy by keeping raw data localized, the exchange of model updates can still pose privacy risks. Attackers may exploit gradient or parameter updates to infer sensitive client information through reconstruction or membership inference attacks. To mitigate such vulnerabilities, privacy-preserving mechanisms have been integrated into FL frameworks. Differential privacy (DP) introduces carefully calibrated noise into model updates or intermediate computations to provide quantifiable privacy guarantees, ensuring that individual data contributions remain indistinguishable [93]. Secure aggregation protocols further safeguard privacy by enabling the server to aggregate client updates in an encrypted manner, preventing exposure of individual updates even if the server is compromised. Complementary security measures include robust anomaly detection techniques to identify malicious clients and Byzantine-resilient aggregation algorithms that can tolerate adversarial behaviors without corrupting the global model [94].

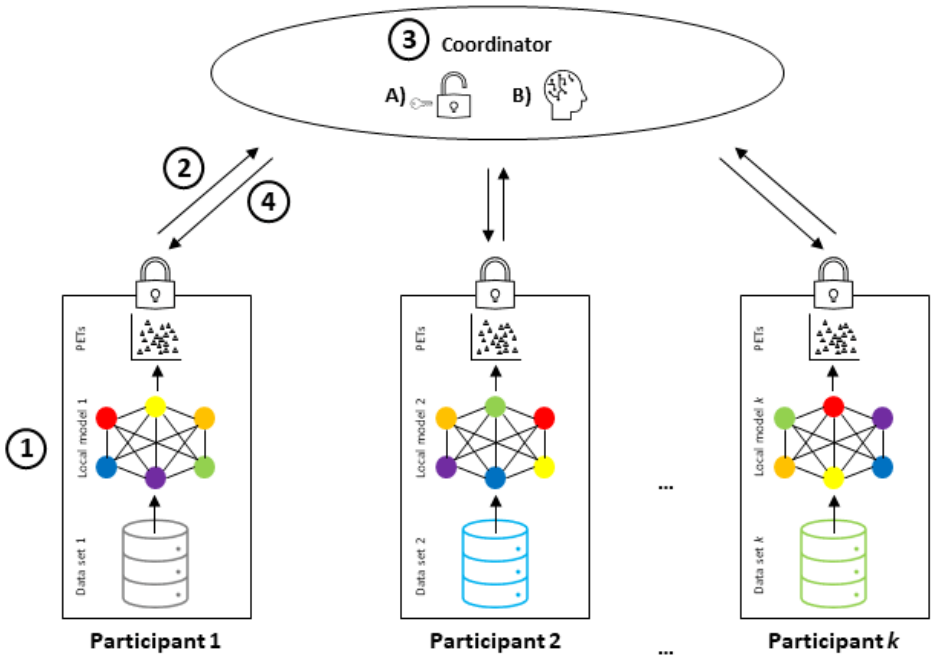


Figure 3.11: Federated Learning Combined with Privacy-Enhancing Technologies.

In summary, the practical adoption of Federated Learning necessitates overcoming fundamental challenges associated with data heterogeneity, communication constraints, and privacy risks. Ongoing research continues to propose novel algorithms and protocols that balance these competing demands, striving for FL systems that are both effective in learning and resilient in diverse, real-world deployment scenarios.

### 3.6.8 Federated Learning in Comparison to Traditional Machine Learning and Distributed Machine Learning

Federated Learning (FL) represents a significant evolution in the field of machine learning by addressing critical limitations associated with data privacy, ownership, and communication overhead. To fully appreciate its advantages and trade-offs, it is essential to distinguish FL from traditional centralized Machine Learning (ML) and conventional Distributed Machine Learning (DML) paradigms [78], [80].

#### 3.6.8.1 Traditional Machine Learning

In traditional machine learning, all training data is centrally collected and stored in a single repository where the model is trained. This approach assumes unrestricted access to complete and clean datasets, typically maintained in high-performance data centers. While this centralized method allows for efficient training and model optimization, it poses substantial privacy and scalability challenges. Sensitive data such as healthcare records, financial transactions, or user behavior logs must be transferred to the central server, violating privacy regulations such as GDPR or HIPAA and increasing the risk of data breaches [79]. Moreover, centralizing vast volumes of data can be computationally intensive and inefficient, particularly in scenarios involving edge devices or geographically distributed sources.

#### 3.6.8.2 Distributed Machine Learning

Distributed Machine Learning aims to overcome the scalability bottlenecks of traditional ML by dividing the training workload across multiple compute nodes. In DML, data is often partitioned and distributed across worker nodes, which collaborate under a parameter server or similar architecture [95]. Each node trains a subset of the data, and their gradients or model parameters are periodically synchronized through a central coordinator. Although DML improves computational efficiency and reduces training time, it still typically requires access to raw data, which may be transmitted across networked systems, raising similar privacy concerns as centralized ML [80]. Furthermore, DML architectures often assume a homogeneous and stable network environment, making them less suitable for heterogeneous and dynamic edge devices.

#### 3.6.8.3 Federated Learning

Federated Learning introduces a paradigm shift by enabling collaborative model training without requiring data centralization. Instead of sending raw data to the server, FL orchestrates local training on client devices and transmits only model updates (e.g., weights or gradients) to a central aggregator [77]. This fundamental difference offers strong privacy benefits, as raw data remains on the client side, and supports compliance with data protection laws [78]. FL is inherently more resilient to data heterogeneity, network variability, and client unavailability, as it is designed to operate in decentralized and resource-constrained environments such as mobile phones, IoT devices, or distributed sensors [79].

Compared to DML, FL introduces additional challenges, including non-IID data distributions, limited communication bandwidth, and variable client reliability. However, recent advances in model compression, secure aggregation, and personalization techniques are addressing these limitations, making FL increasingly viable for real-world applications [80], [92].

#### 3.6.8.4 Comparative Summary

| Aspect | Traditional ML | Distributed ML | Federated Learning |
|---|---|---|---|
| Data Location | Centralized | Centralized/Partitioned | Decentralized (local devices) |
| Privacy | Low | Moderate | High |
| Communication Overhead | Moderate | High | Low (with compression) |
| Computation Distribution | Centralized | Across workers | Across clients |
| Fault Tolerance | Low | Moderate | High |
| Suitable Environments | Data centers | HPC clusters | Edge/IoT/mobile systems |
| Security | Weak (centralized risk) | Improved | Strong (with secure aggregation) |
| Data Heterogeneity Tolerance | Low | Low to Moderate | High |

Figure 3.12: Comparison of Traditional ML, Distributed ML, and Federated Learning.

In summary, while traditional ML and DML approaches are effective in high-performance centralized settings, FL emerges as a promising alternative for privacy-sensitive, dis-

tributed, and resource-constrained environments. Its unique design enables collaborative intelligence without compromising user privacy or data sovereignty [78], [80].

## 3.7 Conclusion

Federated learning represents a transformative solution for developing secure and privacy-preserving intrusion detection systems in distributed environments. By eliminating the need for centralized data aggregation, it reduces the risks of data breaches and ensures compliance with privacy regulations. The integration of advanced preprocessing, feature selection, and deep learning within the federated setting enables the construction of robust models that can adapt to diverse network conditions and attack patterns. Overall, this methodology offers an effective balance between data privacy, computational efficiency, and detection accuracy in modern cybersecurity applications.

# Chapter 4

# Implementation and Results

## 4.1 Introduction

This chapter outlines the implementation and evaluation of a federated learning system for cybersecurity using the Flower framework, TensorFlow, Python, and Jupyter Notebooks in Anaconda Navigator. We prepared distributed datasets across five clients, trained a neural network using Federated Averaging (FedAvg) and Federated Proximal (FedProx) strategies, and evaluated performance through metrics like accuracy, precision, recall, and F1-score. Visualizations, including confusion matrices, support the analysis. The results show the system's effectiveness in handling non-IID data, making it suitable for privacy-sensitive cybersecurity applications.

## 4.2 Environment and development tools

For the implementation of the process presented in the previous chapter, we used a set of languages, programming environments, and tools that are often used in deep learning projects.

### 4.2.1 Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface included with the Anaconda distribution, which facilitates the management of environments, packages, and applications without requiring command-line interactions. It simplifies dependency handling and environment isolation, which is particularly beneficial in machine learning workflows that rely on specific library versions. Anaconda also streamlines the installation and execution of Jupyter Notebooks, making it a preferred choice for rapid prototyping and experimentation.

### 4.2.2 Python

Python is a high-level, interpreted programming language widely adopted in the field of artificial intelligence and data science. Developed by Guido van Rossum in the late 1980s, Python emphasizes code readability and simplicity of syntax, allowing developers to express complex logic with fewer lines of code compared to languages like Java or C++ **artima**. Its extensive ecosystem of libraries, including NumPy, Pandas, Scikit-learn, and TensorFlow, makes it an indispensable tool for implementing and experimenting with deep learning models.

### 4.2.3 Jupyter Notebook

Jupyter Notebook is an open-source web-based interactive computing environment that allows users to combine live code, explanatory text, and visualizations in a single document. It is particularly advantageous for data exploration, debugging, and iterative development of machine learning models. Its support for multiple programming languages and seamless integration with libraries like Matplotlib and Seaborn enables detailed data analysis and model evaluation in real time.

