الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة قاصدي مرباح ورقلة
كلية التكنولوجيات الحديثة للمعلومات والاتصال
قسم الإعلام الآلي وتكنولوجيا المعلومات

Republic of Algeria Democratic and People's
Ministry of Higher Education and Scientific Research
University of KASDI Merbah – Ouargla
Faculty of New Technologies of Information and Communication
Department of Computer Science and Information Technology

Professional Master Thesis

Domain: Mathematics and Computer Science
Sector: Computer Science
Specialty: Network Administration and Security

Theme

A Blockchain-Federated Learning for Privacy-Preserving Intrusion Detection in IoMT

Presented by : ABBAZI ZINEB and BOUHNIK KATIA
Publicly discussed:

Jury member:

President	**Dr.Boukhamla Akram**	**UKM Ouargla**
Supervisor	**Dr. Benkaddour Mohammed Kamel**	**UKM Ouargla**
Examiner	**Dr.Messiaid Abdessalam**	**UKM Ouargla**

Academic : 2025/2024

Acknowledgment

﴿ وَقُلِ ٱعۡمَلُواْ فَسَيَرَى ٱللَّهُ عَمَلَكُمۡ وَرَسُولُهُۥ وَٱلۡمُؤۡمِنُونَ ﴾

[At-Tawbah: 105]

We praise Allah Almighty and thank Him for granting us patience and strength to complete this work.

We extend our deepest gratitude and appreciation to our parents, who have been our greatest support and accompanied us with their prayers and encouragement at every step of this journey.

We also express our sincere thanks and appreciation to our esteemed professor, **Dr. BENKADDOUR Mohammed Kamel**, who was a model of patience and understanding, guiding us with open-hearted advice and continuous support; for this, we owe him our utmost respect and gratitude.

Finally, we would like to thank our families and everyone who encouraged and stood by us with a word, advice, or support throughout our academic journey. To them, we offer our sincerest feelings of gratitude and respect.

إهداء

«أولًا وقبل كل شيء، وبكل امتنان لله، أهدي هذا العمل إلى أولئك الذين كانوا دائمًا سندي ونبض قلبي والدافع الحقيقي للاستمرار – إلى والدتي الحبيبة، التي أضاءت دعواتها وحنانها طريقي؛ وإلى والدي العزيز، الذي علّمني قيمة المثابرة والعمل الجاد، وإلى إخوتي الذين كانوا دائمًا مصدر قوتي ورفاق دربي. كما أوجّه شكري الخالص إلى ... الأستاذة الفاضلة خديجة عامر، التي كان لإلهامها وتشجيعها دور كبير في تنمية حبي لهذا التخصص وتعزيز إيماني بنفسي وبأحلامي. وأعبر عن عميق امتناني لعائلتي العزيزة، ولكل معلم ومعلمة، ولكل من وقف إلى جانبي، وآمن بي، وساندني طوال هذه الرحلة.»

``ZINEB ABBAZI''

«قال تعالى:
﴿ فَرِحِينَ بِمَا آتَاهُمُ اللَّهُ مِنْ فَضْلِهِ ﴾»

«الحمدلله الذي بنعمته تتم الصالحات، وبتوفيقه تتحقق الغايات. أهدي هذا العمل المتواضع عربون محبة وامتنان إلى من كان لهم الفضل بعد الله في وجودي وتربيتي وتعليمي... إلى من غرسا في نفسي الأخلاق والصدق والإصرار... إلى أبي العزيز... إلى أمي الحبيبة... أسأل الله أن يحفظكما. إلى شريك دربي وسندي الذي كان لي دعماً وسنداً في كل الأوقات ... إلى من كان لي مصدر إلهام ودعم ... إلى زوجي الحبيب. إلى سندي في الحياة وقوتي ... إلى ملهمي نجاحي إلى خيرة أيامي ... إخوتي محمد الطيب، ضياء الدين ...، أختي شروق. إلى الذي كان على الدوام ناصحاً أميناً ... عمي محمد رحمه الله ... إلى الصديق والمعلم، فهو بمثابة الأب ... عمي يونس. إلى صاحبة العزم والمثابرة، لكِ مني كل الشكر والتقدير ... إلى شريكتي في هذا العمل زينب عبازي. إلى كل من أحب.»

``KATIA BOUHENIK''

Abstract

This project aims to develop an intelligent intrusion detection system for Internet of Medical Things (IoMT) environments by integrating three key technologies: Artificial Intelligence, Federated Learning, and Blockchain. Initially, centralized learning models were adopted; however, they demonstrated limitations in preserving data privacy and posed a single point of failure that threatens system stability. To address these challenges, Federated Learning was employed as an alternative that enables training models locally on edge devices without sharing raw data, thereby enhancing privacy and reducing reliance on centralized servers. Nevertheless, the presence of a central server in traditional federated learning remains a critical security vulnerability. Therefore, Blockchain technology was integrated to provide a decentralized and secure infrastructure for transparently exchanging and verifying model updates. This integration has contributed to enhancing the security and reliability of the system, mitigating the risks of cyberattacks, and offering a promising solution for securing intelligent healthcare systems based on IoMT technologies.

Keywords:

Intrusion Detection System (IDS), Internet of Medical Things (IoMT), Federated Learning (FL), Blockchain, Privacy-Preserving,

ملخص

يهدف هذا المشروع إلى تطوير نظام ذكي لكشف التسللات في بيئات الإنترنت للأشياء الطبية ،(IoMT) من خلال دمج ثلاث تقنيات رئيسية: الذكاء الاصطناعي، والتعلم الفيدرالي، وتقنية البلوكشين. في المرحلة الأولى، تم اعتماد نماذج التعلم المركزي، إلا أنها أظهرت محدودية في الحفاظ على خصوصية البيانات، بالإضافة إلى وجود نقطة فشل واحدة تهدد استقرار النظام. ولتجاوز هذه الإشكالية، تم اعتماد التعلم الفيدرالي كبديل يُمكّن من تدريب النماذج محليًا على أجهزة الأطراف دون الحاجة إلى مشاركة البيانات الخام، مما ساهم في تعزيز الخصوصية وتقليل الاعتماد على الخوادم المركزية. ومع ذلك، فإن وجود خادم مركزي في هذا النمط من التعلم ظل يشكّل نقطة ضعف أمنية قائمة. لذا، تم دمج تقنية البلوكشين لتوفير بنية لامركزية وآمنة لتبادل وتوثيق تحديثات النماذج بطريقة شفافة وغير قابلة للتلاعب. ساهم هذا التكامل في رفع مستوى الأمان والموثوقية في النظام، والحد من المخاطر المرتبطة بالهجمات السيبرانية، مما يجعله حلاً واعدًا لتأمين أنظمة الرعاية الصحية الذكية المعتمدة على تقنيات .IoMT

الكلمات المفتاحية :
نظام كشف التسللات , إنترنت الأشياء الطبية ,(IoMT) التعلم الفيدرالي , ،(FL) تقنية البلوكشين، الحفاظ على الخصوصية.

Contents

List of Figures

List of Tables

List of abbreviation

AI	Artificial Intelligence
BCFL	*Blockchain-based Federated Learning*
CIANA	*Confidentiality, Integrity, Availability, Non-Repudiation, and Authentication*
CIC	*Canadian Institute for Cybersecurity*
CL	*Centralized Learning*
CNN	*Convolutional Neural Network*
DApps	*Decentralized Applications*
DDoS	*Distributed Denial of Service*
DL	*Deep Learning*
DoS	*Denial of Service*
ECG	*Electrocardiogram*
FedAvg	*Federated Averaging*
FL	*Federated Learning*
FN	*False Negative*
FP	*False Positive*
FTL	*Federated Transfer Learning*
GDPR	*General Data Protection Regulation*
HFL	*Horizontal Federated Learning*
HIDS	*Host-Based Intrusion Detection System*
HIPAA	*Health Insurance Portability and Accountability Act*
IDS	*Intrusion Detection System*
IID	*Independent and Identically Distributed*
IMDs	*Implantable Medical Devices*
IoMT	*Internet of Medical Things*
IoWDs	*Internet of Wearable Devices*
JSON	*JavaScript Object Notation*
KNN	*K-Nearest Neighbors*
LAN	*Local Area Network*
LSTM	*Long Short-Term Memory*
ML	*Machine Learning*
MQTT	*Message Queuing Telemetry Transport*
NIDS	*Network-Based Intrusion Detection System*
NLP	*Natural Language Processing*
P2P	*Peer-to-Peer*
PBFT	*Practical Byzantine Fault Tolerance*
PCA	*Principal Component Analysis*
PoS	*Proof of Stake*
PoW	*Proof of Work*

ReLU	*Rectified Linear Unit*
RNN	*Recurrent Neural Network*
ROC	*Receiver Operator Characteristic*
SGD	*Stochastic Gradient Descent*
SIEM	*Security Information and Event Management*
SMOTE	*Synthetic Minority Oversampling Technique*
SVM	*Support Vector Machine*
TFF	*TensorFlow Federated*
TN	*True Negative*
TP	*True Positive*
VFL	*Vertical Federated Learning*

General Introduction

In recent decades, the exponential growth in computer science and communication technologies has led to the generation and exchange of vast amounts of data across various domains. One of the most prominent sectors affected by this advancement is the Internet of Medical Things (IoMT), which integrates medical devices and healthcare systems via the Internet to enhance patient care and improve the efficiency of medical operations. However, this extensive connectivity exposes such systems to a wide range of cyber threats, including malware, denial-of-service (DoS) attacks, and data breaches, which can severely compromise patient privacy and reduce the availability of critical medical services [7][8].

Traditional security solutions, such as firewalls and signature-based intrusion detection systems (IDS), have proven insufficient in the face of sophisticated and zero-day attacks. This has paved the way for the integration of artificial intelligence (AI), particularly machine learning (ML), to enhance IDS by leveraging its ability to recognize patterns and detect anomalous behavior based on historical data. Despite their effectiveness, most ML-based IDS solutions still rely on centralized training models, which introduce challenges related to privacy violations, communication overhead, and single points of failure[9][10] .

To address these challenges, federated learning (FL) has emerged as a privacy-preserving distributed learning paradigm that enables local training of models on edge devices without sharing raw data. This reduces privacy risks and resource requirements for data storage and transmission. However, conventional FL frameworks still depend on a central coordinator to aggregate model updates, reintroducing the issue of centralized failure points [11][12].

In this thesis, we address the pressing security and privacy challenges posed by the growing adoption of Internet of Medical Things (IoMT) technologies. After analyzing the limitations of traditional centralized intrusion detection systems and machine learning approaches, we explored the advantages of federated learning (FL) as a decentralized and privacy-aware alternative. However, recognizing the security limitations inherent in FL's reliance on a central server, we proposed a hybrid approach by integrating blockchain technology into the FL pipeline.

Therefore, this research aims to integrate blockchain technology with federated learning to enhance the reliability and security of IDS in IoMT environments. Blockchain, as a decentralized and tamper-proof distributed ledger, provides a secure and transparent framework for recording model updates and facilitating trustless collaboration among participating entities without relying on a central authority. This integration effectively addresses the limitations of centralized coordination, enhances data integrity, and improves the trustworthiness of the learning process . Accordingly, this thesis seeks to design and develop an intelligent IDS tailored for IoMT by leveraging the combined advantages of federated learning and blockchain. The proposed system aims to ensure data privacy, scalability, robustness against attacks, and improved accuracy in detecting modern cybersecurity threats .

Roadmap of the Thesis

This thesis is organized into four main chapters, each addressing a distinct aspect of the proposed intelligent intrusion detection system in the context of the Internet of Medical Things (IoMT):

- **Chapter 1:**
  This chapter introduces the Internet of Medical Things (IoMT) and its growing impact on modern healthcare. It discusses the types, architecture, and applications of IoMT, as well as the critical security and privacy challenges associated with connected medical devices. The chapter also explains the need for Intrusion Detection Systems (IDS) in IoMT, outlines their types and functionalities, and highlights the importance of integrating IDS into healthcare systems to protect sensitive patient data and ensure system reliability.

- **Chapter 2:**
  This chapter reviews the role of Artificial Intelligence (AI), including Machine Learning (ML) and Deep Learning (DL), in enhancing IDS capabilities. It introduces key AI models, common evaluation metrics for IDS performance, and presents state-of-the-art research in applying ML/DL techniques to detect intrusions in IoMT environments. The chapter also addresses existing challenges and limitations, such as data imbalance, computational overhead, and high false-positive rates.

- **Chapter 3:**
  This chapter focuses on the concept of federated learning and the blockchain technology, explaining the mechanisms of each and their respective advantages. It explores how both technologies can be integrated to build a decentralized and secure intrusion detection system known as BCFL. This integration enhances privacy preservation while ensuring the transparency and immutability of model updates, thus improving the security and reliability of IoMT environments.

- **Chapter 4:**
  This chapter presents the practical aspects of the study, including the implementation environment, tools used, experiment setup, and evaluation metrics. It provides a detailed analysis of the obtained results to assess the performance of the proposed system in terms of accuracy, efficiency, and reliability.

Chapter 1

An Overview of Internet of Medical Things and Intrusion Detection Systems

1.1 Introduction

IoT technologies are used across a wide range of industries, each with its own unique security needs. However, these technologies are not always developed with the specific risks of each sector in mind. In the healthcare domain, IoT devices-referred to as the Internet of Medical Things (IoMT)-play a vital role in supporting essential medical functions and services. IoMT integrates IoT communication protocols with healthcare systems and medical devices, enabling more connected and responsive care.[13]

As healthcare systems become more integrated, they become more vulnerable to cyber threats. Patients' sensitive information is now on network and the security as well as privacy of medical data have become major concerns to say least. This is where Intrusion Detection Systems come into play. For monitoring network activity and uncovering unauthorized access occurring or perhaps even some form of malicious behavior, IDS is a necessity.

This chapter aims to provide an overview of the Internet of Medical Things and the role of Intrusion Detection Systems in securing such environments. It introduces the concept of IoMT, outlines the security challenges it faces, and explores how IDS solutions can help safeguard sesitive medical data and ensure reliable healthcare delivery.

1.2 Overview of IoMT

The Internet of Medical Things (IoMT) is a specialized domain of IoT that connects medical devices, software, sensors, and networks to healthcare systems. These embedded systems, often integrated with technologies like surgical robots, enable real-time data collection, transmission, and analysis to enhance patient care and clinical efficiency. However, their growing complexity and involvement in critical medical procedures raise important concerns about data security and patient privacy.[14]

1.3 Types of IoMT Devices and Applications

IoMT systems play a vital role in supporting a wide range of medical needs. Some devices are essential, such as implantable medical devices (IMDs) like pacemakers used for managing heart conditions. Others are designed to enhance the overall healthcare experience—these are often wearable devices like smartwatches that help monitor health in real time. Based on these functions, IoMT systems can generally be grouped into two main types: implantable medical devices (IMDs) and Internet of Wearable Devices (IoWDs).[1]

1.3.1 Implantable Medical Devices (IMDs)

Any device that is implanted to replace, support, or enhance a biological structure is an IMD. For example, a pacemaker is an IMD that helps control abnormal heart rhythms, i.e., by promoting the heart to beat at a normal rate if it is beating too fast or too slow . Fig. shows several popular IMDs and their placement locations in the human body. Recently, wireless IMDs have been proposed to solve problems associated with wired IMDs, e.g., infection and cable breakage . IMDs are mostly very small and have very long battery life. Hence, low power consumption, small storage space, and small batteries that last long are essential requirements for these devices to stay inside a human body for a long time.

1.3.2 Internet of Wearable Devices(IoWDs)

These are devices worn by individuals to monitor their biometrics, e.g., heart rate, and may help improve individuals' overall health. Examples include smartwatches, fall detection band, electrocardiogram (ECG) monitors, and blood pressure monitors. Smartwatches are currently one of the most known forms of IoWDs

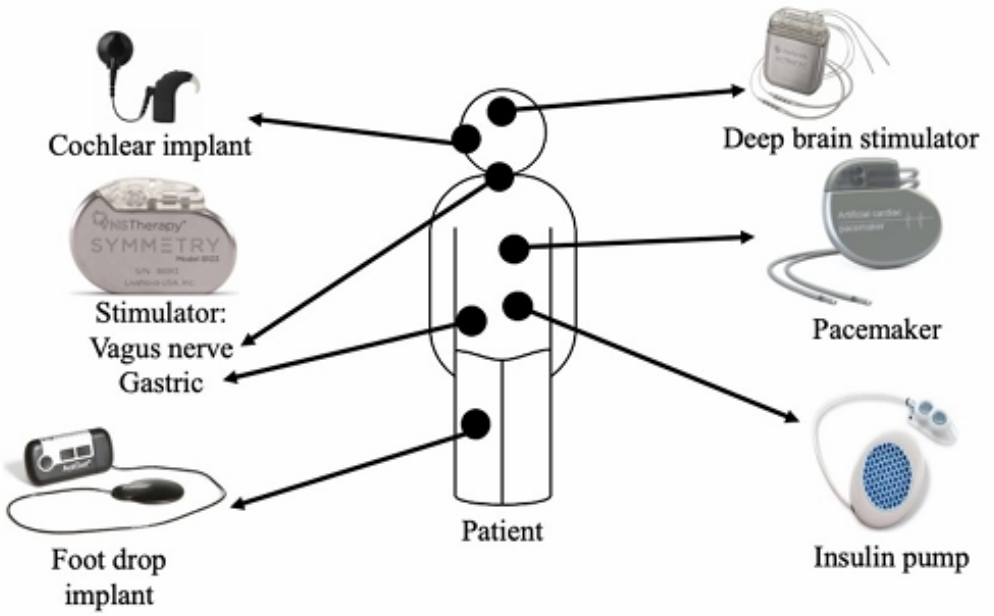


Figure 1.1: Examples of IMDs and their locations in the human body.[1]

to monitor biometrics such as heart rate and movement. The monitoring can be used to detect slow and fast heartbeats when the individual is not active. The new watches also support fall detection and ECG readings to detect atrial fibrillation (irregular heartbeat) medical conditions. They are currently widely used for non-critical patient monitoring. However, these devices have sensor accuracy and battery life limitations; thus, not likely to replace IMDs in critical conditions.

