Enhancing Security and Privacy in the Internet of Things

A Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

by

Vidushi Agarwal

(2019csz0010)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY ROPAR

June, 2024

Dedicated to my family.

Declaration of Originality

I hereby declare that the work which is being presented in the thesis entitled **"Enhancing Security and Privacy in the Internet of Things"** has been solely authored by me. It presents the result of my own independent research conducted during the time period from July 2019 to March 2024 under the supervision of **Dr. Sujata Pal, Assistant Professor, IIT Ropar**. To the best of my knowledge, it is an original work, both in terms of research content and narrative, and has not been submitted or accepted elsewhere, in part or in full, for the award of any degree, diploma, fellowship, associateship, or similar title of any university or institution. Further, due credit has been attributed to the relevant state-of-the-art and collaborations with appropriate citations and acknowledgments, in line with established ethical norms and practices. I also declare that any idea/data/fact/source stated in my thesis has not been fabricated/ falsified/ misrepresented. All the principles of academic honesty and integrity have been followed. I fully understand that if the thesis is found to be unoriginal, fabricated, or plagiarized, the Institute reserves the right to withdraw the thesis from its archive and revoke the associated Degree conferred. Additionally, the Institute also reserves the right to appraise all concerned sections of society of the matter for their information and necessary action (if any). If accepted, I hereby consent for my thesis to be available online in the Institute's Open Access repository, inter-library loan, and the title & abstract to be made available to outside organizations.

Signature

Name: Vidushi Agarwal
Entry Number: 2019csz0010
Program: PhD
Department: Computer Science and Engineering
Indian Institute of Technology Ropar
Rupnagar, Punjab 140001

Date: 20 June 2024

Acknowledgement

With heartfelt gratitude for the divine blessings graciously provided to me throughout my academic journey, I would like to express my sincere appreciation to all those who have contributed to the successful completion of this thesis. First and foremost, my deepest appreciation goes to my supervisor, Dr. Sujata Pal, whose expertise, guidance, and unwavering support have been invaluable throughout this research journey. I have been extremely lucky to have a supervisor who motivated me throughout my PhD and supported me in every step. I owe her way more than my words could ever express.

I must extend my special gratitude to Dr. Omid Ardakanian, who has been an exceptional mentor and collaborator throughout this journey. His expertise and willingness to share knowledge have significantly contributed to my professional growth and will always be a cherished aspect of my academic journey.

I am also immensely grateful to my doctoral committee members: Dr. Nitin Auluck (Chairperson), Dr. Neeraj Goel, Dr. Shashi Shekhar Jha, and Dr. Sam Darshi for evaluating my research work periodically, their support and invaluable feedback that significantly enriched this research.

I am indebted to IIT Ropar for providing me with all the necessary facilities for this research. I am also deeply grateful to Tata Consultancy Services (TCS) and the Science and Engineering Research Board (SERB) for their financial support and belief in the potential of my research.

Special thanks to my colleagues and peers, Ashish Kaushal, Avani Vyas, Vivek Sethi, Sweta Dey, Shruti Mishra, Gulshan Sharma, Pratibha Kumari, and Sanju Xaviar who have been a source of motivation, inspiration, and unwavering support throughout this journey.

My warmest appreciation extends to my friends and family for their endless love, understanding, and encouragement. Their belief in me has been a constant source of strength and resilience during the challenging moments of this academic pursuit. This accomplishment would not have been possible without all of you. Thank you all for being a part of this journey.

Vidushi Agarwal
Indian Institute of Technology Ropar

Certificate

This is to certify that the thesis entitled **"Enhancing Security and Privacy in the Internet of Things"**, submitted by **Vidushi Agarwal (2019csz0010)** for the award of the degree of **Doctor of Philosophy** of Indian Institute of Technology Ropar, is a record of bonafide research work carried out under my guidance and supervision. To the best of my knowledge and belief, the work presented in this thesis is original and has not been submitted, either in part or full, for the award of any other degree, diploma, fellowship, associateship or similar title of any university or institution.

In my opinion, the thesis has reached the standard fulfilling the requirements of the regulations relating to the Degree.

Sujata Pal.

Signature of the Supervisor(s)
Dr. Sujata Pal
Computer Science and Engineering
Indian Institute of Technology Ropar
Rupnagar, Punjab 140001
Date: 20 June 2024

Lay Summary

With the expanding digital age, our world is becoming increasingly connected each day. Today, it's not just our phones and computers that connect to the internet, but also a wide array of devices in our homes, offices, and cities—from smart refrigerators to traffic lights. This network of connected devices, known as the Internet of Things (IoT), helps in making our lives more convenient and our environments more intelligent. However, this advancement also brings with it a challenge: keeping all the data these devices collect and share safe from spying eyes and potential cyber threats. Given the sheer volume of data and the diversity of devices involved, traditional security measures often fail to address these challenges. Imagine trying to use a single type of lock to secure every door, window, and gate in a bustling, ever-changing city — it's a challenging task that requires a more dynamic approach.

Therefore, our research explores new strategies that adapt to the unique needs of the IoT. We examine how we can protect data from unauthorized access and ensure that our private information stays private, all while maintaining the functionality that makes these connected devices so beneficial. This involves looking at how data is stored, how it is shared across entities, and how we can build systems that are not only secure but also resilient to attacks. The goal of this work is to ensure that the information collected by our devices is used to improve our lives, not compromise our privacy. In essence, this thesis seeks to build a foundation for a future where the interconnectedness of devices enhances our world without endangering our digital safety.

Abstract

The Internet of Things (IoT) is revolutionizing the way we interact with the world around us, connecting everyday objects to the Internet for collecting and exchanging data. It offers immense potential for enhancing efficiency, convenience, and technological innovation. However, the rapid expansion of IoT devices also introduces significant security and privacy challenges. These challenges arise because of the diverse and ubiquitous nature of IoT devices with limited processing capabilities and open networks, making them vulnerable to cyberattacks, data breaches, and unauthorized access.

This thesis is based on novel security and privacy solutions for the unique constraints and requirements of the IoT ecosystem. First, we explore the application of blockchain technology as a foundational measure to secure and decentralize IoT systems. Recognizing the scalability challenges inherent in existing blockchain solutions when faced with the voluminous data generated by IoT devices, we propose an architectural framework that employs sidechains and offline data storage. Extending this work further, we propose a hierarchical blockchain-based framework specifically designed for the healthcare sector. This framework addresses the issue of securely storing and sharing sensitive medical data without compromising its integrity and privacy. It optimizes data storage solutions and employs fog nodes for computational services to ensure efficient data management.

Next, we explore the privacy aspect of IoT by leveraging Federated Learning (FL) to protect user data. We present an FL-based framework that enables collaborative training of a deep neural network on decentralized data, effectively mitigating privacy risks associated with data sharing. We further expand our discussion to decentralize the FL framework by eliminating the need for a central server for data aggregation. Lastly, we develop a sustainable and secure framework for IoT-based applications by integrating the characteristics of both blockchain and FL. This mechanism enables the dynamic optimization of blockchain performance parameters by training a decentralized FL model, thereby preserving the distributed nature of blockchain while enhancing its efficiency. Through theoretical analysis and practical implementation, we validate the feasibility and performance of proposed frameworks for different applications of IoT.

Keywords: Internet of Things (IoT), Blockchain, Federated Learning (FL), Data Security, Decentralization, Privacy Preservation.

List of Publications

Journals

1. **V. Agarwal** and S. Pal, “HierChain: A Hierarchical-Blockchain-Based Data Management System for Smart Healthcare,” in *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 2924-2934, 15 Jan.15, 2024, doi: 10.1109/JIOT.2023.3295847.

2. **V. Agarwal**, O. Ardakanian, and S. Pal, “A Robust and Privacy-Aware Federated Learning Framework for Non-Intrusive Load Monitoring,” in *IEEE Transactions on Sustainable Computing*, doi: 10.1109/TSUSC.2024.3370837.

3. **V. Agarwal** and S. Pal, “Towards a Sustainable Blockchain: A Peer-to-Peer Federated Learning based Approach,” in *ACM Transactions on Internet Technology (TOIT). (minor revision)*

Conferences

1. **V. Agarwal** and S. Pal, “Blockchain meets IoT: A Scalable Architecture for Security and Maintenance,” *IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2020, pp. 53-61, doi: 10.1109/MASS50613.2020.00017.

2. **V. Agarwal** and S. Pal, “FLEC: Federated Learning for Cloud/Edge-Based Smart Industry via Batch Normalization,” *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2023, pp. 1576-1581, doi: 10.1109/ICCWorkshops57953.2023.10283541.

Book Chapters

1. **V. Agarwal** and S. Pal, “Securing IoT with blockchain: Challenges, applications, and techniques,” *Securing IoT and Big Data*, pp. 15–38, 2020.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Internet of Things

The Internet of Things (IoT), evolving rapidly in the industrial and academic areas is a promising technology aiming to revamp human life completely. The IoT is a heterogeneous network of interconnected devices with the capabilities to actuate, sense, store and communicate, thereby connecting real-life to the digital world. IoT can help in enhancing the quality of products and gain profits through autonomous operations and increased product throughput. In today's era, an IoT device has varying functionalities ranging from a wearable device to complete industrial automation. IoT has the power to transform a home into smart home, energy grids to smart energy grids, a city to smart city, and much more. These smart environments represent a significant evolution in how we interact with our surroundings, driven by the integration of IoT. They utilize interconnected devices and systems to enhance operational efficiency, improve resource management, and elevate the quality of services provided to users. For instance, a smart home uses IoT to connect various devices and appliances in a household to the Internet, allowing them to be controlled remotely via smartphones or other networked devices. This can include lighting, heating, air conditioning, TVs, computers, entertainment audio and video systems, security cameras, and alarms. Smart homes can improve energy efficiency, convenience, and security. On a larger scale, a smart city employs similar IoT technologies to optimize urban functionality and promote sustainable development. This includes sophisticated management of city services such as transportation, utilities, and public safety through real-time data analysis and connectivity.

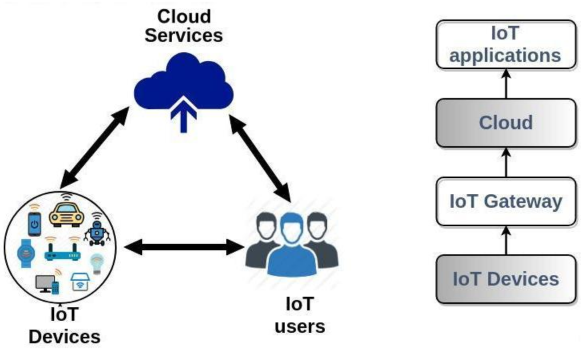


Figure 1.1: Traditional IoT services.

Basically, IoT systems are made up of four layers as shown in Figure 1.1.

- The first layer is made up of the battery-controlled, resource-constrained, small devices with attached actuators and sensors. These devices sense and collect data, perform light-weight processing on it if required and then communicate the data into the network.
- The second layer, the IoT gateway, is a powerful processing device which aggregates data from various devices and delivers this data further to the cloud services.
- The next layer is the Cloud which is used as a storage server for all the collected sensor data. The data can be analyzed as well as processed in these cloud servers for decision making.
- The fourth layer is the platform layer where the raw data is used for business services and user applications by applying data analytics, statistical methods, machine learning and other complex algorithms to extract corrective results from the data.

1.2 Applications of IoT

The applications of IoT are in diverse sectors, offering numerous opportunities for efficiency, automation, and connectivity. The key applications of IoT and its impact across various domains are outlined as follows:

- **Smart Homes:** IoT technology is central to the development of smart homes, where appliances, heating, lighting, and multimedia devices are interconnected and remotely controlled via the internet. These smart systems offer homeowners convenience, energy efficiency, and security. For example, thermostats can adjust the temperature based on the homeowner's habits, while smart locks provide enhanced security through remote monitoring and control.
- **Healthcare:** In healthcare, IoT devices play a critical role in monitoring patient health, managing chronic conditions, and enhancing drug delivery systems. Wearable devices can track vital signs, sending real-time data to healthcare providers for timely intervention. Furthermore, IoT facilitates remote patient monitoring, reducing the need for hospital visits and allowing patients to receive care in the comfort of their homes.
- **Industrial Automation:** Sensors and smart machines in factories enable real-time monitoring of production lines, predictive maintenance, and inventory management, leading to increased efficiency and reduced operational costs. Moreover, IoT facilitates the creation of digital twins, virtual replicas of physical devices that can be used for simulation and analysis.
- **Agriculture:** In agriculture, IoT applications include precision farming, livestock monitoring, and smart irrigation systems. Sensors can measure soil moisture,

nutrient levels, and weather conditions, enabling farmers to make informed decisions about planting and irrigation. Livestock monitoring devices track the health and location of animals, improving herd management.

- **Smart Cities:** IoT technology is also helpful in developing smart cities, enhancing urban infrastructure, transportation, and public services. Smart sensors manage traffic flow, reduce energy consumption through intelligent lighting, and monitor environmental pollution. Additionally, IoT devices support waste management by optimizing collection routes and schedules.

- **Energy Management:** IoT devices play a significant role in energy management, both in residential and industrial settings. Smart grids use IoT technology to balance electricity supply and demand, improve grid reliability, and support the integration of renewable energy sources. Smart meters provide real-time information on energy usage, encouraging consumers to adopt energy-saving behaviors.

These applications highlight the versatility and transformative potential of IoT across multiple sectors. However, the data collected by IoT devices is not entirely secure and tamper-proof which may pose a threat to the privacy and confidentiality of users [1]. Moreover, as the number of IoT devices are increasing, the traditional cryptographic methods are no longer able to perpetuate data integrity against the threats and risks they are exposed to.

1.3 Security and Privacy Challenges in IoT

In the context of the IoT, the concepts of security and privacy are critically important yet distinctly different, each addressing separate concerns that arise from the interconnected nature of devices and the vast amounts of data they handle.

1.3.1 Security in IoT

Security in IoT refers to the measures and techniques implemented to protect IoT devices and networks from malicious attacks, unauthorized access, and other cyber threats. The goal of security is to ensure the integrity, confidentiality, and availability of data as it is collected, processed, transmitted, and stored by IoT devices and systems. Key aspects of IoT security include:

- Integrity: Protecting data from being altered or tampered with by unauthorized parties.

- Confidentiality: Ensuring that sensitive information is accessible only to those who are authorized to view it.

- Availability: Guaranteeing that devices and data are accessible to authorized users when needed, preventing disruptions to IoT services.

Security challenges in IoT involve protecting the devices themselves from being hijacked or compromised, securing the data they generate from interception or theft, and safeguarding the networks they operate on from attacks like denial of service (DoS).

1.3.2 Privacy in IoT

Privacy in IoT, on the other hand, focuses on the rights and expectations of individuals regarding their personal information that is collected, stored, and used by IoT devices and services. It involves considerations about what data is collected, how it is used, who has access to it, and how individuals can control their own information. Privacy concerns in IoT include:

- Consent and Choice: Ensuring that individuals are informed about what data is collected and have the choice to consent to its collection and use.
- Data Minimization: Collecting only the data that is necessary for the intended purpose, and not retaining it longer than needed.
- Transparency and Accountability: Making it clear to users how their data is used, who it is shared with, and implementing measures to hold entities accountable for protecting user data.

Privacy challenges in IoT revolve around the potential for pervasive surveillance and data collection without explicit consent, the risk of sensitive information being shared or sold without users' knowledge, and the difficulty users may face in managing their privacy settings across multiple devices and platforms.

Several incidents and attacks to IoT devices have been devised in the past few years which makes it difficult to trust them. In 2018, it was discovered that smart assistants like Alexa and Google home were squandered by hackers to snoop on users without their knowledge. A smart refrigerator sent a large number of email spams in 2014 without the awareness of owners. The large data generated by IoT devices, open wireless channels and complexity of IoT systems further adds up to their security risks. Some of the typical challenges faced by IoT networks are described as follows:

1. **IoT Malware:** IoT devices are exposed to various security problems which can compromise the services provided by them. One such risk that makes them vulnerable is the attack by malware which are created by hackers for the purpose of stealing data, damaging devices or simply causing a mess. A malware is a malicious software which can be of many types such as, viruses, worms, ransomware, Trojans etc. Wang et al. [2] categorized IoT malware into two types based on how the IoT devices are corrupted, one is to exploit the IoT devices through their unfixed or imprudent vulnerabilities; while the other is to use brute force methods to infect devices. The most common IoT vulnerabilities are buffer overflow, poorly implemented encryption, unencrypted services and denial of service attacks. One of the reasons that makes Brute force attack successful is the use of weak passwords.

2. **Device Updates Management:** Although the IoT devices when purchased could contain the latest security software, but it is not possible to avoid any new risks or attacks which may arise later. Therefore, keeping the IoT devices updated to the latest software becomes a necessity instead of an option. However, IoT updates are still not delivered as efficiently like smartphones and computers and the manufactures of IoT devices pay less attention to the security risks. The updates should be delivered automatically because the customers feel it is not their duty to do so and we cannot even expect the users to stay on top of every software update. For the IoT devices which are deployed in remote areas or are difficult to access, over-the-air updates is an option, but it can also pave a path for the hackers to use malicious software for updating IoT devices.

3. **Data Privacy Issues:** IoT devices can collect detailed information about individuals' habits, health, and preferences, often without explicit consent. Users may not be fully aware of the extent of data collection or how to control it, raising concerns about personal privacy and autonomy. Once collected, data can be shared with or sold to third parties, including advertisers and data brokers. Without strict privacy controls, there is a risk that this data will be used in ways that users have not agreed to, potentially leading to unwanted surveillance and targeting. Moreover, even if data is collected with consent, the way it is stored and transmitted can be insecure, leading to unauthorized access and breaches. Ensuring that data is encrypted and securely managed throughout its lifecycle is a significant challenge.

4. **Manufacturing Defects:** IoT device manufacturers are more eager to release their product in the market than testing for possible security leaks. This is among one of the major security issues of IoT. Without any standard security mechanism, IoT manufacturers will keep producing devices that are vulnerable to attacks. For example, a smart refrigerator can compromise a user's login credentials of their Gmail-IDs, most fitness bands available today remain visible to be paired by Bluetooth even after they have been paired by the user. Moreover, these resource constraint devices cannot support heavy methods for security because complex encryption and decryption methods cannot be performed fast enough by them to transfer data in real-time securely.

5. **Security of Massively Generated Data:** With the increase in IoT devices each day, the data generated by them is also increasing. Devices like smart thermostats, smart TVs, lightning systems, speakers etc. constantly produce data leading to a problem of processing, transmitting, and storing it securely. This data should be kept encrypted or anonymous before it is stored or sent to the unsecure cloud servers so that the personal information of users is not revealed. Disposing of cached or unwanted data is also a security challenge so that it does not fall into wrong hands. Moreover, data integrity should also be maintained by using digital signatures or checksums. It is a common practice nowadays that the service providers sell or

share this data with other companies without the permission of users violating their privacy and trust. Therefore, ensuring the privacy of user's crucial data has become a serious concern to protect them from hackers.

6. **Authorization and Authentication Issues:** Authentication is the process of identifying whether the user is the same person/entity he is claiming to be. Authorization means checking whether the user has the required permission for the data or resources he wants to access. In machine-to-machine communications, authentication and authorization should be provided by the use of cryptographic techniques [3]. Many IoT devices fail to authenticate other devices properly because the basic means of authentication is the use of passwords and users can use weak passwords and even default passwords which are easily predictable. Authorization mechanisms available currently are mostly centralized and they do not provide efficient, effective, manageable, and scalable mechanisms to control and verify the access permissions of the users accurately. The increasing number of IoT users and data produced by them makes it even more necessary to devise scalable and manageable authorization techniques.

7. **Botnet Attacks:** A botnet is a cluster of devices connected through the internet whose security has been compromised by an attacker. Hackers create botnets by using malware to corrupt internet-connected devices and then controlling them through a server. If one device has been compromised on a network, all the other connected devices pose a threat of infection. One such example which affected IoT devices is the Mirai botnet (2016) [4]. Mirai botnet lead to a denial of service attack with 620 gigabit-per-second of traffic lead by a group of IoT devices, security cameras, and routers. At almost the same time, another DDoS attack targeted the French webhost which used the Mirai malware with a peak rate of 1.1 Tbps traffic. To avoid the common vulnerabilities, it is necessary to adopt some standard security mechanisms to ensure that IoT devices are not used as zombies. Proper intrusion detection systems should be installed because IoT software codes might become obsolete when not updated for long periods of time.

1.4 Motivation

Traditional security and privacy solutions often fail to address the unique complexities of IoT applications, leading to a need for novel approaches that can safeguard sensitive data and ensure robust protection against cyber threats. Numerous solutions have been proposed to mitigate these challenges, ranging from advanced encryption techniques and secure communication protocols to privacy-preserving data aggregation methods as shown in Figure 1.2. Despite these advancements, the decentralized nature of IoT networks along with the vast amount of data generated by IoT devices, continues to pose significant security and privacy risks. These include vulnerabilities to data breaches, unauthorized access, and the potential for misuse of personal information. These issues are critical because they pose significant threats to the integrity and functionality of IoT systems and have far-reaching consequences across interconnected networks. For instance, IoT malware and botnet attacks not only compromise individual devices but can also escalate to disrupt entire networks, making them critical concerns that necessitate robust and immediate solutions. Moreover, the increasing integration of IoT devices into sensitive and personal aspects of users' lives is leading to privacy violations and data breaches.

In response to these challenges, this thesis is motivated by the potential of integrating blockchain technology and Federated Learning (FL) as a comprehensive solution for enhancing IoT security and privacy. Blockchain, with its decentralized architecture and tamper-resistant ledger, offers a robust framework for securing data transactions and ensuring data integrity across distributed networks. Its inherent characteristics—such as decentralization, transparency, and immutability—make it an ideal candidate for addressing security vulnerabilities inherent in IoT applications. Conversely, FL presents a novel approach to privacy-preserving data analysis. By enabling machine learning models to be trained directly on devices without the need to centralize data, FL mitigates traditional privacy concerns associated with data aggregation and processing. This localized approach to data handling not only enhances user privacy but also reduces the latency and bandwidth requirements associated with transmitting large volumes of data to centralized servers. However, the integration of blockchain and FL within IoT ecosystems is not without its challenges. The scalability of blockchain, when handling the high throughput of IoT data, and the efficiency of FL, particularly in terms of computational and communication overheads, present significant obstacles. These challenges are compounded by the need to maintain a delicate balance between security, privacy, and operational efficiency in IoT systems. Acknowledging these complexities, our motivation evolved to not only explore the integration of blockchain and FL as a

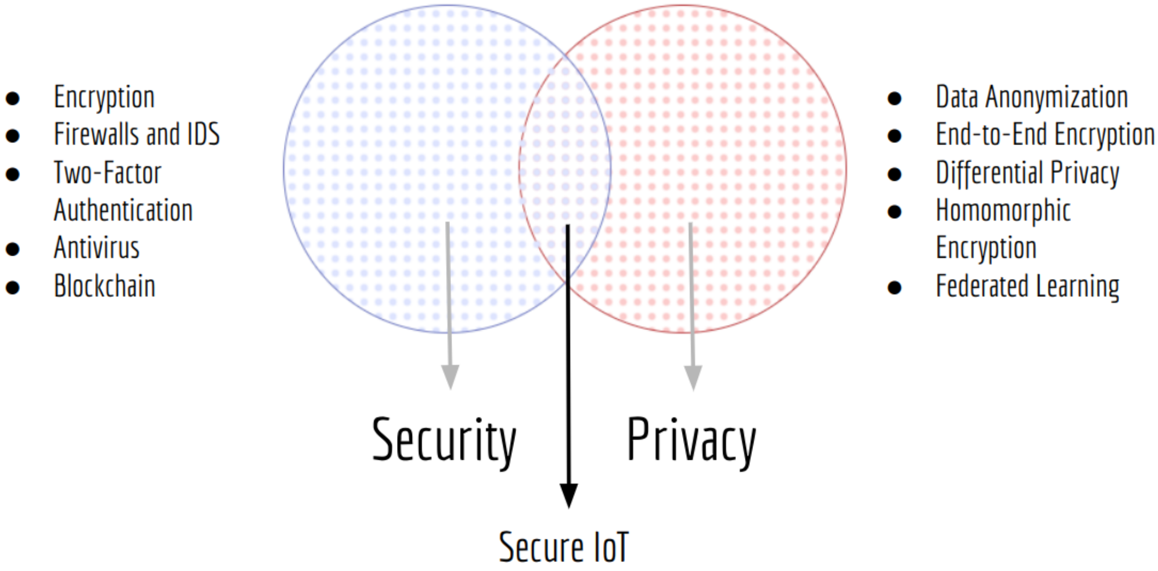


Figure 1.2: Techniques for a Secure IoT

robust solution for IoT security and privacy but also to address the inherent challenges associated with their implementation in IoT contexts. This thesis aims to navigate these hurdles by proposing innovative approaches that enhance the scalability of blockchain and the efficiency of FL, ensuring their viability as integrated solutions for IoT applications. We describe these technologies in more details in the next sections.

1.5 Blockchain and IoT

A blockchain is basically a record of blocks which contain information secured via "digital signatures". It is a distributed ledger used to save information about transactions between two parties in an incorruptible way. The distributed ledger technology offers an advanced way of identity management through public and private keys instead of passwords which could be cracked easily. Each block contains a hash of the previous block which means that if an intruder wants to modify one block, he will have to modify all the preceding blocks, thus making it permanent. Blockchains offer a secure and transparent way to exchange information as every node in the blockchain has a local copy of the blockchain and any user can verify the identities himself.
The transactions in blockchain are validated by certain nodes called as "miners". The miners use a consensus mechanism such as proof-of-work to validate the transactions. The transaction is valid only if the hash of a block with reference to the hash of its preceding block is correct. The validated transaction is then combined with other transactions and once the consensus is reached by a majority nodes, the new block is added to the existing blockchain network. The existing local copies of the blockchain are then updated on all the nodes with the new block. When two parties wish to exchange services or information, blockchain uses smart contracts similar to paper contracts for stating the terms and conditions of the deal. Smart contracts are digital, self-executable codes that run automatically when some predefined conditions are met. Smart contracts stored on the blockchain are helpful to set-up trustworthy relationships among parties without the need of any third party. Ethereum is a public, open source, decentralized blockchain-based platform featuring the functionality smart contracts. Smart contracts have applications in the field of healthcare for health insurances, industries for financial agreements, real estate for documents of property owner, etc. There are mainly three kinds of blockchain namely, public blockchain, private blockchain and consortium blockchain.