### 4.2.4 TensorFlow

TensorFlow is an open-source deep learning library developed by the Google Brain team and released in 2015. Designed to support a wide range of machine learning tasks, TensorFlow is particularly efficient for constructing and deploying deep neural networks. It provides a flexible architecture that enables deployment across various platforms—from desktops to edge devices—and supports both eager execution and computational graph-based workflows. Its scalability and optimization capabilities make it well-suited for training deep models on large datasets in distributed environments [96].

### 4.2.5 Flower

Flower (FLWR) is a federated learning framework designed to enable efficient experimentation and production-scale deployment of FL systems. It provides a flexible and extensible API that supports the implementation of custom client and server strategies, including aggregation methods, evaluation protocols, and communication patterns. Flower abstracts much of the underlying complexity involved in federated learning, allowing researchers to focus on model design and performance evaluation. Its compatibility with existing machine learning frameworks such as TensorFlow and PyTorch further facilitates integration into established workflows [97].

One of Flower's key strengths is its modular architecture, which separates the roles of clients and servers while enabling seamless interaction. The framework supports a variety of federated optimization algorithms, such as Federated Averaging (FedAvg) and Federated Proximal (FedProx), and allows users to define custom strategies tailored to specific use cases, like handling non-IID data distributions or ensuring privacy constraints. This modularity is particularly valuable in domains like cybersecurity, where data heterogeneity and regulatory compliance are critical.

Flower's communication layer is built to be robust and scalable, supporting synchronous and asynchronous training modes over TCP/IP. It includes built-in tools for client sampling, configuration broadcasting, and metrics aggregation, which streamline the coordination of distributed training across multiple devices. The framework also emphasizes security and privacy, leveraging techniques like secure aggregation and local differential privacy when integrated with appropriate libraries, making it suitable for sensitive applications.

Additionally, Flower offers extensive documentation and a growing ecosystem of extensions, such as Flower-SuperLink for production-grade deployments, enhancing its usability for both research and real-world applications. Its open-source nature fosters community contributions, ensuring continuous improvement and adaptation to emerging FL challenges. By providing a unified platform, Flower lowers the barrier to entry for federated

learning, enabling rapid prototyping and deployment across diverse hardware environments.

## 4.3 Implementation of Federated Learning using Flower Framework

The proposed system utilizes a federated learning (FL) architecture implemented through the Flower framework [97], an open-source platform renowned for its flexibility and scalability in decentralized machine learning. Flower facilitates the training of models across multiple heterogeneous clients while ensuring data privacy by keeping raw data local to each device—a critical feature for cybersecurity applications where sensitive network traffic data must comply with strict locality and regulatory constraints. The implementation comprises five geographically and statistically diverse clients, each operating on local datasets, coordinated by a central server that aggregates model updates.

Flower's modular design allows for the integration of custom strategies and evaluation protocols. Two federated optimization approaches are employed: a customized Federated Averaging (FedAvg) algorithm and a Federated Proximal (FedProx) variant with a proximal regularization term. This dual-strategy setup enables a comparative analysis of weighted averaging versus proximal regularization, particularly effective for handling non-independent and identically distributed (non-IID) data, a prevalent challenge in cybersecurity due to varying attack patterns and network conditions.

### 4.3.1 Server Design

The server extends the `flwr.server.strategy.FedAvg` and `flwr.server.strategy.FedProx` classes, leveraging Flower's strategy interface to implement custom aggregation and evaluation logic.

- **FedAvg Aggregation:** The `aggregate_fit` method is overridden to perform weighted averaging of client model parameters based on local sample sizes. Comprehensive logging tracks performance metrics—accuracy, precision, recall, F1-score, area under the ROC curve (AUC-ROC), and log loss—across up to 50 communication rounds, facilitating convergence monitoring.
- **FedProx Integration:** The FedProx strategy augments the local loss with a proximal term $\mathcal{L}_{prox} = \mathcal{L}_{local} + \frac{\mu}{2}w - w^2_{global}$, where $\mu = 0.1$ penalizes deviations from the global model, enhancing stability in non-IID settings. Aggregation mirrors FedAvg, with the proximal parameter broadcast to clients.
- **Model Checkpointing and Visualization:** Confusion matrices and trend plots of metrics are generated per round, stored in a timestamped directory, supporting longitudinal analysis.
- **Client Sampling and Configuration:** All five clients participate per round (`fraction_fit` = 1.0) to maximize data utilization. A configuration dictionary, including feature indices and hyperparameters (e.g., `prox_mu`), is broadcast to synchronize client environments.

- **Convergence-Based Stopping:** Training can terminate before the 50-round limit if convergence is detected. The server monitors accuracy and log loss differences between consecutive rounds, stopping when both differences are $\leq 0.001$. This adaptive mechanism, activated after the second round, optimizes resource use by halting when further improvements are negligible.

The training spans up to 50 rounds, with weights updated and evaluated after each round, though it may end earlier due to convergence, allowing a robust comparison of FedAvg and FedProx efficacy on non-IID cybersecurity data.

### 4.3.2 Client Configuration

Clients, subclassed from `flwr.client.NumPyClient`, operate autonomously on local datasets loaded from CSV files (e.g., `subset_1.xls` to `subset_5.xls`).

- **Data Processing:** Local datasets undergo MinMax normalization and are split into training and validation subsets (80%/20%) with stratification.
- **Model Architecture:** A feedforward neural network is used, featuring:
  - An input layer with 20 features,
  - Three hidden layers (64, 32, 16 neurons) with ReLU activations, L2 regularization, batch normalization, and dropout rates (0.5, 0.4, 0.3),
  - A sigmoid output for binary classification.
- **Training Protocol:** The AdamW optimizer (learning rate: 0.0003, weight decay: 0.005) drives 6 local epochs with a batch size of 32, enhanced by early stopping and learning rate scheduling.
- **FedProx Regularization:** Clients implement the proximal term with $\mu = 0.1$, stabilizing training in heterogeneous contexts.
- **Metrics and Visualization:** Class weights address imbalance, and clients log metrics (accuracy, precision, recall, F1-score, AUC-ROC, log loss), generating confusion matrices and plots locally.
- **Robustness:** Data and weights are cast to `float32`, with error handling for inconsistencies.

### 4.3.3 Communication Flow and Synchronization

The FL system operates over TCP on `localhost:8082`, with clients registering and exchanging global weights and configurations synchronously.

- Clients receive global weights and strategy-specific parameters (e.g., `prox_mu`).
- Local training aligns with the active strategy (FedAvg or FedProx).
- Updated parameters and metrics are sent back to the server, ensuring temporal alignment and robust evaluation.

This implementation harnesses Flower's scalability and security, with the convergence-based stopping enhancing efficiency for decentralized cybersecurity model training.

## 4.4 Results

This section shows the obtained results after training and testing our federated learning based model.

### 4.4.1 Performance Evaluation

#### 4.4.1.1 FedAvg

The following analysis evaluates the performance of a federated learning experiment using the FedAvg algorithm, conducted over 15 rounds with 5 clients, as indicated by the successful aggregation of results without failures.

1. **Clients Metrics:**
   - **Client1:**

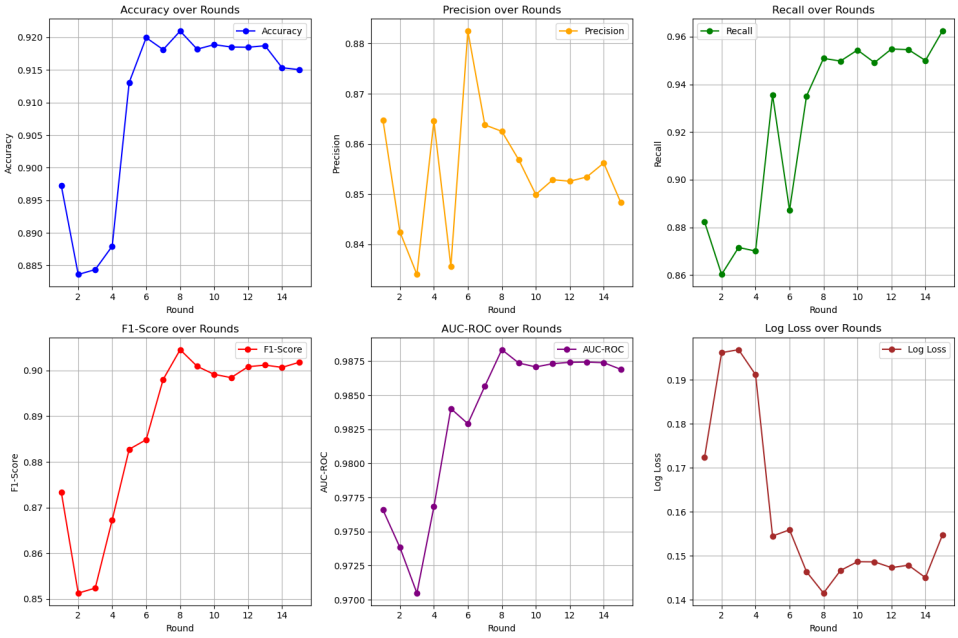


Figure 4.1: Metrics Plot Client 1.