1.4 IoMT Systems Architecture

Most modern IoMT systems are structured into four main layers, covering the entire data journey—from collecting biometric information from the patient to storing and displaying it for medical analysis. These layers enable both doctors and patients to access and monitor health data in real-time, often through cloud platforms. With recent advancements, implantable medical devices (IMDs) and wearable devices (IoWDs) now often share a similar architecture. For instance, devices like Medtronic pacemakers can communicate directly with gateways, allowing seamless integration into the IoMT ecosystem. Each layer is described below:[1]

1.4.1 Sensor Layer

This is where everything begins. Small sensors, either worn or implanted in the body, continuously collect vital biometric data such as heart rate or glucose levels. These sensors communicate wirelessly using protocols like Bluetooth, Wi-Fi, or MedRadio.[1]

1.4.2 Gateway Layer

Since sensors have limited processing power, the raw data they collect is sent to a more capable device—like a smartphone or a dedicated access point. Here, the data can be briefly stored, validated, and even undergo basic AI-driven analysis before being securely sent to the cloud.[1]

1.4.3 Cloud Layer

In this layer, data is stored long-term and analyzed more deeply. It enables remote monitoring, detects potential health issues, and ensures secure access for healthcare providers. It also manages system security by generating unique IDs and encryption keys.[1]

1.4.4 Visualization/Action Layer

Finally, the processed data is presented to physicians and patients in an understandable format. Based on this information, healthcare professionals can make informed decisions, such as adjusting treatments or prescribing medications.[1]

Figure 1.2: IoMT system architecture.[1]

1.5 Privacy and security issue in IoMT

The Internet of Medical Things (IoMT) brings great advantages to patient care, but it also raises serious concerns about the security and privacy of sensitive medical data. Because this data contains deeply personal information, mishandling it can have serious consequences for both patients and healthcare providers. That's why it's essential to treat medical data with care and to use smart, secure IoMT solutions that ensure its protection and confidentiality at all times.[3]

We focused on the security and privacy challenges of IoMT, emphasizing the need for strong external protection. Since smart medical devices can't detect threats on their own, it's essential to apply defense mechanisms that match the type of data being handled. To guide this, we use the CIANA principles—Confidentiality, Integrity, Availability, Non-Repudiation, and Authentication—to determine the right level of protection needed for sensitive health information.[3]

1.6 IoMT Threats at Different Stages

IoMT systems must ensure that patients' data is protected throughout every stage—whether it's being collected, transmitted, or stored. Each of these stages involves different components of the four-layer architecture,

working together to safeguard sensitive health information at every step of the process.[1] [15]

1.6.1 Data Collection

The first step in any IoMT system is gathering patient information through sensors—either worn or implanted. At this point, both software-based attacks (like tampering with data) and hardware manipulation can occur. Since these sensors deal directly with a patient's health, any compromise could be life-threatening. That's why strong protection at this stage is critical.[1]

1.6.2 Data in Transit

As data moves between devices across the four architectural layers—such as from a sensor to a smartphone or gateway—it becomes vulnerable to interception, alteration, or blocking. Securing this communication path ensures the integrity and reliability of health data as it travels through the system.[1]

1.6.3 Data in Storage

Once transmitted, data is stored—often in the cloud—where it becomes a prime target for cyber threats like credential theft or denial-of-service attacks. Since stored data tends to remain static, it's especially susceptible to unauthorized access. Keeping this information safe is crucial to preserving patient privacy and system trustworthiness.[1]

1.6.4 IoMT Security Requirements

Security is a vital issue for data exchange between medical devices in IoMT-based systems. The data of the patient are very sensitive, as they are particularly personal. The medical history and symptoms of the patients are processed and recorded in IoMT-based systems.[1]

Due to the patient data's sensitivity and safety, a set of requirements that can ensure IoMT systems' security at all layers is needed.[1]

These set has been derived from CIANA considerations and consists of the following security requirements explained below:

- **Confidentiality/Privacy**: Confidentiality, or privacy, is the top priority, as a huge amount of sensitive and personal data is processed and stored across IoMT devices. These data should be accessible to the authorized user via a proper authentication mechanism; furthermore, the stored data should be encrypted to avoid ease of access by an adversary. The encryption adopted must be secure enough to safeguard from attackers.[15]

- **Integrity**: The integrity of data in the IoMT is essential, as these inputs are used for the treatment of the patients. Integrity ensures that the data has not been modified, either during transmission or during the storage process. The modification of data may consist of deleting it, adding false values, etc. It is important to safeguard the sensitive data of the IoMT to stop unauthorized access.[15]

- **Authentication**: The validation of authorized users for communication is key to performing identity authentication. To authenticate an identity, both communicating parties must mutually verify themselves. The transfer of data and information occurs after mutual authentication. The IoMT consists of various services, including the cloud, that need adequate authentication. The authentication mechanism may vary according to the various IoMT-based applications.[15]

- **Non-Repudiation**: This is particularly crucial because an illegal entity could not deny the validity of the messages. To validate the messages, the proof of origin is mentioned along with the integrity of the

data. The denying of the message becomes extremely tough when the source or origin is mentioned. The concept of a digital signature is widely used for implementing non-repudiation.[15]

- **Availability**: This feature ensures that the information and services are accessible to authorized users only. The availability feature is exploited by the adversary or attacker by executing a denial-of-service (DoS) attack. This attack is generally launched when confidentiality and integrity of the system remain intact and the attacker is unable to compromise these two features.[15]

Figure 1.3: CIANA: The pillars of information security [2]

- **Authorization**: The ability to allow authenticated users to only execute commands to which they are authorized. Similar to confidentiality, authorization can be achieved using proper data encryption and access control techniques.[1]

- **Anonymity**: Anonymity in IoMT refers to concealing the identity of patients and healthcare providers (Lone et al., 2020). Protecting the user's end and maintaining the integrity of the system is crucial. It can help prevent passive attacks, which often occur when the attacker has knowledge of the user's activity without knowing their identity.[1]

- **Backward and Forward Secrecy**: The backward and forward features are an integral part of the IoMT-based system since it consists of hardware devices in large numbers. Forward secrecy suggests that, if any device leaves the IoMT system, then it should be discontinued, so that it could not access any communication within the existing system. Moreover, in case of backward secrecy, newly installed devices in IoMT systems should not have any access to previously transmitted messages.[15]

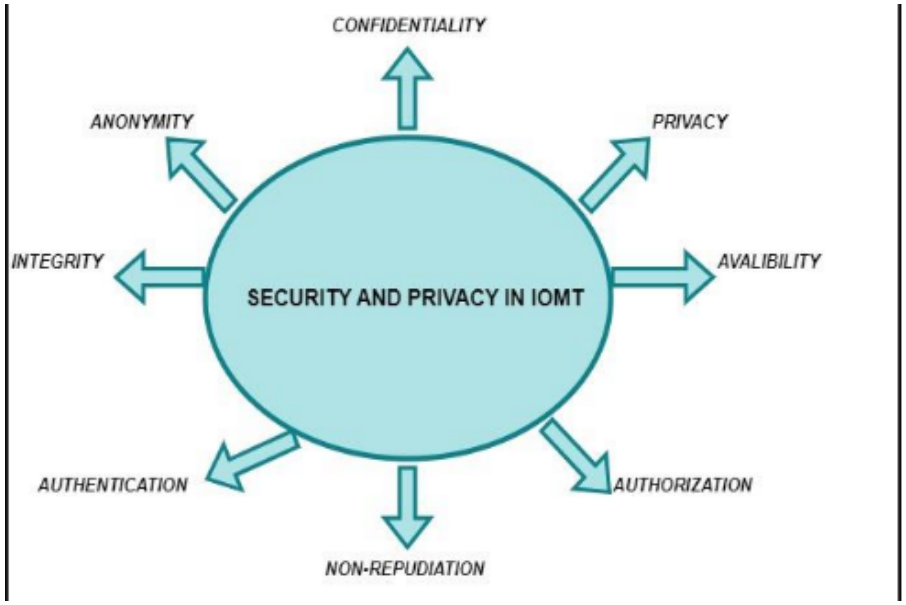


Figure 1.4: Counter measurement for security and privacy in IoMT [3]

1.7 Challenges and Security Threats in IoMT

The proliferation of the Internet of Medical Things (IoMT) offers transformative potential for enhancing patient care through interconnected medical devices, sensors, and software applications. However, integrating IoMT into healthcare systems introduces a range of domain-specific challenges and security threats. These must be carefully addressed to ensure the safety, privacy, and reliability of medical data and IoMT-enabled devices.[16]

1.7.1 Challenges in IoMT:

This section provides a focused look at some of the major challenges and considerations in the Internet of Medical Things (IoMT). These include issues related to interoperability, Data Management, Scalability, and regulatory compliance-each of which plays a critical role in ensuring that IoMT systems are safe, effective, and aligned with the demands of modern healthcare environments.

- **Interoperability:**IoMT devices come from different manufacturers and often use a variety of communication protocols and standards. To ensure smooth integration and data exchange across these diverse systems, strong interoperability frameworks are essential. Without them, coordinated care and continuous patient monitoring become difficult.
- **Data Management:**With IoMT devices generating large volumes of data in real time, managing this information efficiently is a major challenge. Ensuring data quality, handling storage needs, and enabling fast processing are all critical to support accurate and timely clinical decisions.
- **Regulatory Compliance:**IoMT systems must comply with strict healthcare regulations like HIPAA, which require safeguarding patient data privacy and security. Meeting these standards involves not only secure system design but also regular audits and strong data protection practices.
- **Scalability:**As IoMT adoption expands, healthcare systems need to scale effectively. This means upgrading network infrastructure, boosting processing power, and ensuring performance doesn't suffer as more devices and data come online. Scalability is key to delivering consistent, high-quality care.

1.7.2 Security Threats in IoMT:

This section discusses various security threats and vulnerabilities faced by the IoMT:

- **Data Breaches:**IoMT devices carry sensitive patient data, making them a prime target for hackers. Breaches can lead to serious privacy violations like identity theft.
- **Malware Ransomware:**Malicious software can disrupt device function or lock systems, threatening patient care and demanding ransom.
- **Device Hijacking:**Attackers may take over devices for harmful purposes, including launching DDoS attacks or altering treatments.
- **Device Hijacking:**Attackers may take over devices for harmful purposes, including launching DDoS attacks or altering treatments
- **Insider Threats:**Risks can also come from within healthcare organizations, whether through negligence or intentional harm.
- **Physical Attacks:**Devices can be stolen or tampered with, so strong physical security is essential.

Figure 1.5: Reasons for IoMT vulnerabilities

1.8 Overview of Intrusion Detection Systems (IDS)

An Intrusion Detection System (IDS) acts as a digital guardian at the network's entry and exit points, carefully monitoring traffic to spot suspicious or harmful activity. Its main purpose is to detect malicious behavior in real time and help reduce the impact of potential cyberattacks. As Elrawy (2018) described, an IDS analyzes data packets traveling through the network to identify threats and protect sensitive systems. In the context of IoMT environments—where patient safety and data privacy are paramount—the role of IDS becomes even more critical. It helps safeguard the interconnected medical devices by monitoring for unusual activity, flagging potential breaches, and supporting timely response to threats.[17]

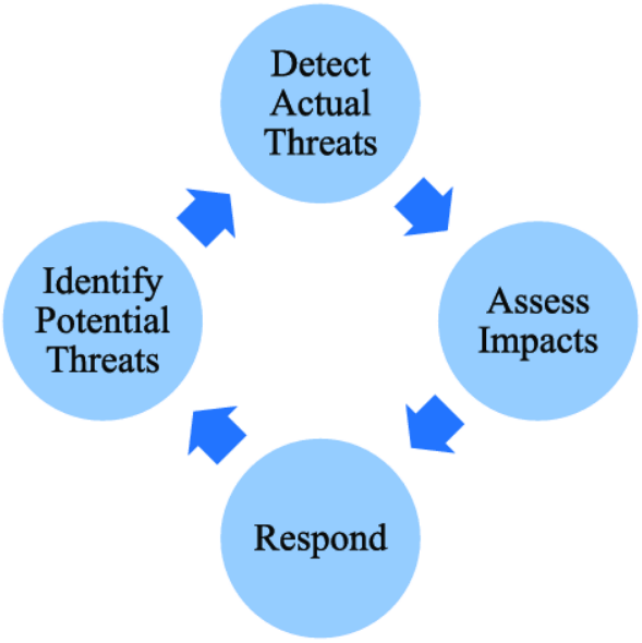


Figure 1.6: Intrusion Detection Systems (IDS)

1.9 Core Functions of an Intrusion Detection System (IDS)

Basically, an IDS can provide two main functions:

- **Information Recording**: An IDS is designed to monitor target systems and record critical information locally. This data can then be shared with centralized analysis systems to support deeper insights and informed security decisions.[4]
- **Alert Generation**: One of the key roles of an IDS is to generate alerts that notify security teams of detected anomalies. The effectiveness of an IDS often depends on how accurately it raises alarms, with false positives being a critical factor to manage.[4]

1.10 Classifying the Guardians: Types of Intrusion Detection Systems

An Intrusion Detection System (IDS) is generally classified into two main types: Host-based IDS (HIDS) and Network-based IDS (NIDS). HIDS monitors activity on individual devices, while NIDS keeps an eye on network traffic. Most real-world IDS solutions combine these approaches to provide more complete and reliable protection across the network.[4]

1. **Host-Based Intrusion Detection System (HIDS)**: It works directly on individual devices, such as computers or nodes, to monitor and analyze their internal activities. Its main job is to detect unusual or suspicious behavior that could indicate a security breach. For instance, it might flag a program that's trying to access system resources in an unusual way or has made harmful changes to system settings.[18]

   HIDS was actually the first type of IDS developed and offers some advantages over Network-Based IDS (NIDS). Because it operates within the system itself, it can analyze complete data sessions, making it immune to common evasion techniques like session splicing. It can even monitor encrypted communications before they're encrypted.

Beyond detecting intrusions, HIDS can perform deeper system-level checks, such as verifying file integrity, analyzing logs, monitoring the registry, detecting rootkits, and even taking direct action in response to threats.

2. **Network-Based Intrusion Detection System (NIDS)**: It monitors traffic across a local area network (LAN) to spot suspicious or unauthorized activity. Unlike HIDS, which works within a single device, NIDS is typically positioned along the network wire, watching the data as it flows between multiple devices.

   It scans incoming and outgoing packets for unusual patterns that could indicate an attack. If something suspicious is found, NIDS can alert administrators or even take action—like blocking the source of the threat—to help protect the network in real time.[18]

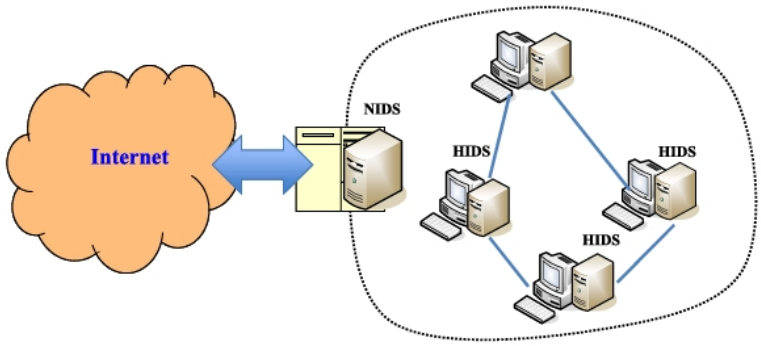


Figure 1.7: The deployment of HIDS and NIDS in a network environment.[4]

Based on the detection approaches, intrusion Detection Systems (IDS) typically use either signature-based or anomaly-based detection system[4]:

3. **Signature-based Intrusion Detection System**: It works by matching system or network activity against a list of known attack patterns, or signatures. It's fast, efficient, and produces fewer false alarms since it only flags what's already recognized as malicious. However, it can't detect new or unknown threats—called zero-day attacks—because those signatures don't exist yet. So, its effectiveness depends heavily on how current the signature database .[18]

4. **Anomaly-based Intrusion Detection System**:works by learning what "normal" behavior looks like on a system or network, and then flagging anything that strays from that pattern. It builds a baseline by observing regular activity, like typical usage or bandwidth, and treats significant deviations as potential threats . Its biggest strength is the ability to detect zero-day attacks, since it doesn't rely on known signatures. Each system has its own unique behavior profile, making it harder for attackers to predict what might trigger an alert.

   However, this approach can be complex. It needs time to learn normal behavior, especially in a new environment, and it often produces more false alarms. Plus, it can be challenging to trace an alert back to the exact event that caused it .[18]

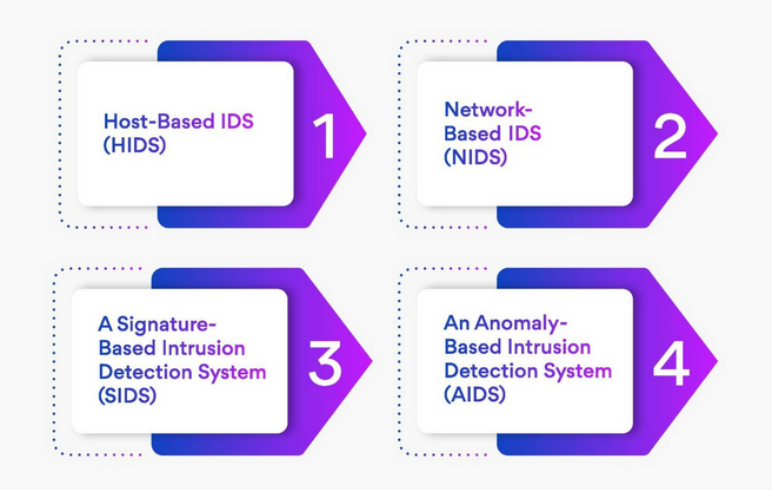


Figure 1.8: Types of Intrusion Detection Systems (IDS) [5]

Table 1.1: Comparison between different IDS classifications

Type	Advantages	Disadvantages
HIDS (Host-Based IDS)	1. More accurate in intrusion detection 2. Able to detect encrypted attacks 3. Does not require additional hardware	1. Higher cost 2. May cause performance issues or resource hogging
NIDS (Network-Based IDS)	1. Low cost 2. Detects network-based attacks such as denial-of-service attacks	1. High fluctuations in network traffic can cause packet loss 2. Requires more CPU power and resources in large-scale LANs 3. Unable to analyze encrypted packets
Anomaly-Based Detection	1. Ability to detect zero-day attack attempts 2. Low false negative rate	1. Slow to work when placed in a new environment 2. High false positive rate 3. Low detection rate for known attacks
Signature-Based Detection	1. High response time for known attacks 2. Low false positive rate	1. Limited capability to detect zero-day attacks 2. Signature database must be updated frequently

1.11 Core Components of an IDS:Understanding Its Five Pillars

Most IDS implementations are built upon five essential components, each playing a critical role in detecting and managing security threats:

1. **Sensors (Data Acquisition Units):**These are responsible for collecting raw data. Network sensors monitor

traffic at key points in the network, while host-based sensors observe system-level activity such as file access, logs, and running processes.

2. **Data Processing and Analysis Engine:**This is the core of the IDS. It analyzes the collected data using two main techniques:
   - **Signature-based Detection.**
   - **Anomaly Detection.**
3. **Alert Generation Engine:**When potential intrusions are detected, this component generates detailed alerts containing key information like the intrusion type, timestamp, and source IP. These alerts are sent to:
   - **Security Personnel:** For investigation and response actions.
   - **Security Information and Event Management (SIEM) System:**A central repository that aggregates security events from various sources, including IDS alerts, to facilitate a comprehensive view of security posture.
4. **Management Interface:**This provides administrators with tools to interact with the IDS. It supports tasks like setting detection rules, configuring sensors, adjusting alert thresholds, and monitoring both real-time and historical system activity. Note: Not all IDS solutions include a management interface.
5. **Knowledge Base:** The IDS maintains a repository that includes:
   - **Attack Signatures:**A well-maintained database of known attack signatures that facilitates signature-based detection.
   - **Security Rules:**Custom rules defined by the security administrator to identify suspicious behavior specific to the organization's network or system.
   - **Alert History:**A chronological record of all generated alerts, including timestamps, details of the detected activity, and the current investigation status.

1.12 The Role of IDS in IoMT Security

Intrusion Detection Systems play a crucial role in securing IoMT environments by monitoring traffic, identifying threats, and enabling timely response to potential attacks.

1. **Anomaly Detection:**Machine learning-based IDSs help detect unusual patterns in IoMT traffic, indicating possible threats or intrusions.
2. **Real-Time Alerts:** They generate timely alerts, allowing healthcare staff to respond quickly to security incidents.
3. **Enhanced Visibility:**Continuous monitoring improves insight into network activity, helping identify weaknesses and strengthen defenses.

1.13 Conclusion

This chapter laid the foundation for understanding the Internet of Medical Things (IoMT) and the crucial role of Intrusion Detection Systems (IDS) in safeguarding its ecosystem. We explored the structure and types of IoMT systems—implantable and wearable devices—organized across four key architectural layers. While IoMT brings numerous benefits to healthcare through real-time monitoring and improved diagnostics, it also introduces

significant challenges such as data privacy, system interoperability, and exposure to diverse cyber threats. The CIANA framework was highlighted as a guiding standard for securing sensitive data throughout its lifecycle, and the integration of intelligent IDS solutions—particularly those enhanced with machine learning—was presented as a vital defense mechanism.

The next chapter will begin by exploring the broader field of Artificial Intelligence (AI), with a particular focus on the fundamentals of Machine Learning (ML) and Deep Learning (DL). It will introduce common models, techniques, and their relevance in cybersecurity. The discussion will then narrow to how ML and DL are specifically applied in Intrusion Detection Systems (IDS) within Internet of Medical Things (IoMT) environments. Finally, the chapter will review existing literature and related works in this domain to position our study within the current scientific landscape.

Chapter 2

Background and literature review

2.1 Introduction

As healthcare technologies continue to evolve, the integration of Artificial Intelligence (AI) into medical systems has become increasingly important. In particular, the Internet of Medical Things (IoMT) has opened up new opportunities for real-time monitoring, diagnosis, and treatment—but with this innovation comes an expanded attack surface and growing security concerns. Traditional security mechanisms often fall short in such dynamic environments, prompting researchers to explore intelligent and adaptive solutions.

This chapter provides an overview of AI, focusing on Machine Learning (ML) and Deep Learning (DL) techniques, and their growing relevance in cybersecurity. It then narrows its scope to the role of AI in enhancing Intrusion Detection Systems (IDS) tailored to IoMT environments. Finally, it reviews significant research efforts and related works that apply AI models to IDS in healthcare, establishing a foundation for the methods proposed in this study.