1.5.1 Public Blockchain

Public blockchains are open to everyone for accessing, reading and writing to the ledger. Anyone can join the blockchain network and verify the transactions to be added to the blockchain. Decision making in such an open and decentralized system is managed by consensus methods such as proof of stake (POS) and proof of work. These blockchains are completely decentralized without intervention of any central authority to modify the ledger or smart contracts which makes it reliable against any single point of failure. Also called as permissionless blockchains, public blockchains are used mainly for exchanging and mining cryptocurrencies such as Bitcoin and Ethereum.

1.5.2 Private Blockchain

Private blockchains, also known as permissioned blockchains, have restricted access on who can participate in the network. Only the users chosen by the respective authority or the network administrator have the permission to read or write in the ledger. Such blockchains are used mainly by centralized organizations who want to store their transactions privately in a closed network. In private blockchains, identity of users are known to everyone but transactions can only be viewed by those with the appropriate permission. Such blockchains are more efficient and have higher throughput due to finite number of users. Some examples of permissioned blockchains are Hyperledger, Multichain etc.

1.5.3 Consortium Blockchain

Consortium blockchains are basically a hybrid of public and private blockchains where some nodes are in-charge of the consensus mechanisms while other nodes have access to the transactions. Unlike private blockchains, these blockchains are governed by more than more one organization instead of a single entity. In these blockchains, transactions can be made and blockchain can be edited/reviewed only by chosen members of the consortium (federation). This approach has the benefits of a private blockchain and is very efficient since all the parties involved work together for the overall benefit of the network. These blockchains are used by governments and central banks to supply chains. Some examples are Energy Web Foundation (EWF) and R3 blockchains.

1.5.4 Challenges in Integration

Although blockchain can be used to resolve major security risks associated with IoT, it faces several challenges while integration:

- **Scalability and Performance Issues:** Blockchain's decentralized nature, while enhancing security and trust, introduces scalability challenges, especially when integrated with IoT systems that generate vast amounts of data. Traditional blockchain networks, such as those used for cryptocurrencies, can handle only a limited number of transactions per second, which cannot handle the needs of large-scale IoT deployments. As the number of IoT devices and the data they generate continue to grow exponentially, the blockchain's ability to process and record transactions efficiently becomes a critical concern. This scalability issue can lead to increased transaction processing times and higher costs, potentially hindering the performance and adoption of blockchain in IoT applications.

- **Data Privacy and Protection:** While blockchain provides a secure and immutable ledger, ensuring the privacy of the data recorded on the blockchain is a significant challenge. Blockchain's transparency means that once data is added to the ledger, it is visible to all participants in the network. This characteristic poses a privacy risk for IoT applications that handle sensitive or personal data. Ensuring that data stored on the blockchain is protected and complies with privacy regulations (e.g., GDPR) requires innovative approaches, such as the use of private or permissioned blockchains, advanced encryption techniques, or zero-knowledge proofs, to balance transparency with privacy.

- **Latency Issues:** The consensus mechanisms of blockchain, such as Proof of Work (PoW) or Proof of Stake (PoS), ensure network security and data integrity but also introduce latency in transaction processing and data recording. In IoT scenarios where real-time or near-real-time data processing is crucial—for instance, in autonomous vehicles, health monitoring systems, or industrial automation—this latency can be problematic. The delay in confirming transactions and recording data on the blockchain may not meet the stringent latency requirements of certain IoT applications, affecting their functionality and user experience.

- **Energy Consumption and Efficiency:** Blockchain networks, especially those utilizing energy-intensive consensus mechanisms like PoW, consume significant amounts of energy. The environmental impact of such high energy consumption becomes a concern when scaling blockchain applications for IoT. Moreover, many IoT devices are constrained by battery life and operate in energy-efficient modes. The added energy demands of blockchain operations could pose challenges in terms of the sustainability and efficiency of IoT devices integrated with blockchain, necessitating the exploration of more energy-efficient consensus mechanisms or hybrid models that minimize energy consumption while maintaining security and integrity.

1.6 Federated Learning and IoT

FL allows IoT devices to learn a shared model while keeping all the training data on the device, only exchanging model updates with a central server or among each other. This approach significantly enhances data privacy, as raw data never leave the device, reducing the risk of personal data exposure as shown in Figure 1.3. However, this integration is not without its challenges. These challenges primarily revolve around ensuring the integrity and effectiveness of the learning process in the face of potential security threats and the complexity of IoT data ecosystem.

- **Malicious IoT Devices:** In an FL environment, IoT devices collaboratively contribute to the development of a shared model by processing data locally and sending model updates to a central aggregator or directly to each other. This distributed approach introduces the risk of malicious IoT devices participating in the learning process. Such devices could intentionally send incorrect or manipulated updates, aiming to degrade the model's performance or bias its outcomes. Ensuring

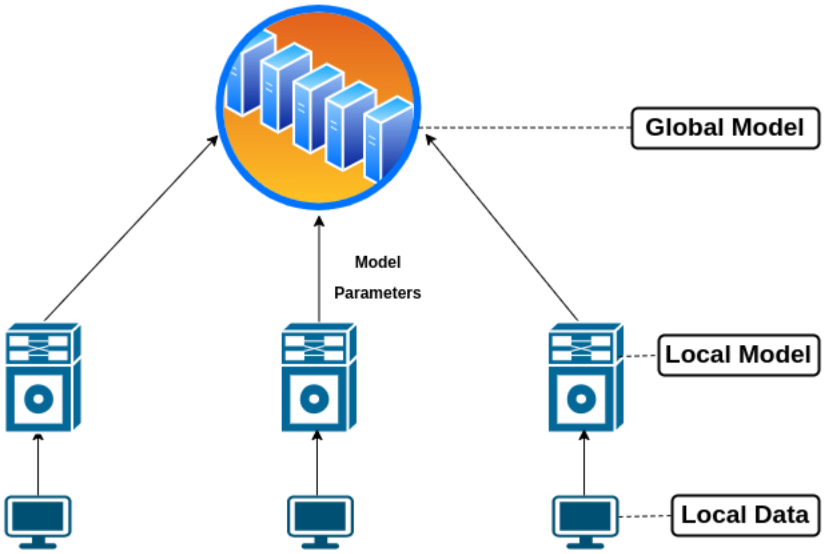


Figure 1.3: General framework of FL.

the reliability and trustworthiness of devices in such a decentralized setup is a significant challenge.

- **Model Poisoning:** Similar to the challenge posed by malicious devices is the threat of model poisoning. In this scenario, adversaries manipulate the learning process by introducing harmful model updates, which, when aggregated, compromise the model's integrity. Detecting and mitigating model poisoning attempts require sophisticated anomaly detection and filtering mechanisms to ensure that only legitimate updates contribute to the model, without compromising the decentralized ethos of FL.

- **Vulnerabilities in the Central Server:** Although FL significantly reduces the need for centralized data storage, a central server is often still involved in aggregating model updates and distributing the updated model to participating devices. This server becomes a critical point of vulnerability, where security breaches or system failures can disrupt the entire learning process. Protecting this server and ensuring its resilience against attacks and operational failures is essential for the successful integration of FL and IoT.

- **Non-IID Data:** One of the intrinsic characteristics of IoT environments is the generation of non-IID (not independently and identically distributed) data. IoT devices, deployed across varied environments and serving diverse functions, generate data that reflect these differences, leading to significant disparities in the distribution of data across devices. This heterogeneity poses a challenge for FL, as conventional machine learning algorithms assume data homogeneity for optimal performance. Developing learning algorithms that can effectively handle non-IID data, ensuring fair and accurate model training, is a critical area of focus in integrating FL with

IoT.

1.7 Research Problem and Objectives

This thesis aims to explore and enhance the domains of security and privacy within IoT by utilizing progressive technologies and methodologies to safeguard IoT ecosystems. In order to fulfill this goal, we aim to achieve the following key objectives:

How can we develop an architectural framework that enhances data security in IoT systems using blockchain technology? This objective focuses on overcoming the challenges of scalability and efficiency in blockchain for real-time data processing in IoT. By introducing a novel architectural framework that incorporates the concept of sidechains and offline data storage, we aim to ensure the secure handling of vast data volumes generated by IoT devices. This exploration, detailed in Chapter 3, also extends to the development of 'HierChain,' a hierarchical blockchain system tailored for the security of sensitive healthcare data, integrating differential privacy techniques to further protect individual privacy.

Can we devise innovative strategies for ensuring data privacy within IoT-based ecosystems? This question investigates the application of FL-based approaches to share model updates based on local training of private IoT data. We aim to develop and detail techniques that maintain user privacy, leveraging meta-learning to address data heterogeneity challenges. The mechanisms and strategies devised are comprehensively discussed in Chapter 4.

How might we enhance privacy in decentralized IoT environments through peer-to-peer networks? This objective seeks to eradicate central points of trust in FL and improve data privacy by deploying a robust aggregation mechanism that protects clients from adversarial updates and malicious attacks. By simulating the efficiency of our algorithm for IoT-based smart applications, we explore the viability of this approach in enhancing privacy within decentralized IoT contexts, as elaborated in Chapter 5.

What integrated approach can combine the strengths of blockchain and peer-to-peer FL to create a sustainable and secure infrastructure for IoT applications? This research question aims to develop a comprehensive solution that integrates the security features of blockchain with the privacy-preserving characteristics of peer-to-peer FL. By implementing load balancing across multiple blockchains based on several optimization parameters, we design a sustainable and secure framework suitable for diverse IoT applications. The methodologies and outcomes of this integrated approach are thoroughly examined in Chapter 6.

By addressing these objectives, this thesis contributes to the advancement of IoT, proposing scalable, efficient solutions that do not compromise system performance or user privacy. Each chapter builds upon the preceding to articulate a coherent and detailed examination of these critical issues, concluding in Chapter 7, which provides the contributions made and gives insights into future directions for research in this rapidly evolving field.

1.8 Thesis Contributions and Organisation

The major contributions of this work can be summarized as follows:

1. Proposed an architectural framework that utilizes sidechains and offline data storage to address the scalability challenges faced by blockchain systems in handling vast amounts of IoT data. This framework ensures scalability without compromising on speed and efficiency.

2. Introduced a hierarchical blockchain-based system for data storage and sharing in healthcare. It optimizes data storage, classifies health data by sensitivity levels, uses differential privacy techniques, and employs fog nodes for computational services. This system enhances the efficiency and security of medical data management in IoT environments reducing latency by more than 70%, and a throughput increase to handle upwards of 100 transactions per second for high-volume scenarios.

3. Proposed a robust and privacy-preserving FL framework for data sharing of smart meters in the energy domain. This framework incorporates meta-learning to manage data heterogeneity and maintains user privacy, achieving accuracy comparable to centralized models. Specifically, the accuracy of the model showed a drop of only 8.7% in case of malicious clients with the attack detection mechanism demonstrating robustness against increasing security threats.

4. Proposed a novel peer-to-peer FL approach that eliminates central points of trust and strengthens data privacy. This method leverages the Bidirectional Encoder Representations from Transformers (BERT) architecture, and a decision strategy based on similarity scores and model update accuracy to provide protection against adversarial attacks on sensitive IoT data. The success rate of poisoning attacks decreased to below 0.2 (20%) after 100 training rounds in the proposed system, compared to approximately 0.4 (40%) in traditional peer-to-peer FL and around 0.6 (60%) in standard FL validating the enhancement in security.

5. Introduced an approach that combines peer-to-peer FL with blockchain for a sustainable and secure data-sharing network. This strategy transforms blockchain nodes from passive components to active participants in the optimization process, creating a responsive and self-optimizing blockchain ecosystem that improves transactional efficiency and reduces energy usage. Specifically, energy usage was

reduced by up to 30% over traditional methods due to the efficient load distribution across nodes, which minimized redundant processing and energy expenditure.

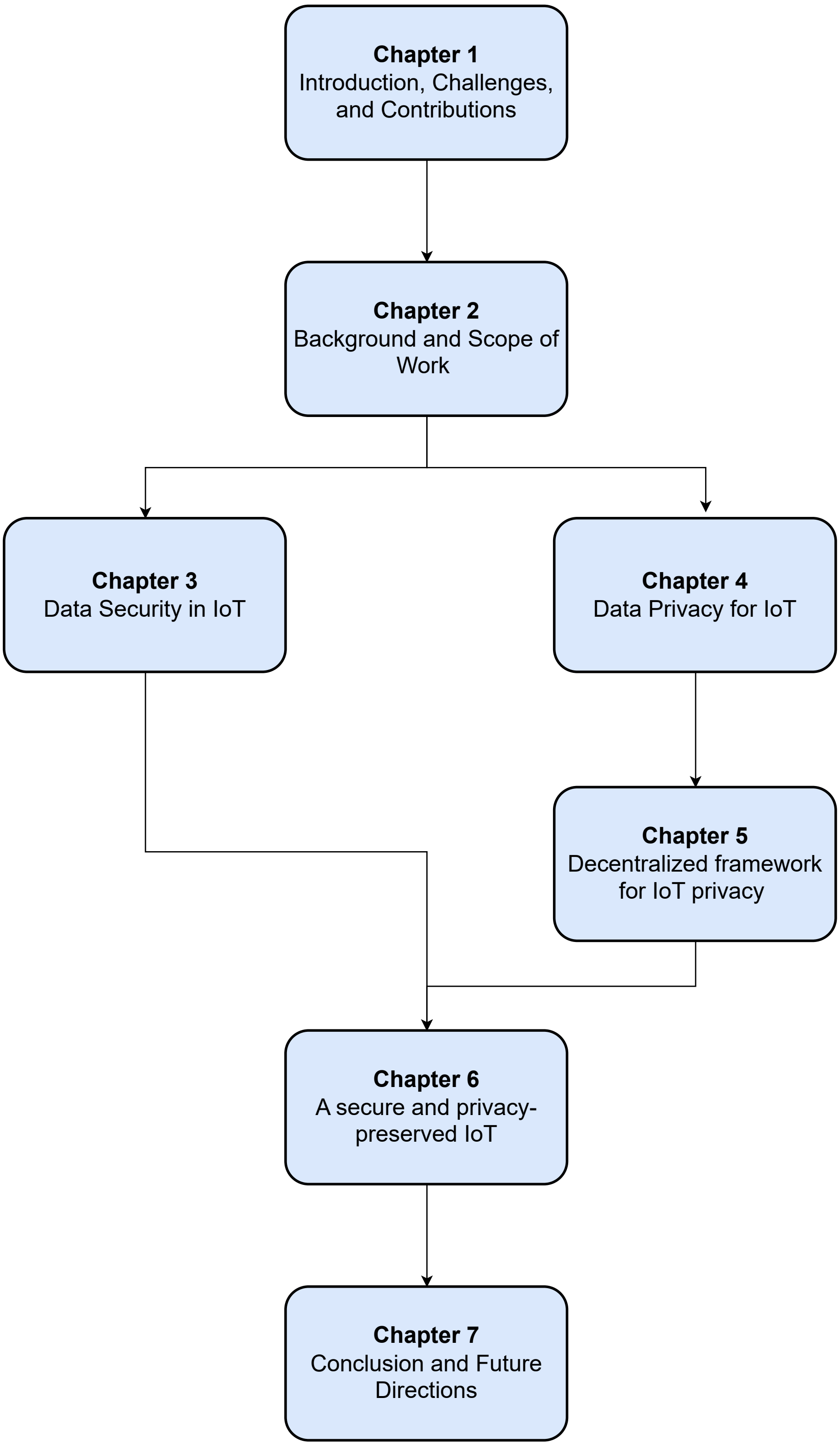

Chapter 1
Introduction, Challenges, and Contributions
Chapter 2
Background and Scope of Work
Chapter 3
Data Security in IoT
Chapter 4
Data Privacy for IoT
Chapter 5
Decentralized framework for IoT privacy
Chapter 6
A secure and privacy-preserved IoT
Chapter 7
Conclusion and Future Directions

Figure 1.4: Structure of the Thesis

Figure 1.4 shows the organization of thesis and how the chapters are related to each other. Here, CHAPTER 2 provides a comprehensive literature survey of the existing works in the domain of providing security and privacy to IoT-based applications. Additionally, it discusses the challenges and limitations of related literature and identifies the research gaps. CHAPTER 3 introduces a novel approach that addresses the significant challenge of security in IoT systems. Specifically, we focus on using blockchain technology to address this challenge for a secure IoT. CHAPTER 4 expands the discussion to privacy concerns in IoT applications by using innovative strategies for data sharing within IoT-based ecosystems. CHAPTER 5 extends the approach of decentralizing the FL framework built in Chapter 4 using a peer-to-peer network aimed at eradicating central points of trust and enhancing data privacy. CHAPTER 6 builds on the frameworks designed in CHAPTER 3 and CHAPTER 5 to create a sustainable and secure infrastructure for IoT applications. Finally, CHAPTER 7 concludes with the challenges and contributions made in this thesis. This chapter ends with giving some highlights on the future direction of the work.

Chapter 2

Literature Review

2.1 Introduction

As IoT devices become more autonomous and intelligent, safeguarding them against potential device corruption and broader network implications has become critical. Several cryptographic measures and procedural strategies have been proposed to ensure confidentiality, integrity, and availability in IoT. These mechanisms safeguard confidentiality through encryption and assure integrity and authenticity via message authentication codes. However, IoT systems require not just these foundational security services but also a focus on the execution and optimization of these solutions [5, 6, 7]. The transition from Wireless Sensor Networks (WSNs) to internet-connected configurations has made it necessary to devise robust defense mechanisms, particularly where sensor nodes present vulnerabilities due to limited computational resources [8, 9]. Standards such as IEEE 802.15.4 offer guidelines to enhance security at physical and medium access control layers, adding layers of confidentiality, integrity, and availability to the system [10]. However, the existing threats and security countermeasures, have certain limitations when it comes to real-life applications [11]. The organization of the literature review in our thesis is depicted in Figure 2.1.

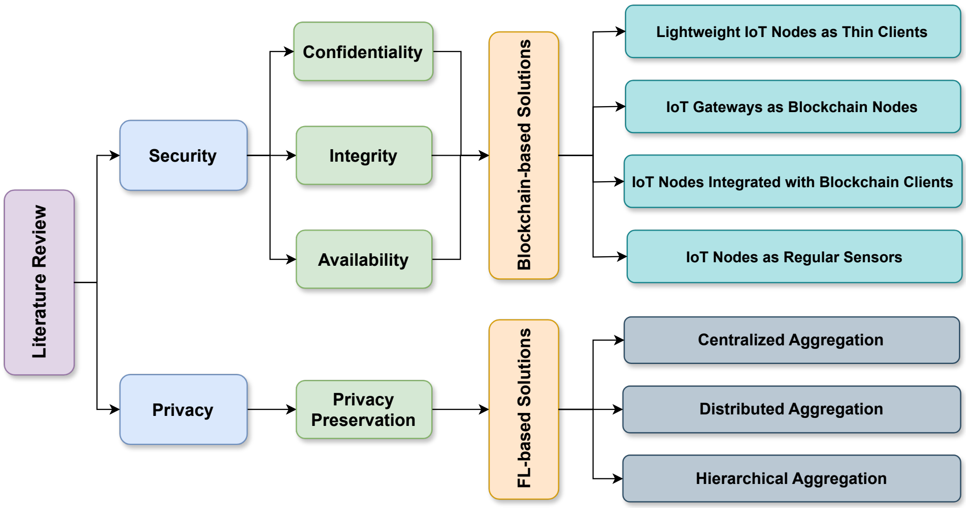


Figure 2.1: Categorization in the Literature Review.

2.1.1 Confidentiality in IoT

Data confidentiality is an important parameter for IoT implementations, particularly in commercial environments [12]. The existing confidentiality solutions are no longer adequate due to challenges related to the vast amount of data generated by IoT devices and the dynamic nature of data streams, which complicate access control. For confidential data, encryption whether through symmetric or asymmetric algorithms, is essential for securing communication channels, though the selection of an appropriate encryption method depends on specific IoT application requirements and system criticality [13]. Wireless IoT communications are susceptible to eavesdropping, potentially compromising data confidentiality and affecting the entire network. Mayer [14] identifies localization/tracking, storage, communication, and identification as areas particularly sensitive to confidentiality issues within IoT, contrasting them with sensors, devices, actuators, and processing aspects.

IoT solutions should implement security mechanisms allowing users to control access to predefined resources, a concept referred to as data ownership [15]. Current IoT technologies' approach to data security, including key management, may place undue strain on IoT resources, risking diminished system capabilities. To counteract this, lightweight cryptographic algorithms are proposed for resource-constrained IoT devices to safeguard data, thereby ensuring confidentiality [16]. Datagram Transport Layer Security (DTLS) is highlighted as a viable solution for confidentiality issues, offering end-to-end security at the application layer with minimal resource impact [5]. Cloud-based solutions that employ cryptographic measures relying on Public Key Infrastructure (PKI) and information flow control have been suggested to enhance data and communication confidentiality [17]. The concept of confidentiality is also linked with privacy by using techniques such as anonymizing data collection and minimizing data collection as a means to protect against unauthorized access.

2.1.2 Integrity in IoT

In the domain of the IoT, integrity comprises the safeguarding of IoT systems against physical damages and tampering, including the use of counterfeit components and sabotage [18]. The resilience and fault tolerance of an IoT system play a vital role in maintaining data integrity, highlighting the need for robust designs capable of withstanding various integrity threats. Sensor systems such as RFID networks are more vulnerable, as their components are often unmonitored, making them easy targets for data modification either in storage or transit. Commonly employed protective measures, such as read-write protections and authentication techniques face limitations due to the inherent resource constraints of many IoT devices, which also hinders the application of standard cryptographic methods [19].

To ensure the integrity of processes, safeguarding device operations, communications, and algorithm implementations are employed, critical for reliable data processing and service delivery [14]. Software integrity can be achieved through hardware isolation and the use of secure hardware for software attestation. However, they face challenges due to hardware-based attacks such as fault attacks if the system does not have protection against them. Identity management systems can help in authentication and resource control but they face deployment challenges related to scalability and management in IoT infrastructures [20]. Integrity maintenance also requires standardized procedures and extensive collaboration among IoT stakeholders to ensure data trustworthiness and system quality. Such collaboration is essential for developing common schemes that address the unique integrity and quality requirements of the IoT ecosystem [12].

2.1.3 Availability in IoT

Device availability stands as a basis for information networks, defining it as the “most important factor” [21]. Therefore, it becomes necessary for IoT systems to maintain operational persistence, uphold service levels, and meet application performance expectations. The concept of availability in IoT extends to ensuring ubiquitous access, suggesting that IoT devices must address key challenges including consistent availability, which consists of both hardware and software dimensions to satisfy user needs [22].

IoT devices might also be victim of attacks such as Denial-of-Service (DoS) due to an overwhelming number of requests resulting in a hindrance to their operation. DoS and Distributed Denial-of-Service (DDoS) attacks pose significant threats to the availability of IoT networks, disrupting device communications and access to network resources. Such attacks range from simple jamming to more complex strategies designed to exhaust resources or interrupt data transmission among IoT nodes [21]. These attacks not only jeopardize network resources and applications but may also lead to energy depletion in devices constrained by power. Current communication protocols endorsed such as the Constrained Application Protocol (CoAP) might not be able to handle such limitations [23]. Therefore, a shift towards Service Oriented Architecture (SOA) was proposed as a preventive strategy, emphasizing the efficiency of random sampling in monitoring network traffic to mitigate such threats [24]. Despite the array of defense mechanisms available for WSNs[114],there is the need of a foolproof solution against such risks. Intrusion-Detection System (IDS) aims to facilitate early detection of DoS activities, thereby preserving network operations for 6LoWPAN solutions [21]. Moreover, distributed architectures over centralized models can enhance service availability and eliminate single points of failure [25].

2.1.4 Privacy in IoT

The exponential growth of IoT in recent years has raised significant privacy concerns due to the increasing availability of data [12]. Currently, these devices lack the necessary safety mechanisms, leading to a need for enhanced protection of personal information related to individuals’ movements, habits, and interactions [26]. Moreover, inadequate measures for ensuring data confidentiality and integrity may lead to privacy breaches leading to unauthorized access to sensitive information, thereby jeopardizing the potential for IoT technologies' widespread acceptance [19, 12].

Roman et al. [27] highlighted potential risks from overreaching entities that could collect and share data without consent. To mitigate privacy violations, several key challenges, including the development of self-aware devices, ensuring data integrity, enforcing authentication, accommodating data heterogeneity, applying efficient encryption, securing cloud services, and establishing clear data ownership, governance, and policy management need to be addressed. Proposed solutions to these privacy concerns include adopting a "privacy by design" approach, allowing users dynamic control over their data and ensuring their preferences align with existing policies before granting data access. Another crucial factor is transparency enabling users to understand who manages and utilizes their data [28]. Alternative measures can be allowing users to opt out of untrusted sensor networks and introducing "privacy brokers" as intermediaries between users and the network [19]. However, technological measures alone are insufficient to address the privacy issues, necessitating consideration of the IoT's economic and socio-ethical dimensions. Therefore, there is a need to revise existing privacy regulations at both private and governmental levels, along with enhancing public awareness about how IoT devices manage data [29].