**Confusion Matrix:**

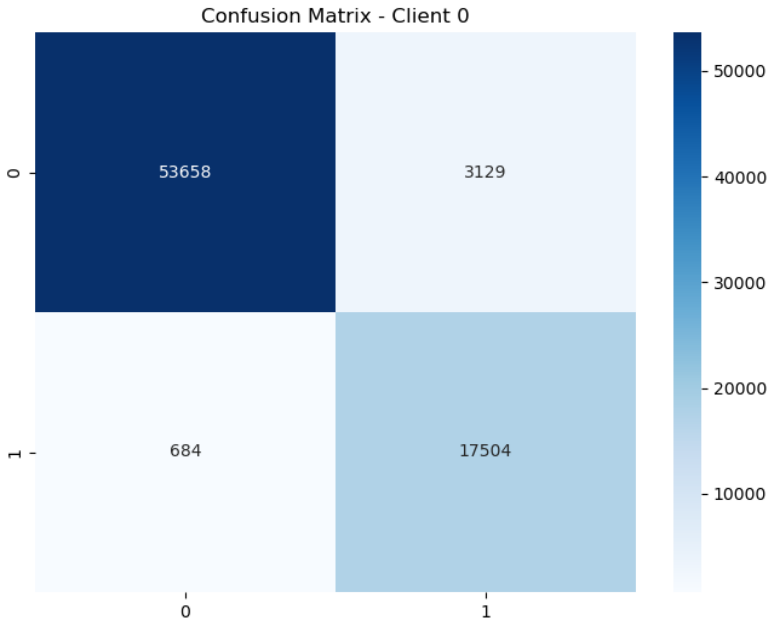


Figure 4.2: Confusion Matrix Client 1.

- **Client3:**

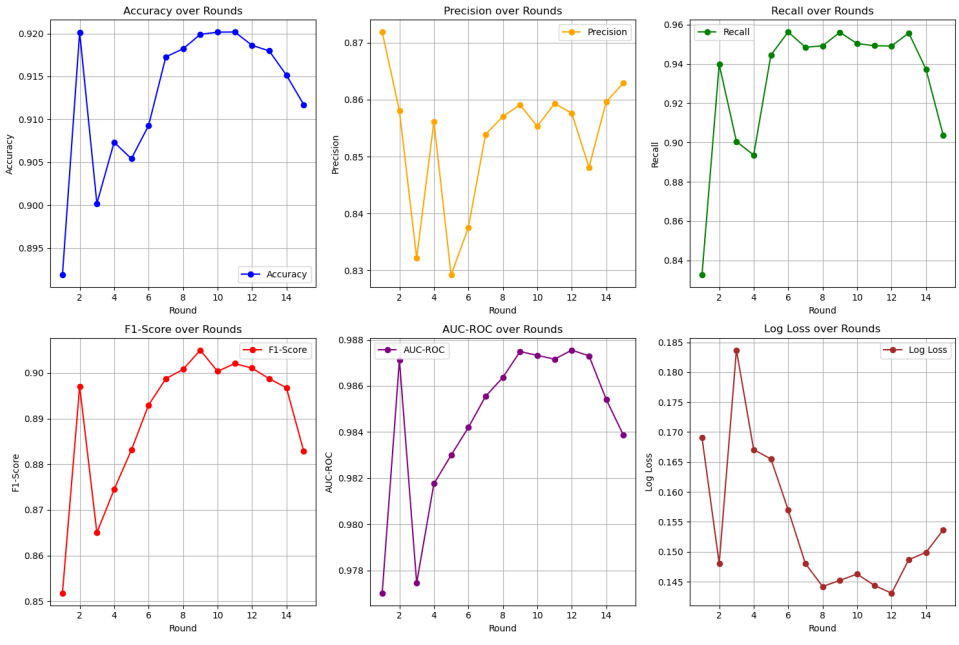


Figure 4.3: Metrics Plot Client 3.

**Confusion Matrix:**

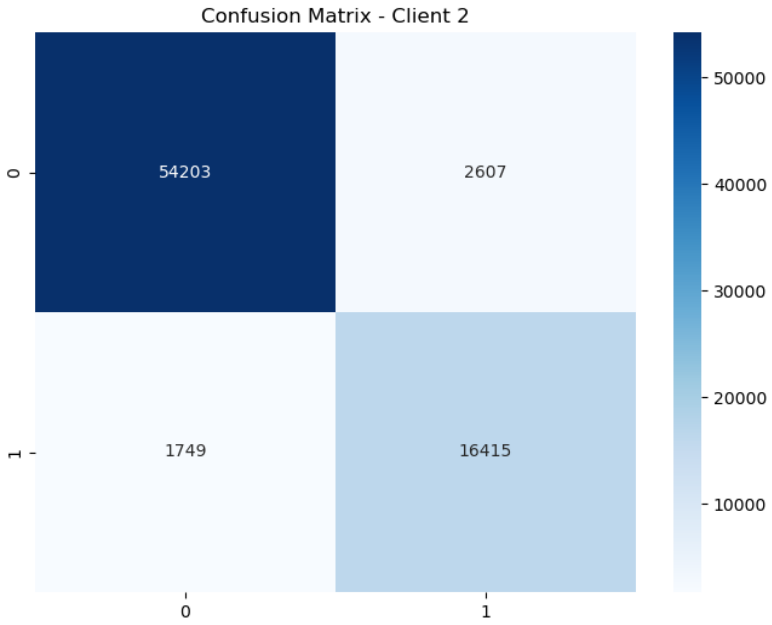


Figure 4.4: Confusion Matrix Client 3.

- **Client5:**

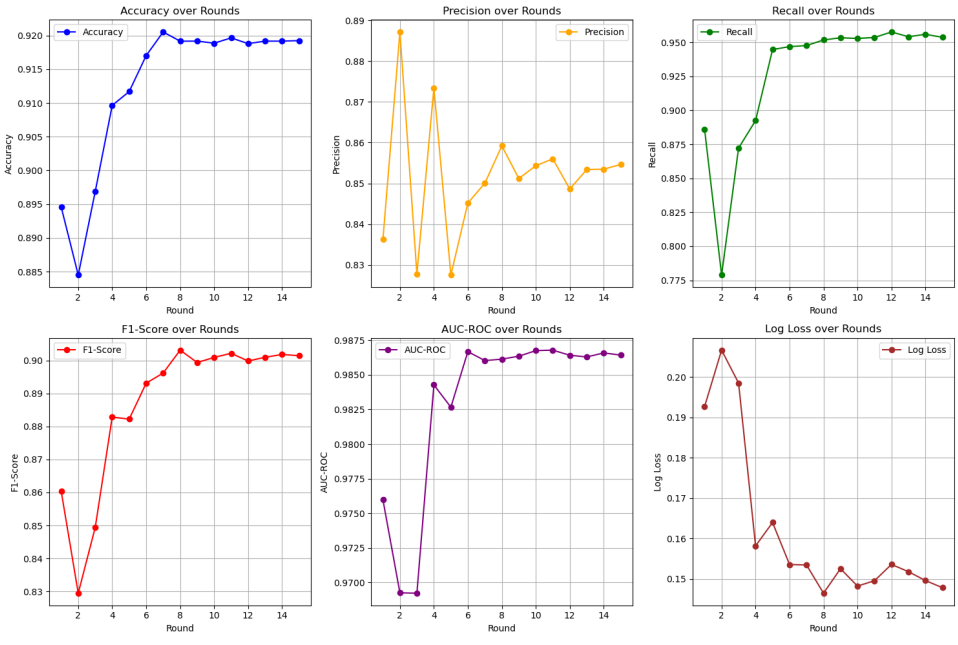


Figure 4.5: Metrics Plot Client 5.

**Confusion Matrix:**

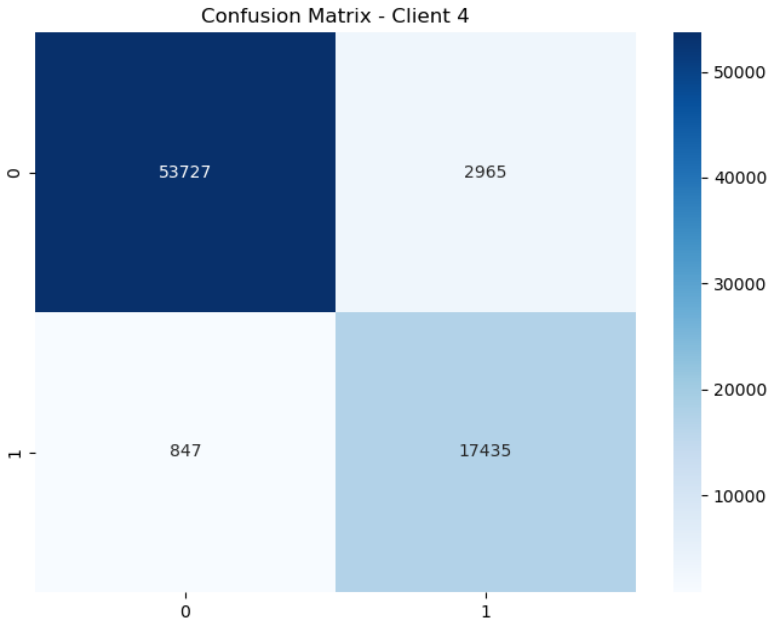


Figure 4.6: Confusion Matrix Client 5.

**Analysis:**

- **Loss:** The log loss performance is characterized by an overall decreasing trend, starting from around 0.18 and dropping to approximately 0.15 across the 15 rounds.

  **Observation:** The log loss decreases overall with some fluctuations, indicating improved model performance over the rounds.

- **Accuracy:** The accuracy performance shows an increase from approximately 0.985 to 0.92, with a peak around round 6 before stabilizing with slight variations.

  **Observation:** Accuracy rises initially, peaking around round 6, and then stabilizes with minor fluctuations throughout the rounds.

- **F1 Score:** The F1 score performance improves from about 0.85 to 0.90, peaking around round 6 and then stabilizing with minor fluctuations.