2.2 Overview of Artificial Intelligence

2.2.1 Introduction to Artificial Intelligence

Artificial intelligence often sparks curiosity and emotion, largely because of our deep interest in understanding intelligence itself—what it is, how it works, and what makes it uniquely human. While these questions are important, especially in fields like neuroscience and psychology, for computer scientists and engineers, the key focus is more practical: how to design machines that can mimic human-like intelligent behavior in real-world tasks, so what is AI ?[19]

2.2.2 Defining the Digital Mind: Understanding Artificial Intelligence

Artificial Intelligence (AI) is a scientific field concerned with developing systems capable of simulating human cognitive abilities such as reasoning, learning, and decision-making [20]. The field began in the mid-20th century, when Alan Turing laid the theoretical foundations with the famous Turing Test for machine intelligence [21]. Since then, AI has undergone significant advancements, especially with the emergence of deep learning and big data technologies [22].

In the digital era, AI has become a fundamental pillar for enhancing system performance across various domains including healthcare, security, and industry [23]. It also plays a critical role in developing intelligent networked systems that rely on data analysis and automated decision-making to improve efficiency [24]. These capabilities are largely powered by subfields such as Machine Learning (ML) and Deep Learning (DL), which provide the computational models and algorithms that enable modern AI systems to learn from data and adapt over time.

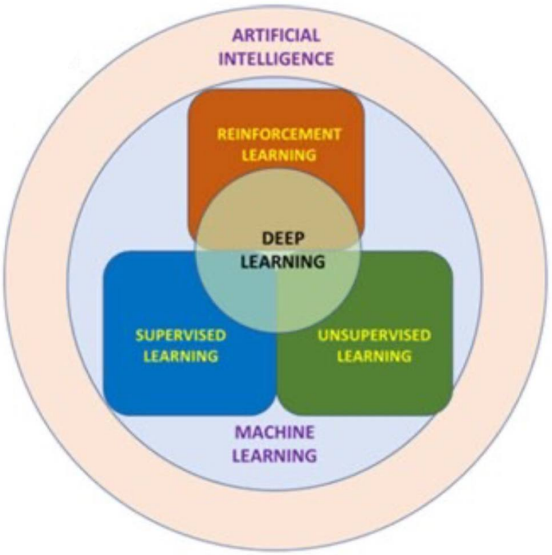


Figure 2.1: Schematic representation of relation between AI, ML, and DL. [6]

2.3 Applications of Artificial Intelligence (AI)

AI finds extensive applications across various sectors including E-commerce, Education, Robotics, Healthcare, and Social Media. We highlight some of them providing examples for each.[25]

- **Artificial Intelligence in Robotics:**Artificial Intelligence has become a driving force in advancing robotics, allowing machines to perform tasks with greater autonomy and precision. In robotics, AI enables real-time decision-making, boosting both efficiency and productivity.

  For example, in autonomous navigation, AI helps robots like drones and self-driving cars understand and react to their surroundings using sensors and computer vision. In healthcare, AI-powered robots assist with surgeries and patient care, adding accuracy and support to critical medical procedures. In industrial settings, AI enhances automation by allowing robots to learn from experience, improve quality control, and streamline repetitive tasks.[25]

- **Artificial Intelligence in Education:**AI is reshaping the education landscape by enhancing how both students and teachers interact with content. Voice assistants make information more accessible and allow for on-demand help. AI tools also support engagement and accessibility, using interactive technologies and personalized content to cater to diverse learning needs, including students with disabilities. With smart content creation, AI can design tailored learning materials—like videos or infographics—and even incorporate AR/VR to create immersive educational experiences. This integration not only improves understanding but also makes learning more dynamic and inclusive.[25]

- **Artificial Intelligence in Healthcare:**AI is transforming healthcare by enabling faster, more accurate diagnoses and improving patient care. In medical imaging, AI can detect diseases such as cancer or heart conditions by analyzing scans like MRIs or X-rays, often spotting patterns that might be missed by human eyes. Predictive analytics uses patient data to anticipate health risks like diabetes or heart attacks, allowing for earlier intervention. Additionally, virtual health assistants—like AI-powered chatbots—offer around-the-clock support, helping with scheduling, answering questions, and monitoring chronic conditions, ultimately easing the workload on healthcare professionals.[25]

2.4 Types of Artificial Intelligence

Artificial Intelligence (AI) can be categorized based on different criteria such as intelligence level and methodology.

2.4.1 Based on Intelligence Level

- **Narrow AI (Weak AI):** These systems are designed to perform a specific task, such as image recognition or language translation. They operate under a limited set of constraints and do not possess general intelligence [20].
- **General AI (Strong AI):** This type of AI aims to perform any intellectual task that a human can do. It remains largely theoretical and has not yet been realized [26].
- **Super AI (Artificial Superintelligence):** This refers to AI that surpasses human intelligence in all aspects, including creativity, problem-solving, and emotional intelligence. It is a hypothetical future development [27].

2.4.2 Based on Methodology and Technique

- **Symbolic AI:** Relies on explicit rules and logic to represent knowledge and reasoning [28].
- **Statistical AI:** Uses mathematical models and statistical methods to make inferences and predictions [29].
- **Learning-Based AI:** Encompasses machine learning and deep learning techniques, where the system learns patterns from data [22].

2.5 Machine Learning (ML)

2.5.1 Definition of Machine Learning

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that enables computers to learn from data and improve their performance without explicit programming [30]. According to Mitchell, ML is defined as a system's ability to improve at a task T based on experience E measured by a performance metric P [30].

Unlike traditional programming, ML systems learn patterns from data to make decisions or predictions [31]. It underpins many modern AI applications such as image recognition and speech processing [32, 33].

2.5.2 How Machine Learning Works

Machine Learning algorithms learn patterns from data to make predictions or decisions. The learning process typically involves feeding a model with input data and corresponding outputs (labels), enabling the model to adjust its internal parameters to minimize errors [33]. This adjustment is commonly achieved through optimization techniques such as gradient descent [31].

The choice of the learning paradigm depends on the nature of the problem and the available data.

2.5.3 Types of Machine Learning

Machine Learning can be broadly categorized into three main types: supervised learning, unsupervised learning, and reinforcement learning [31, 34].

- **Supervised Learning**: In this type, the model is trained on a labeled dataset, where each input is paired with the correct output. The goal is to learn a mapping from inputs to outputs to predict labels on new, unseen data. Common tasks include classification and regression [30].

- **Regression** involves predicting continuous numerical values. Examples include forecasting house prices based on features such as size and location, or predicting temperature over time.
- **Classification** refers to predicting discrete categories or classes. For example, an email can be classified as spam or not spam, or an image can be classified as containing a cat or a dog.

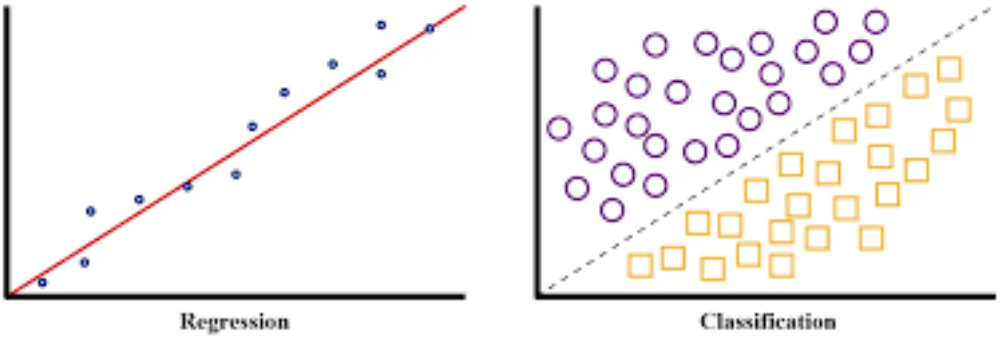


Figure 2.2: regression VS classification

- **Unsupervised Learning**: Here, the data is unlabeled, and the model attempts to find hidden patterns or groupings within the data. Techniques such as clustering and dimensionality reduction are typical examples [31].
- **Reinforcement Learning**: This paradigm involves an agent learning to make decisions by interacting with an environment and receiving feedback in the form of rewards or penalties. It is widely used in areas like robotics and game playing [34].

Each type addresses different problem domains and requires different data availability and learning strategies.

2.5.4 Popular Machine Learning Models

Machine learning encompasses a variety of models that differ in complexity, assumptions, and applications [30, 31]. Some of the most widely used models include:

- **Linear Regression**: A simple yet powerful model used for regression tasks that assumes a linear relationship between input features and the target variable.
- **Logistic Regression**: Despite its name, it is used for classification problems, modeling the probability of class membership using the logistic function.
- **Decision Trees**: These models partition the input space into regions based on feature values, enabling both classification and regression tasks.
- **Support Vector Machines (SVM)**: SVMs find the hyperplane that best separates classes by maximizing the margin between them, effective especially in high-dimensional spaces.
- **K-Nearest Neighbors (KNN)**: A non-parametric method that classifies a data point based on the majority class among its k closest neighbors in the feature space.
- **Naive Bayes**: A probabilistic classifier based on applying Bayes' theorem with strong independence assumptions between features.

- **Ensemble Methods**: Techniques such as Random Forests and Gradient Boosting combine multiple models to improve predictive performance and reduce overfitting.

These models form the foundation of many machine learning applications, with each suited to different types of data and problem requirements [31].

2.5.5 Applications of Machine Learning

Machine learning has become a cornerstone in numerous domains due to its ability to learn patterns from data and make intelligent predictions or decisions without being explicitly programmed [30, 31]. Its applications span across a wide range of fields, including but not limited to:

- **Healthcare**: ML models are used for disease diagnosis, patient risk assessment, and personalized treatment recommendations. For instance, supervised learning algorithms can detect anomalies in medical imaging or predict the likelihood of disease based on patient data.
- **Finance**: Applications include credit scoring, fraud detection, algorithmic trading, and customer segmentation. Machine learning enables real-time analysis of financial transactions to uncover fraudulent activities.
- **Natural Language Processing (NLP)**: ML techniques power tasks such as language translation, sentiment analysis, and chatbots. Models like logistic regression and deep learning architectures are employed for text classification and speech recognition.
- **Computer Vision**: ML plays a crucial role in image classification, object detection, and facial recognition. Convolutional Neural Networks (CNNs) are particularly effective in these tasks [32].
- **Autonomous Systems**: In robotics and autonomous vehicles, reinforcement learning helps agents learn optimal behaviors through interactions with the environment [34].
- **Recommender Systems**: Widely used by platforms like Netflix and Amazon, ML algorithms suggest content or products to users based on their behavior and preferences.

These diverse applications demonstrate the transformative potential of machine learning in solving real-world problems and automating decision-making processes across industries.

2.6 Deep Learning (DL)

2.6.1 Definition of Deep Learning

Deep Learning (DL) is a specialized subfield of Machine Learning that focuses on the use of artificial neural networks with multiple layers (hence the term ``deep'') to model complex patterns in data [33]. Inspired by the structure and function of the human brain, deep learning models are composed of interconnected layers of artificial neurons, each learning increasingly abstract representations of the input.

Unlike traditional machine learning algorithms that often rely on handcrafted features, deep learning methods are capable of automatic feature extraction directly from raw data, making them highly effective for tasks involving unstructured data such as images, audio, and natural language [32]. This capacity to learn hierarchical feature representations enables deep networks to generalize well to a variety of domains.

Overall, deep learning has significantly advanced the field of artificial intelligence by enabling machines to achieve or surpass human-level performance in various complex tasks.

2.6.2 Difference between Machine Learning and Deep Learning

The following table summarizes the key differences between Machine Learning (ML) and Deep Learning (DL)[35] based on their characteristics, performance, and requirements:

Table 2.1: Comparison between Machine Learning and Deep Learning

Aspect	**Machine Learning**	**Deep Learning**
Feature Processing	Relies heavily on manual feature engineering before model training.	Automatically extracts features from raw data without much human intervention.
Model Structure	Includes simpler algorithms such as decision trees, logistic regression, SVM, random forests, etc.	Based on deep neural networks with many hidden layers.
Data Requirements	Performs well with small to medium-sized datasets.	Requires large amounts of data for good performance due to model complexity.
Computational Power	Less computationally demanding, can run on less powerful devices.	Requires high computational resources like GPUs because of complex models.
Accuracy and Performance	Good for many tasks but can be limited on complex, high-dimensional data.	Excels at complex, nonlinear data and usually achieves higher accuracy, especially in computer vision and speech tasks.
Interpretability	More interpretable and explainable, decisions can be traced.	Often considered a ``black box" with difficult-to-interpret decision processes.

2.6.3 Deep Learning Architectures

Deep learning encompasses several architectures designed to tackle different types of data and tasks. The most prominent architectures include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory networks (LSTMs) [33, 36, 37].

- **Convolutional Neural Networks (CNNs)** are primarily designed for processing grid-like data such as images. CNNs use convolutional layers to automatically and efficiently extract spatial hierarchies of features by applying filters over the input [37]. They have been highly successful in computer vision tasks like image classification and object detection.
- **Recurrent Neural Networks (RNNs)** are specialized for sequential data, such as time series or natural language. RNNs maintain a hidden state that captures information from previous inputs, enabling them to model temporal dependencies [33]. However, standard RNNs suffer from difficulties in learning long-term dependencies due to vanishing or exploding gradients.
- **Long Short-Term Memory networks (LSTMs)** are a type of RNN architecture designed to overcome these limitations. Introduced by Hochreiter and Schmidhuber [36], LSTMs use gated cells to regulate the flow of information, allowing the network to retain relevant information over long sequences effectively. This makes LSTMs widely used in applications such as speech recognition, language modeling, and machine translation.

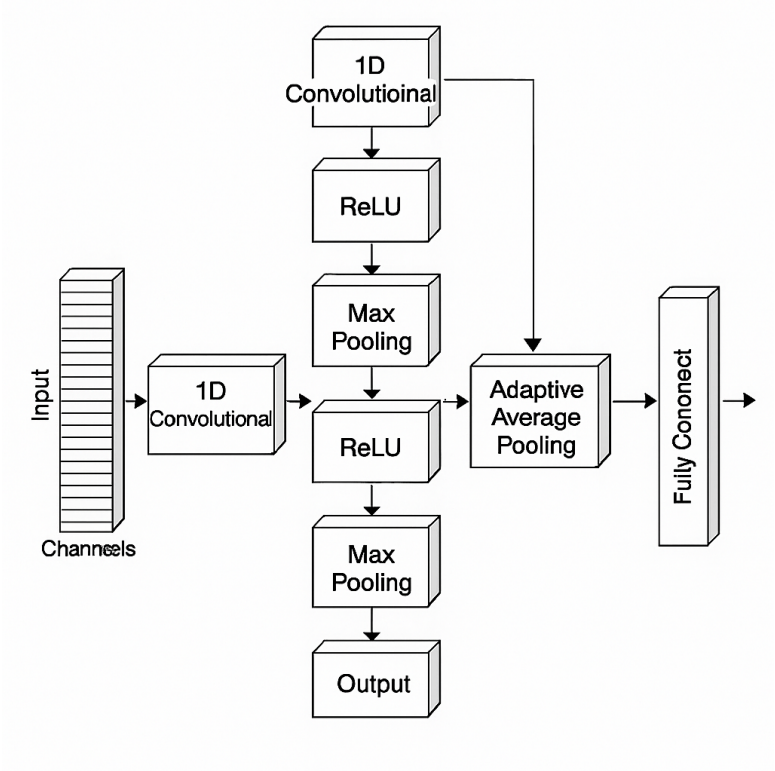


Figure 2.3: CNN Model

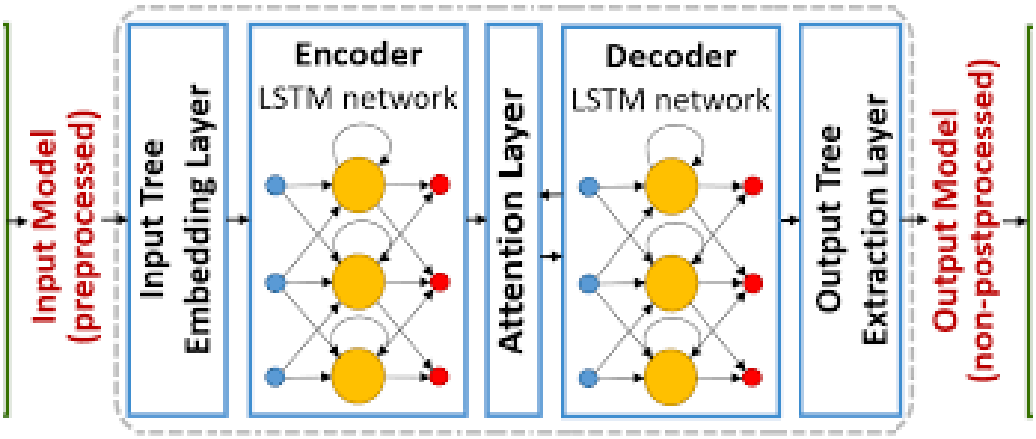


Figure 2.4: LSTM Model

2.7 Evaluation Metrics for ML/DL

This section shows the evaluation metrics used to evaluate the performance of the models. We employed different evaluation metrics of precision, accuracy, F1-score, recall, receiver operator characteristic (ROC), true positive rate, and false positive rate to assess the effectiveness of the suggested IDS as defined below based on the parameters of false positives (TP), true positives (TP), false negatives (FN), and true negatives (TN).[38]

1. **Accuracy:** This metric measures the model's capacity to accurately classify benign instances as displayed in Equation (2.1) below[38]:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

2. **Recall:** Also known as the detection rate or sensitivity, this metric measures the model's ability to recognize attacks as displayed in Equation (2.2) below[38]:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.2}$$

3. **Precision:**This metric refers to the model's capacity to produce accurate predic- tions, specifically, the number of correctly detected positive predictions (attacks), as displayed in Equation (2.3) below[38]:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.3}$$

4. **F1-score:** This metric effectively addresses the trade-off between recall and preci- sion by balancing them over all instances. This is demonstrated in Equation (2.4) below[38]:

$$\text{F1-score} = \frac{2 * Recall * Precision}{Recall + Precision} \tag{2.4}$$

2.8 Empowering IoMT Security: The Vital Role of ML/DL in Intrusion Detection

Machine Learning (ML) and Deep Learning (DL) play a key role in securing IoMT by enabling smart, real-time intrusion detection. This section highlights how these AI techniques enhance threat detection, reduce false alarms, and support resource-limited medical devices.[39][38]

1. **Automated Threat Detection:**ML/DL models learn from patterns in network or device behavior to automatically detect anomalies or intrusions, without requiring manually defined rules. This is essential in IoMT, where real-time monitoring is crucial to protect patient data and device integrity.

2. **Real-Time and Adaptive Response:**IoMT systems generate vast amounts of real-time data. DL models (like LSTM or CNN) are capable of handling streaming data and evolving attack patterns, ensuring that the IDS adapts to new and unknown threats (zero-day attacks).

3. **High Detection Accuracy:**ML/DL-based IDS have demonstrated higher accuracy in distinguishing between benign and malicious activity compared to traditional systems, especially in detecting subtle or low-frequency attacks that might go unnoticed otherwise.

4. **Reduced False Positives:**With proper training and feature engineering, ML/DL models can reduce false alarms, which is crucial in a medical context where unnecessary disruptions can delay care or impact patient safety.

5. **Efficient Processing of Complex IoMT Data:**DL architectures are especially useful for handling heterogeneous and high-dimensional data from diverse IoMT devices (wearables, implantables, etc.), enabling comprehensive security analysis without oversimplification.

6. **Scalability:**As IoMT ecosystems expand, ML/DL approaches offer scalable solutions that can be deployed across large, distributed networks with minimal human intervention.

7. **Support for Resource-Constrained Devices:**Lightweight ML models can be optimized for edge computing, enabling on-device intrusion detection even on low-power IoMT devices—ensuring that security is not sacrificed due to hardware limitations.

2.9 Barriers to Intelligence: ML and DL Challenges in IoMT-based IDS

Integrating ML and DL into IDS for IoMT brings promising benefits, but also presents key challenges. Issues like limited data, constrained device resources, and the need for real-time precision make implementation complex. This section outlines the main hurdles faced when applying these technologies in healthcare environments.[40][41][42]

1. **Data Scarcity and Quality**

   - **Limited Labeled Data:** IoMT-specific datasets are often scarce or not publicly available, making it difficult to train accurate models.
   - **Imbalanced Data:** Attack data is usually rare compared to normal traffic, leading to bias in model predictions.
   - **Privacy Concerns:** Patient data is highly sensitive, limiting data sharing and hindering model development.