The challenge of distinguishing between regular data and Personally Identifiable Information (PII), which is legally protected complicates efforts to safeguard privacy in IoT-based systems [30]. The debate continues over whether IoT privacy should be regulated by governments or self-regulatory bodies, especially given the transboundary nature of IoT data [28].

2.2 Existing Solutions for IoT

The limitations of IoT devices such as their resource-constrained nature, limited processing power, and often weak security protocols make them prime targets for cyberattacks, data breaches, and unauthorized access [22]. Some of the techniques explored by various researchers to address the security and privacy concerns are described as follows:

2.2.1 Lightweight Cryptography

Lightweight cryptography addresses the need for secure encryption in IoT devices constrained by limited computational resources [31]. Unlike traditional encryption methods, which impose significant processing demands, lightweight cryptography is designed to offer robust data confidentiality with minimal computational overhead. This adaptation is crucial for the IoT ecosystem, where devices range from simple sensors to more complex systems, all of which require efficient data protection measures. These techniques are especially vital in scenarios where the energy consumption and processing capabilities of devices are limited, ensuring that security measures do not drain the device's critical resources [32]. While lightweight cryptography provides an effective solution for devices with limited computational resources, it often faces trade-offs between security strength and computational efficiency.

2.2.2 Secure Communication Protocols

The integrity of data exchange between IoT devices can be assured through the implementation of secure communication protocols [33]. Protocols such as Secure Sockets Layer/Transport Layer Security (SSL/TLS) and Datagram Transport Layer Security (DTLS) have been adapted to fit the IoT context [34], providing encrypted channels that secure data in transit. These protocols prevent eavesdropping and tampering by encrypting the data before transmission, ensuring that sensitive information remains confidential between the communicating devices. Their adaptation for IoT takes into account the need for efficiency and low overhead, making secure communication feasible even for devices with stringent resource constraints. However, these protocols do not inherently address issues of data privacy once data reaches its destination, leaving a gap in end-to-end data protection.

2.2.3 Identity and Access Management (IAM)

Traditional IAM frameworks are often ill-equipped to manage a wide variety of devices, from low-power sensors to advanced smart devices, thereby needing innovative solutions for IoT. These solutions can be based on certificate-based authentication and role-based access controls, ensuring that only authorized devices and users can access the network and perform actions based on their defined roles [35]. By addressing the unique challenges of device diversity and scalability, IAM solutions for IoT secure the ecosystem against unauthorized access and potential security breaches.

2.2.4 Intrusion Detection Systems (IDS)

Intrusion detection systems [36] are designed to continuously monitor the network for signs of malicious activity for proactive threat detection and mitigation. By analyzing device behavior patterns, network traffic anomalies, and communication patterns, IoT-specific IDS can effectively identify potential threats and initiate responses to protect the network. The development of IDS for IoT focuses on leveraging the vast amounts of data generated by devices to intelligently detect anomalies, ensuring the security and resilience of IoT networks against a wide range of cyber threats. However, the effectiveness of IDS often depends on the system's ability to accurately differentiate between normal and malicious behavior, which can be particularly challenging in IoT due to the diversity of devices and traffic patterns.

2.2.5 Privacy-Preserving Techniques

The protection of user privacy in the IoT, where devices collect extensive amounts of personal data, is an essential requirement [37]. Research in privacy-preserving techniques explores methods like differential privacy and FL to safeguard individual privacy [38].

Differential privacy introduces noise into the data analysis process, obscuring individual data points without significantly impacting the overall analytical outcomes. This approach prevents the identification of specific individuals from aggregated data. FL [39], on the other hand, enables machine learning models to be trained directly on devices, using data that never leaves its source. Together, these privacy-preserving techniques can help maintain the confidentiality and integrity of user data in the IoT landscape.

Amidst these solutions and their limitations, the integration of blockchain technology and FL offers a comprehensive approach to addressing both security and privacy challenges in IoT. Blockchain in IoT can prevent data tampering, facilitate secure peer-to-peer transactions, and enable transparent audit trails, thereby providing a secure infrastructure for IoT networks. On the other hand, FL minimizes data exposure and transmission, significantly enhancing user privacy by keeping sensitive information localized and reducing the risk of data breaches. We discuss the state-of-art works in these technologies in the succeeding sections.

2.3 Blockchain-based Solutions for IoT Security

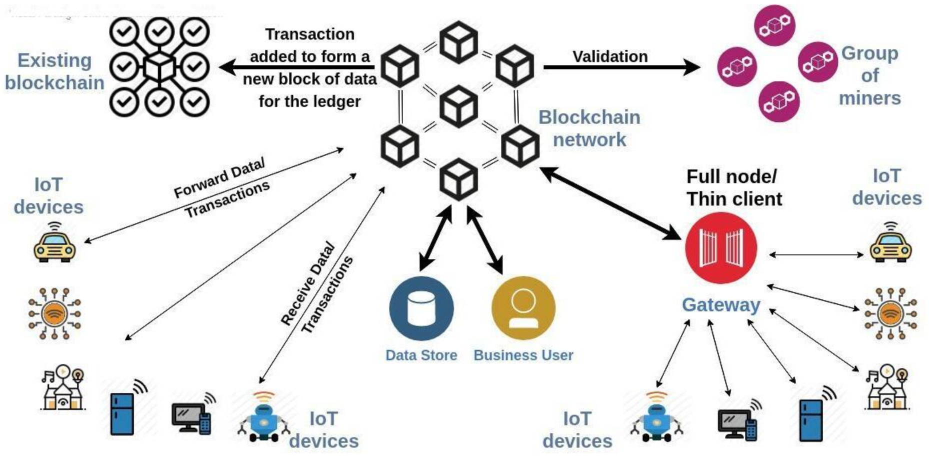


Figure 2.2: Blockchain integration methods.

Integration of blockchain with IoT gateways and end devices can be achieved through several methods, depending upon the available resources and capabilities of IoT hardware. Figure 2.2 shows how IoT nodes can communicate directly or through gateways to a blockchain network. Existing methods of integration strategies are summarized in the following subsections based on how IoT nodes are combined together with blockchain.

2.3.1 Lightweight IoT Nodes as Thin Clients

IoT end devices which are always on can function as a thin client in relaying the transactions to full blockchain nodes. In this case, IoT gateways can function either as full blockchain nodes or thin clients. Danzi et al. [40] have used a method of aggregation at the blockchain nodes to reduce the amount of data to be transferred to the IoT nodes.

Resource constrained IoT devices store only a small subset of data from the blockchain nodes (BNs) such as block headers when they synchronise with the BNs. IoT devices are connected via a base station to the blockchain nodes. The information is sent downlink periodically after every T seconds in an aggregated manner to the IoT devices from the blockchain nodes. The basic protocol used for blockchains is Ethereum protocol which uses Merkle-Patricia trees for providing proof of inclusion for the modified data.

An efficient Lightweight integrated Blockchain (ELIB) model is presented in [41] for meeting the requirements of IoT. The model is applied to a resource constrained smart home environment to verify its credibility. The ELIB model functions in three levels i.e., certificateless cryptography (CC), Distributed Throughput Management (DTM) and consensus algorithm. CC method is used to reduce the overhead associated with the security of new blocks. The new blocks creation is restricted by consensus algorithm and the DTM method takes control of altering specific system variables dynamically to make sure that the throughput of public blockchain does not vary considerably from the standard load.

Wang et al. [42] proposed a data sharing and storage model for decentralized access of data derived from the FISCO-BCOS blockchain. The proposed architecture consists of applications, IPFS and FISCO-BCOS. The working of this blockchain-based IoT network is as follows: Initially, when any IoT device wants to join a network, it provides the necessary information such as deice ID, service type, serial number etc. for identity authentication by the route chain. According to this identity authentication, the route chain allocates it to a specific set chain. The application then gathers data from the IoT device and encrypts it for storage at the IPFS. Therefore, the storage expenditure of the blockchain is reduced significantly because only IPFS file indexes are stored rather than the whole raw data.

Pohrmen et al. [43] proposed an IoT-blockchain architecture consisting of Application, Infrastructure, and Control layers. User specific application services like smart railways, smart grids, and smart campus etc., are provided by the Application layer. The Control layer consists of several connected mining nodes containing high computational powers. In this level, high security methods like asymmetric cryptography can be used (Global blockchain). The Infrastructure layer is made up of networks like WSN, IOT, LTE etc. and networking elements like routers, sensor nodes, switches etc. The network element uses local blockchain to transfer collected data to Control layer from the network. When the packet is forwarded, security methods should be implemented so that they are not modified by any intruder. Lightweight hash functions are used for resource constrained network elements.

A blockchain-based IoT system called BeeKeeper is proposed in [44]. In this threshold-based system, three parties function together namely a leader, it's devices and certain servers. Initially, the leader performs a BeeKeeper protocol which is a (t,n) threshold protocol having a complete control over several devices. The servers have the potential to do homomorphic computations over the data encrypted on the blockchain sent by the devices. However, if honest servers are more than n-t, the servers do not have the capability to learn from the data. A leader sends a query through transactions in the blockchain to the servers when it wants a result from the encrypted data. The sensors which are active can respond to the query with the appropriate encrypted data through blockchain transactions. The encrypted responses can be decrypted only by the leader because the decryption key is only available to the leader. The desired outcome can be obtained by the leader if it collects t or more correct responses. Then, the corresponding servers will gain some reward through the smart contract of the leader automatically.

Li et al. [45] proposed a scheme for storage and protection of IoT data using blockchain. Since IoT devices are low power devices with constrained resources, so edge computing is used in this method to carry out computations at the network edge on behalf of IoT devices and store the data in Distributed Hash Tables (DHTs). Before forwarding this data for storage to the DHT, the edge device announces this data by posting to the blockchain, a transaction, stating that this data belongs to a certain IoT device. This transaction is then verified by the blockchain and details like the ID of IoT device and storage address are recorded. Similarly, when data is required by an IoT device from the DHT, it will first place a transaction which will be authenticated by the blockchain. Certificateless cryptography (CC) is used as an authentication method for the identification of an IoT device. The drawback of CC is outlived through the use of blockchain's public ledger system which helps to broadcast the public key of any IoT device in a convenient manner.

NormaChain [46] is a transaction settlement scheme based on blockchains for IoT E-commerce. The regulators need certain information about user's transactions to keep a check on their legitimacy. This dilemma of choosing between privacy and legitimacy is solved by NormaChain through the use of searchable encryption (SE). SE is a method which allows the regulators to search for any particular keywords from the encrypted data without uncovering the whole plaintext. This helps in searching for any criminal manifests while keeping the privacy of the users intact. The authors propose a modified version of public key encryption with keyword search (PEKS) method which is decentralized without the need of any central supervision unlike the original scheme. This supervision is distributed among n parties and the target keyword can be searched only when all the n parties authorize it simultaneously. Therefore, this scheme tolerates corruptions upto collusion of n-1 parties. This scheme is designed to overcome the efficiency issue for E-commerce based IoT. A layered blockchain is used in which each layer has a different role to achieve scalability and more transaction speed. Experimentally, this scheme can achieve a supervision accuracy of 100% and the average transaction-per-second is 113.

Xu et al. [47] proposed a nonrepudiation service provisioning approach for Industrial IoT based on blockchain for evidence recording and providing service. Each service required by client is divided in advance into two non-executable sections. They are delivered through off-chain and on-chain channels separately after proper recognition of evidence submissions. Specifically, a major portion of the program is conveyed privately through off-chain channels after submitting valid evidence to the blockchain and the remaining part will be posted on the blockchain after recognition of valid evidence of the preceding steps.

Service verification method used in this chapter for Industrial IoT do not require complete program codes due to the lightweight cryptographic techniques and homomorphic hash solutions used. Moreover, fair automated smart contracts are used to aid the Industrial IoT clients and services providers in solving service disputes efficiently.

A blockchain-based analytical model for wireless IoT systems has been proposed in [48]. The network consists of two types of nodes namely, full function nodes (FNs) and IoT transaction nodes (TNs). TNs are the traditional IoT devices with limited resources and are of two kinds, idle TNs and active TNs based on whether they are currently transmitting data or not. FNs are the nodes which have sufficient memory and computational powers. They can implement blockchain protocols fully by taking charge of data storage, transaction confirmation and forming new blocks. FNs are connected through wireless communications with the TNs and via high data links with each other. Information between TNs is transmitted in the form of transactions and they are validated and written in blocks by FNs. Moreover, they have also deduced a probability density function (PDF) of signal-to-interference-plus-noise ratio for transferring data from an IoT to a full node. The security of the proposed model is analyzed under different attacks i.e., random FN, random link, and eclipse attack. The model proved to be valid under these attacks.

BPIIoT [49], is a light-weighted Blockchain-based platform to solve the problems of trust, security and island connection in the Industrial IoT construction process. BPIIoT consists of an on-chain as well as an off-chain network to manage latency and loads in the network. The on-chain network performs all the transactions such as, programmable licenses, digital signatures, etc. while the off-chain network can solve problems of complex processing, storage etc. Involvement of third parties is avoided by using SMPC (Secure Multi Party Computation) by the on-chain network. Data processing and computations are divided among several nodes which perform the given tasks without revealing the information. The IoT nodes, act as lightweight devices consisting of four layers: application, service, network and device layer. The service layer supports services for blockchain like smart contracts, asymmetric encryption and other general services like controller service, I/O interface etc. Network layer is used for communication between nodes and the blockchain. The device layer consists of actuators and sensors that are deployed in the given field. These IoT devices connect blockchain to the machines through a plug and play solution. IoT devices make it possible for the machine to post data to the blockchain, send and receive transactions in the blockchain.

2.3.2 IoT Gateways as Blockchain Nodes

IoT gateways can function either as a full blockchain node or a thin client. IoT gateway can route data and verify integrity of data in case it functions as a full blockchain node. If used as thin clients, IoT gateways store only few relevant data parts. Mehedi et al. [50] proposed a reliable and polished blockchain-IoT infrastructure which cultivates a breather from centralized systems and memory overhead while maintaining the privacy and security effectively. They used the standard IoT infrastructure along with decentralized blockchain technology for storing the data and accessing it. For integration of blockchain and IoT, they used Ethereum as blockchain platform and terminal devices enlisting the network technology. Whenever a request is made for storing a transaction, the proposed approach uses the distributed ledger which gets executed and stored on its own. To protect the users identity, the terminal devices are organized in a better way.

Loukil et al. [51] designed a semantic IoT gateway which improved the control over the private resources along with providing protection for collected personal data. This is done by first matching the terms and conditions of service at customer's end with the privacy preferences at the data owner's end and then inducing a policy adaptable to the described conditions. This privacy policy is converted into smart contracts and then to host the generated smart contracts, the set of resources are connected to a decentralized network using blockchain technology. There are nine core components of this blockchain framework named as public IoT network, private IoT network, public blockchain, smart contract, private ledger, transaction, Semantic IoT Gateway, storage node and local storage. The Semantic IoT Gateway binds together the blockchain network, the actuators and the IoT terminals. When the experiment was performed on real world, use-case data shows positive result and proves that custom generated smart contracts can be added to the blockchain technology with high success rate.

A credit-based mechanism has been proposed in [52] which ensures better efficiency for simultaneous transactions and confirms system security. For ensuring the confidentiality of sensitive data, a mechanism for managing the data authority is defined which controls the sensor data access. The mechanism is designed on structured blockchains based on directed acyclic graphs which gives better performance and improved throughput compared to chain-structured blockchain like Satoshi-style, etc. The case-study was conducted for smart factory and system was implemented on Raspberry Pi model 3B. The architecture design of smart-factory consists of four major components named tangle network, wireless sensors, managers and gateways. The designed architecture is impenetrable to various attacks like DoS, DDoS, Sybil, etc. To guarantee the optimal trade-off between system security and transactional efficiency, a Proof of Work mechanism has been designed which ensures that the honest nodes always devour limited number of resources while enforcing the malicious nodes with increased attack cost. To ensure confidentiality, the authors designed a mechanism for data authority management in which the nodes which collect sensitive data are given the secret key by managers and with the help of this key, sensor data is encrypted before being posted on the blockchain.

Biswas et al. [53] have used a network of local peers which narrows the gap between blockchain peers and IoT devices. Without affecting the transactional validation policy followed by peers at both the local and global level, the number of transactions entering the global blockchain are restricted using local ledger. The authors proposed a framework based on blockchain technology for IoT which considers both the inter and intra transactions for the corresponding organization. Each IoT device is registered by the certification authority and associated with one of the organizations. Instead of using peers belonging to global blockchain, a local peer was structured to achieve the interaction with peers belonging to global blockchain network. The designed framework targets to handle the indirect rise in transaction per second for the global blockchain network and increase in ledger storage requirements at peer level. The size of ledger is limited under this framework and is distributed between local and global peers. The transactions between two organizations are validated via global blockchain network validating it for 100% peer validation. In this work, they clearly demonstrated that if the issue of scalability is not addressed, blockchain and IoT cannot be integrated together and creating a network of local peers allowed the blockchain ledger to spread across all the peers and hence improved scalability.

A blockchain connected gateway is designed in [54] to maintain the privacy preference of IoT devices securely and adaptively within the blockchain network. It can prevent the leakage of sensitive data by ensuring that the data is not accessed without the user's permission. There are three major participants in the proposed framework namely IoT device administrator, gateway administrator and end user. The IoT device administrator stores the device information along with device's privacy policy at the blockchain network before the user gets the access to it. The list of attributes uploaded on the blockchain network includes device's name, manufacturer's information, device description, device images list, etc., and the privacy policy includes information related to preference, policy identifier,etc. The signature scheme proposed by authors is robust and has interacting skills similar to DLP based on elliptic curves. Each security component of this signature scheme is implemented and tested on Model B of Raspberry Pi III and computational cost for each security component is calculated. The results show that while legacy devices are in use, the proposed framework increases the trust among IoT applications and improves user privacy.

Badr et al. [55] designed a novel protocol named pseudonym based on encryption for providing privacy to the patient's data available in the e-Healthcare system. The encryption mechanism is blockchain-based in which high end different authority encryption technique is used for securing the patient's confidential data. The public blockchain tier between the healthcare cloud providers and the blockchain tier handling the sensors on patients body along with patient's system on the platform are considered in this approach. The work elevates the anonymity factor in patient's data by considering blockchain as the anonymity enhancement technology using the multi-tier architectural model which prevents system from various attacks like block enquiry infringement etc. The solution was evaluated using the MIRACL library which keeps track of the processing time for all the functions executing within the communication channel. The architecture is a three tier where the first tier shows how all the sensors, devices are connected to patient through a gateway or aggregator. The second tier analyzes the distribution of ledger and handles communication within the members of health record member and provider. In tier three, the compliance with the cloud providers is considered and analyzed. This framework can handle some of the security vulnerabilities but not all. Therefore, in the future, this model can be modified to handle larger cluster of security issues.

2.3.3 IoT Nodes Integrated with Blockchain Clients

An IoT battery-powered device may be integrated directly with blockchain client. This allows blockchain features to be embedded in IoT devices itself for direct interaction between them. A multilevel blockchain system (MBS) is proposed in [56] to secure IoT which uses mobile agents to enforce the flexibility and speed of transactions in the blockchain. Mobile agents roam throughout the network of IoT devices to aggregate useful data and generate hashed blocks of data reducing time delays and solving other issues like scalability and synchronization. MBS consists of three hierarchical levels through which IoT devices can send their data securely: micro-level consisting of IoT devices, meso-level consisting of cluster-heads and macro-level consisting of the blockchain platform. MBS platform is made up of four entities: IoT device (collect and transmit data), ordering service (accept transactions and create blocks), endorsing peers (check validity of smart-contracts), and committing peers (run validation). It includes meso, macro and micro agents with different roles and locations in the architecture. Simulation is done using Hyperledger Fabric with 1000 nodes and the end results are satisfactory in terms of energy consumption and response time.

Qian et al. [57] divide IoT into three parts i.e, network layer, application layer, and perception layer and propose a security scheme for IoT using blockchain by considering the security issues in these layers. The application layer consisting of smart home, smart healthcare, automatic driving includes access and authentication control, privacy protection and software handling. The network layer consists of low power WANs and mobile networks. The perception layer requires security of devices, authentication and access control and consists of IoT gateway and terminal devices. To manage the security and other issues of IoT, blockchain based platforms for IoT devices can be constructed along-with the integration of cloud services. This structure consisting of union nodes, IoT devices, cloud providers etc. communicate through high speed links. Links between IoT and blockchain devices can be secured through authentication techniques to guarantee reliability. They have also discussed two open issues namely identity verification and machine learning-based monitoring of abnormal network traffic.

A blockchain-based IoT structure is proposed in [58] using smart contacts which aids users in keeping a complete control over their useful data and also on how it is used by third-party clients. The given system model consists of three entities: Aggregates are the users owning IoT devices who post transaction into the blockchain to publish data or grant permissions; Subscribers (third-party) who want to access the data posted on the blockchain by issuing transactions; Vendors are the IoT devices' manufacturers liable for producing official images of firmware. All these three entities are recognized through public-private key pairs when they want to communicate via the blockchain network. Aggregators store their published data in the off-chain network using content based addressing. A hash is calculated for each data piece corresponding to the address of the data which is used as an index for data search and retrieval. Two smart contracts namely, FirmwareUpdate and AccessControl are introduced for controlling updates and proving access permissions. This blockchain-based update scheme of firmware ensures that the IoT devices are not tampered with and are designed through authentic firmwares.

Another hierarchical structure of blockchain is discussed in [59] for tamper-proof storage and retrieval of data in IoT systems. In this architecture, along with the resource-constrained IoT nodes, some additional devices are used for "data collection" which have more storage and computational power. This model ensures that the data is sent securely to the edge servers through the resource-constrained devices for data verification. An authentication and access control method for IoT is proposed in [60] based on a distinct blockchain-dependent architecture. The authentication process is done through smart contracts. If found valid, then the sender's address and access token are broadcasted by the smart contract through which the user can receive this information. A package is then crafted by the user and signed using the ethereum private key. The authors have shown that this method outperforms existing methods in terms of decentralization and tamper proof records. This approach can also withstand attacks attempted to guess credentials through brute force and control legitimate sessions.

Bubbles of Trust [61] is an authentication method for IoT devices based on public blockchains and smart contracts. In this approach, secure virtual zones called bubbles of trust are created where each device trusts only the devices within its zone. Each zone is non-accessible and protected from non-member devices. Communications in this network are through transactions validated by the blockchain. In the initialization phase, a Master device is designated which owns a public and private key-pair. All objects of the system are called followers. Each follower is given a ticket that contains objectID (identifier of follower), groupID (identifier of object's bubble), pubAddr (public address of follower) and a signature. The master of the bubble initiates a transaction containing the identifier of master and the group created. This transaction is validated by the blockchain to check the uniqueness of identifiers. After creation of a bubble, the followers send transactions to get linked to their bubble.The follower's identifier is also verified and validated by the blockchain using smart contracts. This approach satisfies the security requirements of IoT, its cost and efficiency when implemented using Ethereum and C++ language.

2.3.4 IoT Nodes as Regular Sensors

IoT devices with insufficient resources to tolerate any additional logic function as regular sensors. They simple collect the data and forward it to the blockchain structure through the gateway. Dorri et al. [62] showed that blockchain can be designed scalable as well as lightweight by optimizing it with respect to the requirements of the IoT with proper end-to-end security. They proposed a Distributed Throughput Management (DTM) algorithm which lowers the delay and processing overhead for mining. The cluster heads deploy the distributed trust technique such that the verification of new blocks does not result in high processing overhead. To handle scalability, all the overlay nodes including service providers, cloud storage and IoT devices are made a part of clusters and the blockchain is managed by the cluster heads only. For reducing the packet overhead as well as memory footprints, the data of all the nodes is stored off-the-chain on cloud storage. Each cluster head tries to develop trust with the other cluster heads by validating them on the basis of generated new blocks using the distributed algorithm of trust. By considering the transaction load of the network, DTM algorithm ensures that throughput of blockchain remains stable as far as system parameters are handled dynamically. They also performed analysis of proposed approach under 8 significant cyber attacks and estimated the likelihood of these attacks along with their respective defence mechanisms which ensured that the algorithm is prone to all of them. The simulation was performed on NS3 and the results showed that in comparison to other approaches, this approach provides higher scalability and reduces the overall delay and packet overhead.

A highly scalable and distributed access management system is proposed in [63] using blockchain technology. The authors compare the existing approaches with the proposed approach and illustrate that when it is tested and analyzed on different configurations of the blockchain system, the delay was reduced and the throughput was improved. When the sensor nodes are connected through multiple hubs, this approach works better for horizontal scalability and gives better result. The architecture for this model was designed keeping in mind various parameters like concurrency, mobility, lightweight nature, accessibility, scalability, etc. It constitutes of various components where each component performs a specific task and all of them when allied together will result in complete working of the system. Sensor network, agent-node, managers, blockchain network, smart contract and management hub are the various components of this architecture. To test and retrieve the best results, they performed the experiment multiple times using Docker on Ethereum platform. The results were almost similar in each of the trial and showed positive results.

An access management scheme is proposed in [64] based on attributes for IoT systems. Blockchain is used to store attribute's distribution to maintain data integrity and avoid single point of failures. Two main entities used in this approach are: IoT devices and attribute authorities. The attributes authorities distribute attributes and manage the blockchain. Attribute-based access extracts the identities (roles) into an attribute-set managed by the attribute authorities. These authorities maintain a public ledger jointly using a consensus mechanism. The attributes are authorized and posted to blockchain through "transactions". For the registration of IoT devices into the network, the attribute authorities create a public/private key pair. The IoT devices collect and share the data in the network. They do not take take part in verifying transactions and can only view the blockchain. The access control mechanism is simplified through the use of basic signature and hash techniques to make this approach efficient for resource constrained IoT systems.