  **Observation:** The F1 score increases steadily, reaching a peak around round 6, followed by stabilization with slight variations.

- **AUC-ROC, Precision, Recall :** AUC-ROC performance improves from 0.974 to 0.986, peaking around round 6 and then stabilizing. Precision performance fluctuates between 0.83 and 0.90 without a clear upward trend. Recall performance increases from 0.775 to 0.96, showing a general upward trend with some variability.

  **Observation:** AUC-ROC improves and stabilizes after an initial rise, precision remains volatile, and recall shows a consistent upward trend with some fluctuations.

**Clients Performance:**

The overall performance of the clients demonstrates improvement across most metrics over the 15 rounds. Accuracy, F1 score, and AUC-ROC show initial gains with subsequent stabilization, while recall exhibits a consistent upward trend. Log loss decreases, suggesting better model fit, whereas precision remains inconsistent without a clear improvement pattern.

2. **Global Metrics:**

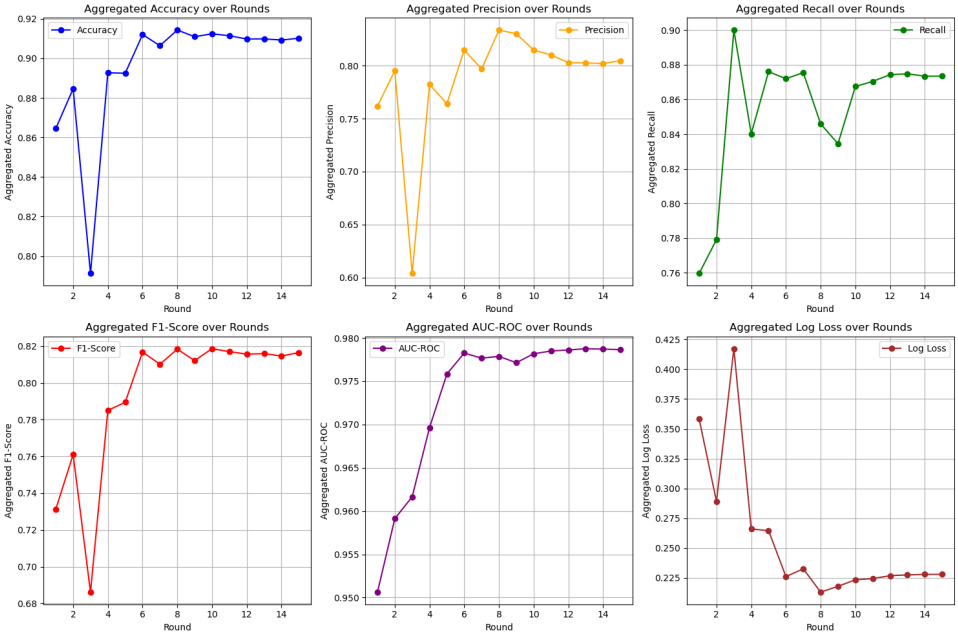


Figure 4.7: Server Metrics Plot.

**Confusion Matrix:**

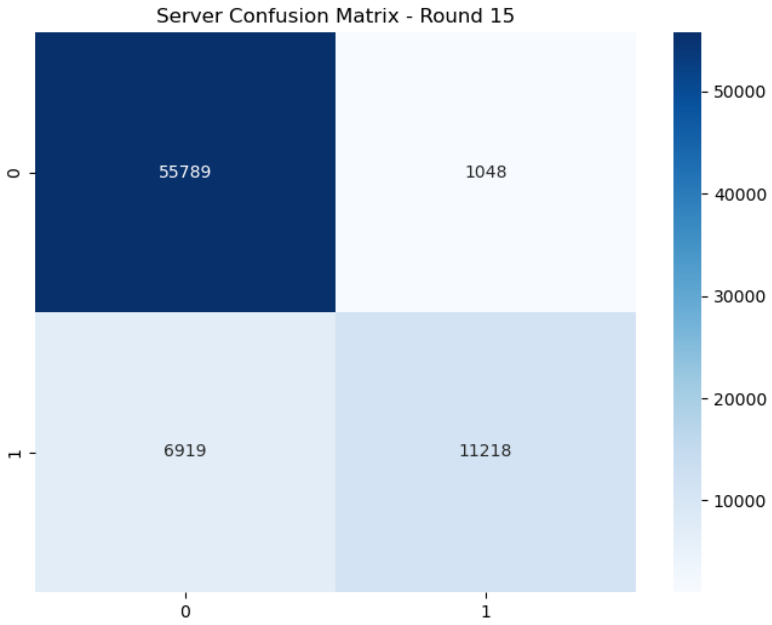


Figure 4.8: Server Confusion Matrix.

**Analysis:**

- **Loss:** The aggregated log loss performance starts at around 0.425 and decreases to approximately 0.25 across the 15 rounds, with noticeable fluctuations.

  **Observation:** The log loss shows a general downward trend with some variability, suggesting an overall improvement in model performance.

- **Accuracy:** The aggregated accuracy performance increases from about 0.81 to 0.92, peaking around round 6 before stabilizing with slight variations.

  **Observation:** Accuracy improves significantly, peaking around round 6, and then remains relatively stable with minor fluctuations.

- **F1 Score:** The aggregated F1 score performance rises from approximately 0.68 to 0.82, peaking around round 6 and then stabilizing with minor fluctuations.

  **Observation:** The F1 score increases steadily, reaching a peak around round 6, followed by stabilization with slight variations.

- **AUC-ROC, Precision, Recall:** AUC-ROC performance improves from 0.950 to 0.990, peaking around round 6 and then stabilizing. Precision performance fluctuates between 0.60 and 0.80 with no clear upward trend. Recall performance increases from 0.775 to 0.90, showing a general upward trend with some variability.

  **Observation:** AUC-ROC improves and stabilizes after an initial rise, precision remains volatile without a consistent trend, and recall shows a consistent upward trend with some fluctuations.

**Global Performance:**

The global performance indicates overall improvement across most metrics over the 15 rounds. Accuracy, F1 score, and AUC-ROC show initial gains with subsequent stabilization, while recall exhibits a consistent upward trend. Log loss decreases, reflecting better model fit, whereas precision remains inconsistent without a clear improvement pattern.

**Overall Assessment:**

The overall assessment of the federated learning experiment using the FedAvg algorithm reveals a generally positive trend in model performance across both client and global metrics over the 15 rounds. For clients, accuracy, F1 score, and AUC-ROC show initial improvements with stabilization, while recall exhibits a consistent upward trend, and log loss decreases, indicating enhanced model fit. Precision, however, remains volatile without a clear improvement pattern. Similarly, at the global level, accuracy, F1 score, and AUC-ROC demonstrate significant initial gains followed by stabilization, with recall showing a steady increase. Global log loss also decreases, reflecting better overall model performance, while precision continues to fluctuate without a consistent trend. The alignment between client and global metrics suggests effective aggregation, with the model benefiting from the federated learning approach, though precision's instability across both levels indicates a potential area for further optimization.

#### 4.4.1.2 FedProx

The following analysis evaluates the performance of a federated learning experiment using the FedProx algorithm, conducted over 15 rounds with 5 clients, as indicated by the successful aggregation of results without failures.

1. **Clients Metrics:**
   - **Client1:**

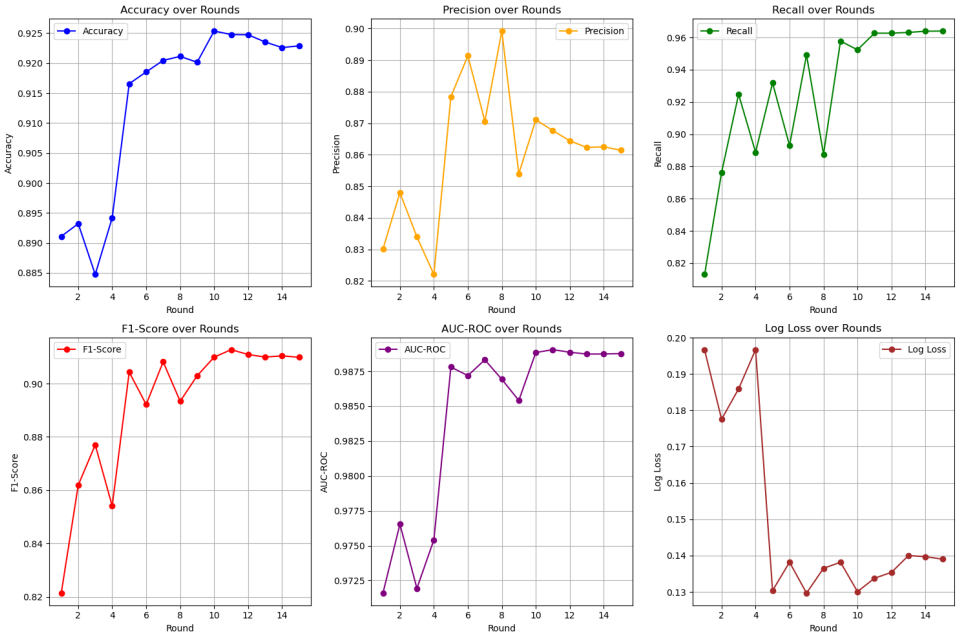


Figure 4.9: Metrics Plot Client 1.