2. **High False Positives**ML/DL models can produce high false-positive rates, overwhelming healthcare professionals with alerts and reducing trust in the system.

3. **Resource Constraints**IoMT devices often have limited computational power, memory, and battery life, making it challenging to deploy complex DL models at the edge.

4. **Real-Time Detection Requirements**IDS in IoMT must respond to threats in real-time to avoid putting patients at risk. Achieving this speed while maintaining accuracy is a significant challenge.

5. **Adaptability to New Threats**ML/DL models often require retraining to handle new or evolving attacks, which is not always practical in IoMT environments. Zero-day attacks are especially difficult to detect without constant model updates.

6. **Interpretability and Explainability**Deep learning models are often ``black boxes,'' making it hard to understand why a prediction was made. In healthcare, explainability is crucial to support decision-making and compliance with regulations.

7. **Regulatory and Compliance Challenges**The use of ML/DL with patient data must comply with regulations such as HIPAA or GDPR. Ensuring privacy, fairness, and auditability of the models is essential.

8. **Integration and Interoperability** Integrating ML/DL-based IDS into diverse and heterogeneous IoMT ecosystems can be complex. Compatibility with various devices, networks, and data formats remains a challenge.

2.10 Related Works

In healthcare and medical systems, security measures must be implemented with exceptional care and rigor due to the critical nature of medical data and the growing frequency of cyberattacks targeting healthcare infrastructures. Conse- quently, ensuring the security of the IoMT has become a significant research focus, motivated by both the sensitivity of medical information and the increasing threat landscape. Intrusion detection systems (IDS) play a crucial role in safeguarding sensitive medical data within IoMT environ- ments by identifying malicious activities and unauthorized access.

Various studies have explored the application of deep learning (DL) and machine learning (ML) methods to bolster IDS effectiveness in IoMT networks.

For example, Anitha et al [43] explored some ML techniques to detect attacks, concluding that kNN yielded the best results with an accuracy of 89.79% . Their work utilized IEEE Data Port datasets for binary classification purposes.

In another study, Ksibi et al [44] employed the novel ECU- IoHT dataset. Their binary classification efforts, leveraging the Random Forest ML algorithm, achieved an accuracy of 99.76% after using SMOTE for data balancing .

Tanzila Saba [45] advocated for ensemble classifiers, such as bagged decision trees based on the bagging algorithm, to detect attacks against Smart City Hospitals. Her model achieved 93.2% accuracy using the KDDCup'99 dataset, which encompasses 5 classes.

Alsalman [46] proposed FusionNet, a model combining Support Vector Machine, K-Nearest Neighbors, Random Forest, and Multi-Layer Perceptron for anomaly detection. According to his paper, this model reached 98.5% accuracy on the WUSTL EHMS 2020 Dataset and 99.5% on ICU-IoMT for binary classification .

Sun et al[38] attained a 98.5% accuracy by using Particle Swarm Optimization and AdaBoost on the NSL-KDD dataset, which includes 5 classes, including normal traffic .

Balhareth et al [47] worked with the CICIDS2017 dataset, applying Mutual Information and XGBoost for feature selection, resulting in a binary classification accuracy of 98.79% . Deep learning, known for its ability to autonomously uncover complex patterns, has also been widely used in IDS for IoMT.

Awotunde et al [48] used a Deep Auto Encoder on the NF-ToN-IoT dataset as an intrusion detection mechanism for secured IoMT systems, achieving 89% accuracy for a 10-class multi-classification .

Kulshrestha and Kumar [49], in their study using the ToN–IoT dataset, claimed around 99% accuracy for 4-class classification with the AdaBoost classifier .

Khan et al [50] proposed a hybrid CNN-LSTM model to address feature interdependencies and improve feature learning, which they applied to the IoT Malware dataset, achieving an approximate 99% accuracy for binary classification . Combining more classifiers often uses additional system resources, yet hybrid AI techniques have demonstrated effi- cacy in enhancing IDS performance.

Liaqat et al [51] proposed a hybrid DL architecture combining CNN and cuDNNLSTM for the IoMT environment employing the Bot–IoT dataset. They compared multiple configurations, concluding that CNN with cuDNNLSTM was the most effective, achieving an accuracy close to 99.99% for 3-class classification .

Faruqui et al [52] devised a model combining CNN and LSTM to improve cybersecurity in IoMT, reporting an average accuracy rate of 97.63% for 12 classes on the CIC-IDS2017 dataset, although it is not specific to IoMT .

Otoum et al [53] employed a Federated Transfer Learning- based IDS, and at most with achieving 95.1% accuracy rate for 3-classes multi-classification on the CICIDS2017- Tuesday dataset .

Ravi et al's [54] CNN-LSTM model on the WUSTL EHMS 2020 dataset achieved 99% accuracy with 10-fold cross-validation in binary classification .

Khan et al [55] developed the XSRU-IoMT model based on a bidirectional simple recurrent unit (Bid-SRU), achieving a 99.38% accuracy for 8-class multi-classification on the ToN- IoT dataset .

Dadkhah et al [56] generated the CICIoMT2024 dataset to simulate an IoMT environment for IDS research. They tested Logistic Regression, AdaBoost, DNN, and Random Forest, achieving nearly 100% accuracy in binary classification but observing a decrease in accuracy to 73.3% in 19-class scenario .

Sánchez et al [57] studied the application of fine-tuning Transformer designs also using the CICIoMT2024 dataset and assessed it using the Aposemat IoT-23 dataset. Their experiments on their proposed model reached up to 96% for Accuracy, Precision, Recall and F1- Score metrics .

This overview highlights ongoing research into IoMT security and the intersection of advanced AI techniques and IDS frameworks continues to show promising advancements in the field. By reviewing the related works on IoMT, it is shown that the studies are continuing and this special part of IoT still attracts researchers to solve different research and security problems.

2.11 Conclusion

In this chapter, we explored the foundational concepts of Artificial Intelligence, focusing on Machine Learning (ML) and Deep Learning (DL) and their growing relevance in securing Internet of Medical Things (IoMT) environments. We reviewed various ML/DL techniques used in IDS solutions, discussed standard evaluation metrics like accuracy, precision, recall, and F1-score, and highlighted the core challenges faced when applying these methods in resource-constrained and privacy-sensitive healthcare settings.

Furthermore, we examined related works that demonstrate how ML/DL models are being adapted to meet the unique demands of IoMT-based intrusion detection. While promising, these approaches face notable limitations, emphasizing the need for more robust, scalable, and privacy-preserving solutions. Building on this foundation, the next chapter will introduce a novel direction—Blockchain-based Federated Learning—for enhancing IDS in IoMT, aiming to address data privacy, distributed training, and trust issues in modern healthcare networks.

Chapter 3

Blockchain-based Federated Learning for Intrusion Detection in IoMT

3.1 Introduction

This chapter focuses on reviewing and analyzing three pivotal technologies that play a fundamental role in the development of intelligent intrusion detection systems: Centralized Learning, Federated Learning, and Blockchain. It begins by presenting the traditional approach based on centralized data processing, highlighting its advantages and limitations. It then transitions to Federated Learning as an alternative paradigm that enhances data privacy and enables collaboration between multiple entities without the need to share raw data. Finally, the chapter explores Blockchain technology, which has emerged as an effective means to strengthen the security and transparency of artificial intelligence systems, particularly when integrated with Federated Learning to create decentralized and secure learning environments.

3.2 Centralized Learning

In centralized learning, data generated by IoMT devices—such as wearable sensors and patient monitoring systems—are transmitted to a central server or cloud for training machine learning models. This approach leverages large-scale datasets and powerful computing resources, which enhances the effectiveness of intrusion detection systems (IDS) in identifying malicious activities . However, centralized learning in IoMT environments poses significant challenges due to the sensitivity of medical data and the need for low latency. Transmitting such data to the cloud increases the risk of privacy breaches, unauthorized access, and non-compliance with healthcare regulations such as HIPAA . Therefore, despite its performance benefits, centralized learning is often unsuitable for IoMT-based IDS, encouraging the adoption of privacy-preserving and decentralized alternatives.

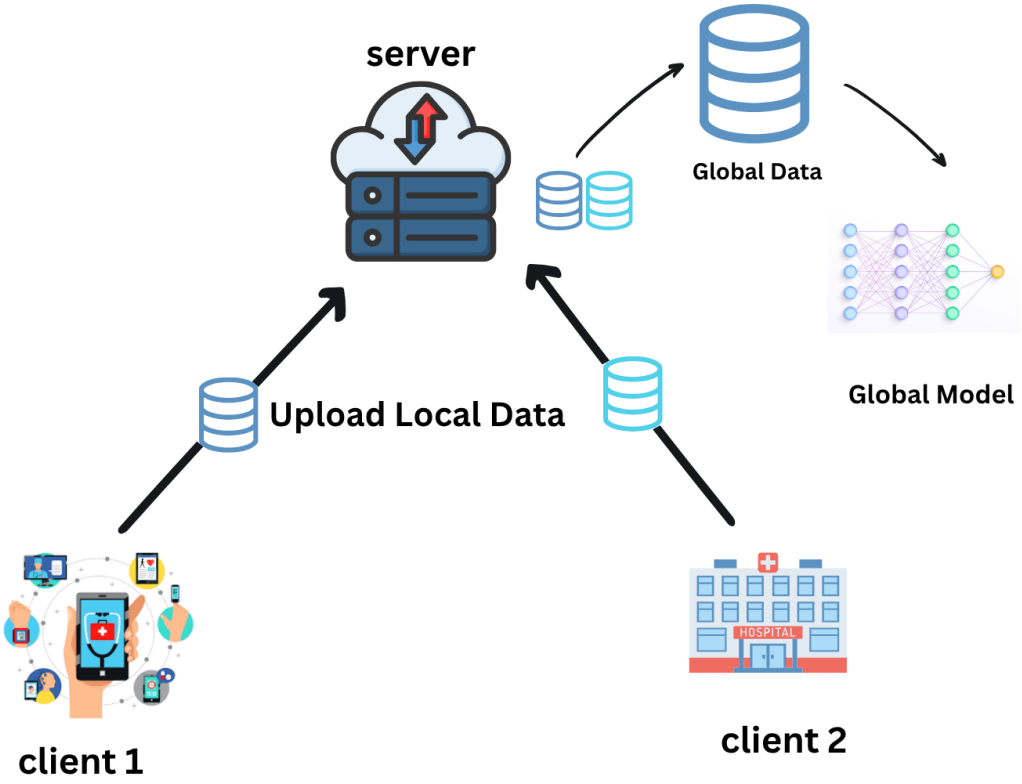


Figure 3.1: Centralized Learning

3.3 Federated Learning

3.3.1 Definition of Federated Learning

Federated Learning (FL) is a decentralized machine learning paradigm that enables multiple devices or entities to collaboratively train a shared model while keeping their raw data locally stored. This approach was formally introduced by [58] as a solution to privacy-preserving distributed learning.

3.3.2 Mechanism of Federated Learning

Federated Learning (FL) operates through a decentralized training process that enables multiple clients (e.g., edge devices or local institutions) to collaboratively train a shared machine learning model while keeping their data localized. This mechanism ensures data privacy and compliance with regulatory constraints by avoiding direct data exchange.

The foundational workflow of FL can be summarized in four main steps:

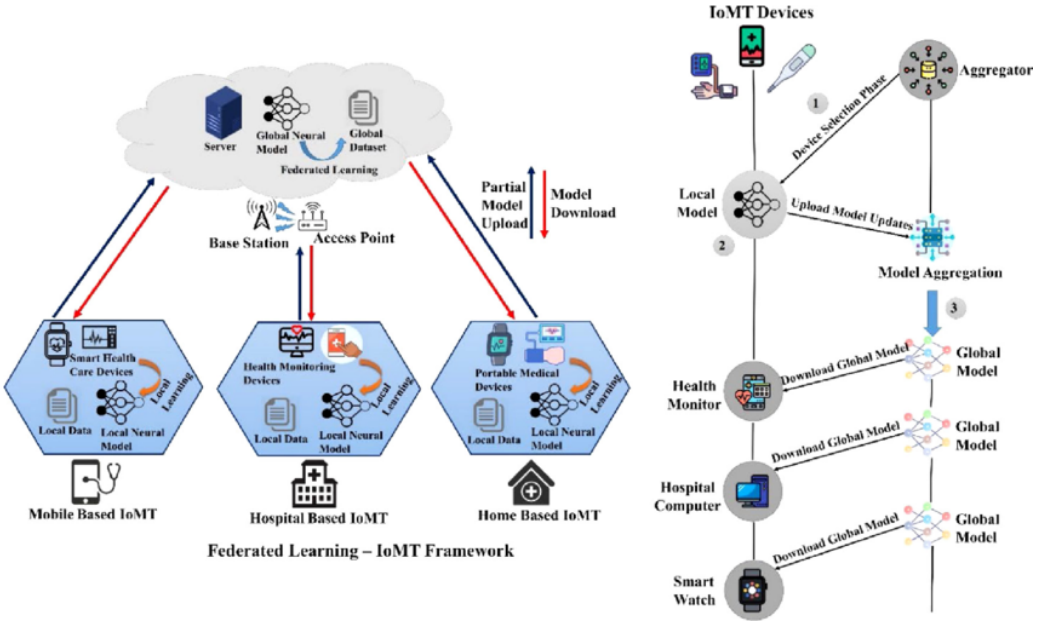


Figure 3.2: Federated Learning

1. **Initialization**: A central server initializes a global model and broadcasts it to selected clients.
2. **Local Training**: Each selected client trains the received model using its local dataset for a few epochs, typically by applying stochastic gradient descent (SGD) or its variants.
3. **Model Upload**: After local training, clients send their model updates (e.g., gradients or weights) to the central server rather than their raw data.
4. **Aggregation**: The server aggregates the collected updates to form an improved global model, commonly using the Federated Averaging (FedAvg) algorithm proposed which performs a weighted average of the client updates based on the size of their local datasets.

This iterative process continues for multiple rounds until convergence.

3.3.3 Characteristics of Federated Learning

Federated Learning (FL) possesses several unique characteristics that distinguish it from traditional centralized learning paradigms. According to [59], the key characteristics of FL include the following:

- **Decentralized Data**: FL operates under a decentralized data setting, where training data remains distributed across multiple clients (e.g., mobile devices, institutions) and is not collected or stored centrally.
- **Privacy Preservation**: Since raw data never leaves the client devices, FL inherently provides privacy protection and supports compliance with data protection regulations such as GDPR and HIPAA.
- **Heterogeneity**: FL faces significant challenges due to system heterogeneity (differences in hardware, network connectivity, and power constraints across clients) and statistical heterogeneity (non-IID and unbalanced data distributions across clients).
- **Communication Efficiency**: As communication between clients and server is a major bottleneck, FL emphasizes the need for efficient communication strategies, including model compression, update sparsification, and fewer communication rounds.
- **Scalability**: FL systems are designed to support training across a large number of clients, sometimes involving thousands or millions of devices, which necessitates scalable algorithms and infrastructure.

3.3.4 Comparison between Centralized and Federated Learning

The following table summarizes the key differences between Centralized Learning (CL) and Federated Learning (FL). [60]. The comparison highlights important aspects such as data privacy, resource consumption, training efficiency, and more.

Table 3.1: Comparison between Centralized Learning and Federated Learning

Property	**Centralized Learning (CL)**	**Federated Learning (FL)**
Data Storage Location	Data aggregated on a central server	Data remains local on devices
Data Privacy	Low privacy due to data transfer	Higher privacy as data never leaves devices
Bandwidth Consumption	High due to data transfer	Lower, only model updates sent
Data Distribution	Data is centralized	Non-IID distribution affects performance
Training Efficiency	Faster and more stable	Can be slower due to distribution and communication
System Flexibility	Less flexible, depends on server	More flexible with many clients
Security Risks	Server vulnerable to attacks	Risks from update manipulation and malicious clients
System Complexity	Less complex to manage	More complex due to coordination
Typical Applications	Traditional ML systems	Distributed applications like healthcare

3.3.5 Types of Federated Learning

Federated Learning (FL) can be categorized based on the type of participating entities (devices or organizations) or the structure of the data distributed across them. Two common taxonomies are described below.

1. **Device-Based Categorization**

   - **Cross-Device Federated Learning:** This type involves a large number of unreliable and resource-constrained devices, such as smartphones, tablets, or IoT devices, each with a small amount of local data. The key challenge here lies in managing scalability, communication efficiency, and device availability kairouz2019advances.

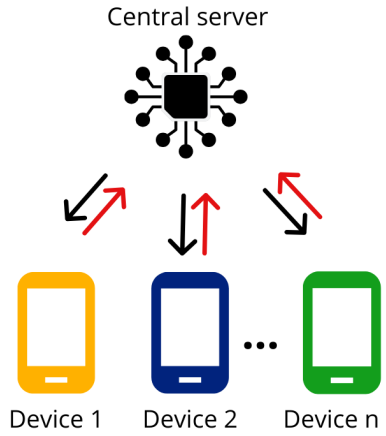


Figure 3.3: Cross-Device Federated Learning

   - **Cross-Silo Federated Learning:** In this scenario, the participating entities are relatively few, but more stable and reliable, such as organizations (e.g., hospitals, banks, or universities). These entities often have larger datasets and more computational power. Cross-silo FL emphasizes data security and regulatory compliance liu2021federated.

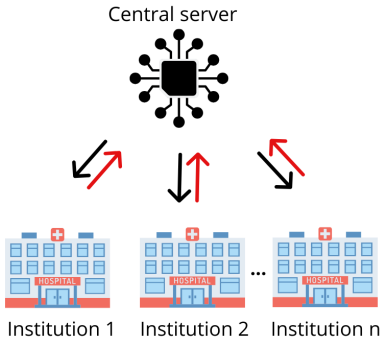


Figure 3.4: Cross-Silo Federated Learning

2. **Data-Based Categorization**

   - **Horizontal Federated Learning (HFL):** This applies when data across different clients share the same feature space but differ in the sample space (i.e., different users or samples). For instance, two banks in different regions may have similar customer information fields but serve different individuals yang2019federated.
   - **Vertical Federated Learning (VFL):** This setting involves participants that share the same sample space (i.e., they have data on the same users), but their datasets have different feature spaces. An

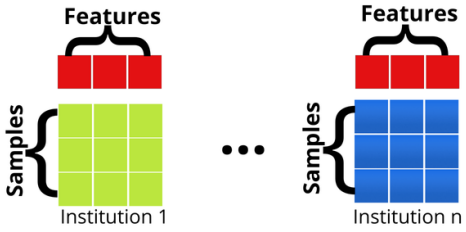


Figure 3.5: Horizontal Federated Learning

example is a bank and an e-commerce company that both have information on the same users but from different perspectives (financial vs. purchasing behavior) yang2019federated.

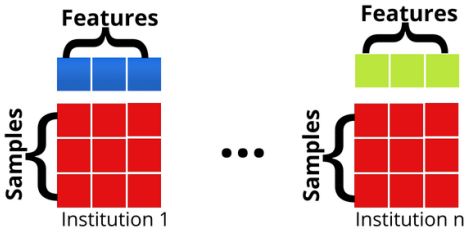


Figure 3.6: Vertical Federated Learning

- **Federated Transfer Learning (FTL):** When both the sample space and the feature space overlap only partially, FTL is employed. It leverages transfer learning techniques to align and share knowledge across domains with limited overlap. FTL is particularly useful in cross-domain collaborative learning scenarios yang2019federated.

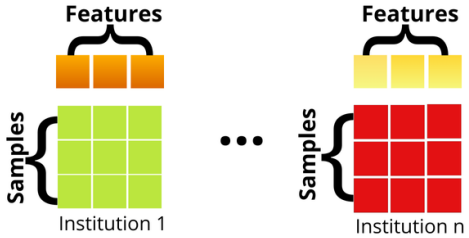


Figure 3.7: Federated Transfer Learning

3.3.6 Federated Learning Strategies

► **Federated Averaging (FedAvg)**

The first and most widely used strategy, introduced by [58]:

- **Mechanism**:
  1. Server sends the current model to selected devices.
  2. Each device trains locally for E epochs:

$$W_k^{t+1} = W_k^t - \eta \nabla \mathcal{L}(W_k^t; \mathcal{D}_k) \tag{3.1}$$

3. Server computes weighted average of updates:

$$W_{global}^{t+1} = \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} W_k^{t+1} \tag{3.2}$$

- **Advantages**:
  - Communication-efficient (reduces server rounds).
  - Suitable for resource-constrained devices.