BB-DIS (Blockchain and Bilinear mapping based Data Integrity) is proposed in [65] for data integrity of large-scale data of IoT systems. Framework of BB-DIS includes four entities namely Cloud Service Providers (CSPs), Data Consumer Devices (DCDs), Data

Owner Devices (DODs), and Smart Contracts. Smart contracts used in this approach are of different kinds and they are used to verify the data integrity in blockchain. DCDs and DODs are used to produce key pairs at the time of initialization of the blockchain. CSPs can provide mining services by functioning as miner nodes and receiving the required rewards. Storage services like Microsoft Azure, Amazon S3 etc., are also provided by CSPs. BB-DIS after dividing the IoT data into shards generates homomorphic verifiable tags (HVTs) for verfication of sampling. Edge computing is used to reduce the computation and communication costs by preprocessing the large-scale generated IoT data.

A distributed key management architecture is proposed in [66] to reduce the latency and satisfy the scalability, hierarchical access control and decentralization requirements in IoT. In this architecture, the blockchain mechanism is governed by the SAMs (security access managers) instead of central authorities. SAMs are used for storing the logical topology whereas the blockchain is used to store the key management activities. Moreover, multiblockchains are used by cloud managers to reduce latency and support scalability operations. This multiblockchain approach improves the scalability and system performance as shown through simulation results.

2.4 FL-based Solutions for IoT Privacy

The integration techniques for FL within IoT networks can be broadly classified into three categories: centralized, distributed, and hierarchical aggregation methods as shown in Figure 2.3. In the centralized approach, a singular edge or cloud server is responsible for compiling the local learning models from all connected devices. Meanwhile, the distributed method employs multiple servers to collect updates from local learning models. These updates are then shared and combined across different servers before a collective global model is formed. Hierarchical aggregation, alternatively, involves an initial layer of aggregation at closer nodes, such as edge servers, before these pre-aggregated models are further combined into a final global model.

2.4.1 Centralized Aggregation

Centralized aggregation, as explored in several studies [67, 68, 69, 70], involves the collection of local learning models from IoT devices by a singular edge or cloud server to construct a global learning model. This model of aggregation, adaptable for edge or cloud computing, offers a streamlined approach but raises concerns regarding scalability and potential bottlenecks.

Wang et al. [67] proposed an In-Edge-AI framework that integrates Deep Q-Learning agents within edge computing-enabled IoT networks for tasks such as computation offloading and edge caching. While this framework demonstrates the potential to reduce training costs significantly, it does not fully address security and privacy concerns that might lead to potential exploitation by malicious entities. The study in [71] presents a framework aimed at enhancing both learning and communication efficiency in wireless

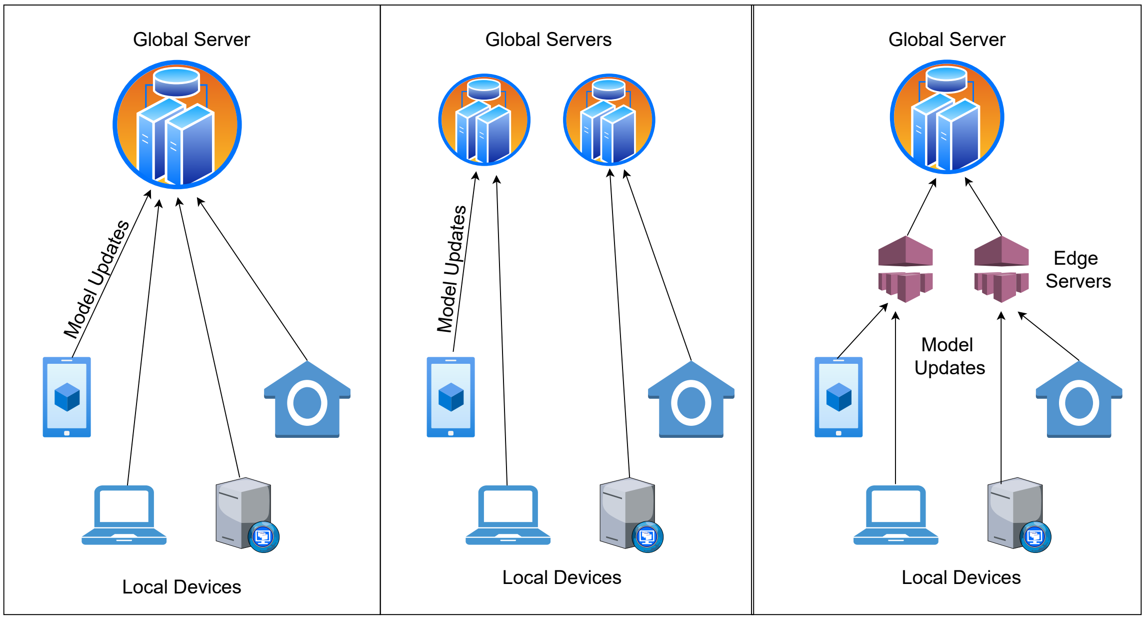


Figure 2.3: Types of FL aggregations: central, distributed and hierarchical.

FL environments. It formulates an optimization problem for joint wireless resource allocation, highlighting the influence of wireless factors on learning performance and suggesting strategies for optimal resource distribution to minimize learning loss. Wang et al. [72] explored the convergence properties of FL with non-IID data and proposed a control algorithm to minimize learning loss, thereby enhancing learning efficiency within edge networks. However, it emphasizes the need for further research into handling non-convex loss functions more commonly encountered in IoT applications. A significant advancement in client selection for FL is introduced by the FedCS protocol [68], which optimizes client selection based on device capabilities and data characteristics to improve learning efficiency. However, it necessitates additional measures to protect device privacy and better manage the transmission latency to accommodate more participants in the learning process. To address the challenge of asynchronous participation and the presence of malicious devices, the architecture CoLearn [73] utilizes a FL framework to ensure secure and efficient learning among resource-constrained IoT devices. Despite its advantages, there's an opportunity to enhance its performance further through lightweight authentication mechanisms for IoT devices.

2.4.2 Distributed Aggregation

Unlike centralized models, where a single server aggregates learning updates, distributed methods utilize multiple aggregation points, allowing for a more flexible and resilient framework. In distributed aggregation [74, 75], IoT devices generate their local learning models and transmit these updates to a network of distributed servers. These servers then collaborate, sharing the gathered learning models among themselves before performing a collective aggregation to formulate a global model. This method stands out for its ability to operate without relying on a singular central server thereby enhancing decentralization.

Qu et al. [74] introduced FL-Block, which leverages blockchain technology within fog computing environments to enhance the robustness of FL against poisoning attacks. By adopting a decentralized aggregation method, FL-Block aims to mitigate risks associated with centralized server failures or security breaches. The FL process under FL-Block comprises local model development by devices, cross-verification at fog servers, and global model updates through collaborative fog server aggregation. However, the latency due to blockchain's consensus algorithms poses a challenge, suggesting a need for innovative FL schemes that can accommodate the decentralized nature of blockchain more efficiently.

Furthermore, the study in [75] addresses the limitations of device connectivity to fog servers by proposing an FL scheme that facilitates device cooperation, eliminating the reliance on centralized servers for model aggregation. This scheme introduces two strategies: consensus-based federated averaging and an enhanced version that includes gradient exchange, aiming to reduce convergence times despite added complexity. The proposed model was tested within an industrial IoT setup, showcasing its potential for scalability and robustness in dense Device-to-Device (D2D) networks. Although the proposed approach has several advantages, the authors do not consider the importance of efficient resource allocation and power control within such networks. Optimizing resource distribution and adjusting power levels could significantly reduce packet error rates, thereby enhancing the accuracy of the global federated learning model.

2.4.3 Hierarchical Aggregation

Hierarchical FL integrates multiple layers of model aggregation, typically involving initial aggregations at edge servers before a final, comprehensive aggregation occurs at a cloud server. This method has been explored in various studies [76, 77, 78] which illustrate its application in enhancing IoT networks' efficiency and security.

Zhao et al. [77] introduced an FL framework for intelligent fog radio access networks. This framework aims to address the challenges associated with centralized machine learning approaches, such as elevated communication costs and privacy concerns. By adopting both traditional and hierarchical FL processes, the framework proposes a solution where model aggregations occur initially at fog nodes, followed by a concluding aggregation at a remote cloud server. This setup facilitates the inclusion of a greater number of end-devices in the learning process through the compression of local model neural network architectures and parameters. The authors further discussed the crucial technologies required to implement FL in intelligent fog radio access networks, along with highlighting relevant research challenges. Similarly, the approaches in [78] and [76] proposed hierarchical FL frameworks designed for mobile networks, emphasizing the benefits of local aggregations before global cloud-based aggregation. This approach significantly benefits performance, enabling end-devices to utilize existing communication resources more efficiently. However, the implementation of hierarchical FL has several design considerations, such as the optimal frequency of edge server aggregations prior to central cloud aggregation. Hierarchical FL introduces potential divergences in model weights, categorized on the level of aggregation

— at the edge due to client-edge discrepancies and at the cloud due to edge-cloud discrepancies. As described in [76], the extent of these divergences is influenced by the balance between the number of local learning iterations and edge aggregations. Optimizing the number of edge aggregations typically enhances learning outcomes, particularly for non-IID data distributions. The performance of hierarchical FL is further dependent on the chosen federated optimization scheme (e.g., FedAvg, FedProx) and how it manages data heterogeneity.

2.5 Research Gaps

Based on the existing state-of-the-art literature on IoT security and privacy, a thorough examination reveals that there are significant gaps that need to be addressed to enhance the robustness of IoT systems.

Firstly, the scalability of blockchain technology within IoT-based applications is still a critical concern. Despite the recognition of blockchain's potential to enhance security measures, its application in IoT environments is limited due to scalability challenges. The substantial data volumes generated by IoT devices demand a blockchain framework that not only accommodates it but does so without compromising security.

Secondly, preserving privacy through FL-based techniques in IoT presents another area requiring further research. While FL proposes a methodology for training ML models on decentralized data sources, current approaches lack in protecting the model from adversarial attacks and maintaining model accuracy even in the presence of heterogeneous IoT data.

Thirdly, the decentralization of FL systems to a peer-to-peer (P2P) architecture is sparsely covered in existing literature. This decentralization is aimed at mitigating single points of failure and strengthening the system's defense against adversarial attacks. This gap signifies the need for innovative approaches to ensure secure, robust, and decentralized collaborative learning within IoT ecosystems.

Lastly, a detailed exploration regarding the integration of P2P-FL that can address blockchain's energy consumption—is understudied. This research gap highlights the need for a solution that combines blockchain and P2P-FL to build a sustainable, secure, and privacy-focused infrastructure for IoT applications.

Chapter 3

Data Security for IoT

In this chapter, we delve into the intersection of data security and IoT, focusing on the application of blockchain technology as a secure data storage solution.

3.1 Blockchain for Data Security

The trending smart living IoT environments though beneficial introduce several security and privacy issues [41]. For instance, in a smart healthcare scenario, on one hand, we want to ensure the privacy and security of user's medical data and on the other hand, we will have to disclose their personal information to the doctors and staff so that they can examine and treat the patients. Another example is the safety and privacy of user's data in a smart home system. As shown in Figure 3.1, the various devices of a smart home (such as smart refrigerators, smart lightning systems, smoke detectors [79], audio system, thermostat, power control, etc.) can regulate themselves automatically and be remotely controlled through the Internet. However, they pose several security risks, such as, smart assistants like Alexa can be used by intruders to steal the private data of users by snooping upon them. Therefore, it is critical to ensure database security, data encryption, information integrity, service validation and secure communication links for the smart applications of IoT systems.

Moreover, the IoT systems are mostly centralized and dependent on cloud servers for storage and processing capabilities. Although this centralized cloud architecture has been prevalent for decades, it will not be able to support the ever increasing demands of the IoT systems tomorrow.

A decentralized approach with a standard peer-to-peer (P2P) model would help in solving many issues of the IoT ecosystems. In a decentralized IoT network, each device can operate independently and communicate with its peers directly without needing a central authority. This P2P model reduces the risk of centralized data breaches and system failures, while also potentially decreasing latency and improving system responsiveness. Decentralized systems make it much harder for any single point of failure to cripple the network thereby enhancing the resilience of the network against attacks or technical failures. Moreover, it can also reduce the cost of setting up and running large data centers, which are usually needed in centralized systems. Blockchain technology is a distributed approach that can be used to resolve the security issues related to IoT [40]. By keeping an immutable history of the data collected and exchanged by smart devices, blockchain can help make the IoT system secure along with improving the trust and security through "smart contracts"

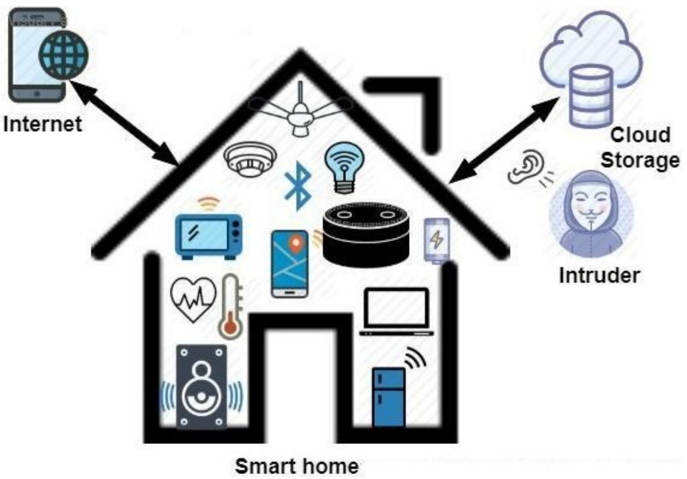


Figure 3.1: A smart home being eavesdropped by an intruder.

and “digital signatures”. Developing IoT systems are using the blockchain technology to provide security, verify and validate users, record data, construct decentralized platforms and manage transactions. However, the adoption of blockchain technology in IoT-based systems still pose the issue of scalability due to the tremendous amount of data generated. In this chapter, we propose a framework guideline for IoT-blockchain systems. We use the concept of sidechains to address the scaling problem of blockchain thus making it appropriate for IoT-based systems. Moreover, we utilize the concept of edge servers to process and store the data offline. Sidechains exist alongwith main blockchains generally attached to it via two-way pegs. They can be used to perform some specific task independently and then later be rejoined to the main blockchain. Sidechains increase the efficiency of blockchains by offloading some work of the main chain thus making it faster and smaller. To exemplify our idea, we use the scenario of healthcare systems in the rest of this chapter. Healthcare is a very typical and crucial use-case of IoT systems, where the amount of data collected is very extensive and voluminous. However, the architecture proposed is application-agnostic for usage in diverse IoT-based systems.

Along with scalability, highly sensitive and private healthcare data requires strong authorization and authentication procedures for storage. Generally, users have very limited control over their data because they don’t know how it will be stored or accessed once it is sent to the cloud. In such cases, patients who are unfamiliar with the norms and procedures for protecting private data may avoid full-disclosure of their medical data or refuse treatment altogether. Interoperability challenges between hospital systems and different providers pose additional barriers to effective data access and sharing.

Since blockchain is not fundamentally designed for high throughput, therefore it needs to be optimized for use in real-time IoT applications such as smart healthcare [80]. In light of existing state-of-the-art research, there are three major issues in the current mechanisms: 1) ledger storage constraints 2) data management difficulties, and 3) high latencies.

To address the above challenges, we propose HierChain, a novel **hier**archical block**chain**-based framework, for secure health data management and storage. According to the scalability trilemma given by Vitalik Buterin, all three components i.e. scalability, security, and decentralization cannot be maximized simultaneously [81]. Figure 3.2 represents the hierarchical storage taxonomy used in HierChain. The taxonomy utilized in HierChain states that as the security of a blockchain increases, its scalability decreases and vice versa [82]. Following this constraint, HierChain manages data storage in a manner such that more sensitive data is sent to a highly secured blockchain (even if it is less scalable) whereas less sensitive data is sent to a highly scalable blockchain (that could be less secure). Therefore, the proposed strategy helps to perpetuate the trade-off between security and scalability in a coherent manner. Furthermore, our approach integrates fog nodes to facilitate data pre-processing and encryption techniques specifically designed for resource-constrained IoT nodes. By leveraging fog nodes, we optimize the data flow and ensure that only relevant and necessary data is communicated to the blockchain network. Overall, HierChain introduces a key innovation by leveraging multiple blockchains in a novel approach that has not been explored in the existing literature. By harnessing the potential of a multi-blockchain framework, we present a robust solution that effectively addresses the prevailing issues. HierChain addresses the challenge of ledger storage constraints by categorizing health data based on its specific features and storage requirements and storing it on different blockchains accordingly. Moreover, it tackles data management difficulties by integrating fog nodes for data preprocessing, enabling the elimination of redundant information and optimizing the data flow. Additionally, HierChain mitigates high latencies by intelligently selecting the blockchain based on network latency and choosing the optimal one for data storage and sharing. Therefore, our approach offers a fresh perspective and contributes to the advancement of blockchain technology in the healthcare domain.

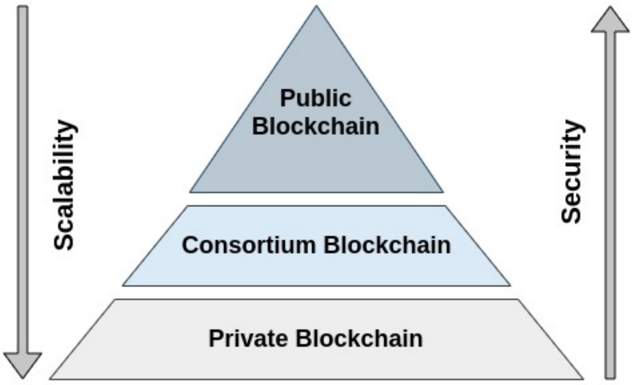


Figure 3.2: HierChain Taxonomy.

It is important to note that our approach is highly relevant for healthcare due to its ability to ensure the privacy and security of sensitive healthcare information. By integrating differential privacy, secure blockchain allocation, alignment with compliance standards, and optimized data storage techniques, HierChain effectively addresses the specific challenges and regulatory requirements associated with medical data, providing a robust and tailored solution for healthcare applications.

3.2 System Model

This section presents the technical framework designed for integrating IoT with blockchain in a scalable and storage optimized manner. The proposed framework, HierChain, is configured in the form of layers to store the massive IoT data effectively and securely while maintaining the performance of application.

3.2.1 Different Layers

- End-user layer encompasses a sensing network that contains various sensors and actuators at edge of the layer. It includes distinct types of devices such as health monitors, smart wearables, medical equipment, body sensors, and other devices that sense the environment and collect real-time data.

- Fog layer is the central layer of proposed hierarchical architecture. It consists of reliable computing and powered devices called fog nodes. Any device which is a few hops away from the end-user layer can be converted into a fog node. Each fog node is connected to a local group of devices for performing the computation and storage of perceived data.

- Cloud layer enables storage and computation by providing resources to end users over the internet. Most of the existing applications nowadays require abstraction, encapsulation, and processing of real-time data that can be provided by the cloud. It consists of a centralized, shared pool of cloud servers that are delivered to the end users in an on-demand manner.

3.2.2 Data Types

In certain parts of the world, data protection regulations have been implemented to safeguard personal health records against security attacks and threats. Two of the most commonly enforced data protection regulations are GDPR and HIPAA [83]. The European Union has implemented the General Data Protection Regulation (GDPR), according to which companies should protect the Personally Identifiable Information (PII) of individuals. The GDPR categorizes data into three types: Sensitive Personal information, Personal information, and Non-Personal information, each with its own requirements for protection. Health insurance portability and Accountability Act (HIPAA) governs the usage and disclosure of U.S.-protected health information (PHI) and applies to healthcare plans, payment systems, and business associates. PHI pertains to mental or physical health information, and prior patient approval is typically required for its disclosure or use. IoT data collected from clinical sources is unstructured and voluminous. In our system model, the health data from different sources are classified into three categories as follows.

- **Public data (least sensitive):** Information that does not require high confidentiality or privacy is termed public data. Data set for which no confidentiality or privacy is expected is classified as public data. Although this public information does not require protection, care should be taken while sharing health-related data. Data examples: Hospital department's information, dates of current employment, job postings, campus maps, list of publications, specializations of doctors, medicines available currently, and tests facilitated.

- **Restricted data (moderately sensitive):** Restricted data refers to information that is restricted to a group of personnel only. It may be a group of authorized employees, an organization, or a group of organizations. This information is not available to the general public, hence, some level of authentication is required to access this data. Restricted data refers to the information restricted to a group of authorized people or an organization and not available in the public domain.
  Data examples: Insurance information, appointment details, pharmaceutical supplies details, performance reviews, personnel records, policies and procedures, scheduling, and billing details.

- **Confidential data (highly sensitive):** Confidential data is information that needs to be protected at all costs and is accessible only through approved authorization. Such information is required to be kept confidential even by the laws and regulations of healthcare institutions. Data examples: Credit/debit card details, personal details such as date of birth, age, sex, and address, patient's medical history, doctor-patient communication, images collected during diagnosis, service records and file progress notes.

3.2.3 Blockchain Categorization

A blockchain is essentially a chain of blocks (as depicted by Figure 3.3), each containing data, the hash of the block (a unique digital fingerprint), and the hash of the previous block. Blocks are secured through hash functions, specifically SHA-256 in the case of Bitcoin, which generates a fixed-size string that appears random and changes entirely with any modification to input data. Hashes are critical for maintaining the blockchain's integrity and ensuring that each block is permanently recorded and unalterable once added to the chain. Each block in the chain contains the hash of the previous block, which links the blocks together in a chronological and unchangeable sequence as shown in Figure 3.4. This chaining defends against block tampering, as altering any block would require recalculating all subsequent block hashes, a task that becomes computationally impractical as the chain lengthens. To create a block, the following steps are performed:

- When a transaction occurs, it is verified through a consensus mechanism by thousands of computers around the network.

- The verified transaction is stored in a block and combined with other transactions to form a data set.

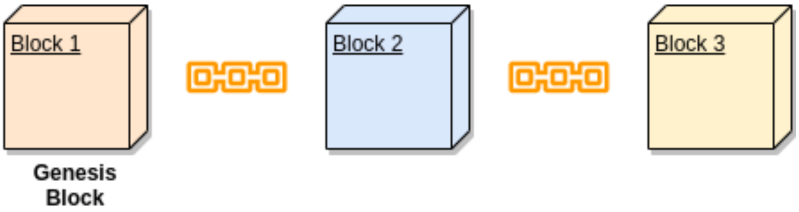


Figure 3.3: Basic Structure of a Blockchain.

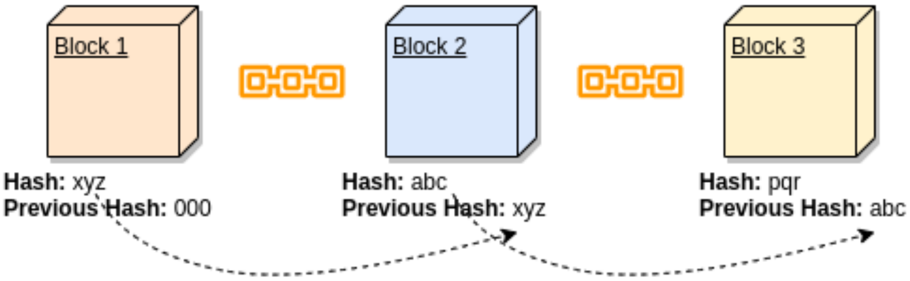


Figure 3.4: Linking different blocks.

- The block is then given an exact hash. Once added to the blockchain, it's published across the network.

The types of blockchains available for storing the previously described categories of data can be either public, private, or consortium blockchains. A public blockchain is a decentralized and distributed digital ledger where anyone is free to join the network and perform transactions. On the other hand, a private blockchain is operated and controlled by an entity with restrictions on who can participate in the network. Finally, consortium blockchains are a hybrid between private and public blockchains with a semi-decentralized structure. The data can be stored in these blockchains in two forms: Sidechains and main chain. A sidechain is a secondary blockchain, that exists alongside its parent chain. This mechanism allows assets to be transferred from one blockchain to be used on a second blockchain and vice versa.