**Confusion Matrix:**

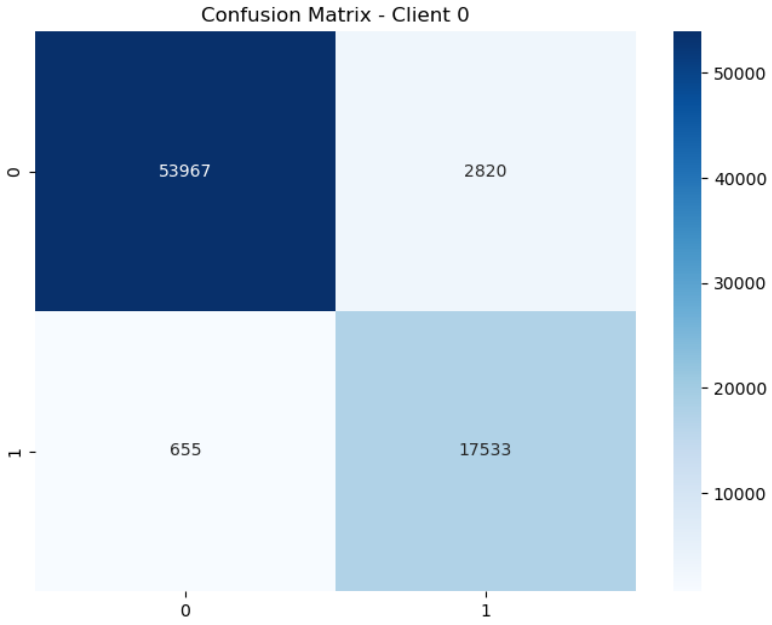


Figure 4.10: Confusion Matrix Client 1.

- **Client2:**

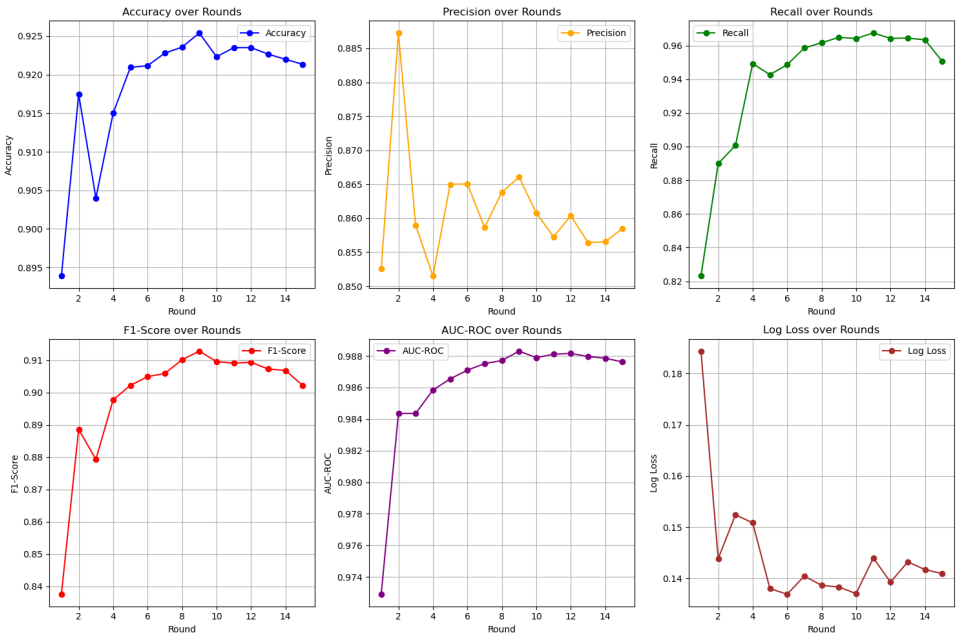


Figure 4.11: Metrics Plot Client 2.

**Confusion Matrix:**

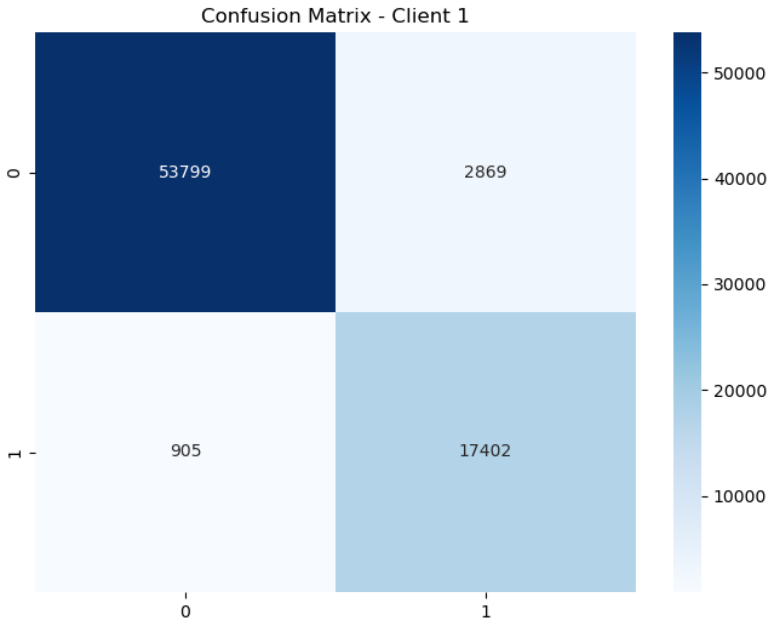


Figure 4.12: Confusion Matrix Client 2.

- **Client5:**

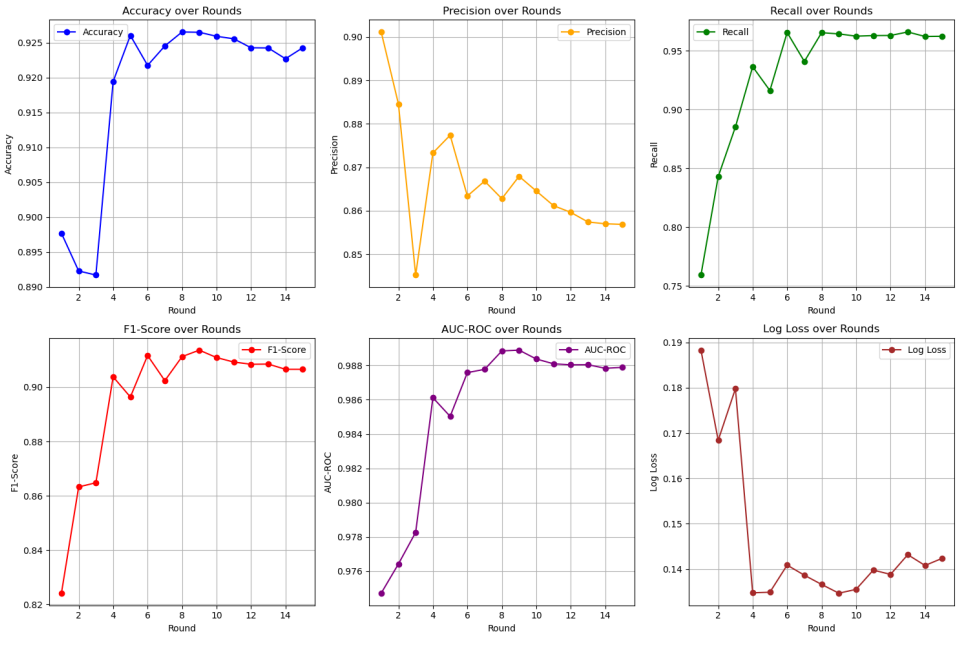


Figure 4.13: Metrics Plot Client 5.

**Confusion Matrix:**

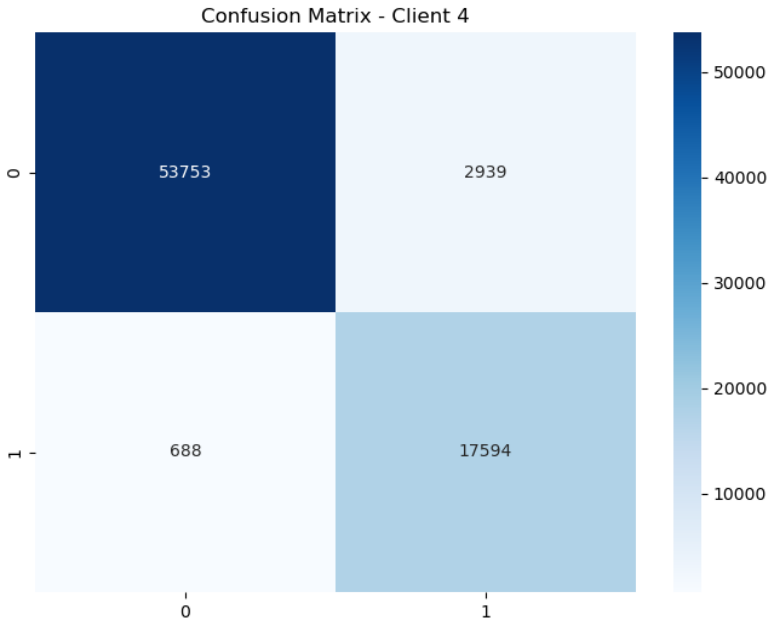


Figure 4.14: Confusion Matrix Client 5.

**Analysis:**

- **Loss:** The log loss performance starts at around 0.19 and decreases to approximately 0.13 across the 15 rounds, with some fluctuations.

  **Observation:** The log loss shows a general downward trend with some variability, indicating improved model performance over the rounds.

- **Accuracy:** The accuracy performance increases from about 0.89 to 0.925, peaking around round 6 before stabilizing with slight variations.