► **SCAFFOLD**

Proposed by [61] to address **client drift** in Non-IID data:

- **Improvement**: Uses control variates for correction:

$$W_k^{t+1} = W_k^t - \eta \left(\nabla \mathcal{L}(W_k^t; \mathcal{D}_k) + c_k - c_{global} \right) \tag{3.3}$$

  where c_k and c_{global} are local/global control variables.
- **Performance**:
  - Reduces client drift by up to 30% vs. FedAvg.
  - Requires slightly more communication.

► **FedProx**

Introduced by [59] for heterogeneous environments:

- **Mechanism**: Adds a proximal term to the loss function:

$$\min_W \mathcal{L}(W; \mathcal{D}_k) + \frac{\mu}{2} \|W - W_{global}^t\|^2 \tag{3.4}$$

  where μ is a regularization parameter.
- **Best Use Cases**:
  - Asynchronous device participation.
  - Non-IID data distributions.

3.3.7 Applications of Federated Learning

Federated Learning (FL) has gained significant traction across various domains due to its ability to enable collaborative model training without sharing raw data. This privacy-preserving paradigm is particularly beneficial in scenarios where data are sensitive, distributed, or subject to legal and regulatory constraints.

- **Healthcare:** In the medical field, FL enables institutions such as hospitals and research centers to collaboratively train machine learning models on patient data without transferring it to a central server. This is crucial for maintaining patient privacy and complying with data protection regulations (e.g., GDPR, HIPAA). For instance, rieke2020future demonstrated how FL can be applied to medical imaging tasks across multiple institutions, improving diagnostic performance while preserving data locality.
- **Keyboard Prediction and Mobile Devices:** One of the earliest large-scale applications of FL was in personalized language modeling on mobile devices. hard2018federated implemented FL to train on-device models for next-word prediction and keyboard auto-correction without collecting users' typing data on central servers. This approach provided both privacy and personalization, paving the way for broader adoption of FL in edge computing.

- **Finance and Banking:** FL allows banks and financial institutions to collaboratively detect fraud or assess credit risk while preserving user confidentiality and complying with strict data sharing regulations.
- **Smart IoT and Edge Devices:** In the context of smart homes and industrial IoT, FL facilitates learning from distributed sensor data without central aggregation. This supports real-time decision-making while maintaining privacy and reducing latency.
- **Autonomous Vehicles:** FL can help manufacturers and service providers train models across fleets of autonomous vehicles by aggregating knowledge from each vehicle's local observations, enhancing driving performance and safety without exposing raw telemetry data.

Table 3.2: Key Properties of Federated Learning Frameworks

Framework	Developer	Open Source	Data Sharing	Integration
TFF	Google	Yes	H	T
PySyft	OpenMined	Yes	H & V	K, P, T
NVFlare	NVIDIA	Yes	H & V	P, T
FATE	WeBank	Yes	H & V	K, P, T
Flower	Adap GmbH	Yes	H	K, P, T
IBM FL	IBM	No	H	K, P, T
FedLab	FedLab	Yes	H	K, P, T
FedML	FedML Inc.	Yes	H	J, M, P, T
FLUTE	Microsoft	Yes	H	P, T
OpenFL	Intel	Yes	H	P, T

Notes: H = Horizontal, V = Vertical, T = TensorFlow, K = PyTorch, P = Python, J = JAX, M = MXNet.

3.3.8 Federated Learning for Intrusion Detection Systems in IoMT

Intrusion Detection Systems (IDS) are a critical component in securing Internet of Medical Things (IoMT) environments, where protecting sensitive health data and ensuring the operational integrity of connected medical devices is paramount. Traditionally, IDS models rely on centralized machine learning approaches, where data from distributed devices is collected and processed at a central server. However, this paradigm poses substantial challenges, particularly regarding data privacy, communication overhead, and reliance on continuous connectivity, making it unsuitable for many real-world IoMT deployments.

Federated Learning (FL) offers a novel solution to these challenges by enabling distributed training of intrusion detection models without sharing raw data. In FL, each device (or client) trains a local model using its private data and only transmits model updates (e.g., gradients or weights) to a coordinating server or peer nodes. This approach preserves data privacy while collaboratively improving a global IDS model.

Key advantages of applying FL in IoMT-based IDS include:

- **Enhanced privacy**: Raw data remains on-device, minimizing exposure to data breaches.
- **Reduced bandwidth consumption**: Only model updates are shared, not the data itself.
- **Local adaptability**: Devices can tailor models to specific local contexts or threats.
- **Handling of non-IID data**: FL naturally accommodates heterogeneous data distributions, common in medical networks.

3.3.9 Security Risks in Federated Learning

Despite the decentralized nature of Federated Learning (FL), the paradigm is still susceptible to a variety of security threats due to its distributed architecture, the presence of partially trusted participants, and the necessity of communication among nodes.

- **Model Poisoning Attacks:** Malicious clients may upload manipulated model updates to influence the global model's behavior, either degrading its performance or inserting backdoors.
- **Data Poisoning Attacks:** Adversaries inject corrupted or mislabeled data into their local datasets, thereby indirectly compromising the global model during aggregation.
- **Inference Attacks:** Attackers attempt to extract sensitive information about the training data from shared gradients or model parameters using gradient inversion or membership inference techniques.
- **Free-riding Attacks:** Some clients might participate in FL without contributing meaningful updates, aiming to benefit from the aggregated model without local training, harming the fairness of collaboration.
- **Communication Hijacking:** Without secure communication protocols, model updates transmitted over the network can be intercepted, modified, or spoofed by external attackers.
- **Server Compromise:** In server-based FL architectures, the central aggregator becomes a critical point of vulnerability. If compromised, the attacker can manipulate the aggregation process, inject malicious models, or extract sensitive information from client updates.

To mitigate these risks, a range of defense mechanisms has been proposed, including differential privacy, secure multiparty computation, homomorphic encryption, anomaly detection, and Byzantine-resilient aggregation algorithms. Robust FL system design requires balancing privacy, efficiency, and security considerations lyu2020threats.

3.4 Blockchain Technology

3.4.1 Types of Networks Based on Distribution

Network architecture plays a fundamental role in determining how information flows and how resilient a system is to failures or attacks. Based on distribution and control, there are three primary types of networks: **centralized**, **decentralized**, and **distributed** [62].

- **Centralized Network:** In this architecture, a single central node is responsible for managing and controlling the communication and operations of the entire system. All data passes through the central authority, which creates a single point of control—and potentially, a single point of failure. This model is simple but not fault-tolerant or scalable.
- **Decentralized Network:** Control and data processing are distributed among several nodes, each capable of making decisions and communicating with others. There is no complete reliance on a single authority, enhancing fault tolerance and reliability compared to centralized systems.
- **Distributed Network:** Every node in the system holds equal responsibility, and data is fully replicated across nodes. This structure is the most robust in terms of fault tolerance, transparency, and resistance to single-point failures. It is also the foundational design for blockchain systems.

Understanding these three models provides the groundwork for exploring how blockchain leverages distributed architectures to ensure trust, resilience, and decentralization.

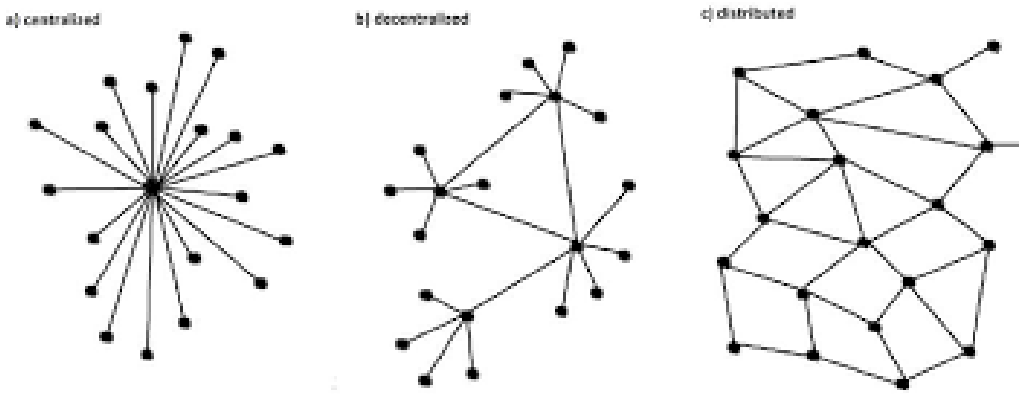


Figure 3.8: Centralized VS Decentralized VS Distributed

3.4.2 Definition of Blockchain

Blockchain is a decentralized, distributed digital ledger technology that records transactions across multiple nodes in a tamper-proof and transparent manner. Each transaction is grouped into a block, and these blocks are cryptographically linked in a chronological chain. Once data is recorded in a block and added to the chain, it becomes immutable and verifiable by all participants in the network [63].

Unlike traditional centralized systems where a single entity has control, blockchain operates in a peer-to-peer (P2P) architecture, eliminating the need for intermediaries. This makes it highly suitable for applications requiring trust, transparency, and data integrity among untrusted parties.

3.4.3 Key Features Of Blockchain :

- **Decentralization:** Data is not stored or controlled by a single entity; it is distributed across the network, reducing the risks of central points of failure.
- **Transparency:** All transactions recorded on the blockchain are visible to participants and can be independently verified.
- **Immutability:** Once a block is added to the chain, its data cannot be altered without modifying all subsequent blocks, which is computationally infeasible.
- **Security:** Blockchain uses cryptographic techniques such as hashing and digital signatures to ensure the integrity and authenticity of data.
- **Consensus Mechanisms:** Blockchain networks rely on protocols like Proof of Work (PoW), Proof of Stake (PoS), or others to agree on the state of the ledger.

These characteristics make blockchain a powerful infrastructure for enabling secure, auditable, and decentralized applications across various domains.

3.4.4 Types of Blockchain

Blockchain networks can be classified into four primary categories based on access and control mechanisms:

- **Public Blockchain:** Open and permissionless networks where any participant can join, validate transactions, and access the ledger. Notable examples include Bitcoin and Ethereum. [64, 65]

- **Private Blockchain:** Restricted and permissioned networks governed by a single entity. Participants must be granted access, making them suitable for enterprise settings. [64, 66]
- **Consortium Blockchain:** Semi-decentralized networks managed by a group of trusted organizations. Used in inter-organizational collaboration scenarios such as finance or healthcare. [65]
- **Hybrid Blockchain:** A combination of public and private features, allowing controlled access to some data while keeping other data publicly verifiable. [65]

Table provides a comparative overview of the three main types of blockchain networks: Public, Consortium, and Private blockchains. The comparison focuses on key properties such as consensus determination, read permissions, immutability, efficiency, centralization, and the nature of the consensus process. Each type demonstrates distinct trade-offs between transparency, control, and performance, which influence its suitability for various applications.[67]

Table 3.3: Comparisons among Public, Consortium, and Private Blockchains

Property	**Public Blockchain**	**Consortium Blockchain**	**Private Blockchain**
Consensus Determination	All miners	Selected set of nodes	One organization
Read Permission	Public	Could be public or restricted	Could be public or restricted
Immutability	Nearly impossible to tamper	Could be tampered	Could be tampered
Efficiency	Low	High	High
Centralized	No	Partial	Yes
Consensus Process	Permissionless	Permissioned	Permissioned

3.4.5 Block Structure

In blockchain technology, each **block** is a fundamental unit that stores a batch of transactions and connects to the previous block via a cryptographic hash, forming a secure and immutable chain [68]. A typical block is divided into two main components:

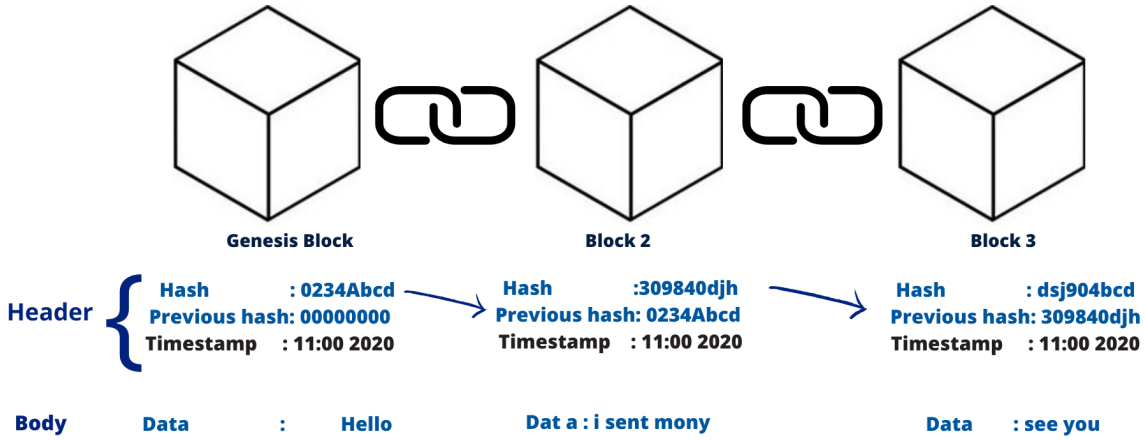


Figure 3.9: Block Structure

- **Block Header:** This part includes essential metadata such as:
  - **Previous Block Hash:** A cryptographic reference to the hash of the previous block, maintaining the chain's continuity.
  - **Timestamp:** The time when the block was created.
  - **Nonce:** A number used during mining to find a valid hash.
  - **Merkle Root:** The root hash of the Merkle tree built from all transactions in the block.
- **Block Body:** This part contains the list of transactions that are included in the block. These transactions are hashed and organized in a Merkle tree structure to ensure integrity and facilitate efficient verification.

3.4.6 Nodes and Miners

- **Nodes** are the participating devices in a blockchain network that store and validate copies of the distributed ledger. There are different types of nodes such as full nodes, light nodes, and mining nodes.
- **Miners** are specialized nodes that participate in the process of adding new blocks to the chain. They compete to solve complex mathematical puzzles in Proof-of-Work (PoW) systems, or are selected based on stake in Proof-of-Stake (PoS) systems [69]. The mining process ensures consensus and secures the network from tampering.

3.4.7 Consensus Mechanisms

Consensus mechanisms are protocols that ensure all nodes in a decentralized system agree on the state of the blockchain. Popular mechanisms include:

- **Proof of Work (PoW):** Nodes (miners) solve computational puzzles to validate blocks. Used by Bitcoin.

- **Proof of Stake (PoS):** Nodes are selected to validate blocks based on their stake in the network. Used by Ethereum 2.0.
- **Practical Byzantine Fault Tolerance (PBFT):** Suitable for permissioned blockchains.

These mechanisms ensure that the blockchain remains secure, synchronized, and tamper-resistant across all participating nodes.

3.4.8 Mechanism of Blockchain

The mechanism of blockchain revolves around maintaining a distributed and tamper-resistant ledger through a sequence of cryptographically linked blocks. Each transaction added to the ledger undergoes several stages before becoming immutable. The core steps of blockchain operation are as follows:

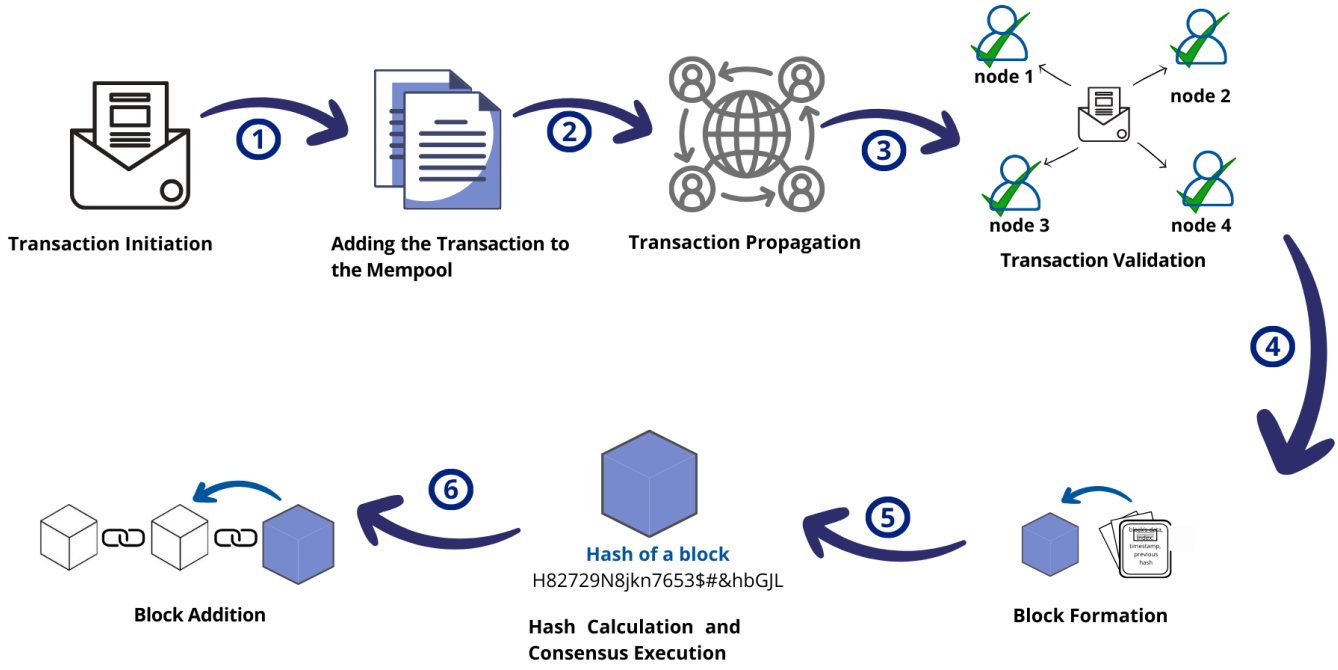


Figure 3.10: Mechanism of Blockchain

1. **Transaction Initiation:** A participant initiates a transaction by digitally signing it using their private key, ensuring authenticity and non-repudiation.
2. **Adding the Transaction to the Mempool**
3. **Transaction Propagation:** The signed transaction is broadcast to the peer-to-peer (P2P) network where nodes validate its structure and authenticity.
4. **Transaction Validation:** Nodes (also called validators or miners) validate transactions according to pre-defined consensus rules (e.g., checking for double spending).
5. **Block Formation:** Validated transactions are grouped into a new block. This block includes a reference (hash) to the previous block, forming a cryptographic chain.

6. **Consensus Execution:** The network applies a consensus mechanism (e.g., Proof of Work, Proof of Stake) to agree on the addition of the new block to the ledger.
7. **Block Addition:** Upon achieving consensus, the new block is appended to the blockchain. All nodes update their local copies of the ledger.
8. **Ledger Continuity:** The block is now part of the permanent record. Altering it would require modifying all subsequent blocks, making tampering practically infeasible.

This decentralized mechanism ensures security, transparency, and resilience without the need for centralized control.

3.4.9 Smart Contract

A smart contract is a software program stored on the blockchain network that is executed automatically when predefined conditions between contracting parties are met. The smart contract aims to execute, verify, or enforce the terms of agreements securely and transparently without the need for intermediaries. Since it is stored on the blockchain, smart contracts provide a high level of trust, as the operations are immutable and permanently recorded [70].

According to Mohanta et al. [70], smart contracts offer significant capabilities in automating contracts and improving efficiency across various domains such as **financial services**,**supply chains** , and **healthcare** by reducing human errors and operational costs.

3.4.10 Comparison between Bitcoin and Ethereum

Blockchain is the foundational technology that underlies various decentralized systems, including cryptocurrencies and smart contract platforms. It provides a distributed, immutable ledger that enables secure and transparent data exchange without intermediaries. Two of the most prominent implementations of blockchain technology are **Bitcoin** and **Ethereum**, which, although both built on the blockchain concept, differ significantly in their design goals, architecture, and use cases.

Table highlights the key differences between Bitcoin and Ethereum based on the study by Jani [71].