3.3 Problem Formulation

In this section, we present a formulation of our research problem in terms of an optimization problem. The amount of IoT healthcare data is increasing day-by-day with time [84]. This data when stored on a blockchain leads to an increase in the number of executed transactions. Each transaction within a block consists of various fields that determine its size. Size of a transaction for any block b can be given as $s_{txn}(b)$. This size is comprised of following values:

$$\begin{aligned} s_{txn}(b) = d_{flag} + \sum_{t=t_{start}}^{t_{lock}} (ctr_{in} + ctr_{out}) \\ + t_{lock} + V, \end{aligned} \tag{3.1}$$

where V = version number, d_{flag} = flags data, t_{start} = start time, ctr_{in} = in-counter, list of inputs, ctr_{out} = out-counter, list of outputs, and t_{lock} = lock time. The size $s_{txn}(b)$ is measured in bytes and includes all bytes from the fields mentioned above. Each component of the transaction adds to the total size, calculated by summing the individual sizes of the version number, flags, counters, and time fields, plus the actual data included in input and output lists. If the number of transactions in a block b are $n_{txn}(b)$, then the total size of transaction data in a block can be given as:

$$S_{txn}(b) = n_{txn}(b) \cdot s_{txn}(b) \tag{3.2}$$

Along with the transaction data, each block contains some metadata too. Metadata consists of a block header and transaction counters. Therefore, size of a block header [85] can be given as:

$$\begin{aligned} S_{hdr}(b) = h\Big[previous\, block\Big] + h\Big[merkle\, root \\ + version + target\, formula \\ + nonce + timestamp\Big], \end{aligned} \tag{3.3}$$

Here, h represents the hash function used in the blockchain, *previous block* refers to the hash value of the previous block in the blockchain, *merkle root* denotes the hash value that represents all the transactions included in the block, *version* indicates the version number of the blockchain protocol or software being used, *target formula* is used to calculate the target value for mining a new block, *nonce* stands for "number only used once", and *timestamp* specifies the timestamp indicating the time when the block was created.
Thus, the total size of metadata for block b can be described as:

$$S_{md}(b) = S_{hdr}(b) + S_{ctr}(b), \tag{3.4}$$

where $S_{ctr}(b)$ denotes the size of transaction counters. Each block of a blockchain contains both transaction data and its metadata. Therefore, from equations (1) – (4), the total size of the block b for blockchain B_i can be represented as,

$$S(b, B_i) = S_{txn}(b) + S_{md}(b), \tag{3.5}$$

As mentioned above, a block consists of a multiple number of transactions [85]. The time it takes for a blockchain to reach a consensus for a particular block depends on the difficulty level of the blockchain. When a block b is posted, it takes some time to be propagated in the blockchain B_i network. So, the propagation delay for sending a block b on blockchain B_i can be defined by:

$$t_p(b, B_{\mathrm{i}}) = t_c(b, B_{\mathrm{i}}) + t_{pr}(b, B_{\mathrm{i}}) + t_q(b, B_{\mathrm{i}}) \tag{3.6}$$

Here, t_c *(b,B_i)* = communication delay, t_{pr} *(b,B_i)* = processing delay, and t_q *(b,B_i)* = queuing delay. Along with the propagation delay, the settling time for a block adds to the total delay before it can be confirmed. This settling delay, $t_s(b, B_{\mathrm{i}})$, can be represented by:

$$t_s(b, B_{\mathrm{i}}) = t_{cs}(b, B_{\mathrm{i}}) + t_{ot}(b, B_{\mathrm{i}}). \tag{3.7}$$

Here, $t_{cs}(b, B_{\mathrm{i}})$ denotes consensus delay and $t_{ot}(b, B_{\mathrm{i}})$ denotes other delays that might be involved such as synchronization delay or network delay. Therefore, by using equations 6 and 7, overall network latency, $T(b, B_i)$, for a block to be generated can be given by:

$$T(b, B_{\mathrm{i}}) = t_p(b, B_{\mathrm{i}}) + t_s(b, B_{\mathrm{i}}). \tag{3.8}$$

Therefore, in order to achieve maximum performance for data storage on a blockchain, the optimization problem can be represented mathematically as:

$$min\ T(b, B_i)\ \ \&\ \ max\ S(b, B_i) \tag{3.9}$$

Thus, this optimization problem provides a mathematical foundation for determining the most efficient allocation of data to different blockchains within the HierChain framework. In order to achieve this, we have used three distinct blockchains in our experimentation. These blockchains are selected such that they have variable network latency and block sizes based on their ability to provide security and scalability. A detailed description of our proposed solution for data access and storage is provided in the subsequent section.

3.4 Proposed Solution

HierChain classifies the data on the basis of its sensitivity and stores it on different blockchains according to it. We build a storage hierarchy consisting of different types of blockchains selected by the given optimization problem making it scalable and efficient. A component-wise description of the proposed hierarchical storage model is shown in Figure 3.5.

3.4.1 Data Initialization and Classification

The health data can be from various sources such as the sensors used for monitoring patients, health insurance data, and pharmaceutical supply data. This vast amount of data if not preprocessed becomes difficult to handle. Therefore, we use filtering and data reduction techniques followed by data classification and tagging before actually storing the data on the blockchain.

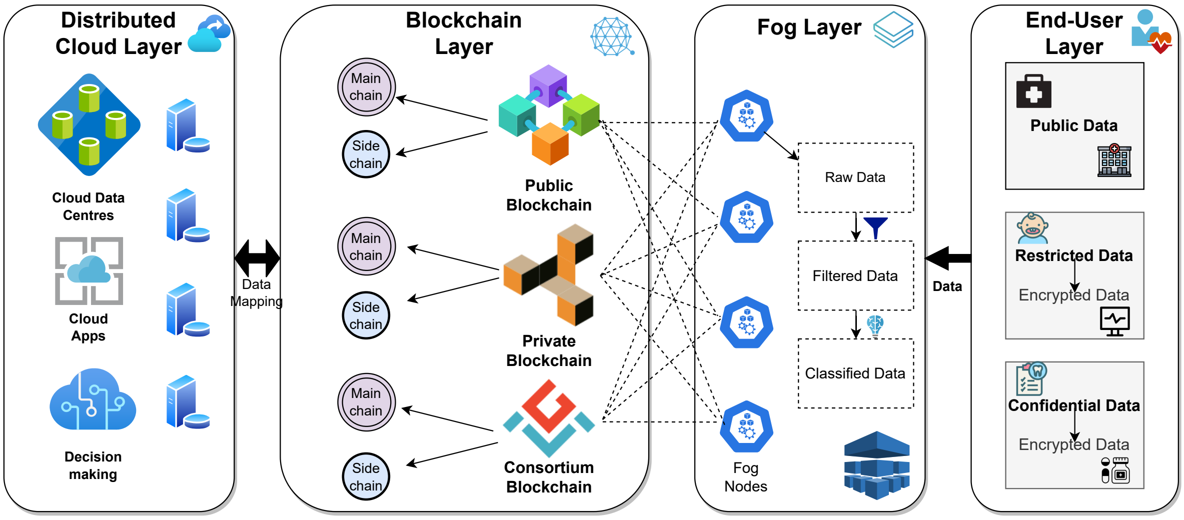


Figure 3.5: The Layered Architecture of HierChain.

The first phase of our proposed approach as given by Algorithm 1 is explained in the following steps:

Algorithm 1 Data Initialization and Classification

Input: Raw data from various data sources.

Output: Classified and filtered data as public ($\mathrm{d_p}$), restricted ($\mathrm{d_r}$) and confidential data ($\mathrm{d_c}$).

```
procedure INITIALIZATION($S$)
    while True do
        for $i = 1$ to sizeof(Nodes) do
            if (data==valid) then
                Data Filtering
                Data Aggregation
                Data Classification
            else
                Data is Invalid
end
```

1. To handle the redundant and voluminous raw data, we first clean the data to filter out the outliers. This is followed by a data aggregation process that integrates multiple data from various sensors into a single data batch. The data from light or resource-constrained IoT devices is preprocessed by the help of fog nodes while full nodes can perform the computation on their own.

2. The information about the storage of the generated data is then mapped according to the sensitivity of the data. Although all kinds of user medical data should be protected from unauthorized access, the levels of sensitivity can vary when it comes to the secure storage of data. We classify the health data into three categories namely public data, restricted data, and confidential data.

3. The data classification is performed by training data sets for a machine learning

based algorithm that further performs data tagging. As described in the next subsection, we use Principal component analysis (PCA) technique for data preprocessing and Naive Bayes for data classification on the fog layer. The data which is highly sensitive is first encrypted by the fog layer and tagged as restricted or confidential accordingly before sending it for storage. If an IoT device wants to send pre-encrypted data, it has to tag the data itself before sending it to the next layer. The data is encrypted using the Advanced Encryption Standard (AES) algorithm where each user has their own unique AES encryption key to encrypt data before uploading it to the blockchain. AES is a widely used symmetric key cipher, which means that the same key is used for both encryption and decryption processes. The encrypted data can then be stored on the blockchain along with a reference to the encryption key used.

3.4.2 Data Preprocessing, Security, and Classification on the Fog Layer

To keep health data private according to the above-mentioned regulations, we use an automated method for the privacy classification of user data. The first step in the process involves the preprocessing of data that is performed by the fog layer in the form of different subtasks described as follows:

- Data cleaning: Some parts of the received health data can be inaccurate or irrelevant for the required application. Data cleansing helps in reducing the complexity and computational time by removing unnecessary data features.
- Filtering: The medical IoT devices may produce imperfect and noisy data that may increase the complexity of the model built, therefore it needs to be filtered first.
- Aggregating: Since there are many sensors used for health monitoring, the fog nodes aggregate all the information from these sensors to reduce bandwidth consumption.
- Wrangling: The raw data from different sensors are not in the same format, therefore it is converted to a common format to make it more convenient for storage and later computations.

We use the PCA technique [86] for data preprocessing which is a multivariate technique that extracts important information by analyzing a data table to display the similarity patterns as data points in maps. PCA transforms the original high-dimensional data into a new set of data that is lower-dimensional, while still retaining the most important information. It achieves this by identifying the principal components in the data, which are linear combinations of the original features that explain the most variation in the data. The first principal component captures the most variation, the second captures the second-most variation, and so on. By reducing the dimensionality of the data, PCA can improve the accuracy of machine learning models and reduce the risk of overfitting.

The Naive Bayes algorithm is a widely used machine learning technique that is known for its speed and ability to perform well in multi-class classification tasks [87]. In our work, since our focus is on classifying user-health data into public, restricted, and confidential, the Naive Bayes algorithm appears to be a suitable choice for meeting our classification needs. The goal of the classifier is to predict the probability of a given instance belonging to a particular class, based on its features.

HierChain also leverages the privacy-preserving ML technique called differential privacy to protect the privacy of sensitive data during analysis [88]. Differential privacy ensures that the presence or absence of an individual's data in the dataset does not significantly impact the outcome of the analysis. It achieves this by injecting controlled noise or randomness into the computations performed on the data. This noise masks the contribution of individual data points, preventing the extraction of sensitive information.

- Sensitivity Analysis: For each category of data (public, restricted, and confidential), a sensitivity analysis is performed to determine the maximum impact that a single data point can have on the analysis results. Let's denote the sensitivity of the analysis function as Δf, which represents the maximum difference in the output of the analysis function f when a single data point is added or removed.

- Privacy Budget: A privacy budget, denoted as ε, is established to quantify the allowable privacy loss. The privacy budget determines the amount of noise that will be added to the analysis results.

- Noise Generation: The noise generation mechanism, i.e., the Laplace noise, is applied based on the selected privacy budget and sensitivity of the analysis. The Laplace mechanism adds noise from a Laplace distribution with a scale determined by the sensitivity (Δf) and privacy budget (ε). The probability density function (PDF) of the Laplace distribution is given by:

  $$Lap(x|\mu, b) = \frac{1}{2b} \cdot \exp\left(-\frac{|x-\mu|}{b}\right) \tag{3.10}$$

  where x is the random variable, μ is the location parameter (mean), the scale parameter b determines the amount of noise added, and it is computed as $b =$ Δf / ε.

- Privacy-Preserving Analysis: The fog nodes perform privacy-preserving analysis on the data using the selected noise generation mechanism. The analysis function, denoted as *f*, is applied to the data, and the noise is added to the results to protect the privacy of individual data points.

To further illustrate the integration of differential privacy in the fog layer, let's consider an example where the fog nodes perform a query to calculate the average BMI (Body Mass Index) of a subset of individuals in the dataset.

Let's define the function f as the average BMI calculation function, and let X be the input dataset. The sensitivity Δf of the average BMI calculation is 1 since adding or removing a single individual's data can change the average by at most 1 unit. Using Laplace noise addition, the privacy-preserving calculation of the average BMI, denoted as $\hat{Z}$, can be represented as:

$$\hat{Z} = f(X) + \text{Lap}(0, \frac{\Delta f}{\varepsilon}) \tag{3.11}$$

where $\hat{Z}$ represents the perturbed output and Lap(0, Δf / ε) represents the Laplace noise with a center at 0 and a scale parameter Δf / ε. By incorporating differential privacy into the fog layer of HierChain, sensitive medical data can be protected during the analysis process.

3.4.3 Secure Data Access and Storage

Every IoT device or fog node that wants to access or store data is first authenticated by a "digital signature" in the blockchain. The blockchain layer consists of different kinds of blockchains with varying levels of security, functionality, and scalability. Moreover, for users to have control over their stored data, we use Access Control Lists (ACLs) to enforce access control policies in the blockchain. They specify which entities are allowed to access specific resources and what actions they are allowed to perform on those resources and are enforced by smart contracts.

To handle the large amount of health data, HierChain exploits the specific features of different existing blockchains to store and access data. A blockchain is selected primarily on the basis of the sensitivity of data. Secondly, the specific features of the blockchain such as the block generation time (speed), security hash rate, fault-tolerance, scalability, etc. determine the kind and amount of data that will be stored on it. Moreover, every blockchain consists of one or two sidechains alongwith the mainchain to perform specific functionalities. To ensure that the system is scalable and cost-effective, we use InterPlanetary File System (IPFS) as an off-chain distributed data storage mechanism that provides faster access to data. IPFS is a protocol and peer-to-peer network designed to provide a decentralized and distributed file storage system. We utilize a hybrid storage approach where sensitive data is stored on the blockchain network itself for immutability and security, and less sensitive data is stored on IPFS for efficiency and scalability.

A prototype for our proposed approach is given in Algorithm 2. Here, we consider three blockchains (which can be increased according to the health data) to securely store and access the classified data (output of Algorithm 1). HierChain stores the classified data on the considered blockchains as follows:

1. Public data: This kind of data is voluminous and has the least security requirements. HierChain uses a private blockchain to store public data. The reason is that private blockchains though less secure, are highly scalable due to their speedy consensus algorithms [89].

2. Restricted data: Restricted data being more sensitive is stored on a consortium blockchain. Restricted data often needs to be kept secret between an organization

Algorithm 2 Secure Data Access and Storage

Input: Classified and filtered data as public ($\mathrm{d_p}$), restricted ($\mathrm{d_r}$) and confidential data ($\mathrm{d_c}$).

Output: Data stored in public ($\mathrm{B_{pu}}$) , private ($\mathrm{B_{pr}}$) and consortium blockchains ($\mathrm{B_{cn}}$).

```
procedure Initialization($S$)
    while True do
        for $i = 1$ to sizeof(Nodes) do
            Validate Node $N_i$
            if ($N_i$== Valid) then
                Add node $N_i$
                Grant storage access to $N_i$
                switch data do
                    case 1: data=$d_p$
                        Store in ($B_{pr}$)
                    case 2: data=$d_r$
                        Store in ($B_{cn}$)
                    case 3: data=$d_c$
                        Store in ($B_{pu}$)
            else
                Access failed
end
```

or a group of people, therefore a consortium blockchain can fulfill its required security needs.

3. Confidential data: This is the most sensitive data which requires the highest amount of security. Although the security requirements of confidential data are very high, the data volume is low. Therefore, HierChain uses a public blockchain to store such data in an encrypted form. A public blockchain provides the highest amount of security due to large number of validators and complex consensus algorithms involved [90].

3.4.4 Sidechains Design and Interactions

In this architecture, the healthcare system is maintained through a main blockchain and three side-chains (could be increased according to requirements of healthcare providers). The three side-chains regulate different components of healthcare system namely:

- **Patient's Data:** This side-chain is a consortium blockchain. Every patient is recognized through a universally unique patient identifier assigned by the Certificate authority (CA). The CA ensures the authentication of every patient so that a person's private data is protected from illegal access. The information to be stored about a patient will include his personal data (contact no, address, blood group, age etc.), medical records of any past illness, allergies and other relevant data collected from wearable IoT devices. This huge amount of data is not feasible to be stored on the blockchain due to scalability issues leading to slow transaction processing.

Therefore, we store the patients data in an off-chain storage using a fog/edge server. The patient can control with whom he wants to share his health-related data and for how much time. This data can also be used for processing and making decisions by applying AI techniques in the cloud server. To secure the data stored in cloud server, we store the hash of information on the side-chain which is a small value. If we want to verify the integrity of our data, we can calculate its hash and match it with the associated hash stored on the blockchain. Whenever new data is collected, the blockchain is updated and all the members affiliated by the patient to be on his care team can access this new information.

- **Pharmaceutical suppliers management:** This is also a consortium blockchain which is used to track medicine supplies and verify their integrity. Drug supply chain [42] can be managed through blockchain by using smart contracts and unique barcodes tagged to every drug. The barcodes present on the drugs are scanned and stored on the blockchain in the form of an immutable distributed ledger. The blockchain records are also updated according to the supply chain when transferred from one party to another. Since this is not a public blockchain, only authorized users can view and update the blockchain. This side-chain will help in enforcing drug traceability through accurate tracking of any drug's chain of custody. Moreover, this will help small retailers in financial issues through automated payments and provide easy access of medicines to hospitals in case of emergencies.

- **Hospital's information:** Being a public blockchain, this sidechain maintains information about every hospital, the services they provide and their specific specialisations. This helps the patients to search for the best available health services near them on the basis of their requirements. This unites all the providers into a decentralized repository which will benefit the citizens as well as the hospitals. Moreover, this chain can also maintain publicly the feedback of the hospitals. This will force the providers to deliver quality services.

The main blockchain is a public blockchain used to initiate transactions amongst the entities of side-chains through the use of "smart contracts". A smart contract, similar to an actual contract, is an agreement in the form of a computer code between two parties without the need of any third party to enforce it. Smart contracts enforce data authentication and authorization. We can track down all the activities associated to a unique ID right from the beginning of the registration of that ID through them. In this manner, a patient or a supplier can be authorized to a particular hospital in a verified manner. The main blockchain stores the records in an encrypted form by generating a hash value for every transaction. We use cloud servers to store the data of patients. Only the hash of the data is stored online in the main blockchain instead of storing the whole data to increase the speed and efficiency of transactions.

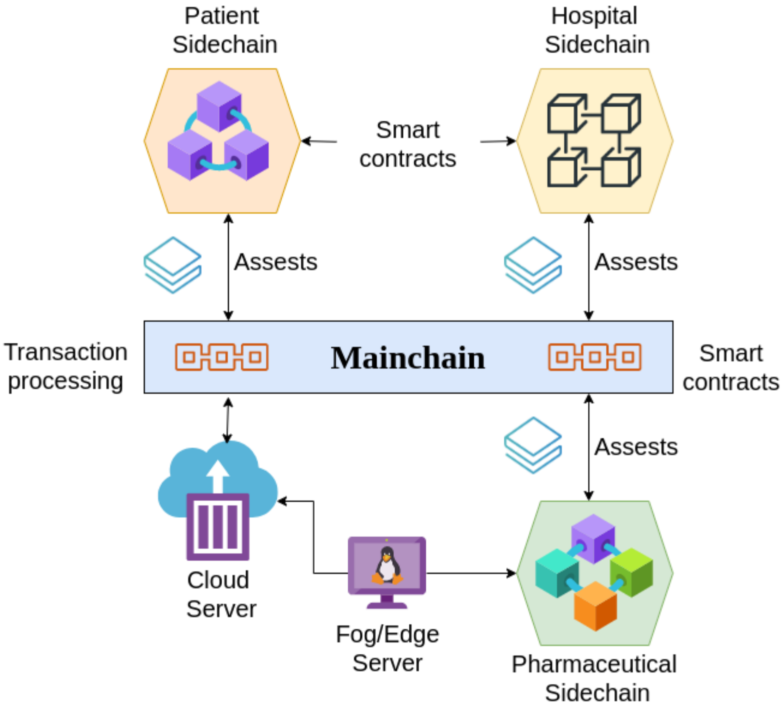


Figure 3.6: Healthcare system architecture.

System Interactions

We outline the system as follows. Doctors, patients and pharmaceutical suppliers first register themselves on the blockchain using their unique addresses. The sidechains are used to store data about specific entities, execute smart contracts, and perform unique functionalities. The main chain stores information in an encrypted form and serves as a common platform to all the sidechains. This framework allows several healthcare entities to participate whilst preserving the decentralization of the system. As shown in Figure 3.6, assets (such as data and cryptocurrency) can be transferred from the sidechain to the main chain and vice versa. For off-chain data storage, the sidechains avail edge and fog servers for faster retrieval and storage of data. However, the data storage and computational facilities required by main chain are extensive, therefore they are connected directly to the cloud servers. Smart contracts are used to store the mapping information and data regarding a particular entity and its interaction with other entities. For instance, the smart contract deployed on the patient's sidechain stores a mapping from the patient's contract address to his unique ID. The account contract of a patient also contains details about his personal information, medical history, and record of healthcare providers allowed to view/modify the patient's medical files. This consortium-based sidechain allows only authorized viewers to execute the smart contracts and read/verify the blocks. Moreover, due to pre-authorized verification, no malicious nodes could post false transactions by colluding, thus eliminating the need for proof-of-work. Therefore, we use Delegated Proof of Stake (DPoS) mechanism [91] for consensus since the nodes are pre-validated and known.

3.5 Performance Analysis

In this section, we evaluate the performance of HierChain through extensive experimentation on real-time data.

3.5.1 Experimental Setup

We created a testbed to simulate a real-time environment for the proposed blockchain-based framework that helps in the management of IoT health-data. The proposed framework, HierChain, consists of the following elements:

- Healthcare datasets
- Raspberry Pi 4 Model B device
- Cryptocurrency Wallet
- Processing units

We used Medical Cost Personal Dataset[1] to access data, write and deploy smart contracts, and perform transactions on HierChain. Three different blockchains are used to store data classified according to its sensitivity. Data pre-processing and classification are performed at the fog nodes using the PCA and Naive Bayes algorithms respectively. HierChain uses Ethereum (public blockchain) to store confidential data, Hyperledger Sawtooth (consortium blockchain) to store restricted data, and MultiChain (private blockchain) to store public data.

The considered dataset consists of 7 features that are classified into public, restricted, and confidential categories as follows:

- Public- *age, BMI, children*: These features are generally not considered sensitive medical information and are often collected in public health surveys. They can be used for demographic analysis and to identify general health trends.
- Restricted- *sex, region*: While *region* may not be considered highly sensitive medical information, it is still personal data that can be used to identify individuals. Also, *sex* can be considered sensitive information in some contexts. As such, it is usually treated as restricted information and is subject to data protection laws.
- Confidential- *smoker, charges*: These features contain highly sensitive medical information that is considered confidential. Smoking status can reveal important health risks, and charges can provide insights into an individual's medical history and treatment. This information should be carefully protected and only shared on a need-to-know basis.

[1] https://gist.github.com/meperezcuello/82a9f1c1c473d6585e750ad2e3c05a41

Framework Element	**Configuration Parameters**	**Values**
Healthcare dataset	Source Number of records Features	Medical Cost Personal Dataset 1338 age, sex, BMI, children, smoker, region, charges
Full Node	Operating System CPU RAM Storage	Ubuntu 18.04 Intel Core i7-7700 CPU @ 3.60GHz 64 GB 1 TB SSD
Fog Node	Operating System CPU RAM Storage	Ubuntu 18.04 Intel Core i7-6600U CPU @ 2.60GHz 8 GB 256 GB SSD
Raspberry Pi-4B (light node)	Number of devices CPU RAM Storage	3 Quad-core ARM Cortex-A72 4GB LPDDR4-3200 SDRAM 32GB MicroSD card
Data processing	Python 3.7 Data pre-processing Data classification	Flask PCA (n components = 3) Naive Bayes
Blockchains	Ethereum Hyperledger Sawtooth MultiChain	PoS, Network: Public, Network ID: 15734 pBFT, Network: Consortium, Initial Nodes: 3 Round-robin, Network: Private, Network ID: 98765

Table 3.1: Configuration details.

It's important to note that the classification of medical information can vary depending on the specific context and regulatory requirements of the situation. This is just a general test-case scenario for the medical insurance dataset.

We created a network of blockchains in which the end-users send medical data to the fog nodes connected to them. We use Raspberry Pi 4B as light nodes that send data for computation to the fog node. The data from these nodes is then sent to the fog nodes for pre-processing and classification. The technical details of all the components used in our experimentation are described in Table 3.1.

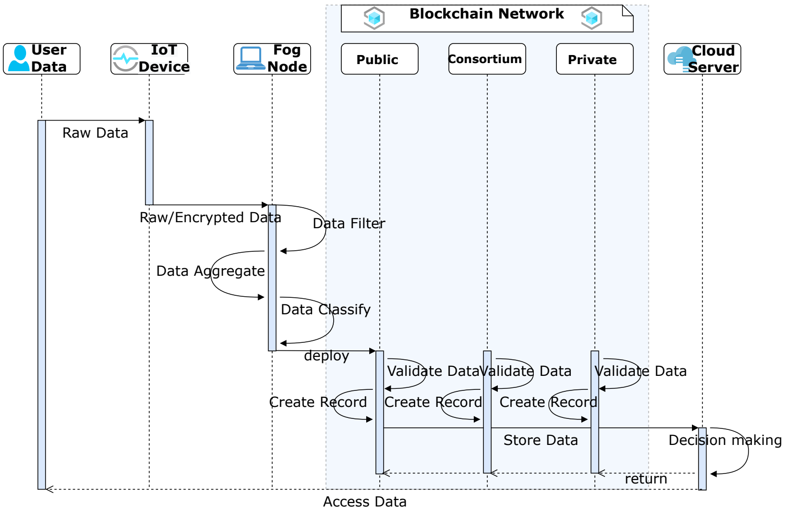


Figure 3.7: Logical flow execution of HierChain.

The steps in the execution flow of HierChain are illustrated in Figure 3.7. Fog nodes perform filtering and classification of data by associating a tag with it. The tags associated with the data can be either public, restricted, or confidential. Moreover, confidential and restricted data is encrypted either by the end-user or the fog node depending upon the preference of the user. We assume that the communication delay of delivering the messages from the source node to the fog node is very less and does not impact the overall execution time of the blockchain. After the raw user data is preprocessed and encrypted by the fog node, it is stored on the particular blockchain platform according to the tag associated with it (Algorithm 2). The data records are then validated by the blockchain's consensus algorithm in the form of blocks and recorded on the blockchain. This data can then be accessed by the cloud servers or users for storage or decision-making purposes according to the ACLs. We enforce ACLs using smart contracts that has conditions to decide whether access to data or resources is granted or denied. HierChain stores the tagged data on different blockchains (selected through the optimization problem) as follows:

- Ethereum: Ethereum is a public, open-source, and decentralized blockchain-based platform that uses the Proof-of-Work (PoW) consensus protocol [82]. Since it is highly secure due to active participation and decentralization of nodes, it is nearly impossible for intruders to gain control over the network. However, due to long transaction times and computationally intensive consensus methods, it faces the issue of scalability at the trade-off of high security. Therefore, HierChain uses it to store confidential data that requires the highest protection but consumes less network bandwidth.