  **Observation:** Accuracy rises initially, peaking around round 6, and then stabilizes with minor fluctuations throughout the rounds.

- **F1 Score:** The F1 score performance improves from approximately 0.84 to 0.91, peaking around round 6 and then stabilizing with minor fluctuations.

  **Observation:** The F1 score increases steadily, reaching a peak around round 6, followed by stabilization with slight variations.

- **AUC-ROC, Log Loss, Precision, Recall :** AUC-ROC performance improves from 0.970 to 0.988, peaking around round 6 and then stabilizing. Precision performance fluctuates between 0.82 and 0.90 with no clear upward trend. Recall performance increases from 0.75 to 0.96, showing a general upward trend with some variability.

  **Observation:** AUC-ROC improves and stabilizes after an initial rise, precision remains volatile without a consistent trend, and recall shows a consistent upward trend with some fluctuations.

**Clients Performance:**

The overall performance of the clients demonstrates improvement across most metrics over the 15 rounds. Accuracy, F1 score, and AUC-ROC show initial gains with subsequent stabilization, while recall exhibits a consistent upward trend. Log loss decreases, suggesting better model fit, whereas precision remains inconsistent without a clear improvement pattern.

2. **Global Metrics:**

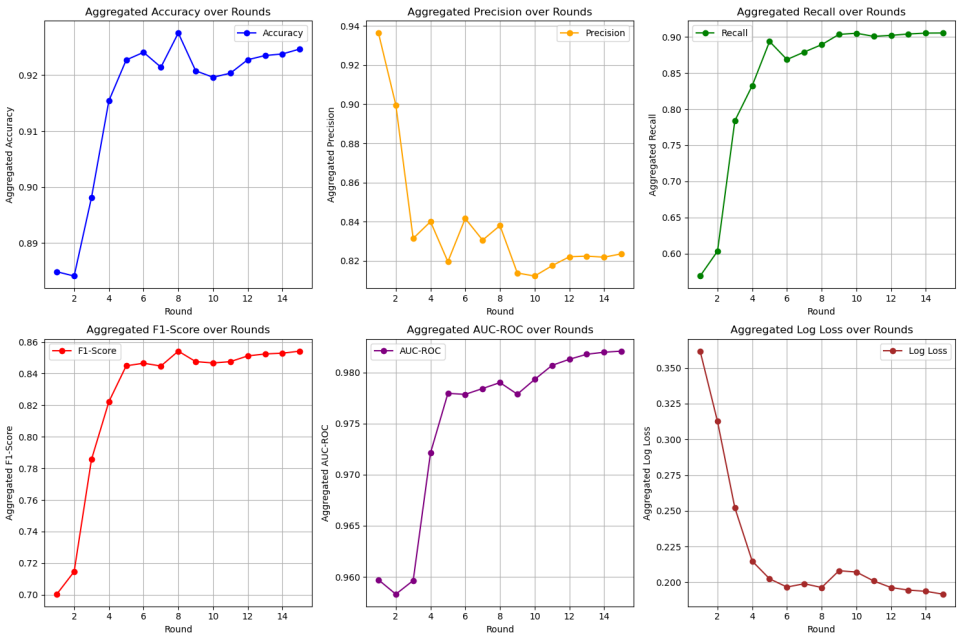


Figure 4.15: Server Metrics Plot.

**Confusion Matrix:**

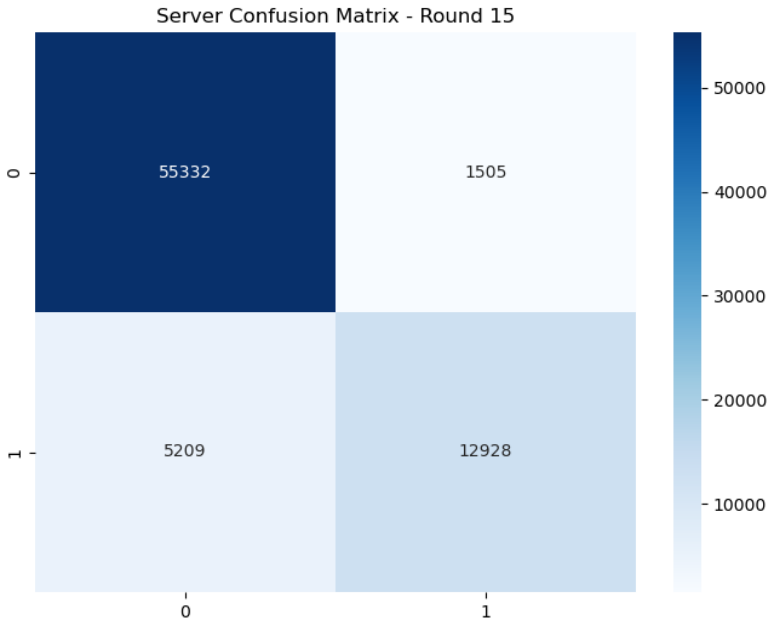


Figure 4.16: Server Confusion Matrix.

**Analysis:**

- **Loss:** The aggregated log loss performance starts at around 0.35 and decreases to approximately 0.20 across the 15 rounds, with some fluctuations.

  **Observation:** The log loss shows a general downward trend with some variability, indicating improved model performance over the rounds.

- **Accuracy:** The aggregated accuracy performance increases from about 0.89 to 0.94, peaking around round 6 before stabilizing with slight variations.

  **Observation:** Accuracy improves significantly, peaking around round 6, and then remains relatively stable with minor fluctuations.

- **F1 Score:** The aggregated F1 score performance rises from approximately 0.70 to 0.86, peaking around round 6 and then stabilizing with minor fluctuations.

  **Observation:** The F1 score increases steadily, reaching a peak around round 6, followed by stabilization with slight variations.

- **AUC-ROC, Log Loss, Precision, Recall:** AUC-ROC performance improves from 0.965 to 0.990, peaking around round 6 and then stabilizing. Precision performance fluctuates between 0.84 and 0.94 with no clear upward trend. Recall performance increases from 0.65 to 0.90, showing a general upward trend with some variability.

  **Observation:** AUC-ROC improves and stabilizes after an initial rise, precision remains volatile without a consistent trend, and recall shows a consistent upward trend with some fluctuations.

**Global Performance:**

The global performance indicates overall improvement across most metrics over the 15 rounds. Accuracy, F1 score, and AUC-ROC show initial gains with subsequent stabilization, while recall exhibits a consistent upward trend. Log loss decreases, reflecting better model fit, whereas precision remains inconsistent without a clear improvement pattern.

**Overall Assessment:** The overall assessment of the federated learning experiment using the FedProx algorithm reveals a generally positive trend in model performance across both client and global metrics over the 15 rounds. For clients, accuracy, F1 score, and AUC-ROC show initial improvements with stabilization, while recall exhibits a consistent upward trend, and log loss decreases, indicating enhanced model fit. Precision, however, remains volatile without a clear improvement pattern. Similarly, at the global level, accuracy, F1 score, and AUC-ROC demonstrate significant initial gains followed by stabilization, with recall showing a steady increase. Global log loss also decreases, reflecting better overall model performance, while precision continues to fluctuate without a consistent trend. The alignment between client and global metrics suggests effective aggregation, with FedProx demonstrating robust performance improvements, though precision's instability across both levels highlights a potential area for further refinement.

## 4.5 Remark

The federated learning (FL) system implemented using the Flower framework effectively addresses decentralized model training for cybersecurity, accommodating data privacy and heterogeneity. Comparative analysis of Federated Averaging (FedAvg) and Federated Proximal (FedProx) shows both achieve significant improvements in accuracy, F1 score, and AUC-ROC within six rounds, with stable performance thereafter and consistent recall gains. Log loss decreases notably, but precision remains volatile, likely due to class imbalances in non-IID cybersecurity data. FedProx slightly outperforms FedAvg, with lower log loss (0.13 vs. 0.15 for clients, 0.20 vs. 0.25 globally) and higher accuracy (0.925 vs. 0.92 for clients, 0.94 vs. 0.92 globally), owing to its proximal regularization ($\mu = 0.1$). Flower's modular design, convergence-based stopping, and robust communication enhance efficiency and scalability. Future work should address precision instability through class rebalancing and explore hybrid strategies or enhanced privacy techniques.

## 4.6 Conclusion

This study demonstrates a robust FL system for cybersecurity using Flower, comparing FedAvg and FedProx across five heterogeneous clients. Both algorithms improve accuracy, F1 score (up to 0.86), and AUC-ROC, with FedProx showing marginal advantages in non-IID settings due to proximal regularization. Log loss decreases significantly, though precision volatility persists, indicating a need for further optimization. Flower's flexibility and efficiency make it ideal for privacy-sensitive applications. Future enhancements could include hybrid algorithms and privacy mechanisms to strengthen performance and applicability in real-world cybersecurity scenarios.

# General Conclusion

In conclusion, this dissertation has comprehensively explored the integration of federated learning (FL) and artificial intelligence (AI) technologies within the domain of cybersecurity, emphasizing their transformative potential in enhancing threat detection and safeguarding modern digital ecosystems. The research has systematically examined the core components of cybersecurity challenges, the architecture of federated learning systems, and the pivotal role of AI, particularly deep learning, in addressing evolving cyber threats.