Table 3.4: Comparison between Bitcoin and Ethereum

Feature	**Bitcoin**	**Ethereum**
Purpose	Digital currency for peer-to-peer transactions	Platform for smart contracts and decentralized applications (DApps)
Launch Year	2009	2015
Scripting Language	Limited, non-Turing complete	Solidity (Turing complete)
Block Time	10 minutes	12–15 seconds
Consensus Mechanism	Proof of Work (PoW)	Initially PoW; transitioning to Proof of Stake (PoS)
Transaction Fees	Based on transaction size	Based on computational cost (Gas)
Flexibility	Primarily monetary transactions	Highly flexible for various decentralized applications

3.4.11 Applications of Blockchain

Blockchain technology has evolved beyond cryptocurrency and is now being adopted across a wide range of industries due to its decentralization, immutability, and transparency. This section highlights key application areas.

- **Financial Services and Banking**
  Blockchain enables secure, real-time, peer-to-peer financial transactions without the need for intermediaries. It is applied in cross-border payments, smart contracts, and decentralized finance (DeFi) platforms [68].

- **Supply Chain Management**
  Blockchain ensures product traceability and transparency throughout the supply chain. It helps combat fraud and counterfeit products, as implemented by IBM Food Trust in tracking food provenance [**?**].

- **Healthcare**
  In healthcare, blockchain is used for secure storage of electronic medical records, management of patient consent, and tracking pharmaceutical supply chains [72].

- **Voting Systems**
  Blockchain-based voting systems enhance transparency and reduce the risk of electoral fraud through immutable vote recording [73].

- **Digital Identity Management**
  Blockchain offers decentralized identity management systems that give users control over their personal data while ensuring privacy and security [74].

- **Intellectual Property and Copyright Protection**
  Artists and creators can register digital works on a blockchain to establish proof of ownership and combat copyright infringement [75].

- **Internet of Things (IoT)**
  Blockchain secures communication among IoT devices by eliminating central points of failure and enabling autonomous machine-to-machine interactions [76].

3.5 Blockchain-Based Federated Learning (BCFL)

3.5.1 Motivation for Integration

The integration of Federated Learning (FL) and Blockchain technologies is driven by their complementary strengths in addressing key challenges in distributed machine learning. FL enables multiple clients to collaboratively train a shared model without sharing raw data, thus preserving privacy. However, conventional FL frameworks rely on a central server for coordination and aggregation, which introduces several limitations:

- **Single Point of Failure:** The central server is a vulnerability target, prone to failures or attacks such as model poisoning or denial-of-service (DoS).
- **Trust Assumptions:** All clients must implicitly trust the central server to behave honestly, which is not always realistic.
- **Scalability Constraints:** As client numbers increase, the server may become a communication or computation bottleneck.

- **Lack of Transparency:** Model aggregation and decision-making are typically opaque to participants.

Blockchain, as a decentralized and tamper-proof ledger, offers a potential solution to these challenges by replacing the central aggregator with a distributed system that enables secure, transparent, and auditable model update coordination.

3.5.2 Blockchain as a Secure Aggregator in Federated Learning

In Blockchain-Based Federated Learning (BCFL), blockchain assumes the role of a decentralized aggregator. Each client's local model updates are submitted to the blockchain network, ensuring traceability, verifiability, and integrity. This design decentralizes trust and enhances security through the following mechanisms:

- **Immutable Logging:** Verified model updates are stored on the blockchain, ensuring tamper-resistance and auditability.
- **Trustless Coordination:** Model updates are validated through decentralized consensus mechanisms, eliminating reliance on a single entity.
- **Attack Resilience:** Blockchain mitigates risks of model poisoning, inference, replay, and Sybil attacks via consensus, timestamping, and cryptographic identity management.
- **Smart Contracts:** Smart contracts automate model verification rules, reward systems, and access control, further strengthening the decentralization and reliability of FL.

This integration is particularly valuable in multi-stakeholder domains such as healthcare and the Internet of Medical Things (IoMT), where data sensitivity, trust, and transparency are critical.

3.5.3 Challenges in FL-Blockchain Integration

While BCFL offers enhanced security, decentralization, and trust, it also introduces several practical and technical challenges:

- **Communication Overhead:** Blockchain introduces latency due to transaction broadcasting, mining, and consensus delays.
- **Resource Constraints:** Blockchain operations such as mining or consensus can be computationally intensive, which is unsuitable for lightweight IoT or IoMT devices.
- **Privacy Leakage via Metadata:** Although raw data is preserved locally, model metadata or update patterns may still leak sensitive information.
- **Consensus Scalability:** Consensus mechanisms must scale efficiently as the number of clients increases without compromising security.
- **Smart Contract Limitations:** Executing complex model-related operations on-chain is restricted by current smart contract capabilities.

Overcoming these challenges requires careful architectural design, such as hybrid on-chain/off-chain strategies, lightweight blockchain protocols, and privacy-preserving techniques like differential privacy and homomorphic encryption.

3.6 Conclusion

In conclusion, this chapter has provided a detailed review and analysis of the three foundational technologies—Centralized Learning, Federated Learning, and Blockchain—that underpin the development of intelligent intrusion detection systems. By examining their respective advantages and limitations, the chapter has established a clear understanding of how these technologies complement each other. Particularly, the integration of Blockchain with Federated Learning offers a promising approach to achieving decentralized, secure, and privacy-preserving systems, which are crucial for advancing intrusion detection in IoMT environments.

Chapter 4

Experiment, Results and Discussion

4.1 Introduction

This chapter presents a comprehensive overview of the practical and experimental findings obtained throughout the study.It begins by highlighting the use of specialized programming languages and libraries to efficiently implement the experiments and data analysis. The chapter then outlines the outcomes of the data preprocessing stage, including dimensionality reduction and class balancing applied to the datasets.

In this context, the study addressed two types of classification tasks: a multiclass classification problem consisting of six classes, and a binary classification problem. The centralized model was first evaluated as a baseline, followed by an assessment of the performance of various federated learning strategies under different experimental conditions, including variations in the number of clients, training rounds, and data partitioning methods.This chapter concludes with a discussion on the integration of blockchain technology into the federated learning framework, emphasizing its impact on the overall efficiency and effectiveness of the learning process.

4.2 Experimental Environment

This section provides a comprehensive description of the experimental environment used in this study, including the computing platform, programming libraries, and tools employed in developing the models and analyzing the results.

4.2.1 programming language used

Python is a high-level and user-friendly programming language widely adopted in artificial intelligence due to its support for powerful libraries such as PyTorch, TensorFlow, and scikit-learn. It enables efficient development of deep learning models and is extensively used in federated learning through frameworks like Flower and TensorFlow Federated, making it an essential tool for building intelligent systems that prioritize privacy.

4.2.2 Computational Platform

- **Kaggle :**All experiments were conducted on the Kaggle platform, a leading cloud-based service for data science and machine learning. Founded in 2010 and acquired by Google in 2017, Kaggle became part of the Google Cloud ecosystem. The platform offers a fully integrated environment for researchers and developers, including access to high-performance computing resources such as GPUs—available free of charge for up to 30 hours per week, with usage limits reset every Saturday at UTC.
  The experiments were executed using **Jupyter Notebooks** provided within Kaggle, which facilitated interactive development, efficient visualization, and iterative debugging of models. This environment was a key factor in enabling the efficient training and evaluation of computationally intensive models without the need for advanced local hardware.

4.2.3 Tools and Libraries

- **NumPy:** For numerical operations and array manipulation.
- **Pandas :**For data loading, cleaning, and manipulation.
- **Matplotlib and Seaborn :** For visualizing data distributions and results.
- **Scikit-learn (sklearn):** For preprocessing (e.g., MinMaxScaler, StandardScaler), dimensionality reduction (PCA), model evaluation metrics , and data splitting.
- **Imbalanced-learn (imblearn)::** For handling class imbalance using SMOTE and RandomUnderSampler.

- **PyTorch (torch)** For building, training, and evaluating deep learning models.
- **Flower (flwr) :** For implementing and simulating federated learning using strategies such as FedAvg, FedProx, FedAdam, and FedYogi.
- **System Utilities** Including os, sys, and warnings for file handling and runtime control.
- **Hashlib :** For generating secure hash values, primarily used in blockchain applications.
- **Datetime:** For timestamp generation in blockchain transactions and logs.
- **JSON:** For data serialization and deserialization.
- **Base64:** For encoding digital signatures and binary data in a JSON-compatible format.
- **Time:** For managing timing and delays during simulation.
- **Typing:** For type hinting and structured code documentation in Python.

4.3 Dataset Description

In this work, the CICIoMT2024 dataset was adopted, which was developed by the Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick. It is considered one of the leading datasets in the field of Internet of Medical Things (IoMT) security. The data was collected in a realistic testbed environment comprising 40 medical devices, including both real and simulated ones. These devices communicated over various protocols such as Wi-Fi, MQTT, and Bluetooth.

The dataset contains 18 types of cyberattacks, grouped into five main categories:

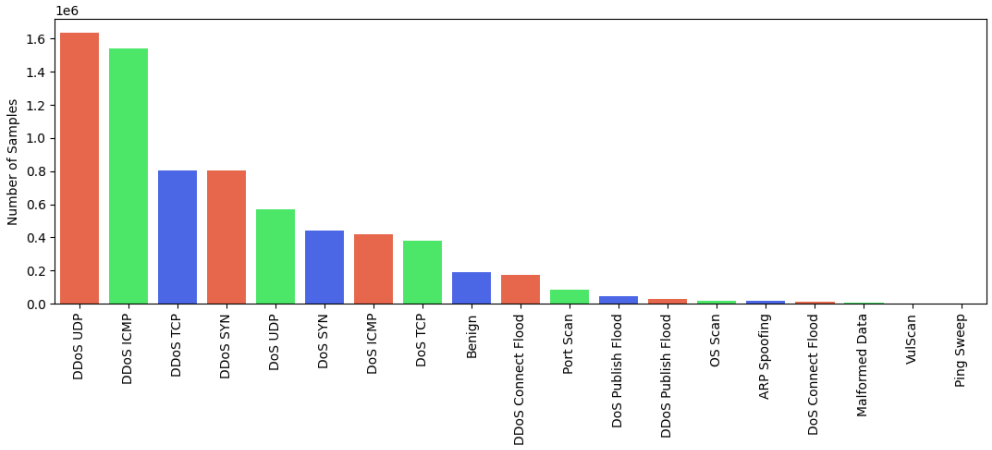


Figure 4.1: Distribution of Attack Types and Benign Traffic in the CICIoMT 2024 Dataset

- **Denial of Service (DoS) attacks**, including: *SYN Flood, TCP Flood, ICMP Flood, and UDP Flood*
- **Distributed Denial of Service (DDoS) attacks**, including: *SYN Flood, TCP Flood, ICMP Flood, and UDP Flood*
- **Reconnaissance attacks**, including: *Ping Sweep, Vulnerability Scan, OS Scan, and Port Scan*
- **MQTT-based attacks**, including: *Malformed Data, DoS Connect Flood, DDoS Connect Flood, DoS Publish Flood, and DDoS Publish Flood*

- **Spoofing attacks**, including: *ARP Spoofing*

The dataset includes 44 features extracted from network traffic, covering various attributes such as:

- Number of packets sent and received
- Amount of data transmitted
- Inter-arrival time between packets
- Type of communication protocol
- MQTT-related features such as `Topic Name`, `QoS Level`, and `Retain Flag`
- Source and destination IP addresses and ports

This dataset provides a rich and practical benchmark for building machine learning models capable of classifying network activities and detecting attacks in sensitive IoMT environments. The dataset and further details are publicly available on the Canadian Institute for Cybersecurity's official website [77].

4.4 Data Preprocessing

Data preprocessing is a fundamental step in preparing data for machine learning and deep learning models, as it directly contributes to improving the accuracy and reliability of the model. Raw data often contains noise, missing values, and class imbalances, which negatively impact model performance.

- **Step 1: Data Collection**
  The CIC 2024 IoMT dataset was obtained directly from the official Canadian Institute for Cybersecurity (CIC) repository and subsequently uploaded to Google Drive to ensure streamlined access and centralized management throughout the preprocessing and analysis pipeline.

- **Step 2: Feature and Target Selection**
  Columns 0–45 were selected as input features, and column 46 (“Category”) was used as the target variable.

- **Step 3: Removal of Non-Numeric and Constant Features**
  Non-numeric features (if present) were excluded. Additionally, features with only a single unique value were removed, as they do not contribute discriminative information.

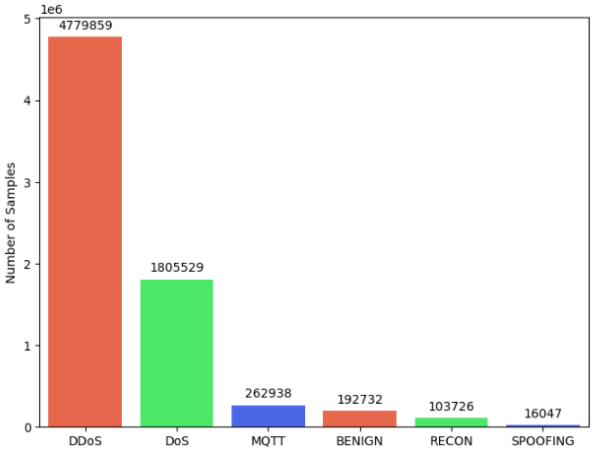
- **Step 4: Standardization** Each feature was standardized to have zero mean and unit variance using the formula:

  where is the mean and is the standard deviation. Any undefined values (e.g., due to division by zero) were replaced with zero.

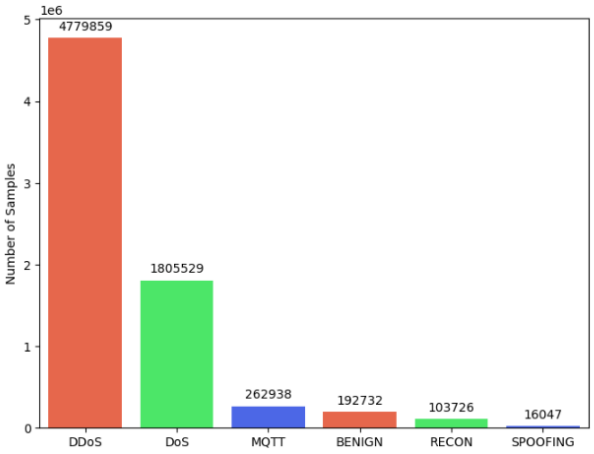
- **Step 5 : Data Balancing**

  - **Synthetic Minority Oversampling Technique (SMOTE) :**
    SMOTE is an oversampling technique used to address class imbalance by generating synthetic samples for minority classes. Instead of simply duplicating existing instances, SMOTE interpolates between a data point and one of its nearest neighbors in the feature space, creating new, plausible instances that preserve the underlying class distribution.

- **Random Undersampling :**
  Random Undersampling is a downsampling method that tackles class imbalance by randomly removing instances from overrepresented classes. This reduces the size of majority classes to match minority class sizes or a specified target, thereby balancing the dataset at the cost of potentially discarding useful information.
- **Target Size:**
  All classes were adjusted to contain 200,500 instances



(a) Before balancing



(b) After balancing

Figure 4.2: Data Balancing

► **Step 6 : MinMaxScaler**
In this step, the MinMaxScaler was applied to the balanced dataset in order to normalize the feature values. The MinMaxScaler is a standard technique for rescaling features, ensuring that all feature values fall within a specified range, typically [0, 1]. This transformation is performed using the following formula:

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

- x' is the scaled (or normalized) value of the feature.

- x is the original value of the feature before scaling.
- $\min(X)$ is the minimum value of the feature X in the dataset.
- $\max(X)$ is the maximum value of the feature X in the dataset.

► **Step 7 : Data Splitting** The data was split into training, testing, and validation sets according to the specified proportions. 70% of the data was allocated to the training set, while the remaining 30% was split into validation and test sets, each receiving 50%.

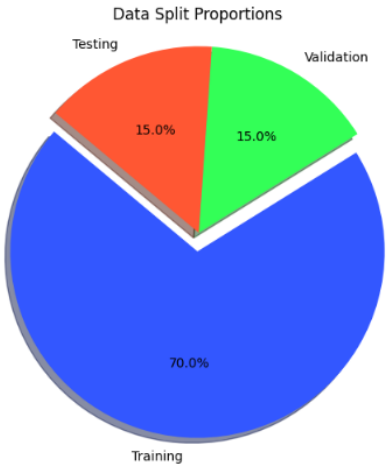


Figure 4.3: Data Splitting

4.5 Centralized Learning (CL)

In the Centralized Learning (CL) setting, three types of models were evaluated: CNN, LSTM, and a hybrid CNN-LSTM. These models were compared based on their performance using several evaluation metrics, including loss,accuracy, precision, recall, and F1-score, in order to determine the most effective architecture.
In all implemented models, the Rectified Linear Unit(ReLU) activation function was utilized . The Cross-Entropy Loss function was adopted as the training criterion. Each model was trained for 20 epochs under a centralized learning setup.

► **Binary Classification Model and Results**

Table 4.1: Binary classification model

Model	**No. of Layers**	**Hidden** Size	**CNN Channels**	**Sequence Length**	**No. of Classes**
CNN	3 Conv + FC	—	64, 128, 256	45	2
LSTM	2 LSTM + 2 FC	256, 64	—	45	2
CNN-LSTM	2 Conv + 1 LSTM + FC	256	64, 128	45	2

Table 4.2: Binary classification results

Model	Loss	Accuracy	Precision	Recall	F1-score	Learning Rate
CNN	0.0097	0.9977	0.9977	0.9977	0.9977	0.001
CNN-LSTM	0.0085	0.9979	0.9979	0.9979	0.9979	0.001
LSTM	0.0085	0.9980	0.9981	0.9980	0.9980	0.01

In the binary classification task, all three models demonstrated excellent performance, with the LSTM model slightly outperforming the others in terms of accuracy, loss, and F1-score, highlighting its strength in capturing temporal dependencies in the data. The CNN-LSTM model ranked second, benefiting from its combined spatial and temporal learning capabilities. Although the CNN model ranked last, it still achieved very high performance, confirming its effectiveness in feature extraction for this type of task.

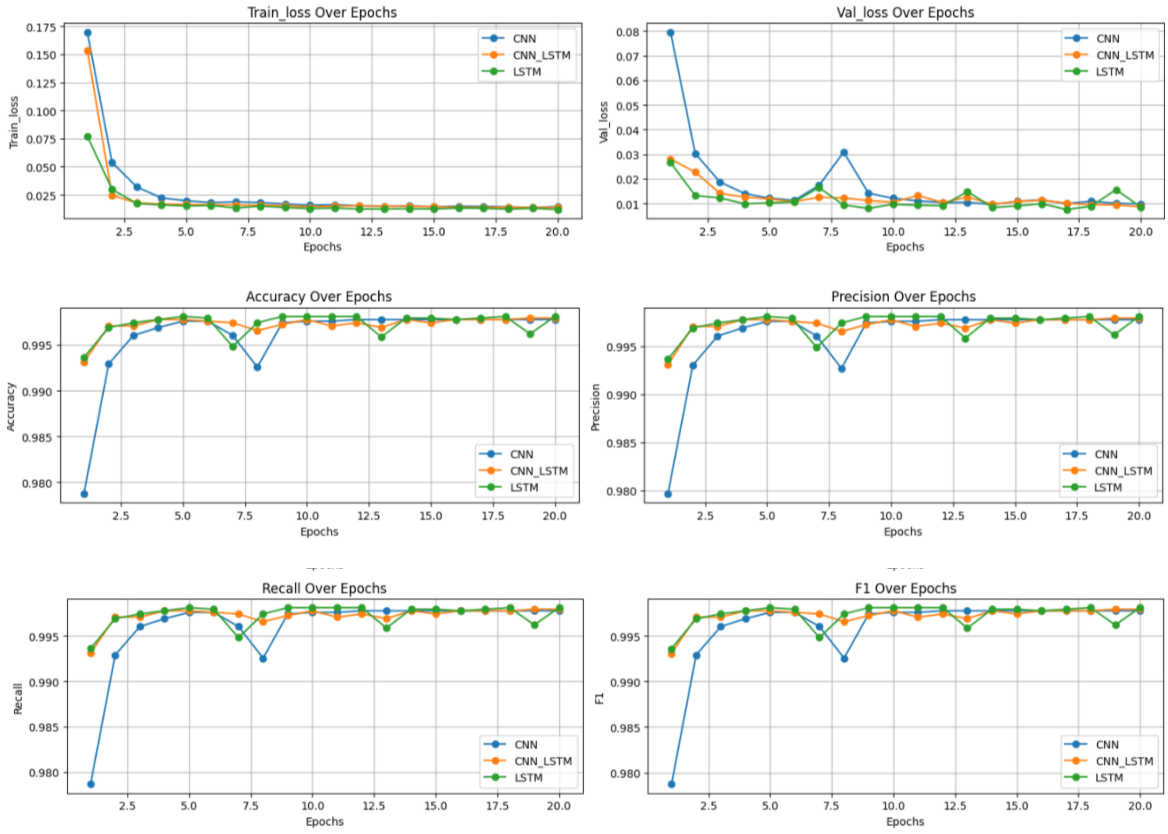


Figure 4.4: Performance Metrics of Binary Classification

► **Multiclass Classification Model and Results**

Table 4.3: Multiclass classification model

Model	**No. of Layers**	**Hidden** Size	**CNN Channels**	**Sequence Length**	**No. of Classes**
CNN	3 (3 Conv + FC)	—	128, 128, 256	44	6
LSTM	2 (LSTM + FC)	128	—	1	6
CNN-LSTM	2 Conv+1 LSTM +FC	256	64, 128	44	6

Table 4.4: Multiclass classification results

Model	**Loss**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Learning Rate**
CNN	0.2843	0.8604	0.8621	0.8604	0.8608	0.001
CNN_LSTM	0.2753	0.8639	0.8658	0.8639	0.8641	0.001
LSTM	0.2873	0.8552	0.8593	0.8552	0.8549	0.001

In the multiclass classification task, a slight decrease in performance was observed across all models, which is expected due to the increased complexity of the task. Nevertheless, the CNN-LSTM model achieved the best overall balance among the evaluation metrics, followed by the CNN model, while the LSTM model recorded the lowest performance among the three. These results indicate that combining convolutional and recurrent layers is particularly beneficial in complex multiclass tasks.