- Hyperledger Sawtooth: Hyperledger Sawtooth is a Hyperledger project that supports two types of consensus algorithms: Practical Byzantine Fault Tolerance (pBFT) and Proof of Elapsed Time (PoET) [43]. Hyperledger Sawtooth is preferred because these consensus protocols are both fast and scalable. Therefore, we use it to store restricted data shared amongst a group of people within an organization.

- MultiChain: MultiChain is an open-source platform for the deployment and creation of private blockchains. The mining process in MultiChain is controlled by a set of identified validators. The round-robin mining protocol uses a simple and efficient consensus mechanism based on a round-robin algorithm where each node takes turns proposing the next block. MultiChain is used to store public data as it provides scalability and faster transactions at the cost of security.

3.5.2 Experimental Procedure

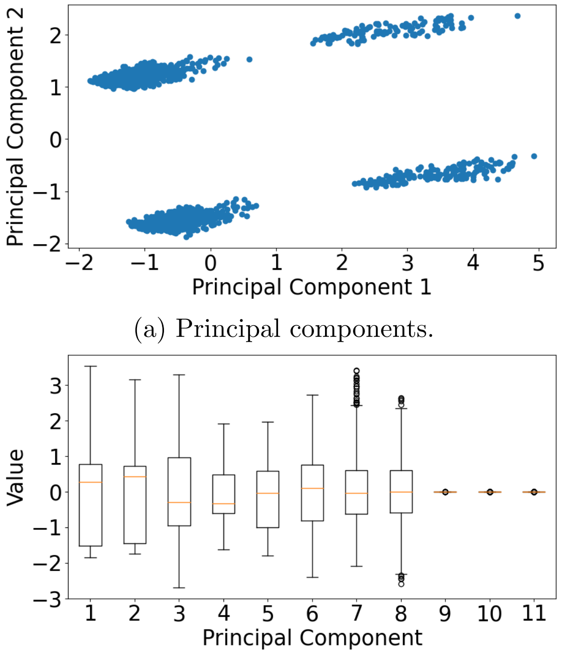
With the testbed and procedure explained in the previous section, we then proceed to test the performance of HierChain for healthcare data storage. We compare the performance of HierChain with 3 scenarios:

- All the data is stored using smart contracts on Ethereum platform.

- All the data is stored using smart contracts, called "chaincode", on Hyperledger Sawtooth platform.

- All the data is stored on MultiChain using streams.

Moreover, we also present the attack evaluation to provide a comprehensive understanding of the framework's resilience and shed light on the capabilities, motivations, and potential attack vectors. This evaluation will help uncover potential weaknesses in the HierChain framework and enable the development of effective countermeasures to mitigate security risks.



(a) Principal components.

(b) Outlier Detection.

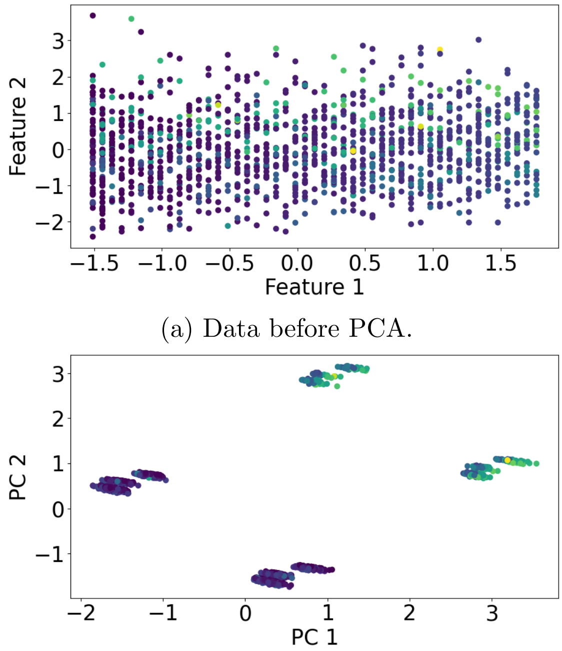
Figure 3.8: Data preprocessing using PCA.

3.6 Results and Discussions

The presented results aim to provide a scalable data management framework without compromising the security and performance of the system. We discuss the evaluation of HierChain in terms of performance and security in the following subsections.

3.6.1 Performance Evaluation

Fog nodes perform data preprocessing of data using the PCA technique to visualize the dataset and detect outliers that may affect the results of subsequent analysis. As we can see from Figure 3.8a, the first two principal components are linear combinations of the original variables, where the first component explains the largest possible variance, and the second component explains the second-largest possible variance. This plot helps to visualize any underlying patterns or clusters in the data that may be difficult to see in the original high-dimensional space. Figure 3.8b is a box plot, that identifies any potential outliers in the data. We can see that there are a few points in some columns that fall outside the whiskers, which suggests that they may be outliers.



(a) Data before PCA.

(b) Data after PCA.

Figure 3.9: Visualization of the effect of PCA on the data.

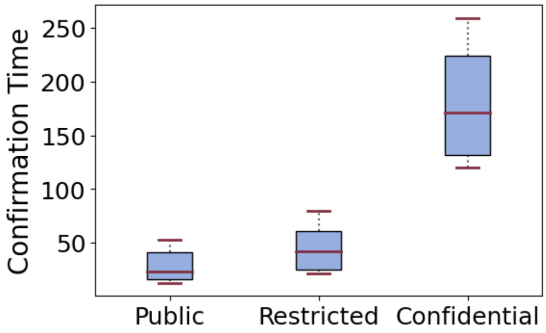


Figure 3.10: Confirmation time vs data sensitivity

The data appears to be relatively clean with a few outliers in the “age”, “bmi”, and “charges” features. These outliers are investigated further to determine if they are errors or legitimate data points. The “children”, “smoker”, and “region” features do not appear to have any outliers. Overall, Figure 3.9 shows a visualization of the effect of PCA to gain insights into the underlying patterns in the data. Figure 3.9a shows the original data with the two most important features (Feature 1 and Feature 2) and the target variable (charges) as the color scale. On the other hand, Figure 3.9b shows the data after PCA with the two principal components (PC 1 and PC 2), where each data point is represented by its two principal component scores, which are linear combinations of the original features. It shows that the data has been transformed and compressed into two dimensions while still preserving the variance of the original data.

We performed experiments with a different number of transactions to compare the performance of HierChain in terms of scalability. Figure 3.10 shows the confirmation times

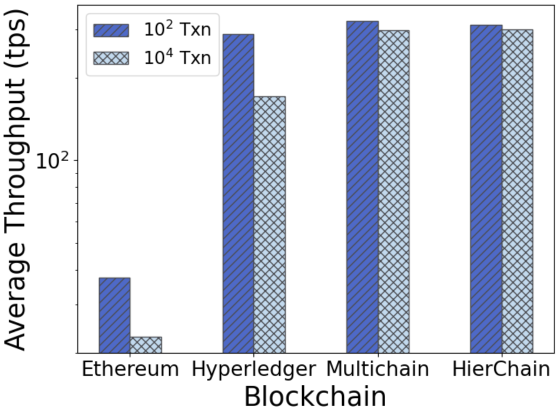


Figure 3.11: Average throughput vs Blockchain type.

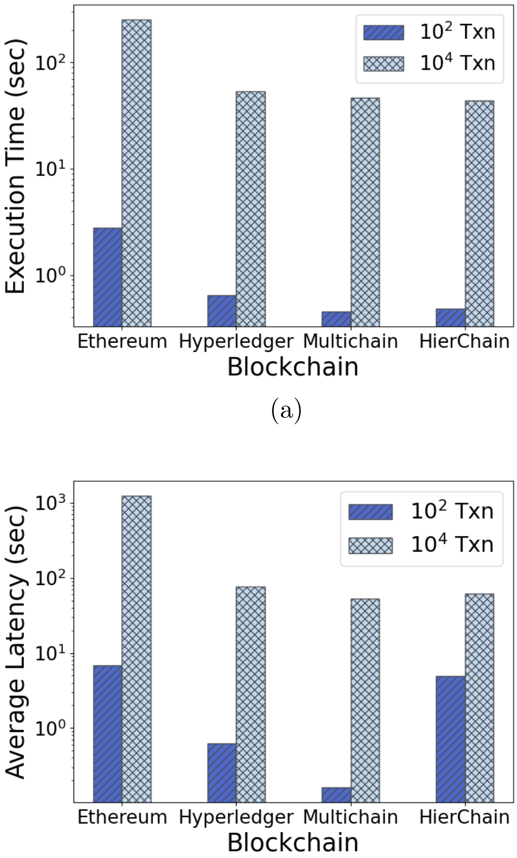
for different kinds of data. It shows that the average confirmation time for confidential data is the highest (around 171 seconds), followed by restricted and public data. This is because confidential data is stored on Ethereum (which is more computation-intensive) as it requires the highest amount of security. To compare the performance of HierChain, we use the following metrics:

- The throughput of a blockchain is the number of transactions or data items that can be processed and stored by the blockchain network within a specific time interval, typically measured in transactions per second (TPS) or data items per second. As shown in Figure 3.11, the throughput of HierChain is comparable to that of Multichain although it provides the security level of an Ethereum blockchain. Since Multichain is a private blockchain, it is controlled by a node or a group of nodes, while Ethereum uses a fully decentralized PoW consensus protocol. Figure 3.11 shows that when the number of transactions increases, the throughput of Ethereum and Hyperledger decreases considerably. However, HierChain provides an average throughput of 299.6 transactions per second even when the number of transactions is 10,000.

- The time it takes for a blockchain to execute and confirm all the transactions in a block is termed as the execution time. As the number of transactions increase, the execution time increases due to clogging of the network. Figure 3.12a shows that the execution time of Ethereum is very high (255.4 seconds) as compared to other methods. This is because the process of mining used in Ethereum is much more exhaustive as compared to Multichain and Hyperledger. HierChain has a comparatively less execution time in the order of 44.2 seconds even when the number of transactions are high due to its selective storage criteria.

- Latency in a blockchain can be defined as the difference between the deployment and completion time for a transaction. As represented by Figure 3.12b, Ethereum has the highest average latency as compared to other approaches because the generated block is broadcasted across all nodes in the network. The average latency of HierChain is 61.2 seconds when the number of transactions are 10,000. Although the security standards HierChain are high, its average latency is much less than Ethereum and comparable to a private blockchain.

Therefore, we can see that HierChain outperforms the existing methods as it provides the required scalability as well as data security through optimal storage of data.



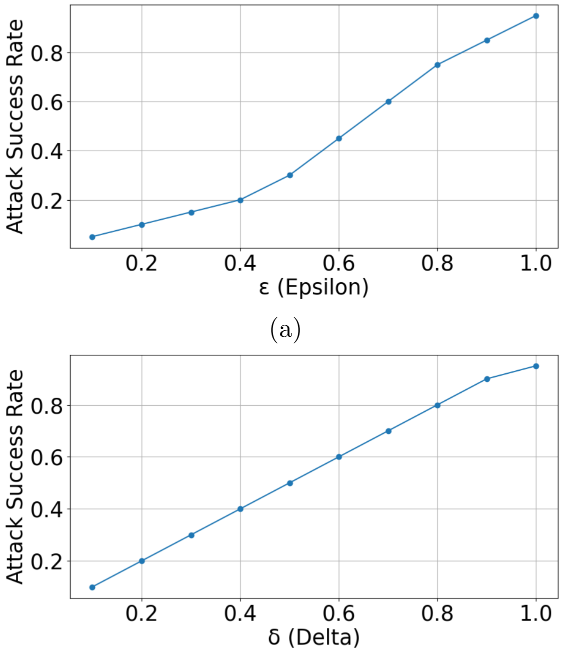
(a)

(b)

Figure 3.12: (a) Execution Time vs Blockchain type, (b) Average latency vs Blockchain type.

3.6.2 Attack Evaluation on Fog Layer

We define the following variables to analyse the security of the fog layer:
D: Original dataset containing sensitive healthcare data.
D': Perturbed dataset obtained by applying differential privacy mechanisms to D.
A: Attacker's knowledge or background information.
Q: Query function representing the analysis or computation performed on the dataset.
δ: Privacy parameter that represents the overall privacy guarantee of the differential privacy mechanism.
The attack model is characterized using the following descriptions:



(a)

(b)

Figure 3.13: Attack success rates by varying the privacy parameters.

1. Attacker's Knowledge: The attacker's knowledge or background information, denoted as A, can be represented as A = {A_1, A_2, ... , A_n}.

2. Attack Objective: The attacker's objective is to infer sensitive information from the perturbed dataset D' using the knowledge A and query function Q.

3. Security Analysis: The security analysis aims to quantify the risk of privacy breaches and the potential for reidentification attacks.

The attack success rate represents the likelihood of an attacker successfully breaching the privacy of the data despite the differential privacy measures in place. By varying the privacy parameters (ε and δ), we analyze the impact of these parameters on the effectiveness of the privacy protection mechanism. Therefore, we evaluate the resilience of fog layer and identify the optimal privacy parameters that minimize the attack success rate through Figures 3.13a and 3.13b. The y-axis of the graphs represents the attack success rate, which indicates the probability of an attacker successfully de-anonymizing or extracting sensitive information from the protected data. It helps determine the level of privacy protection needed for the fog layer model without significantly compromising the usefulness of the data for analysis or computation.

3.6.3 Attack Evaluation on Blockchain Layer

In this section, we examine the attack evaluation for the proposed HierChain framework by describing the related actors as follows:

Threat Actor

A threat actor is an attacker who wants to compromise the integrity and privacy of medical data stored on the HierChain system. It can be an adversary or a group of adversaries trying to steal the sensitive health data of users for unethical purposes.

Attack Vectors

- Sybil attack: The attacker could create multiple fake identities to gain control of the HierChain network and manipulate the classification of health data, leading to the storage of sensitive data on less secure blockchains.

- Data tampering: The attacker could modify the health data stored on the HierChain system, compromising its integrity and accuracy.

- Denial of Service (DoS) attack: The attacker could launch a DoS attack on the HierChain network, making it unavailable to medical professionals and causing disruptions in the diagnostic process.

Evaluation

- Sybil attack: Since HierChain saves the most sensitive data on Ethereum, we compare the Sybil attacks on Ethereum with Hyperledger and Multichain. We detect the Sybils using the k-means clustering algorithm [92] to group nodes that have similar transaction patterns. Nodes that belong to the same cluster are likely to be legitimate nodes, while nodes that do not belong to any cluster may be Sybil nodes. This technique cannot provide a perfect measure of the actual number of Sybil nodes but we use it for the calculation of a rough estimate. As we can see from Figure 3.14, Ethereum is less prone to Sybil attacks than MultiChain, therefore HierChain uses it to save confidential data. This is because public blockchains are generally considered to be more resistant to Sybil attacks than private blockchains since they rely on a decentralized network of nodes to validate transactions and maintain the integrity of the ledger. The decentralized nature of public blockchains makes it difficult for an attacker to create a large number of fake identities without being detected by the rest of the network.

- Data tampering: Although blockchain technology is known for its security and immutability, we use digital signatures and text encryption before storing sensitive data to ensure the integrity of the data stored on blockchain. Moreover, since HierChain preprocesses the data to remove redundancy using fog nodes, any malicious data injection would need to occur before the preprocessing stage to avoid detection.

- DoS attack: Since HierChain is a distributed system, a DoS attack would need to target multiple nodes simultaneously to cause disruption to the network. Moreover, in the event of an attack, we could use load-balancing to use a different blockchain to store data temporarily for instant availability.

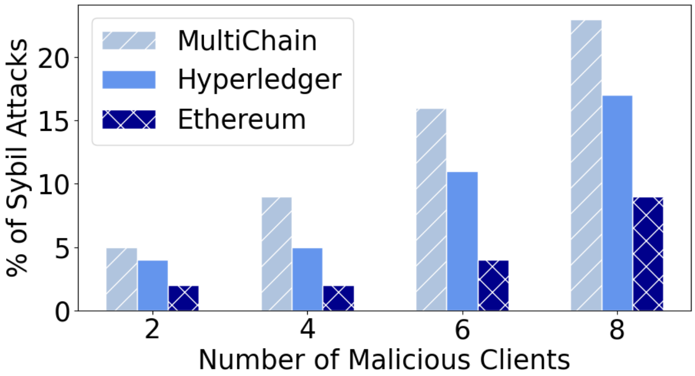


Figure 3.14: Comparison of Sybil attacks on different blockchains.

Overall, HierChain has the potential to address the challenges faced by healthcare systems in the secure storage and sharing of medical data.

3.6.4 Security Assessment

Security of HierChain can be analyzed through the STRIDE threat model [93]. There are two possible ways to perform STRIDE-based threat modeling [93]: STRIDE-per-interaction and STRIDE-per-element. Since STRIDE-per-element requires analysing the operations and behavior of each component in the system, it is not suitable for our framework. Therefore, we perform STRIDE-per-interaction that considers the risks against system interactions and its prevention strategies are generally sufficient to protect a system (as cyber-attacks are normally caused due to malicious interactions).

STRIDE is a mnemonic for six categories of threats– Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service (DoS), and Elevation of Privilege. Every device and user who wants to store or access the data is first authenticated by HierChain through digital signatures. Due to this pre-verification step, no user could spoof or impersonate someone else. Since blockchain is an immutable ledger for data storage, tampering with data is almost impossible without being detected. All the nodes are pre-verified in HierChain and every posted transaction is signed with the identity provided to the node. Therefore, no user can claim non-repudiation of his actions in the network. HierChain prevents information disclosure to unauthorized entities by giving complete control to the users on who can access their data. Since we are storing the data on different blockchains, so even if one blockchain suffers from the DoS attack, the whole system will not come down to a halt. Elevation of Privilege is prevented in HierChain as it gives access to information only to authorized users. Moreover, all the sensitive information is encrypted before being stored on the blockchain layer. Therefore, we can conclude that the proposed framework fulfills the security constraints according to the STRIDE model.

3.7 Summary

In this chapter, we developed a novel approach that combines the advantages of different blockchains and fog computing to store the data records in an optimal manner. The main focus of this work was to overcome the scalability challenges of existing blockchain systems while maintaining data security and the performance of the network. Our comprehensive evaluation aims to provide valuable insights into the practical implications of HierChain in healthcare context proving its viability for real-world deployment and its potential to safeguard sensitive medical data. Although blockchain can keep the data secure, it does not guarantee the privacy of data if an attacker is successful in eavesdropping any unencrypted data. The next chapter explores the privacy aspect of IoT-based applications.

Chapter 4

Data Privacy in IoT Applications

In the previous chapter, we focused on the security challenges associated with IoT, particularly through the potential of blockchain technology to enhance data integrity and confidentiality. While ensuring security is crucial, it is equally important to address the privacy concerns regarding user data in IoT systems. This chapter focuses on privacy concerns within IoT and how to address them using the concept of FL.

In this chapter, we consider another application of IoT, the smart grid, and experiment on energy-related data to enhance energy efficiency while maintaining user data privacy. Specifically, we use Non-Intrusive Load Monitoring (NILM) to analyze detailed energy usage data to identify individual appliance loads, offering significant benefits for energy management and efficiency. While our discussion includes a focus on NILM, the privacy architecture we propose is designed to be application-agnostic, suitable for a wide range of IoT applications. This chapter underlines the essential balance between utilizing the potential of IoT technologies and protecting the privacy of individuals in an increasingly interconnected environment.

4.1 FL for Data Privacy

Electric utilities were historically grappling with the challenge of increasing efficiency and detecting outages, because they had limited visibility into their networks, especially in the last mile of distribution. With the advent of smart meters and advanced metering infrastructure [94], they were finally able to gather high-resolution, aggregate data from their customers for billing purposes in addition to generating useful insights, from monitoring aberrant power usage patterns and developing virtual energy audits to demand response. Some of these applications rely on the power consumption profile of home appliances, which is not readily available and must be inferred from the aggregate data.

Non-Intrusive Load Monitoring (NILM) [95] is the problem of disaggregating the total household energy use, measured by a smart meter, into the load of individual appliances. It helps reduce the energy consumption and electricity bill of homeowners by providing real-time feedback on appliance-level power consumption [96], improving the load forecasting accuracy, and suggesting load shifting strategies during peak hours [97]. While there are several promising techniques for NILM, including signal processing and probabilistic graphical models, deep learning, and in particular the attention mechanism, has been shown to be more effective [98]. One such deep learning model is based on the notion of sequence-to-sequence (seq2seq) translation, in which a sequence of words, e.g. in one language, is mapped to a sequence in another language. By analogy, in energy disaggregation, the seq2seq translation can be used to map the sequence of the aggregate household demand into the power consumption of individual home appliances. However, traditional seq2seq models often struggle with long input sequences because they use Recurrent Neural Networks (RNNs), which process the input one token at a time. This processing can become computationally expensive for long sequences and make it difficult for the model to learn dependencies between tokens that are far apart in the input sequence [99]. Transformers, however, handle long input sequences more efficiently by processing all tokens in parallel [100]. They use a self-attention mechanism to weigh the importance of each token in the input sequence for each output token, allowing the model to capture long-range dependencies more effectively. This makes transformers a more suitable choice for NILM, where the input sequences can be long and complex, and the model needs to learn dependencies between different parts of the sequence to accurately identify individual loads [101].

Regardless of which machine learning model is used for NILM, sending the smart meter data along with groundtruth power profiles of home appliances to a remote server that trains or runs this model could raise privacy concerns. This is because a passive adversary or an intruder can obtain appliance-level information and use this private information to learn the user's habits and lifestyle, such as when they come home, their preferred temperature setting and activities of daily living [102, 103]. To address these privacy concerns and enable training the NILM model on decentralized data that belong to many clients, possibly with unique appliances and usage patterns, Federated Learning (FL) has been adopted in recent work [104, 105, 103, 106]. In this framework, the NILM model can be trained by the clients in a collaborative fashion, without requiring them to share their data with a central server [39]. The global model is built by aggregating the model updates performed independently by the clients on their local data.

While FL addresses the concern related to sharing raw data with a central server, there are still several challenges that limit its real-world application. Firstly, FL algorithms are prone to privacy attacks during the exchange of parameters between clients and the aggregation server [107, 108]. The server might be Honest-But-Curious (HBC), i.e., a passive adversary that follows the aggregation protocol but takes a peek at the clients' updates to extract additional (private) information. Different techniques have been proposed to mitigate this attack, from differential privacy to homomorphic encryption and secure multiparty computation [109]. Secondly, malicious clients may attack the FL model by sending incorrect updates to the server during the training process. This client-side attack has multiple types, namely the poisoning attack, model inversion attack, membership inference attack, and backdoor attack [110]. However, existing defense mechanisms against malicious clients in FL are inadequate and more research must be done on developing robust defense mechanisms against adversarial attacks, particularly those that are difficult to detect or prevent [108]. Moreover, the current FL-based frameworks are not fully effective in the presence of heterogeneity, e.g. when homes contain different

appliances or exhibit dissimilar feature distributions [111, 112].
Recent advances in FL motivated researchers to apply this paradigm to NILM [105, 103, 106], but the related work builds on the standard FL framework, failing to address the specific challenges of FL in NILM, namely dishonest clients and heterogeneity. We address this gap in the literature by introducing a robust and privacy-aware FL-based NILM framework. The main contributions in this chapter are summarized below.

- We propose a robust NILM framework based on FL. We use the bidirectional transformer architecture that follows the pattern of sequence-to-sequence learning for energy disaggregation to achieve better performance in terms of the diverse smart meter clients. To address the challenge of data heterogeneity, we use Model-Agnostic Meta-Learning (MAML) which allows fast and dynamic adaptation of the model according to the client updates.

- To make the model robust against dishonest clients, we devise a reputation scheme for the selective sampling of clients in every round of FL based on their gradient updates during the global model training.

- Through extensive experiments, we corroborate the effectiveness of the proposed FL-based NILM framework by comparing it with the centrally-trained model.

4.2 Preliminaries

4.2.1 Non-Intrusive Load Monitoring (NILM)

The aim of NILM is to identify the electricity consumption of individual appliances using aggregate data, e.g. from a smart meter. Since it requires only a single point of measurement and no extra equipment needs to be installed in the house, this identification technique is deemed non-intrusive. The aggregated power load for a home is time-series data denoted by $P = [p_1, p_2, \cdots, p_T]$ where p_t represents the smart meter reading at time t. This value is the sum total of the energy usage of all appliances that were not in the OFF state. Suppose there are I appliances in a given home, the total power consumption at t is given by

$$p_t = \alpha_t + \sum_{i=1}^{I} e_t^i, \tag{4.1}$$

where e_t^i is the energy usage of i^{th} appliance and α_t represents the measurement error, which is assumed to be small. The disaggregation problem concerns recovering $E^i = [e_1^i, \cdots, e_T^i]$ for every appliance i from the aggregate measurement $P = [p_1 \cdots, p_T]$. Thus, NILM algorithms approximate a function G that performs the following mapping for every time slot:

$$G(p_t) = [e_t^1, \cdots, e_t^i, \cdots, e_t^I]. \tag{4.2}$$

For the disaggregation task, each target appliance must have a threshold value so that we can determine if it is in the ON or OFF state. Hence, the state for each appliance depends on the specified minimum on and off duration, maximum power, and the on-power threshold. In our experiments, the ON/OFF states are determined by a simple comparison with the on-status thresholds, but the status changes are considered valid only when they last longer than the minimum ON and minimum OFF duration. The state s_t^i of an appliance i at time t is determined as

$$s_t^i = \begin{cases} 1, & e_t^i \geq \lambda^i \\ -1, & otherwise \end{cases} \tag{4.3}$$

where λ^i represents the threshold value of the i^{th} appliance, and -1 and 1 correspond to the OFF and ON states, respectively.