The study has demonstrated how FL-powered solutions can be leveraged to enhance cybersecurity performance, ensure data privacy, and address critical challenges such as sophisticated cyber attacks. These solutions manifest in several key ways:

**Cyber Threat Detection:**

- Federated learning enables collaborative training across decentralized entities, allowing the identification of unusual network patterns and potential threats without compromising sensitive data.
- By analyzing distributed datasets, such as network logs and user behavior profiles, FL-based models can flag malicious activities, including malware, phishing, and advanced persistent threats, while preserving the privacy and security of proprietary information.

**Data Privacy and Compliance:**

- FL eliminates the need for centralized data aggregation, significantly reducing the risk of data breaches and ensuring compliance with stringent privacy regulations such as GDPR.
- The exchange of model updates rather than raw data maintains data locality, enabling organizations to meet regulatory and ethical standards while fostering trust in collaborative cybersecurity frameworks.

**Model Robustness and Scalability:**

**Performance Optimization:**

By harnessing these FL-powered capabilities, cybersecurity practitioners can enhance threat detection, improve data privacy, and build resilient defenses against an ever-evolving threat landscape. The integration of AI and FL into cybersecurity frameworks leads to significant improvements in system reliability, regulatory compliance, and operational efficiency, paving the way for secure and privacy-preserving digital infrastructures.

The findings and insights presented in this dissertation serve as a valuable resource for researchers, policymakers, and industry professionals in the cybersecurity sector, guiding the adoption of FL-based solutions for enhanced threat detection and mitigation. As the

cyber threat landscape continues to evolve, the synergistic relationship between federated learning and AI technologies will play a crucial role in shaping the future of cybersecurity. This research provides a robust foundation for further innovations, fostering advancements toward a more secure, adaptive, and ethically responsible digital ecosystem.

# Bibliography

[1] X. Feng and S. Hu, "Cyber-physical zero trust architecture for industrial cyber-physical systems," *IEEE Transactions on Industrial Cyber-Physical Systems*, vol. 1, pp. 394–405, 2023. DOI: 10.1109/TICPS.2023.3333850.

[2] I. Naidji, O. Mosbahi, M. Khalgui, and A. Bachir, "Two-stage game theoretic approach for energy management in networked microgrids," in *Software Technologies*, M. van Sinderen and L. A. Maciaszek, Eds., Cham: Springer International Publishing, 2020, pp. 205–228, ISBN: 978-3-030-52991-8.

[3] I. Naidji, M. B. Smida, M. Khalgui, and A. Bachir, "Non cooperative game theoretic approach for residential energy management in smart grid," in *The 32nd Annual European Simulation and Modelling Conference*, Ghent, Belgium, 2018, pp. 164–170.

[4] I. Naidji, M. Ben Smida, M. Khalgui, A. Bachir, Z. Li, and N. Wu, "Efficient allocation strategy of energy storage systems in power grids considering contingencies," *IEEE Access*, vol. 7, pp. 186 378–186 392, 2019. DOI: 10.1109/ACCESS.2019.2957277.

[5] W. Stallings, *Effective Cybersecurity: A Guide to Using Best Practices and Standards*. Addison-Wesley Professional, 2018.

[6] S. Sharmeen and M. A. Rahman, "Ai in cybersecurity: Threats and applications," *Journal of Cybersecurity and Privacy*, vol. 1, no. 1, pp. 47–62, 2020.

[7] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," *IEEE Symposium on Security and Privacy*, pp. 305–316, 2010. DOI: 10.1109/SP.2010.25.

[8] T. Nguyen, P. N. Pathirana, A. Seneviratne, and M. Ding, "Deep learning for cybersecurity: Challenges and opportunities," *Computers & Security*, vol. 104, p. 102 258, 2021. DOI: 10.1016/j.cose.2021.102258.

[9] M. Conti, N. Dragoni, and V. Lesyk, "A survey of malware detection techniques: From traditional to machine learning," *ACM Computing Surveys (CSUR)*, vol. 48, no. 1, pp. 1–36, 2018.

[10] I. Naidji, C. E. Choucha, and M. Ramdani, "Electricity theft detection techniques using artificial intelligence: A survey," in *2024 IEEE International Conference on Advanced Systems and Emergent Technologies ($IC_ASET$)*, 2024, pp. 1–6. DOI: 10.1109/IC_ASET61847.2024.10596174.

[11] I. Naidji, M. B. Smida, M. Khalgui, and A. Bachir, "Multi agent system-based approach for enhancing cyber-physical security in smart grids," in *Proceedings of the the 33rd Annual European Simulation and Modelling Conference*, pp. 177–182.

[12] N. Abdelhamid, A. Ayesh, and F. Thabtah, "Phishing email detection based on structural properties," *Proceedings of the 2014 IEEE International Conference on Computer Systems and Applications*, pp. 646–653, 2014.

[13] K. Scarfone and P. Mell, "Guide to malware incident prevention and handling for desktops and laptops," *NIST Special Publication*, vol. 800, no. 83, 2016.

[14] J. Mirkovic and P. Reiher, "A taxonomy of ddos attack and ddos defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.

[15] L. Bilge and T. Dumitras, "Before we knew it: An empirical study of zero-day attacks in the real world," *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 833–844, 2012.

[16] C. Hadnagy, *Social engineering: The science of human hacking*. Wiley, 2018.

[17] F. L. Greitzer and D. A. Frincke, "Insider threat detection using behavioral modeling: A survey," *Proceedings of the IEEE*, vol. 102, no. 1, pp. 40–57, 2014.

[18] P. Chen, L. Desmet, and C. Huygens, "Study on advanced persistent threat detection and response," *2014 International Symposium on Research in Grey-Hat Hacking*, 2014.

[19] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[20] I. Naidji., C. Choucha., and M. Ramdani., "Decentralized federated learning architecture for networked microgrids," in *Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO*, INSTICC, SciTePress, 2023, pp. 291–294, ISBN: 978-989-758-670-5. DOI: 10.5220/0012215200003543.

[21] I. Naidji., O. Mosbahi., M. Khalgui., and A. Bachir., "Cooperative energy management software for networked microgrids," in *Proceedings of the 14th International Conference on Software Technologies - ICSOFT*, INSTICC, SciTePress, 2019, pp. 428–438, ISBN: 978-989-758-379-7. DOI: 10.5220/0007965604280438.

[22] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," *Proceedings of the 52nd Annual Design Automation Conference*, 2015.

[23] V. Sharma, S. Sood, and N. Jhanjhi, "Security and privacy issues in cloud computing: A survey," *IEEE Access*, vol. 8, pp. 88 750–88 773, 2020.

[24] E. Kraemer-Mbula and S. Wunsch-Vincent, "Solarwinds and the rise of supply chain attacks," *Nature*, vol. 589, no. 7842, pp. 500–502, 2021.

[25] R. Chesney and D. K. Citron, "Deep fakes: A looming challenge for privacy, democracy, and national security," *California Law Review*, vol. 107, no. 6, pp. 1753–1819, 2019.

[26] M. Mosca, "Cybersecurity in an era with quantum computers: Will we be ready?" *IEEE Security & Privacy*, vol. 16, no. 5, pp. 38–41, 2018.

[27] S. Kaur and H. S. Gill, "Security vulnerabilities in legacy systems: A review and roadmap," in *2023 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 2023, pp. 135–140. DOI: 10.1109/ESCI57804.2023.10214755.

[28] B. Gogoi and T. Ahmed, "Http low and slow dos attack detection using lstm based deep learning," in *2022 IEEE 19th India Council International Conference (INDICON)*, 2022, pp. 1–6. DOI: 10.1109/INDICON56171.2022.10039772.

[29] I. Ahmed and N. Kumar, "Addressing cybersecurity skills shortage in the era of digital transformation," in *2023 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, 2023, pp. 118–123. DOI: 10.1109/CyberSecurity56782.2023.10795421.

[30] Y. Li and X. Chen, "Cloud security challenges and solutions: A comprehensive review," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 3217–3231, 2023. DOI: 10.1109/TNSM.2023.3285611.

[31] R. Tamilkodi, P. S. Rani, L. G. Deepthi, M. P. Jagannadh, K. V. P. Kumar, and T. R. Teja, "Enhanced alert generation system with attacker ip for dos attacks," in *2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, 2024, pp. 590–595. DOI: 10.1109/ICACRS62842.2024.10841575.

[32] S. Sharma and R. Gautam, "Insider threat detection in cybersecurity: A comprehensive review," *IEEE Access*, vol. 11, pp. 4562–4575, 2023. DOI: 10.1109/ACCESS.2023.3241482.

[33] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015. DOI: 10.1126/science.aaa8415.

[34] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006, ISBN: 9780387310732.

[35] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016, ISBN: 9780262035613.

[36] H. Xu, L. Liu, and J. Hu, "Reinforcement learning algorithms with applications in cybersecurity," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 8, pp. 1617–1628, 2018. DOI: 10.1109/TSMC.2018.2814539.

[37] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pp. 4–11, 2014. DOI: 10.1145/2689746.2689747.

[38] E. Müller, M. Quade, M. Scherf, and M. F. Huber, "Natural language processing for intelligent threat detection and response in cybersecurity," *Journal of Information Security and Applications*, vol. 54, p. 102 554, 2020. DOI: 10.1016/j.jisa.2020.102554.

[39] J. Gera, V. K. K. Rejeti, J. N. C. Sekhar, and A. S. Shankar, "Distributed denial of service attack prevention from traffic flow for network performance enhancement," in *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, 2021, pp. 406–413. DOI: 10.1109/ICOSEC51865.2021.9591974.