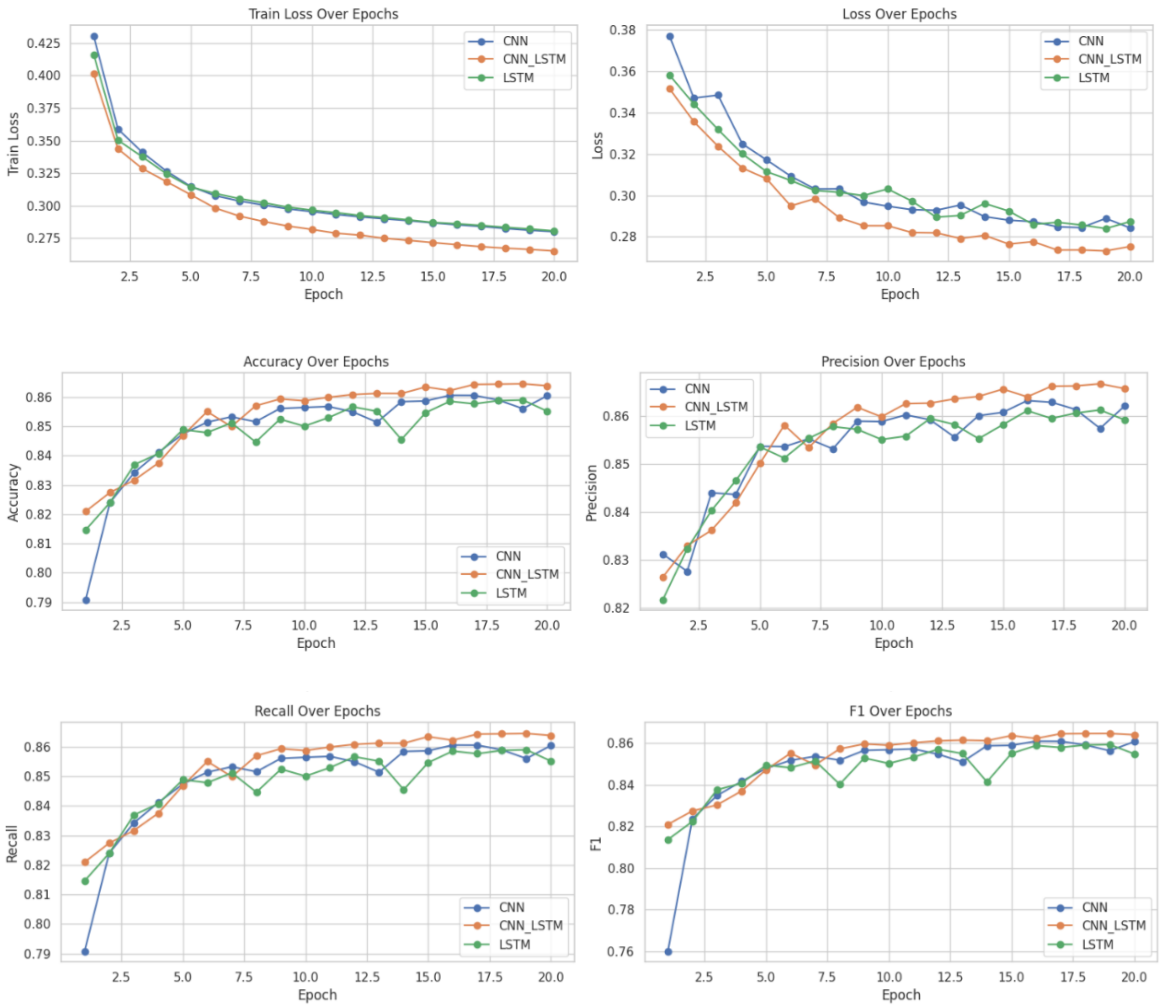


Figure 4.5: Performance Metrics of Multiclass Classification

4.6 Federated Learning implementation

In this study, two Federated Learning strategies—FedAvg and FedProx—were implemented and evaluated under both IID (Independent and Identically Distributed) and Non-IID data settings. The primary objective was to assess how varying the number of clients (5, 10, and 20) and communication rounds (5, 10, and 20) impacts the performance and stability of each strategy. The CNN-LSTM model, which demonstrated the highest accuracy during centralized training, was selected for the federated setup.

Table 4.5 presents the accuracy results when the number of clients is set to 5. Both strategies show a clear improvement in accuracy as the number of communication rounds increases. Under the IID setting, FedProx slightly outperforms FedAvg, especially at higher rounds. In contrast, under Non-IID conditions, FedAvg shows marginally better results, particularly at 20 rounds.

Table 4.5: Federated Learning Accuracy (Num Clients = 5)

Communication Rounds	FedAvg Accuracy	FedProx Accuracy
IID		
5	0.8247	0.8237
10	0.8317	0.8353
20	0.8459	0.8513
Non-IID		
5	0.8298	0.8314
10	0.8426	0.8414
20	0.8649	0.8631

As the number of clients increases to 10, shown in Table 4.6, both FedAvg and FedProx maintain stable performance under IID conditions, with nearly identical accuracy values across different communication rounds. Under the Non-IID setting, FedProx maintains a slight and consistent advantage over FedAvg, particularly as the communication rounds increase.

Table 4.6: Federated Learning Accuracy (Num Clients = 10)

Communication Rounds	FedAvg Accuracy	FedProx Accuracy
IID		
5	0.8179	0.8161
10	0.8261	0.8265
20	0.8379	0.8378
Non-IID		
5	0.8321	0.8332
10	0.8333	0.8349
20	0.8642	0.8645

When the number of clients reaches 20, as illustrated in Table 4.7, a slight decline in accuracy under the IID setting is observed for both strategies. This is likely due to increased data fragmentation across more clients, which makes local training less effective. However, in the Non-IID setting, both strategies maintain high accuracy, with FedProx again demonstrating a marginal advantage, especially at higher communication rounds.

Table 4.7: Federated Learning Accuracy (Num Clients = 20)

Communication Rounds	FedAvg Accuracy	FedProx Accuracy
IID		
5	0.8020	0.7993
10	0.8168	0.8195
20	0.8257	0.8262
Non-IID		
5	0.8349	0.8359
10	0.8437	0.8442
20	0.8638	0.8648

Overall, both FedAvg and FedProx demonstrate strong and stable performance across various configurations. FedProx shows a slight advantage in Non-IID settings across all client numbers, whereas performance differences under IID settings are minimal. The results highlight the robustness of both strategies and the influence of communication rounds and client count on federated model accuracy.

Based on the experimental results, the following observations can be made:

- The results show a gradual improvement in model accuracy as the number of communication rounds increases, indicating that the model becomes more stable and effective with repeated interactions between clients and the central server.
- When comparing the FedAvg and FedProx strategies, FedProx demonstrates slightly better performance in the Non-IID data scenario, which aligns with its design objective to address data heterogeneity across clients.
- Despite the slight performance difference, the results indicate a general similarity in effectiveness between FedAvg and FedProx under the conditions of this study.
- Increasing the number of clients results in a noticeable decrease in model accuracy, especially in the IID data setting. This can be attributed to the reduced amount of data available per client, which may negatively impact the quality of local model updates.

4.7 Blockchain-Based Federated Learning (BCFL)

Traditional federated learning (FL) architectures rely heavily on a central server that is responsible for coordinating the training process, aggregating local model updates, and communicating with all participating clients. However, this centralized design introduces several critical limitations:

- **Single Point of Failure:** The central server constitutes a vulnerability, as it may become a target for attacks such as model poisoning or denial-of-service (DoS).
- **Trust Assumptions:** All clients must place implicit trust in the central aggregator, assuming it will behave honestly and securely, which may not always be guaranteed.
- **Scalability Constraints:** As the number of clients increases, the server may become a bottleneck, limiting the scalability of the system.
- **Lack of Transparency:** The central server operates as a black box, offering limited visibility into the selection, validation, and aggregation of model updates.

To address these issues, this work proposes the integration of blockchain with federated learning to transition toward a decentralized architecture. Blockchain offers several advantages that align with the principles of distributed trust and security:

- **Elimination of the Central Server:** By enabling peer-to-peer (P2P) communication and decentralized consensus mechanisms, blockchain removes the need for a centralized coordinator.
- **Immutable Logging of Model Updates:** Each model update can be recorded on the blockchain ledger, ensuring traceability, verifiability, and transparency.
- **Distributed Trust:** Trust is no longer concentrated in a single entity but is instead distributed among all participants in the network.
- **Enhanced Security and Robustness:** The decentralized nature of blockchain mitigates the risks associated with centralized failures and targeted attacks.

4.7.1 Implementation Strategy

The implementation of Blockchain-Based Federated Learning (BCFL) followed a phased approach:

1. **Initial Implementation Using the Flower Framework :** Federated learning was initially implemented using the Flower framework, which supports centralized orchestration. Experiments were conducted under both IID and Non-IID data distributions with varying numbers of clients to understand core FL dynamics.

2. **Recognition of Centralization Limitations :** As integration with blockchain was pursued, it became evident that Flower's dependence on a central server conflicted with the decentralized nature of blockchain-based systems.

3. **Migration to PyTorch for Flexibility :** To overcome the centralization limitation, the implementation was migrated to PyTorch. This allowed for the development of custom communication protocols simulating peer-to-peer interactions using multithreading, enabling each client to operate independently on a shared machine.

4. **Blockchain Integration :** Blockchain was introduced to handle the consensus process and securely log model updates. This integration replaced the central server with a decentralized validation mechanism, thereby achieving a trustless and resilient federated learning system.

5. **The network topology :**
   I used a ring topology, where each client is connected to its immediate left and right neighbors.

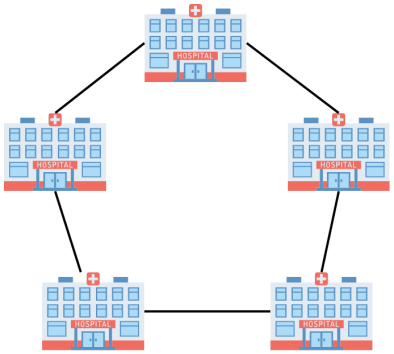


Figure 4.6: The network topology

4.7.2 Decentralized Federated Learning with Blockchain Support

In this work, a decentralized federated learning framework is implemented using a ring topology, where each peer communicates only with its immediate neighbors. The process of each training round involves the following steps:

1. **Start with the latest global model:**
   Each peer begins the round by retrieving the most recent version of the global model, which is recorded and verified on the blockchain.

2. **Train on local data:**
   The peer performs local training using its private dataset. No raw data is shared with any other participant, ensuring data privacy.

3. **Exchange and aggregate updates with neighbors:**
   Each peer sends its locally updated model parameters to its immediate neighbors (left and right) in the ring topology. Then, it computes the average of its own update and its neighbors' updates to generate a locally aggregated model.

4. **Submit the aggregated update to the blockchain:**
   A transaction is created containing the aggregated model update and the corresponding local dataset size. The transaction is signed digitally and submitted to the blockchain for inclusion in the next block.

5. **Block confirmation (mining):**
   A peer is selected to perform mining, during which:

   - It verifies the validity of submitted transactions.
   - It compiles valid updates into a new block.
   - It solves a computational puzzle to append the block to the blockchain.

   Once confirmed, the block becomes a permanent and immutable part of the ledger.

6. **Compute the new global model (using median update):**
   After block confirmation, all peers extract the model updates from the block and compute the new global model by selecting the *median* of all updates. This helps mitigate the impact of outlier models and enhances robustness.

7. **Announce the new global model:**
   A peer publishes a transaction containing the new global model parameters, which serves as the starting point for the next training round.

8. **Evaluate the global model:**
   The published model is evaluated using a separate test dataset to assess its performance.

9. **Start a new training round:**
   The next round begins, repeating the steps with the newly announced global model.

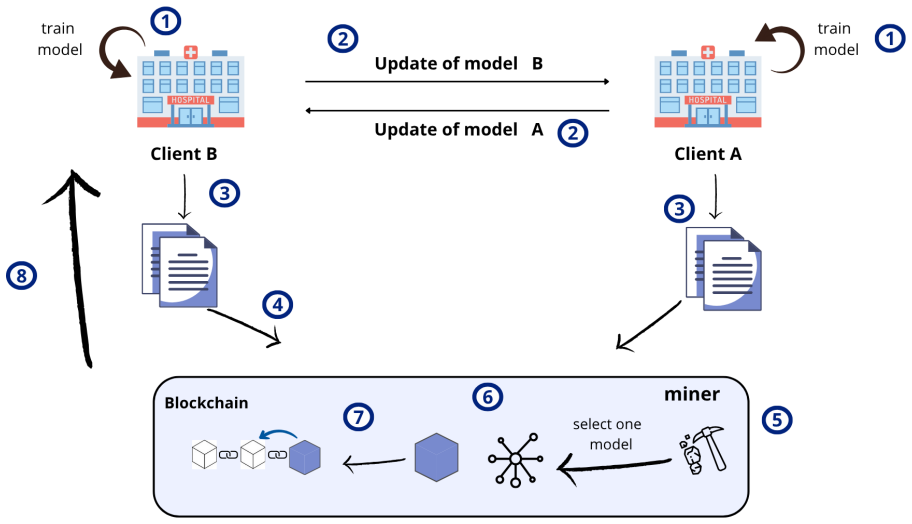


Figure 4.7: Operational Cycle of Blockchain-based Federated Learning

4.7.3 Blockchain Federated Learning (BCFL) Results

This is a block from the blockchain, where distributed learning model transactions from peers are recorded, including the sender ID, transaction type, model data, and digital signature, ensuring timestamped verification and data integrity within a decentralized and secure framework.

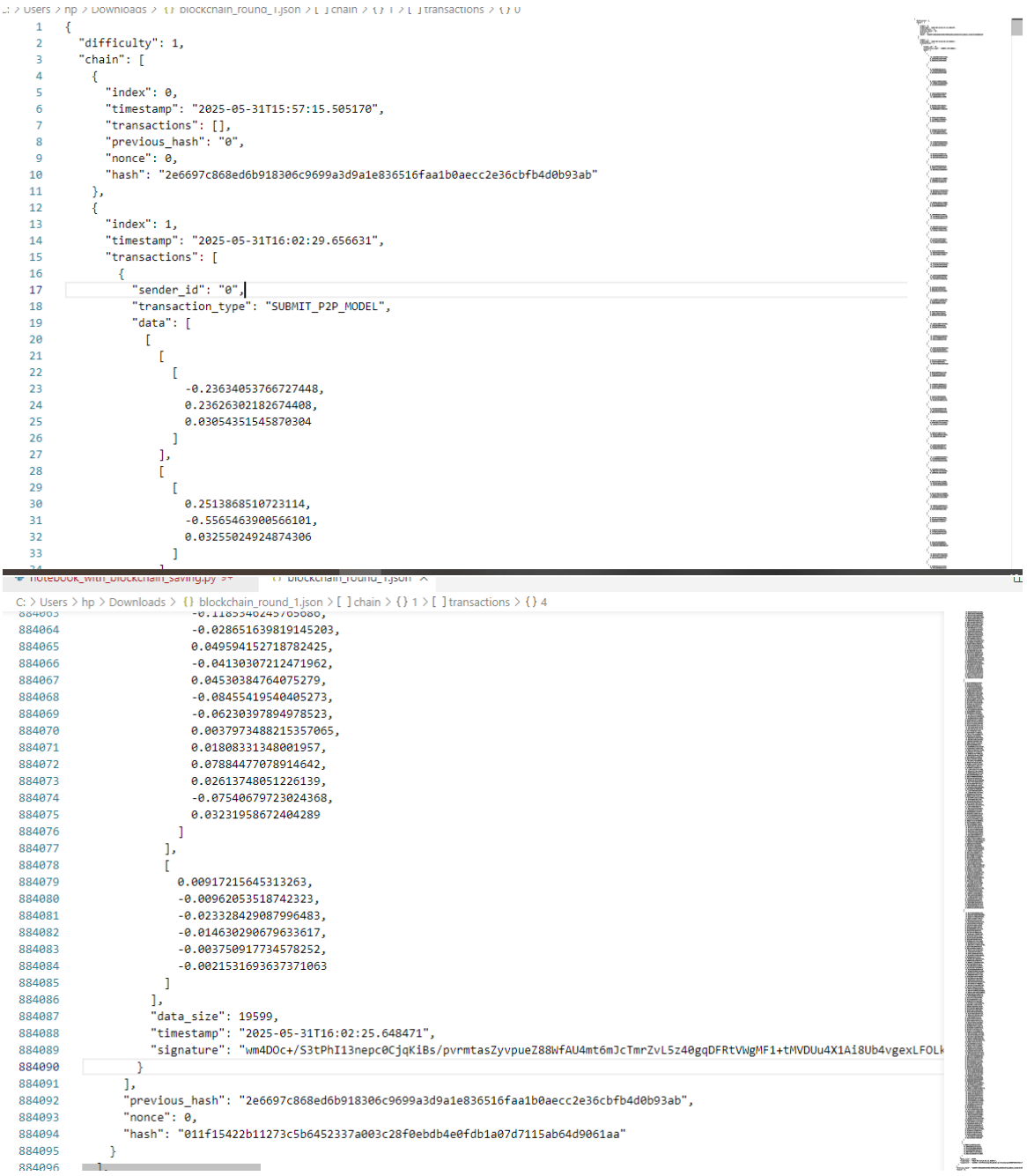


Figure 4.8: Structure of a Blockchain Block

- Figure 4.9 illustrates the evolution of key performance metrics (Accuracy, Precision, Recall, and the F1-score) over twenty training rounds in a Blockchain-based Federated Learning. A significant improvement in all metrics was observed starting from the fifth round, with performance reaching a stable and consistent level by the twelfth round. Both Accuracy and the F1-score attained values around 0.82, reflecting a strong and balanced model performance in classification tasks.