We note that disaggregation is a regression task in which the energy consumption of individual appliances is estimated. Once the estimates are obtained, they are utilized to determine the state of each appliance. Therefore, NILM is simultaneously a regression task and a status classification task. This inspired the choice of the loss function as we discuss in Section 4.4.3.

Ethical Considerations: In our FL framework for NILM, ethical concerns extend beyond data privacy and include fairness. The risk of data breaches and unauthorized access to sensitive information poses significant challenges to the NILM community. Using standard and widely used and recognized datasets, we ensure the adherence to privacy standards. These datasets have undergone necessary ethical reviews and clearances, particularly with respect to data collection, and are anonymized to protect individual privacy, aligning with standards for research involving human subjects. Furthermore, our use of two representative NILM datasets, which are collected in two countries from households that may contain different types of each appliance, is a step towards addressing issues of fair and equitable representation.

4.2.2 Federated Learning (FL) and Non-IID Data

The idea of FL is to take advantage of decentralized data and compute power of client devices to train a machine learning model that achieves high accuracy on the dataset of every individual client. One of the popular algorithms used to aggregate model updates from multiple clients is federated averaging (FedAvg) [39]. Concretely, in FedAvg, a server calculates the average of all updates received from clients in every round to obtain a global model. The global model is sent to the clients that participate in the next round so they can further update it according to their own data distribution. Given a loss function *f*, the FL's objective can be written as:

$$\min_{w} \; f(w), \quad \text{where } f(w) = \frac{1}{n}\sum_{i=1}^{n} f(X_i, Y_i; w), \tag{4.4}$$

where $f(X_i, Y_i; w)$ is the prediction loss on samples of client i, denoted $\{X_i, Y_i\}$, with w being the vector of model parameters and n being the number of participating clients in one round of FL. We assume that clients' datasets have the following characteristics:

- Non-IID: Every client's data may be from a different distribution, such that the data points and labels available locally do not match the global distribution.

- Unbalanced: The number of training samples in the dataset of each client may vary drastically depending on the amount of data they hold.

In the NILM application, clients will likely have different type and number of appliances, e.g., some customers may not have a microwave at all while others have microwaves of different makes and models. Hence, it is reasonable to assume that clients' datasets are independent and not identically distributed.

Some of the existing works use FedProx [113] for handling non-iid data, which introduces an additional regularization term, known as the proximal term, to the standard federated averaging process. However, FedProx relies on hyperparameter tuning that can significantly impact its performance. Selecting appropriate values for hyperparameters, such as the proximal term weight and learning rate, can be nontrivial. Model-Agnostic Meta-Learning (MAML) [114], on the other hand, aims to learn a general initialization that can be quickly fine-tuned to new clients or tasks. This enables MAML to handle variations and imbalances in non-IID data more effectively compared to FedProx. This motivates the use of MAML for handling heterogeneity in this work.

4.2.3 Threat Model

In this chapter, we assume the aggregation server is honest, but clients can be dishonest (malicious). The dishonest clients manipulate their own data, but they cannot observe or manipulate the data of other clients. Multiple dishonest clients may collude and form a group of *sybils* to perform coordinated attacks in federated learning. Non-colluding adversaries can control multiple sets of sybils to carry out poisoning attacks concurrently. We assume that every class of data required in the global model is included in the dataset of at least one honest client. This assumption is necessary because, in the absence of any honest client, the model would not be able to learn anything about the correct classes in the first place.

We primarily focus on targeted poisoning attacks where the attacker sends malicious updates to manipulate the parameters of the global model. The goal of the attacker is to increase the chance of one class being classified incorrectly without changing the probabilities of other classes. This can be done using the label-flipping strategy [110].

4.2.4 Motivations for the Poisoning Attack

The rationale behind poisoning attacks on NILM-based systems is either to obtain economic advantages or avoid regulatory compliance issues. One prominent motivation is the exploitation of virtual energy auditing [115]. Here, NILM applications, which analyze appliance-level energy consumption, could be manipulated by attackers possibly affiliated with appliance manufacturers. Attackers could deceive homeowners into purchasing new, seemingly more efficient models by tampering with data to show false inefficiencies in appliances like air conditioners. This reveals a strong economic reason for engaging in poisoning attacks. Moreover, attackers could manipulate NILM data to show reduced energy consumption to falsely claim incentives offered for buying energy-efficient appliances. In commercial contexts, rival companies might engage in poisoning attacks to impair the NILM systems of competitors, thereby gaining an unfair market advantage. Furthermore, attackers could alter energy consumption patterns to either mask aberrant characteristics or create fictitious profiles of a household's energy usage to satisfy regulatory requirements.

4.3 The Proposed Solution: Reputation-based Aggregation and Selection

Suppose there are several homes that have different kinds of appliances, record their electricity consumption using a smart meter, and are interested in training an accurate NILM model in a collaborative fashion. These homes are the clients in FL. There is also an aggregation server, presumably owned by the NILM service provider, that receives all the updates from the clients. We outline steps of the proposed FL framework below:

1. The model parameters are initialized and the first round of training begins. At the central FL server, parameters of the global model are initialized randomly and sent to participating clients (households) as illustrated in Figure 4.1.

2. The households train the received global model on their own data (locally). They follow the BERT deep learning model for NILM as discussed in the next section. As shown in Step 2 of Figure 4.1, the local model updates are then sent to the global server for model aggregation.

3. The FL server uses the FedAvg algorithm to update the global model. For robust and fault-tolerant aggregation of the updates, we adopt a reputation model and a client sampling technique that takes into account clients' reputation (Steps 3 and 4 of Figure 4.1) as described in Section 4.3.2.

4. The new aggregated optimal model (Step 5 of Figure 4.1) is then broadcast to the clients and this process repeats until a stopping criterion is met (e.g. maximum number of training rounds reached).

4.3.1 The BERT Model for NILM

The model we choose for NILM is the Bidirectional Encoder Representations from Transformers (BERT) model with sequence-to-sequence learning. This model is considered

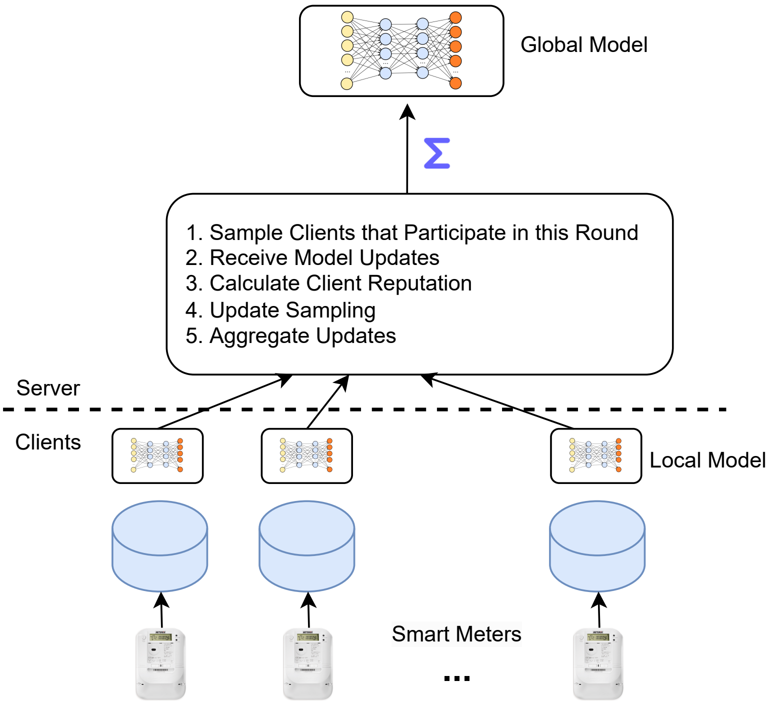


Figure 4.1: Illustration of the proposed mechanism.

the state-of-the-art and outperforms the other NILM techniques, according to different metrics [98]. The BERT model, with its foundation in transformer architectures, presents a significant advancement over traditional neural network models such as RNNs. Transformers employ the self-attention mechanism, allowing the model to process entire sequences of data simultaneously. This contrasts with older sequence processing methods that handle one data point at a time. The attention mechanism in transformers enable dynamic focus on different parts of the input sequence, which is essential for understanding context and relationships within data. BERT builds upon this by analyzing data bidirectionally, considering both prior and subsequent information in the sequence. This bidirectional processing is particularly effective in NILM, where understanding the sequential context of energy usage is crucial for accurate predictions.

The basis of the model is BERT [116] which consists of an embedding module, transformer layers, and an output multilayer perceptron (MLP). It is shown in [116] that the bidirectional model attains a deeper understanding of the context compared to unidirectional models. The model takes fixed-length sequential data as input and predicts the energy usage of individual appliances, an output of the same shape. The on-power thresholds can then be used to determine the state of each appliance.

The transduction model has an encoder-decoder architecture, where the encoder maps an input sequence to a continuous sequence of symbols. The decoder then uses the encoding to generate an output sequence of symbols by spitting out one element at a time. In the BERT model, the input data is first mapped to a convolutional output by extracting relevant features from the one-dimensional input sequence and increasing the dimensionality of the hidden representation sequence. By employing this

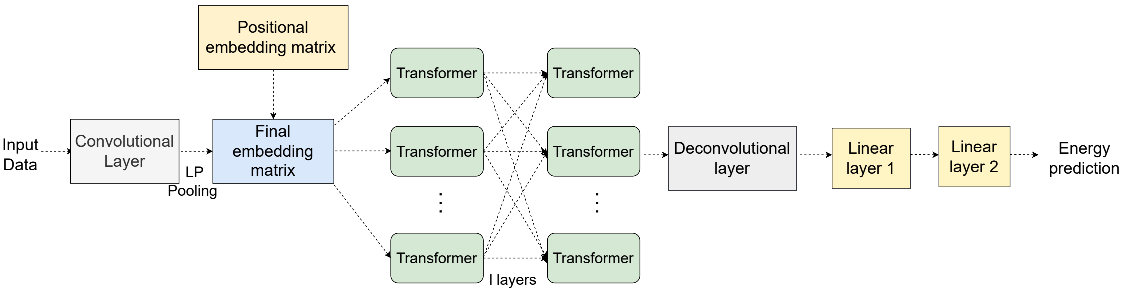


Figure 4.2: Architecture of the BERT model used in this work.

convolutional layer, the network is able to capture important patterns and enhance the latent representation of the input data. This layer is then pooled and added to a positional embedding matrix that makes up the sequence positional encoding. The formed embedding matrix is then sent to a bidirectional transformer consisting of several layers of transformers and attention heads within each layer. After the multi-head attention operation in every transformer layer, the previous matrix is further passed through a position-wise feed-forward network (PFFN). By incorporating the PFFN into the architecture, the model captures intricate relationships and patterns within the matrix, leading to improved learning and representation capabilities. Finally, the output MLP consists of a deconvolutional layer followed by two linear layers. The various layers of the BERT model are shown in Figure 4.2. The training process of BERT involves masking a portion of the input sequence with a special token. This random masking, where a fraction of input elements are masked, allows the model to learn from the surrounding context and predict the masked items. By focusing on the output results from the masked positions, the model is compelled to capture significant patterns from the whole input sequence. Energy prediction is done by the output values multiplied by the maximal device power while corresponding on-power thresholds are used to obtain the appliance status.

4.3.2 Reputation-based Aggregation for FL

We propose a reputation-based mechanism for the detection of dishonest clients at the aggregation server. Let $c_{j,t}$ be the gradient received at the server from client j in iteration t.

Detecting malicious clients

Sybils in FL provide updates that lead to the poisoning of the global model towards a common objective. In non-IID case, one key insight is that the diversity of the gradient updates can be used to separate honest clients from sybils. Specifically, since the data distribution of each client is different, the updates from sybils will be more similar to each other compared to the ones from honest clients [117]. Using this insight, we assign a reputation value to clients that help in the aggregation process by reducing the chances of suspicious clients being selected. As described in Algorithm 3, cosine similarity is used to calculate the angular distance between client updates. We choose cosine similarity over

Euclidean distance because the magnitude of a gradient can be manipulated by sybils to achieve dissimilarity. But they cannot easily change its direction without reducing the effectiveness of the attack. We maintain a history for each client and update the history vector at every iteration into a single aggregated gradient. Thus, instead of using just the update from the current iteration, cosine similarity is calculated using the aggregated historical updates. Finally, the maximum cosine similarity a client has with some other client is compared to the average cosine similarity to calculate the reputation of that client.

Algorithm 3 Detecting malicious clients at server

```
 1: Input: Initial history $h_j = c_{j,1}$, initial reputation $r_j = 1$.
 2: for iteration $t$ do
 3:     for every client $j$ do
 4:         Receive $c_{\mathrm{j,t}}$ for this round and append it to $h_{\mathrm{j}}$
 5:         while $i \neq j$ do
 6:             $s_{\mathrm{j,i}} \leftarrow h_{\mathrm{j}} \cdot h_{\mathrm{i}} / (\|h_{\mathrm{j}}\| \|h_{\mathrm{i}}\|)$
 7:         $w_{\mathrm{j}} \leftarrow max_{\mathrm{i}}(s_{\mathrm{j}})$          ▷ max cosine similarity of client j
 8:     $\tau_{\mathrm{t}} \leftarrow$ average of $w$ for all clients
 9:     for every client j do
10:         if $w_{\mathrm{j}} > \tau_{\mathrm{t}}$ then
11:             $r_{\mathrm{j}} \leftarrow r_{\mathrm{j}} - \delta \cdot t \log t$
12:         else
13:             $r_{\mathrm{j}} \leftarrow r_{\mathrm{j}} + \delta \cdot t \log t$
14:         if $r_{\mathrm{j}} \geq \beta$ then
15:             Client $j$ can be chosen for aggregation
16:     Select the most reputable $K$ clients for aggregation
```

Reputation calculation and client selection

The selection of clients for an iteration is based on their calculated reputation for high accuracy and robust model training. For non-IID data, the reputation is calculated on the basis of cosine similarity with other clients. The value of reputation, r_j is compared to the average of the maximum cosine similarity of all clients and computed according to Steps 11 and 13 of Algorithm 3. In this algorithm, δ represents a hyperparameter that can be tuned to adjust the rate at which reputations are updated. The choice of $t \log t$ as the reputation update function in the algorithm allows for a gradual and balanced growth of the reputation score that reflects the long-term contributions and reliability of each client. It grows faster than linear but slower than quadratic or exponential functions, making it a better choice for updating the reputation. Since the model should converge to the optimal point, changes in reputation become more substantial as the number of iterations increases.

If the reputation score of a client j becomes greater than or equal to the threshold β (determined experimentally), it means this client can be chosen for aggregation, as it is considered reputable (Step 14 of the algorithm). Finally, we select K number of clients with the highest reputation scores for aggregation in this iteration. By iteratively updating the reputation scores based on the similarity between client histories and selecting the most reputable clients for aggregation, the algorithm aims to identify and choose reliable clients, while detecting potentially malicious or unreliable ones. The reputation scores serve as a measure of trustworthiness, and the threshold β determines the minimum reputation required for a client to affect the global model.

Adapting to time-varying clients' behavior

An assumption made in our work is that clients do not change their type, which dictates their behavior, during model training; e.g. an honest client will not start behaving maliciously and vice versa. This can be justified given the short duration of the training phase, which makes type changes unlikely to occur. But this assumption can be relaxed, and clients' reputation can be updated differently to address the following conditions: a) A client initially deemed trustworthy could begin to exhibit malicious behavior; b) A client initially categorized as malicious could later demonstrate normal behavior.

To enhance the adaptability of our framework to behavior changes, our reputation mechanism can be modified to place greater emphasis on the most recent contributions from clients, rather than a cumulative analysis of their entire historical data, enabling a more agile and current reflection of a client's reliability. This can be achieved by considering a sliding window for calculating each user's reputation or implementing exponential smoothing as a forgiveness mechanism.

4.3.3 Model Agnostic Meta-Learning for FL

In meta learning, which is a "learning to learn" approach, the goal is to train a model (i.e., the main task) by learning from multiple related subtasks. This model can adapt quickly to new tasks by making use of only a few training iterations and data points. We use the MAML algorithm of [114] as it is compatible with task definition in FL and allows us to address the data heterogeneity challenge. Concretely, in our NILM framework, each subtask involves training the described BERT model on a client's electricity data. The global NILM model is optimized towards the direction that could quickly adapt to all subtasks, each associated with a user's local data. This enables better generalization to heterogeneous data.

The benefits of meta-learning in the NILM framework are manifold. First, it allows the global NILM model to exploit the knowledge gained from training on multiple client datasets, enabling better generalization to heterogeneous data. This is particularly advantageous in FL, where clients may have different types of appliances, energy usage patterns, or household characteristics. Furthermore, meta-learning can enhance the efficiency of the overall NILM process. Since the global model is pre-trained on a diverse set of subtasks, it can quickly adapt to new client data with only a few iterations and data points. Additionally, meta-learning enables knowledge transfer across clients. The global NILM model can capture common patterns and dependencies across different households, appliances, or energy usage scenarios. Overall, meta-learning in the NILM framework enhances adaptability and generalization capabilities of the global model.

The model parameters are trained by minimizing the meta-loss, which measures the performance of the adapted model $f_{\theta_0 i}$ with respect to the model parameters θ across a set of tasks sampled from the distribution $p(T)$. This meta-loss can be expressed as [114]:

$$\min_{\theta} \sum_{T_i \sim p(T)} \ell_{T_i}(f_{\theta_0 i}) = \sum_{T_i \sim p(T)} \ell_{T_i}(f_{\theta - \alpha \nabla_\theta L_{T_i}(f_\theta)}) \tag{4.5}$$

Here, ℓ_{T_i} represents the loss function for each task T_i (given in Eq. (4.9)), f_θ denotes the model with parameters θ, $\theta_0 i$ represents the updated model parameters obtained through gradient descent updates, and α is the step size or learning rate. In summary, our objective is to find the optimal model parameters θ that lead to the best result when updated using one step of gradient descent on subtasks. This process facilitates the global model's ability to adapt swiftly to individual client tasks with minimal fine-tuning.

4.4 Performance Evaluation

4.4.1 Dataset and Preprocessing

We used two real-world energy datasets to evaluate the performance of our framework:

- REDD [118]: The Reference Energy Disaggregation Dataset consists of the electricity consumption data for 6 real houses in the U.S. over several months, with the sampling period of 1s for mains and 6s for appliances.

- UK-DALE [119]: The UK-Domestic Appliance-Level Electricity consists of data from 5 houses in the UK with a sampling period of 1s for mains and 6s for appliances.

These datasets were selected due to their status as standard benchmarks in the field of NILM, offering comprehensive insights from diverse environments. These datasets include both individual appliance and aggregated consumption data, making them appropriate for use in training and test phases [120]. Moreover, they encompass a broad range of appliances, providing a more extensive and varied dataset compared to others that may lack complete appliance coverage or proper labeling. This diversity not only adds reliability to our evaluation but also allows us to effectively test our framework in non-IID scenarios, which is critical for assessing performance in real-world settings.

For the REDD dataset, we choose four specific appliances for training our model: microwave, dishwasher, washer and dryer, and refrigerator. For the UK-DALE dataset, we also include the kettle along with these four appliances. Similar to the preprocessing of BERT4NILM [98], the raw data is resampled and clamped to specify the minimum on- and off-duration, on-threshold, and maximum power of each appliance as given in Table 4.1. The ON/OFF status of each appliance is determined by a comparison between the received data and the on-power thresholds, with the status changes being valid if they

Table 4.1: Overview of different appliance values.

Dataset	Appliance	Max Power (W)	On-power threshold (W)	Minimum on Duration (s)	Minimum off Duration (s)
REDD	Microwave	1800	200	12	30
	Dishwasher	1200	10	1800	1800
	Washer	500	20	1800	160
	Fridge	400	50	60	12
UK-DALE	Microwave	3000	200	12	30
	Dishwasher	2500	10	1800	1800
	Washer	2500	20	1800	160
	Fridge	300	50	60	12
	Kettle	3100	2000	12	0

Table 4.2: Dataset details for the appliances in REDD Dataset

House No.	1	2	3	4	5	6
# Appliances	18	9	20	18	24	15
Microwave	✓	✓	✓		✓	
Dishwasher	✓	✓	✓	✓	✓	✓
Washer	✓	✓	✓	✓	✓	✓
Fridge	✓	✓	✓		✓	✓

Table 4.3: Dataset details for the appliances in UK-DALE Dataset

House No.	1	2	3	4	5
# Appliances	52	19	4	11	24
Microwave	✓	✓		✓	✓
Dishwasher	✓	✓			✓
Washer	✓	✓		✓	✓
Fridge	✓	✓		✓	✓
Kettle	✓	✓	✓	✓	✓

last longer than the minimum on/off duration. In the case of REDD, house number 1 is used for testing and other houses are used for training, whereas in UK-DALE, house number 2 is used for testing and other houses are used for training. Different appliances present in the given houses of these datasets are listed in Tables 4.2 and 4.3.

4.4.2 Baselines

To assess the performance of our proposed approach, we compare with the disaggregation results of the following learning schemes while keeping the neural network architecture unchanged:

- **Centrally-trained model:** The centrally-trained model is the primitive form of ML training where raw data from all the households is aggregated and processed at a central location. Only one central model is trained and tested for all the clients.

- **FL-based NILM model (using FedAvg):** Each client uses its raw dataset to train a local model and sends model updates to the aggregation server. The server then aggregates the updates via FedAvg and trains the final NILM model over several iterations (100 to 150) between the clients and server. Hence, the clients do not need to share their actual data with the server. This baseline is used for evaluating the following two schemes in the proposed framework for non-iid datasets:

  - FL-based NILM in the presence of malicious clients: An adversary can impersonate the clients participating in the aggregation process leading to a targeted attack, i.e., the label flipping attack.
  - Robust FL-based NILM: The clients can be adversarial aiming to poison the model. We use the aggregation technique outlined in Algorithm 3.

4.4.3 Federated Experiments

We evaluate our framework using the sequence-to-sequence (seq2seq) benchmark evaluation and train the BERT model for NILM. In this chapter, we have used the PyTorch implementation of BERT4NILM as our base model [1]. To train our model, we split the dataset differently for IID and non-IID settings to simulate a higher number of clients for accurate training as follows:

- IID Scenario: The data from a single house in our dataset is split day-wise and distributed proportionally amongst n clients. This is because data from a house will have the same probability distribution and is therefore suitable for the IID scenario.

- Non-IID Scenario: We know that the same appliances from different houses can differ in many aspects, such as voltage and power profiles, energy efficiency, etc. Therefore, for the data to be split in a true non-IID fashion, we assign the data of one appliance from every house to each client.

We set the default sampling period to 6s with a learning rate of 10^{-4} for training the model. We use Adam as the optimization function as it performs better by faster convergence and requires lesser parameters for tuning. The loss function used for training (in all learning schemes) is described next.

[1] `https://github.com/Yueeeeeeee/BERT4NILM`

Loss Function

Following [98], the loss function we use for training the BERT model has multiple terms, each described below.

The first one is the Mean Square Error (MSE) loss for observed and predicted power usage values, which is given by:

$$\ell_{\text{mse}} = \frac{1}{T}\sum_{t=1}^{T}(\hat{e}_t^i - e_t^i)^2, \tag{4.6}$$

where T is the disaggregation length, i.e., the total number of time steps, $\hat{e}_t^i$ is the predicted energy usage of appliance i at time t, and e_t^i is the corresponding observation as described in Section 4.2. The energy usage values $\hat{e}_t^i, e_t^i$ are normalized between 0 and 1 by dividing them by the maximum power limit. This ensures that the power usage sequences are comparable and consistent across different appliances, regardless of their individual power profile.

To minimize the relative entropy between the predicted and observed power usage, we also incorporate the Kullback–Leibler (KL) divergence in the loss function. The tempered softmax operation applies a temperature parameter to the softmax function that converts a vector of real numbers into a probability distribution. Since electrical appliances are frequently in an off state, we have chosen a temperature parameter of 0.1 to account for the distinction between on-loads and off-loads. This adjustment aims to enhance the performance of the model on error metrics, especially for rarely utilized appliances such as the kettle. A hyper-parameter η is introduced to fine-tune the temperature for our designed loss function. It can be given mathematically as:

$$\ell_{\text{kl}} = D_{\text{KL}}(softmax(\frac{\hat{e}_t^i}{\eta})||softmax(\frac{e_t^i}{\eta})) \tag{4.7}$$

Finally, to reduce the effect of misclassification and penalize inconsistent predictions, we consider a soft-margin loss and an L1 term which are given by:

$$\ell_{\text{sm}} = \frac{1}{T}\sum_{t=1}^{T} log(1 + exp(-s_t^i \hat{s}_t^i)) \tag{4.8}$$

such that s_t^i is the state label of an appliance as described in Section 4.2 and $\hat{s}_t^i$ is the corresponding prediction.

Putting these together, the loss function used to train the BERT model for NILM is as follows:

$$\ell_{total} = \ell_{\text{mse}} + \ell_{\text{kl}} + \ell_{\text{sm}} \tag{4.9}$$

This loss is specifically designed to encourage the model to make more accurate and consistent predictions, reducing the impact of misclassification. Note that the above loss function is specific to a single appliance, and during training, the losses for all appliances are summed up to compute the total loss.