[40] V. D. M. Rios, P. R. M. Inácio, D. Magoni, and M. M. Freire, "Detection and mitigation of low-rate denial-of-service attacks: A survey," *IEEE Access*, vol. 10, pp. 76 648–76 668, 2022. DOI: 10.1109/ACCESS.2022.3191430.

[41] R. Rajmohan, A. Meiappane, C. Gupta, and S. Bhavsar, "Intelligent ssh attack detection model using k-clique clustering and reinforcement learning," in *2024 International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2024, pp. 1–6. DOI: 10.1109/ICSCAN62807.2024.10894529.

[42] A. Hajjouz and E. Avksentieva, "Evaluating the effectiveness of the catboost classifier in distinguishing benign traffic, ftp bruteforce and ssh bruteforce traffic," in *2024 9th International Conference on Signal and Image Processing (ICSIP)*, 2024, pp. 351–358. DOI: 10.1109/ICSIP61881.2024.10671552.

[43] N. D. Bobade, V. A. Sinha, and S. S. Sherekar, "A diligent survey of sql injection attacks, detection and evaluation of mitigation techniques," in *2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, 2024, pp. 1–5. DOI: 10.1109/SCEECS61402.2024.10481914.

[44] T. Wang, D. Zhao, and J. Qi, "Research on cross-site scripting vulnerability of xss based on international student website," in *2022 International Conference on Computer Network, Electronic and Automation (ICCNEA)*, 2022, pp. 154–158. DOI: 10.1109/ICCNEA57056.2022.00043.

[45] W. Alsabbagh, S. Amogbonjaye, D. Urrego, and P. Langendörfer, "A stealthy false command injection attack on modbus based scada systems," in *2023 IEEE 20th Consumer Communications Networking Conference (CCNC)*, 2023, pp. 1–9. DOI: 10.1109/CCNC51644.2023.10059804.

[46] B. Aydın, H. Aydın, and S. Görmüş, "A security mechanism against man in the middle attack in 6tisch networks," in *2024 32nd Signal Processing and Communications Applications Conference (SIU)*, 2024, pp. 1–4. DOI: 10.1109/SIU61531.2024.10600741.

[47] M. Saed and A. Aljuhani, "Detection of man in the middle attack using machine learning," in *2022 2nd International Conference on Computing and Information Technology (ICCIT)*, 2022, pp. 388–393. DOI: 10.1109/ICCIT52419.2022.9711555.

[48] B. Peiris D, J. Pathmendre, A. Hasaranga V, A. Athauda B, K. Yapa, and S. Rathnayake, "Codenexa: A novel security framework for federated learning to mitigate man-in-the-middle attacks," in *2024 IEEE 15th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, 2024, pp. 709–714. DOI: 10.1109/UEMCON62879.2024.10754736.

[49] C. Liu, D. Du, C. Zhang, C. Peng, and M. Fei, "Observability analysis of networked control systems under dos attacks," in *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, 2023, pp. 1–6. DOI: 10.1109/IECON51785.2023.10312197.

[50] M. F. Ansari, A. Panigrahi, G. Jakka, A. Pati, and K. Bhattacharya, "Prevention of phishing attacks using ai algorithm," in *2022 2nd Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology (ODICON)*, 2022, pp. 1–5. DOI: 10.1109/ODICON54453.2022.10010185.

[51] J. Dafni Rose, J. N. M, and J. P. S, "Next-gen phishing detection system based on federated learning integrated cnn-lstm for sms communication," in *2024 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2024, pp. 367–372. DOI: 10.1109/ICICV62344.2024.00064.

[52] L. Hou, J. Han, B. Yang, Y. Guo, X. Wang, and G. Yang, "Research on attack trapping methods for multi-stage infiltration attacks," in *2023 International Conference on Computer Simulation and Modeling, Information Security (CSMIS)*, 2023, pp. 556–560. DOI: 10.1109/CSMIS60634.2023.00105.

[53] P. Sharma, Seema, P. S. Rana, M. Singh, A. Kumar, and A. Anand, "Defensive strategies against can-bus infiltrations," in *2025 3rd IEEE International Conference on Industrial Electronics: Developments Applications (ICIDeA)*, 2025, pp. 1–6. DOI: 10.1109/ICIDeA64800.2025.10963196.

[54] P. K, B. Nataraj, and P. Duraisamy, "An investigation on attacks in application layer protocols and ransomeware threats in internet of things," in *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, 2023, pp. 668–672. DOI: 10.1109/ICACCS57279.2023.10112669.

[55] S. Raja and K. Venkatesh, "Using honey pot technique ransomeware get detected," in *2023 International Conference on Computer Communication and Informatics (ICCCI)*, 2023, pp. 1–4. DOI: 10.1109/ICCCI56745.2023.10128365.

[56] Happy, R. Chhikara, and N. Kashyap, "A comparative analysis of machine learning prediction algorithms for detecting iot botnet activities," in *2024 International Conference on Intelligent Systems for Cybersecurity (ISCS)*, 2024, pp. 1–6. DOI: 10.1109/ISCS61804.2024.10581089.

[57] R. T. Wiyono and N. D. W. Cahyani, "Performance analysis of decision tree c4.5 as a classification technique to conduct network forensics for botnet activities in internet of things," in *2020 International Conference on Data Science and Its Applications (ICoDSA)*, 2020, pp. 1–5. DOI: 10.1109/ICoDSA50139.2020.9212932.

[58] M. u. Nisa and K. Kifayat, "Detection of slow port scanning attacks," in *2020 International Conference on Cyber Warfare and Security (ICCWS)*, 2020, pp. 1–7. DOI: 10.1109/ICCWS48432.2020.9292389.

[59] B. Hartpence and A. Kwasinski, "Combating tcp port scan attacks using sequential neural networks," in *2020 International Conference on Computing, Networking and Communications (ICNC)*, 2020, pp. 256–260. DOI: 10.1109/ICNC47757.2020.9049730.

[60] O. Team, "Brute force attacks and detection mechanisms," *Openwall Security*, 2017, Available at: https://www.openwall.com.

[61] Z. Durumeric, J. Kasten, D. Adrian, J. A. Halderman, and M. Bailey, "Security analysis of the https ecosystem," *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, pp. 429–440, 2013.

[62] L. Beaver, "Goldeneye: A layer 7 dos testing tool," *Offensive Security*, 2014.

[63] O. Foundation, "Sql injection," 2023, Available at: https://owasp.org/www-community/attacks/SQ

[64] O. Foundation, "Cross site scripting (xss)," 2023, Available at: https://owasp.org/www-community/attacks/xss/.

[65] M. Roesch, "Snort - lightweight intrusion detection for networks," *Proceedings of LISA '99: 13th Systems Administration Conference*, 1999.

[66] J. Mirkovic and P. Reiher, "Ddos attacks: Classification and countermeasures," *Computer Networks*, vol. 44, no. 5, pp. 657–686, 2004.

[67] M. Antonakakis, T. April, M. Bailey, *et al.*, "Understanding the mirai botnet," *USENIX Security Symposium*, pp. 1093–1110, 2017.

[68] NVD, *Cve-2014-0160: Openssl heartbleed vulnerability*, `https://nvd.nist.gov/vuln/detail/CVE-2014-0160`, 2014.

[69] C. I. for Cybersecurity, *Cicids 2017 dataset*, Available at: `https://www.unb.ca/cic/datasets/ids-2017.html`, 2017.

[70] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147–167, 2019.

[71] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, pp. 108–116, 2018.

[72] W. McKinney, "Data structures for statistical computing in python," *Proceedings of the 9th Python in Science Conference*, vol. 445, no. 1, pp. 51–56, 2010.

[73] M. Abadi, A. Agarwal, P. Barham, *et al.*, *Tensorflow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: `https://www.tensorflow.org/`.

[74] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[75] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[76] B. C. Ross, "Mutual information between discrete and continuous data sets," *PLoS one*, vol. 9, no. 2, e87357, 2014.

[77] B. McMahan, E. Moore, D. Ramage, S. Hampson, *et al.*, "Communication-efficient learning of deep networks from decentralized data," *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

[78] P. Kairouz, B. McMahan, *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[79] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[80] T. Li, A. K. Sahu, M. Zaheer, *et al.*, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[81] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *NeurIPS*, 2020.

[82] S. P. Karimireddy, S. Kale, S. Reddi, S. U. Stich, and A. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," *International Conference on Machine Learning*, 2020.

[83] S. J. Reddi, Z. Charles, M. Zaheer, *et al.*, "Adaptive federated optimization," *International Conference on Learning Representations (ICLR)*, 2021.

[84] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSP*, vol. 1, pp. 108–116, 2018.

[85] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *International Conference on Learning Representations (ICLR)*, 2019.

[86] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.

[87] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[88] K. Cho, B. Van Merriënboer, C. Gulcehre, *et al.*, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[89] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, 1998, pp. 2278–2324.

[90] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[91] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems (MLSys)*, 2020.

[92] J. Konečný, B. McMahan, F. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *arXiv preprint arXiv:1610.05492*, 2016.

[93] R. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," in *NeurIPS Workshop on Machine Learning on the Phone and other Consumer Devices*, 2017.

[94] K. e. a. Bonawitz, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[95] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, *et al.*, "Large scale distributed deep networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[96] M. Abadi, "Tensorflow: Learning functions at scale," in *Proceedings of the 21st ACM SIGPLAN international conference on functional programming*, 2016, pp. 1–1.

[97] D. J. Beutel, T. Topal, A. Mathur, *et al.*, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.