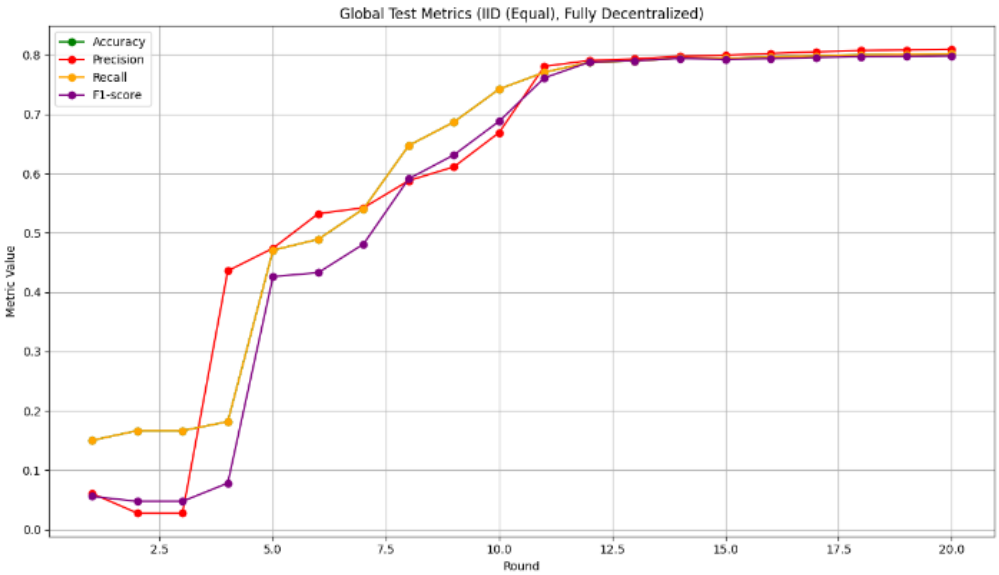


Figure 4.9: Evaluation of BCFL (Accuracy, F1-score, Recall, and Precision)

- Furthermore, the figure 4.10 depicts the loss value on test data throughout the same period. A sharp decline in loss was noted during the first ten rounds, followed by a phase of relative stabilization. This indicates that the model was effectively trained and reached a state of convergence, with a final loss value approximately equal to 0.4.

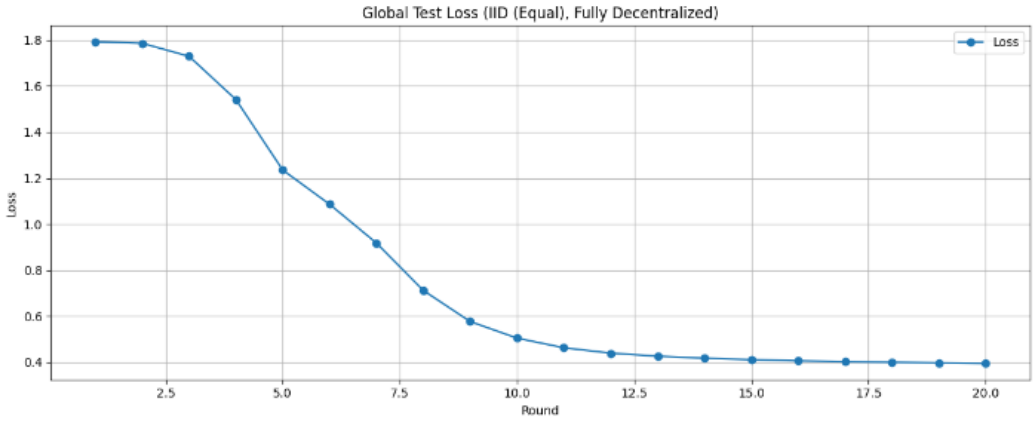


Figure 4.10: The results of Test Loss

Discussion

The experimental evaluations of both FedAvg and FedProx algorithms provide valuable insights into the behavior of federated learning under different conditions regarding the number of communication rounds and participating clients.

First, the results show a gradual improvement in model accuracy as the number of communication rounds increases, indicating that iterative interaction between clients and the central server plays a crucial role in stabilizing and enhancing the global model. This improvement demonstrates that the learning process benefits from continuous collaboration and iterative model refinement over time.

Second, although both FedAvg and FedProx achieved relatively stable and high performance across various settings, FedProx showed a slight advantage in performance, particularly in environments characterized by client heterogeneity (Non-IID).

It is also observed that increasing the number of clients leads to a noticeable decrease in model accuracy, which can be attributed to the reduced amount of data available per client. This may negatively affect the quality of local updates and consequently result in a less accurate global model. This outcome suggests a trade-off between system scalability and model accuracy.

Furthermore, blockchain technology was integrated into the proposed system to eliminate reliance on a central server and reduce single points of failure. This integration enabled decentralized and transparent recording of model updates, enhancing system reliability and resistance to tampering. Simulation results on the IoMT dataset demonstrated that the blockchain-supported system maintained high performance, achieving an accuracy of 82%, which indicates the effectiveness of the proposed approach in enhancing the security and privacy of the federated learning process.

These results confirm the effectiveness of integrating federated learning with blockchain to improve the performance and efficiency of intrusion detection systems.

4.8 Conclusion

To summarize, this chapter has comprehensively presented the practical implementation and experimental evaluation of the proposed system. It demonstrated how various data preprocessing techniques and classification tasks were addressed under different configurations, providing valuable insights into the comparative performance of centralized and federated learning models. Furthermore, the integration of blockchain technology within the federated learning framework was shown to significantly enhance the system's efficiency, security, and robustness, laying a strong foundation for future advancements in secure and reliable intrusion detection for IoMT.

General Conclusion

The rapid advancement of the Internet of Things (IoT) has revolutionized many critical sectors, with healthcare being one of the most impacted. This evolution has given rise to the Internet of Medical Things (IoMT), where interconnected medical devices collect, process, and exchange sensitive patient data in real time to improve diagnostic accuracy, treatment effectiveness, and patient monitoring. However, the integration of IoT into medical infrastructures has significantly increased the attack surface, raising serious concerns about data privacy, device security, and system integrity.

The primary objective of this thesis was to explore and enhance the integration of federated learning with blockchain technology in the context of Network Intrusion Detection Systems (NIDS) within Internet of Medical Things (IoMT) environments. Through a comprehensive analysis covering the fundamentals of information security, artificial intelligence techniques, federated learning principles, and blockchain properties, we proposed an intelligent threat detection model that combines privacy, decentralization, and security.

Our study began by reviewing the foundational concepts of information security and IoMT, highlighting the challenges faced by connected medical systems—particularly in terms of privacy and cyberattacks. We then examined the evolution of intrusion detection systems using machine learning techniques, emphasizing the limitations of traditional centralized models, especially in sensitive environments like IoMT that require high privacy and distributed data sources.

The core contribution of this research was the proposal of a smart intrusion detection system based on blockchain-enhanced federated learning. This system aims to achieve distributed model training without sharing raw data, while ensuring no single point of failure through blockchain integration to securely and immutably record model updates. Blockchain also fosters trust among participants and enables decentralized verification of the updates' integrity.

The proposed model was validated using the CICIoMT2024 dataset, a recent and comprehensive dataset designed specifically to simulate attacks on IoMT devices under realistic network scenarios. Our experiments included data partitioning using both IID and Non-IID distributions to emulate heterogeneous data environments. The results demonstrated that the proposed system achieves high detection accuracy while preserving data privacy and minimizing the security risks associated with centralized aggregation.

The experimental outcomes highlighted the ability of blockchain-assisted federated learning to efficiently handle data heterogeneity and provide a more secure and transparent training environment. Furthermore, the system exhibited scalability and flexibility in addressing various threats, making it suitable for smart healthcare applications.

Future directions of this work include exploring other models within the federated learning framework, such as Split Learning and adaptive collaborative learning. We also aim to enhance the proposed system through advanced techniques for feature and client selection, and the use of smart contracts to automate collaboration among participating entities. Additionally, we plan to broaden the evaluation using more IoMT datasets to further strengthen the reliability of the proposed approach and analyze its performance under more diverse attack scenarios.

Bibliography

[1] Ali Ghubaish, Tara Salman, Maede Zolanvari, Devrim Unal, Abdulla Al-Ali, and Raj Jain. Recent advances in the internet-of-medical-things (iomt) systems security. *IEEE Internet of Things Journal*, 8(11):8707--8718, 2020.

[2] The Open University. What is artificial intelligence?, 2020. Accessed: 2025-06-03.

[3] Shaista Ashraf Farooqi and Fatima Zafar. Security and privacy challenges in the internet of medical things (iomt): A comprehensive review. pages 3007--1909, 10 2024.

[4] Weizhi Meng, Elmar Wolfgang Tischhauser, Qingju Wang, Yu Wang, and Jinguang Han. When intrusion detection meets blockchain technology: a review. *Ieee Access*, 6:10179--10188, 2018.

[5] BotPenguin. Intrusion detection - meaning definition, 2024. Accessed: 2025-06-03.

[6] Pandiaraj Manickam, Siva Ananth Mariappan, Sindhu Monica Murugesan, Shekhar Hansda, Ajeet Kaushik, Ravikumar Shinde, and SP Thipperudraswamy. Artificial intelligence (ai) and internet of medical things (iomt) assisted biomedical systems for intelligent healthcare. *Biosensors*, 12(8):562, 2022.

[7] Ijaz Ahmad, Tanesh Kumar, Madhusanka Liyanage, Jude Okwuibe, Mika Ylianttila, and Andrei Gurtov. Overview of 5g security challenges and solutions. *IEEE Communications Standards Magazine*, 2(1):36--43, 2018.

[8] Vasisht Mothukuri, Reza M. Parizi, Seyedamin Pouriyeh, Y. Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619--640, 2021.

[9] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletarì, Holger R. Roth, Shadi Albarqouni, and Daguang Xu. The future of digital health with federated learning. *NPJ Digital Medicine*, 3(1):119, 2020.

[10] Youyang Qu, Shiva Raj Pokhrel, Sahil Garg, Liang Gao, and Yong Xiang. A blockchain-based federated learning framework for iot. *IEEE Internet of Things Journal*, 8(9):7414--7425, 2021.

[11] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020.

[12] Antonio Reyna, Carlos Martín, Jaime Chen, Enrique Soler, and Manuel Díaz. On blockchain and its integration with iot: Challenges and opportunities. *Future Generation Computer Systems*, 88:173--190, 2018.

[13] Qinyong Lin, Xiaorong Li, Ken Cai, Mohan Prakash, and D. Paulraj. Secure internet of medical things (iomt) based on ecmqv-mac authentication protocol and ekmc-scp blockchain networking. *Information Sciences*, 654:119783, 2024.

[14] Ana Carolina Borges Monteiro, Reinaldo Padilha França, Rangel Arthur, and Yuzo Iano. An overview of the internet of medical things (iomt): Applications, benefits, and challenges. *Security and Privacy Issues in Internet of Medical Things*, pages 83--98, 2023.

[15] Bharat Bhushan, Avinash Kumar, Ambuj Kumar Agarwal, Amit Kumar, Pronaya Bhattacharya, and Arun Kumar. Towards a secure and sustainable internet of medical things (iomt): Requirements, design challenges, security techniques, and future trends. *Sustainability*, 15(7):6177, 2023.

[16] Binariks. Iomt security: Risks & best practices to secure iomt devices, 2023. Accessed: 2025-06-03.

[17] John Morton. *Towards Federated Learning Intrusion Detection Systems* (*IDS*) *Within Internet of Medical Things* (*IoMT*) *Ecosystems*. PhD thesis, The George Washington University, 2024.

[18] Liu Hua Yeo, Xiangdong Che, and Shalini Lakkaraju. Understanding modern intrusion detection systems: a survey. *arXiv preprint arXiv*:1708.07174, 2017.

[19] Elaine Rich and Kevin Knight. *Introduction to Artificial Intelligence*. McGraw-Hill, 1991. PDF version retrieved from https://unidel.edu.ng/focelibrary/books/Introduction

[20] Stuart Russell and Peter Norvig. *Artificial Intelligence*: *A Modern Approach*. Pearson, Upper Saddle River, NJ, 4th edition, 2020.

[21] Alan M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433--460, 1950.

[22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436--444, 2015.

[23] Michael I. Jordan and Tom M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255--260, 2015.

[24] Min Chen, Song Mao, Yunhao Zhang, Victor C.M. Leung, and Mohamed-Slim Alouini. Ai empowered intelligent networking: Concepts, technologies, and applications. *IEEE Network*, 34(1):22--29, 2020.

[25] GeeksforGeeks. Applications of artificial intelligence (ai), 2024. Accessed: 2025-06-03.

[26] Ben Goertzel and Cassio Pennachin. *Engineering General Intelligence, Part* 1: *A Path to Advanced AGI via Embodied Learning and Cognitive Synergy*. Springer, Berlin, 2007.

[27] Nick Bostrom. *Superintelligence*: *Paths, Dangers, Strategies*. Oxford University Press, Oxford, 2014.

[28] Michael Wooldridge. *An Introduction to MultiAgent Systems*. Wiley, 2nd edition, 2020.

[29] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[30] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[31] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[32] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436--444, 2015.

[33] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[34] Richard S Sutton and Andrew G Barto. *Reinforcement learning*: *An introduction*. MIT press, 2018.

[35] Hind Alaskar and Tanzila Saba. Machine learning and deep learning: A comparative review. *International Journal of Advanced Computer Science and Applications*, 12(4):362--371, 2021.

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

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097--1105, 2012.

[38] Z. Sun, G. An, Y. Yang, and Y. Liu. Optimized machine learning enabled intrusion detection system for internet of medical things. *Franklin Open*, 6:100056, March 2024.

[39] Ghaida Balhareth and Mohammad Ilyas. Optimized intrusion detection for iomt networks with tree-based machine learning and filter-based feature selection. *Sensors*, 24(17):5712, 2024.

[40] Abdelouahid Si-Ahmed, Mohammed Ali Al-Garadi, and Nadia Boustia. Survey of machine learning based intrusion detection methods for internet of medical things. *arXiv preprint arXiv*:2202.09657, 2022.

[41] Mohammad Alalhareth and Seung-Chan Hong. Enhancing the internet of medical things (iomt) security with meta-learning: A performance-driven approach for ensemble intrusion detection systems. *Sensors*, 24(11):3519, 2024.

[42] Rose Kimanzi, Peter Kimanga, Dennis Cherori, and Peter K. Gikunda. Deep learning algorithms used in intrusion detection systems -- a review. *arXiv preprint arXiv*:2402.17020, 2024.

[43] C. Anitha, C. V. Vivekanand, S. D. Lalitha, S. Boopathi, and R. Revathi. Artificial intelligence driven security model for internet of medical things (iomt). In *Proc. 3rd Int. Conf. Innov. Practices Technol. Manage.* (*ICIPTM*), pages 1--7, February 2023.

[44] S. Ksibi, F. Jaidi, and A. Bouhoula. Iomt security model based on machine learning and risk assessment techniques. In *Proc. Int. Wireless Commun. Mobile Comput.* (*IWCMC*), pages 614--619, June 2023.

[45] T. Saba. Intrusion detection in smart city hospitals using ensemble classifiers. In *Proc. 13th Int. Conf. Develop. eSystems Eng.* (*DeSE*), pages 418--422, December 2020.

[46] D. Alsalman. A comparative study of anomaly detection techniques for iot security using adaptive machine learning for iot threats. *IEEE Access*, 12:14719--14730, 2024.

[47] G. Balhareth and M. Ilyas. Optimized intrusion detection for iomt networks with tree-based machine learning and filter-based feature selection. *Sensors*, 24(17):5712, September 2024.

[48] J. B. Awotunde, K. M. Abiodun, E. A. Adeniyi, S. O. Folorunso, and R. G. Jimoh. A deep learning-based intrusion detection technique for a secured iomt system. In S. Misra, J. Oluranti, R. Damaševičius, and R. Maskeliunas, editors, *Informatics and Intelligent Applications*, pages 50--62. Springer, Cham, Switzerland, 2022.

[49] P. Kulshrestha and T. V. Vijay Kumar. Machine learning based intrusion detection system for iomt. *Int. J. Syst. Assurance Eng. Manage.*, 15(5):1802--1814, May 2024.

[50] S. Khan and A. Akhunzada. A hybrid dl-driven intelligent sdn-enabled malware detection framework for internet of medical things (iomt). *Comput. Commun.*, 170:209--216, March 2021.

[51] S. Liaqat, A. Akhunzada, F. S. Shaikh, A. Giannetsos, and M. A. Jan. Sdn orchestration to combat evolving cyber threats in internet of medical things (iomt). *Comput. Commun.*, 160:697--705, July 2020.

[52] N. Faruqui, M. A. Yousuf, M. Whaiduzzaman, A. Azad, S. A. Alyami, P. Lió, M. A. Kabir, and M. A. Moni. Safetymed: A novel iomt intrusion detection system using cnn-lstm hybridization. *Electronics*, 12(17):3541, August 2023.

[53] Y. Otoum, Y. Wan, and A. Nayak. Federated transfer learning-based ids for the internet of medical things (iomt). In *Proc. IEEE Globecom Workshops* (*GC Wkshps*), pages 1--6, December 2021.

[54] V. Ravi, T. D. Pham, and M. Alazab. Deep learning-based network intrusion detection system for internet of medical things. *IEEE Internet Things Mag.*, 6(2):50--54, June 2023.

[55] I. A. Khan, N. Moustafa, I. Razzak, M. Tanveer, D. Pi, Y. Pan, and B. S. Ali. Xsru-iomt: Explainable simple recurrent units for threat detection in internet of medical things networks. *Future Gener. Comput. Syst.*, 127:181--193, February 2022.

[56] S. Dadkhah, E. C. P. Neto, R. Ferreira, R. C. Molokwu, S. Sadeghi, and A. A. Ghorbani. Ciciomt2024: A benchmark dataset for multi-protocol security assessment in iomt. *Internet Things*, 28:101351, December 2024.

[57] N. Sánchez, A. Calvo, S. Escuder, J. Escrig, J. Domenech, N. Ortiz, and S. Mhiri. Towards enhanced iot security: Advanced anomaly detection using transformer models. In *Proc. KDD 4th Workshop Artif. Intell.-Enabled Cybersecurity Anal.*, Barcelona, Spain, 2024.

[58] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (*AISTATS*), pages 1273--1282. PMLR, 2017.

[59] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50--60, 2020.

[60] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. A comprehensive experimental comparison between federated learning and centralized learning. In *International Conference on Learning Representations* (*ICLR*), 2019.

[61] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, et al. Scaffold: Stochastic controlled averaging for federated learning. *ICML*, 2020.

[62] LiveAction. Centralized vs. decentralized vs. distributed networks: The history & future, 2023. `https://www.liveaction.com/resources/blog-post/centralized-vs-decentralized-vs-distributed-networks-the-history-future/`.

[63] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain challenges and opportunities: A survey. *International journal of web and grid services*, 14(4):352--375, 2018.

[64] Xiwei Xu, Ingo Weber, and Mark Staples. A taxonomy of blockchain-based systems for architecture design. *IEEE Communications Surveys & Tutorials*, 21(3):2437--2466, 2019.

[65] Fran Casino, Thomas K Dasaklis, and Constantinos Patsakis. A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Future Generation Computer Systems*, 95:302--319, 2019.

[66] Types of blockchain networks. `https://www.ibm.com/topics/types-of-blockchain-networks`. Accessed: 2025-05-25.

[67] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. *Proceedings of the IEEE*, 107(9):1048--1076, 2018.

[68] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and cryptocurrency technologies*: *a comprehensive introduction*. Princeton University Press, 2016.

[69] Andreas M Antonopoulos. *Mastering Bitcoin*: *Programming the Open Blockchain*. O'Reilly Media, Inc., 2017.

[70] Bhabendu Kumar Mohanta, Soumyashree S Panda, and Debasish Jena. An overview of smart contract and use cases in blockchain technology. In 2018 9*th international conference on computing, communication and networking technologies* (*ICCCNT*), pages 1--4. IEEE, 2018.

[71] Shailak Jani. An overview of ethereum & its comparison with bitcoin. *Int. J. Sci. Eng. Res*, 10(8):1--6, 2017.

[72] Chimezie C Agbo, Qusay H Mahmoud, and JM Eklund. Blockchain technology in healthcare: A systematic review. *Healthcare*, 7(2):56, 2019.

[73] Abir Ben Ayed. A conceptual secure blockchain-based electronic voting system. *International Journal of Network Security & Its Applications* (*IJNSA*), 9(3):01--09, 2017.

[74] Guy Zyskind, Oz Nathan, and Alex Pentland. Decentralizing privacy: Using blockchain to protect personal data. In 2015 *IEEE Security and Privacy Workshops*, pages 180--184. IEEE, 2015.

[75] Primavera De Filippi and Samer Hassan. Blockchain technology as a regulatory technology: From code is law to law is code. *First Monday*, 21(12), 2016.

[76] Konstantinos Christidis and Michael Devetsikiotis. Blockchains and smart contracts for the internet of things. *IEEE Access*, 4:2292--2303, 2016.

[77] Ciciomt2024 dataset. `https://www.unb.ca/cic/datasets/iomt-dataset-2024.html`. Accessed: 2025-06-02.