Table 4.4: Parameters used in our experiments

Parameter	**Value/Description**
Datasets	REDD and UK-DALE
Sampling Period	1s for mains, 6s for appliances
Learning Rate	10^{-4}
Optimization Function	Adam
Number of Epochs	150
Batch Size	128
Number of Clients	20 to 100
Model Architecture	BERT4NILM
Software Environment	PyTorch 1.7
Loss Function	MSE + KL divergence + soft-margin loss
Aggregation Algorithm	FedAvg

Table 4.5: Average performance scores for REDD

	Microwave			Dishwasher			Washer			Fridge		
	Accuracy	F1	MAE	Accuracy	F1	MAE	Accuracy	F1	MAE	Accuracy	F1	MAE
Centrally trained model	0.991	0.476	17.58	0.986	0.523	20.49	0.997	0.559	34.96	0.895	0.756	32.35
Proposed (IID)	0.988	0.421	17.21	0.955	0.413	22.13	0.989	0.547	35.13	0.736	0.621	36.91
Proposed (Non-IID)	0.896	0.413	18.408	0.964	0.510	21.56	0.951	0.516	35.87	0.656	0.543	38.34

Evaluation criteria

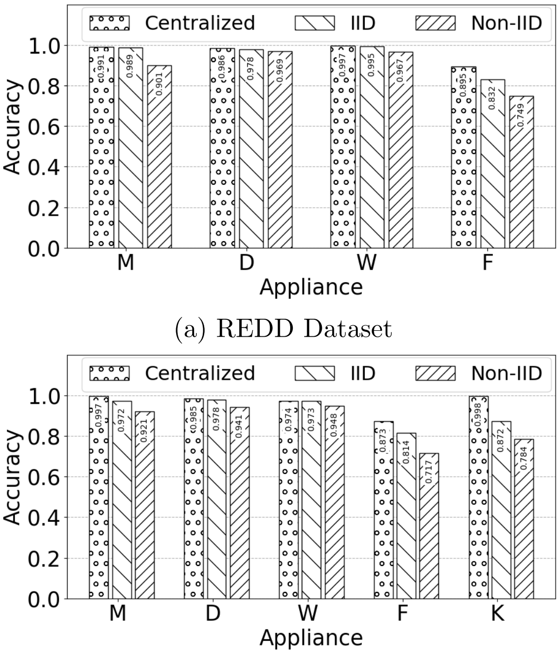
For the evaluation of our framework, we adopt three widely used metrics in NILM research: accuracy, F1 score, and mean absolute error (MAE). Accuracy measures the overall correctness of appliance state predictions. It is calculated as the ratio of correctly predicted states (both ON and OFF) to the total number of appliance state predictions. F1 score, on the other hand, combines precision and recall into a single metric and is particularly useful when the dataset is imbalanced, which is often the case in NILM. Lastly, MAE is calculated by taking the average of the absolute differences between the predicted and true power consumption values per appliance. A summary of the simulation parameters used in our experiments is given in Table 4.4.

4.5 Results and Analysis

For the centrally trained model and our model trained using FL (in the case of IID and non-IID data), we initially set the number of epochs to 70. Table 4.5 and 4.6 show the average performance of these models for different home appliances. It can be seen that the models trained using FL yield satisfactory performance for most of the appliances compared to the traditional models. For fridge, which has a less evident signature due to its relatively low power consumption, the FL-trained models cannot compete with

Table 4.6: Average performance scores for UK-DALE

	Microwave			Dishwasher			Washer			Fridge			Kettle		
	Accuracy	F1	MAE	Accuracy	F1	MAE	Accuracy	F1	MAE	Accuracy	F1	MAE	Accuracy	F1	MAE
Centrally trained model	0.997	0.014	6.57	0.985	0.667	16.18	0.974	0.325	6.98	0.873	0.766	25.49	0.998	0.907	6.82
Proposed (IID)	0.951	0.121	5.45	0.963	0.533	17.02	0.956	0.312	6.99	0.687	0.567	31.26	0.785	0.601	15.71
Proposed (Non-IID)	0.876	0.012	4.95	0.919	0.512	21.45	0.898	0.297	8.96	0.632	0.498	32.43	0.701	0.479	19.21



(a) REDD Dataset

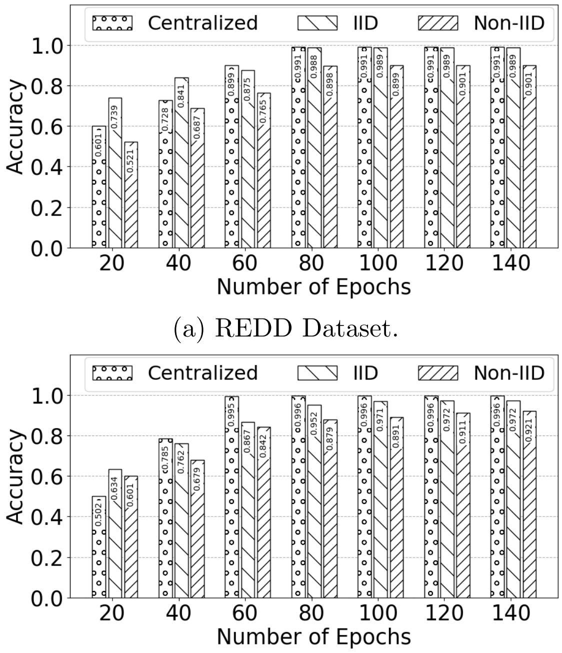
(b) UK-DALE Dataset

Figure 4.3: Disaggregation accuracy at 150 epochs.

the centrally trained model. Moreover, for less often used appliances such as kettle, an improved masking strategy and more training data could help improve the scores, given the BERT model's complex training mechanism. We have found that by fine-tuning model parameters and increasing the number of global rounds, the FL-trained model achieves better performance. Specifically, when we set the number of global rounds to 150, the performance scores improve drastically as can be seen in Figure 4.3a and 4.3b. Moreover, Figure 4.4 shows that the accuracy increases with the increase in the number of epochs. In these figures and the subsequent ones where appliances are represented on x-axis, we abbreviated the appliance name. Thus, microwave, dishwasher, washer, fridge, and kettle are denoted as M, D, W, F, and K, respectively. Overall, we conclude that federated learning makes possible high performance in the NILM task.

4.5.1 Investigating the Non-IID Case

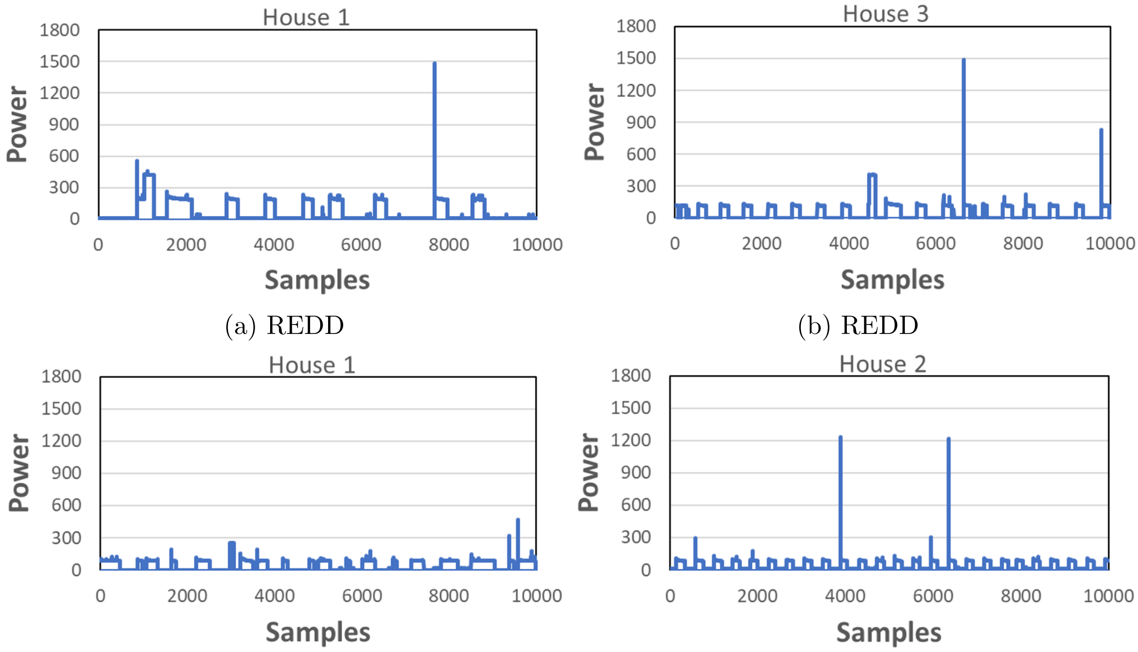
We observe through experimentation that by incorporating MAML into our model training, the accuracy for non-IID data is improved by 4% to 6%. Although the scores for



(a) REDD Dataset.

(b) UK-DALE Dataset.

Figure 4.4: Changes in accuracy when using more epochs



(a) REDD

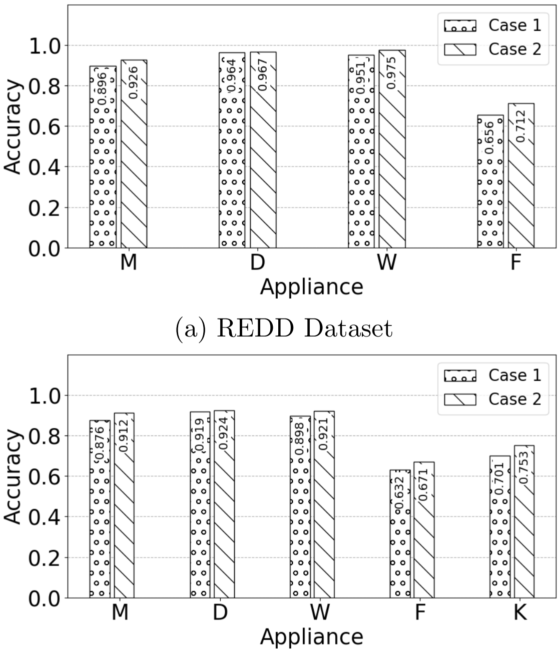
(b) REDD

(c) UK-DALE

(d) UK-DALE

Figure 4.5: Example power profile of a fridge (samples are taken every 6 seconds).

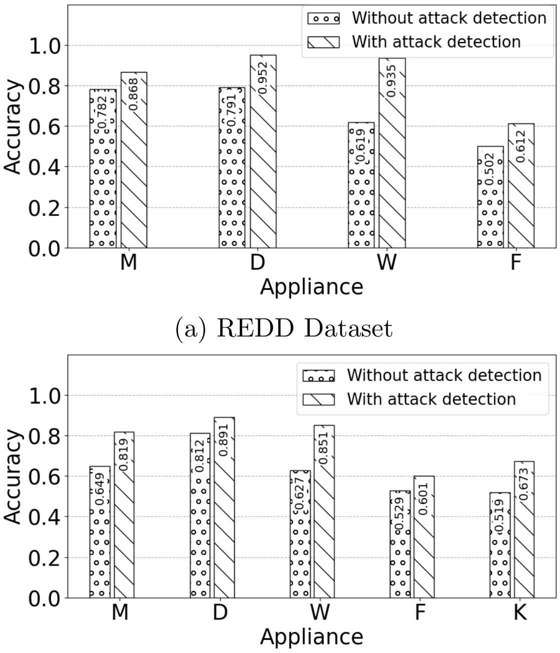
the federated model achieved satisfying performance, the values for non-IID cases are still subpar than both the IID and the centralized NILM models. We speculate that the usage patterns and load consumption signatures of the same appliances from different houses may be dissimilar due to several factors. It can be seen from Figure 4.5 that the load consumption distribution of the appliance fridge for separate houses differs significantly in both datasets. Therefore, we present a comparison of two cases of data heterogeneity



(a) REDD Dataset

(b) UK-DALE Dataset

Figure 4.6: Average accuracy in the non-IID case.



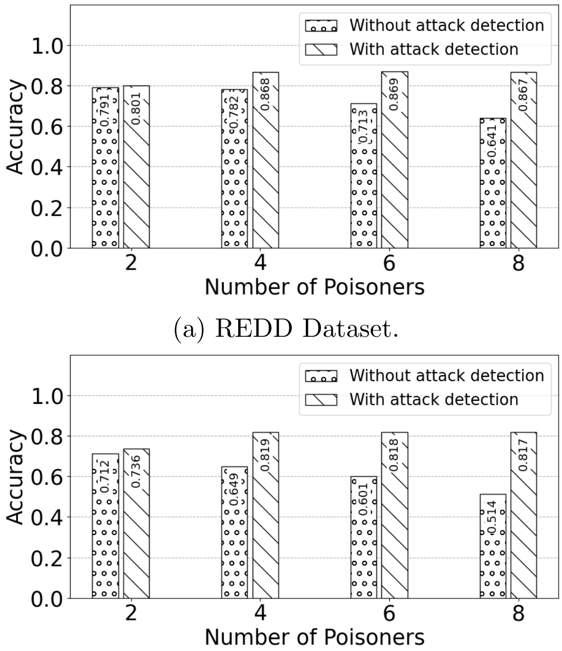
(a) REDD Dataset

(b) UK-DALE Dataset

Figure 4.7: Performance under model poisoning attack.

that can be taken into account for the non-IID data distribution:

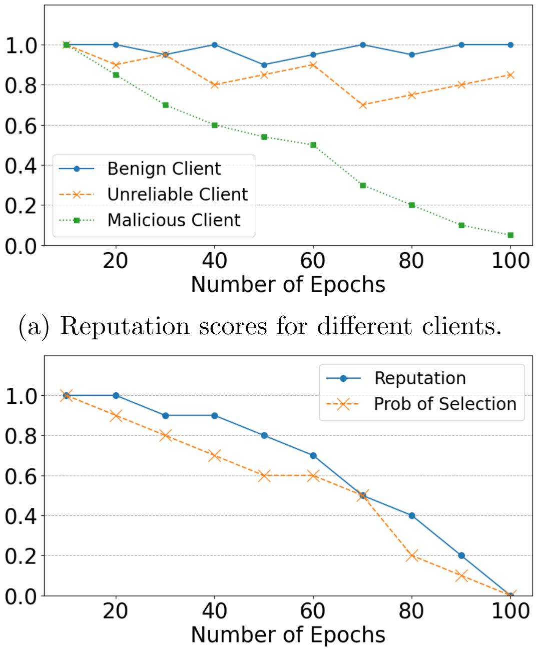
- Case 1 (missing classes): Each client owns the data from a particular appliance of a different house.
- Case 2 (heterogeneous data): 80% of the data owned by a client is from a particular appliance of a specific house and 20% of the data can belong to other heterogeneous



(a) REDD Dataset.

(b) UK-DALE Dataset.

Figure 4.8: Performance of the attack model for varying number of sybils in case of non-iid data.



(a) Reputation scores for different clients.

(b) Probability of a malicious client being selected.

Figure 4.9: Visualization of reputation scores.

classes of appliances.

Figure 4.6a and 4.6b present the average accuracy for the two cases and highlight the improvement when the degree of heterogeneity is decreased.

4.5.2 Evaluating the Robust FL Framework

We now evaluate the performance of our reputation-based aggregation scheme in the non-IID case when there are some dishonest clients. Figure 4.7a and 4.7b show the performance of our NILM model in REDD and UK-DALE datasets, respectively. It can be seen that without using the proposed robust FL framework the model performance drops drastically under the model-poisoning attack. However, using our framework, we can keep the accuracy high despite the model-poisoning attack. Moreover, the model performance does not decline with the increase in the number of dishonest clients as can be seen from Figure 4.8a and 4.8b. We also show that the probability by which a client is selected in a particular round decreases with the decrease in his reputation, thus mitigating the selection of dishonest clients. Figure 4.9a represents the variation of reputation values for a benign client (honest and accurate), unreliable client (honest but might be inaccurate due to non-iid data), and malicious client (dishonest). As we can see from Figure 4.9b, if we set the value of δ to 0.01, the probability of selecting a client falls below 50% if its reputation goes below 0.6. In conclusion, our reputation mechanism demonstrates robust capabilities in identifying dishonest clients, safeguarding the model's integrity and contributing to the overall reliability of our framework.

4.6 Summary

In this chapter, we proposed a FL-based framework for real-life IoT applications. The proposed framework offers a solution for energy consumption monitoring while preserving user privacy. By using a bidirectional transformer architecture, a meta-learning algorithm to handle data heterogeneity, and a reputation mechanism for the selective sampling of clients, we were able to achieve high accuracy and robust against privacy attacks by malicious clients. The experimental results demonstrate the efficacy of the proposed framework on real-world energy datasets in terms of various parameters. In the next chapter, we focus on extending this work for a peer-to-peer network without the presence of a central server.

Chapter 5

Enhancing Data Privacy in a Decentralized IoT

In this chapter, we focus on the decentralization of the learning process in FL. Traditionally, FL relies on a central server to facilitate the aggregation and distribution of model updates as described in Chapter 4. The server introduces bottlenecks and central points of failure that could compromise both the efficiency and privacy of the learning process. To address limitations, we aim to refine our methodology by eliminating the need for a central server such that the clients can directly exchange model updates with each other.

5.1 Peer-to-Peer FL for IoT

In specific IoT configurations, the feasibility of maintaining a central coordinating node becomes difficult due to connectivity constraints among the devices and components. Moreover, the distribution of data across IoT devices is often non-IID (independent and identically distributed). This could hamper the model performance, affecting applications that rely on the disaggregation result.

To address these challenges, peer-to-peer federated learning (P2P-FL) came into existence [121]. Instead of using a central server, P2P-FL encourages direct communication between clients (i.e. nodes in a P2P network) for exchanging model updates. This reduces the total amount of data that must be sent through the network, cutting down on communication time and costs. It also addresses the potential risk of having a single point of failure. However, if the data distribution varies significantly among clients, achieving convergence will be difficult. Furthermore, the decentralized nature of P2P-FL opens up opportunities for adversarial attacks. Malicious nodes could potentially inject false updates, manipulate the learning process, or even attempt to infer sensitive information from the exchanged model updates. The lack of a central coordinating node in P2P-FL also means that there is no central authority to verify the integrity of the model updates being exchanged.

We propose a robust P2P-FL framework to address the above-mentioned challenges in training a deep learning model for NILM. In our approach, we use a trust graph consisting of a network of clients that exchange model updates with each other. To ensure robustness, this exchange is achieved through a mechanism that involves accepting and incorporating model updates based on similarity metrics and accuracy thresholds. To calculate the similarity among clients without sharing their private data, we introduce a novel approach that employs a private set intersection (PSI) [122] protocol based on RSA and Jaccard similarity. Overall, our P2P-FL framework not only protects the privacy and integrity of local data but also promotes collaborative learning, enhancing the overall robustness and adaptability of the system in real-world settings. The main contributions in this chapter are as follows:

- We propose a novel approach to train a transformer-based NILM model by leveraging the P2P-FL framework and a trust graph of clients. In particular, we train a BERT model that is specifically designed to work with non-IID NILM data.
- We introduce a technique to measure the similarity between clients within a peer-to-peer network. This technique uses the statistical features shared by clients to compute the similarity scores through the Blind RSA-based PSI Protocol [122]. The model updates from the neighbors of a client are accepted according to the similarity scores and the received model's accuracy on the client's dataset.
- We evaluate the effectiveness of our approach through experiments using two prominent real-world NILM datasets, namely REDD and UK-DALE. Furthermore, we develop a threat model and analyze the robustness of our P2P-FL approach against three specific attacks, namely model poisoning, membership inference attack (MIA), and PSI inversion.

5.2 Problem Definition

Consider a dataset of energy consumption measurements from $\mathcal{U}$ users or households. This dataset consists of time-series data, denoted as S_u, representing the total energy consumption of each user $u \in \mathcal{U}$. Thus, the aggregate energy consumption of this user at time t is denoted as $S_u(t)$. The core idea behind NILM is the decomposition of this aggregate signal into the energy consumption of individual appliances:

$$S_u(t) = \sum_{a \in \mathcal{A}_u} L_{u,a}(t) + \epsilon(t) \tag{5.1}$$

where $\mathcal{A}_u$ is the set of appliances owned by user u, $L_{u,a}(t)$ is the power consumption of appliance a at time t, and $\epsilon(t)$ is the measurement noise or any latent component at time t.

The fundamental assumption made in NILM is that loads and appliances have a distinct power consumption signature. However, in the real world, data collected from different appliances of the same type exhibit non-IID characteristics, making it difficult to solve the disaggregation problem. This variability can be attributed:

- Diversity of appliances and environments: Consider an appliance as ubiquitous as a refrigerator. While its fundamental operation remains consistent, its energy consumption pattern can vary considerably between households due to the model and age of the appliance, operating voltage, variations in power supply, and standards in effect in different regions.

- Anomalies and unique usage patterns: Data quality issues that occur during data capture or transmission can appear in time series energy data collected from some households. Furthermore, in many households, some appliances are used infrequently, leading to imbalanced classes in a NILM dataset. For instance, the air conditioner might be used only during the summer season in some regions, resulting in sparse data for such appliances compared to others that are used on a daily basis.

5.3 Proposed Solution

In our approach, we address the NILM problem through P2P interactions between clients facilitated by a trust graph. Each user within the graph might have a different degree, meaning that the number of connections or neighbors for each user can vary. It can be visualized through Figure 5.1(a) which shows various houses connected to each other in a graph. We assume that this trust graph is not dynamically evolving during the learning process. Every user is already aware of their adjacent neighbors within the graph before the training is initiated. The steps involved in our proposed P2P-FL approach are described in the following subsections.

5.3.1 Model Update

In the proposed approach, we use the Bidirectional Encoder Representations from Transformers (BERT) model [98] as described in the previous chapter. The learning process begins with clients training a local BERT model on their raw data and then sharing the obtained model updates with neighboring clients. The decision to accept these updates or not is determined by evaluating the similarity scores between the models and the accuracy of the received model updates. This process as described in Algorithm 4 is explained in the following steps:

Local Training

In our setup, every client i starts by training a local model M_i using its raw data. Furthermore, client i also trains another model M'_j using the updates received from its neighbor j and its own local dataset. By doing this, it determines the performance of the external model received from j on the client's dataset, which is expressed quantitatively as accuracy, $\mathrm{acc}(i, j)$.

Sharing Model Updates with Neighbors

The locally updated model M_i is shared with the client's neighbors in the trust graph. The trust graph maintains a distributed learning network ensuring localized interactions and

Algorithm 4 Client-Side P2P Federated Learning

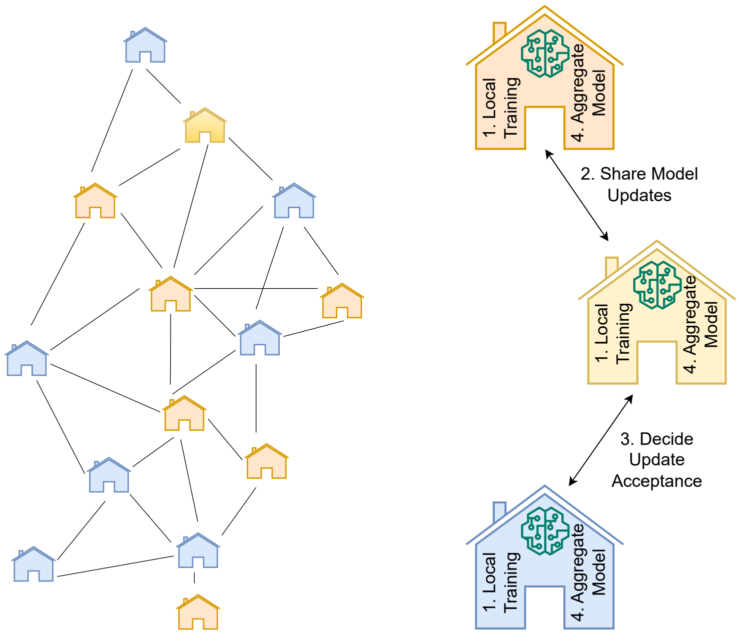
```
Initialize Trust Graph $G$
Identify Neighbors based on $G$
$Mi \leftarrow$ Train($LocalData$)
for $j$ in Neighbors do
    Send($Mi, j$)
    $M'_j \leftarrow$ Receive($j$)
    $M'_j \leftarrow$ Train($M'_j, LocalData$)
    acc($i, j$) $\leftarrow$ CalculateAccuracy($M'_j, LocalData$)
    $s(i, j) \leftarrow$ CalculateSimilarity($Mi, M'_j$)
    if $s(i, j) > \theta_{\text{sim}}$ and acc$(i, j) > \theta_{\text{acc\_h}}$ then
        $U_i \leftarrow \{M'_j\}$
    else if $s(i, j) \leq \theta_{\text{sim}}$ and acc$(i, j) > \theta_{\text{acc\_l}}$ then
        $U_i \leftarrow \{M'_j\}$
    else
        $U_i \leftarrow \{\phi\}$
$M_i^{\text{new}} \leftarrow Mi + \alpha \cdot U_i$
```

reducing the network overhead. This mechanism propagates model updates throughout the network.

Conditional Acceptance of Model Updates

A critical aspect of the proposed solution is the approach used by nodes to determine which updates from neighboring nodes should be accepted. This decision is rooted in two primary factors: the similarity between clients and the accuracy of the shared model update on their local data. We delve deeper into the details regarding the computation of similarity metrics in the next subsection. A higher similarity score suggests that the nodes have analogous data distributions. In such cases, only updates with high accuracy are accepted, ensuring that model updates are both relevant and beneficial. Conversely, for nodes with lower similarity scores, the accuracy threshold is relaxed slightly, under the assumption that these updates offer diverse insights which could potentially improve the model's generalization.

We denote the similarity threshold as θ_{sim}, and the high and low accuracy thresholds as θ_{acc_h} and θ_{acc_l}, respectively. Updates from nodes with similarity scores greater than θ_{sim} are accepted if their accuracy exceeds θ_{acc_h}, whereas those with scores less than or equal to θ_{sim} are accepted if their accuracy surpasses θ_{acc_l}. As shown in Figure 5.1(a), houses of the same color have passed both the similarity and accuracy threshold, indicating their updates are closely aligned. Houses of different colors (orange and blue), though not that similar, have achieved a high value of accuracy threshold, and as a result, their updates are also accepted. But the yellow-colored houses do not meet either the similarity or accuracy criteria, hence their updates are disregarded by the neighboring clients.



(a) P2P network (trust graph).

(b) Training process.

Figure 5.1: Proposed P2P approach overview.

Aggregation

After deciding which updates to accept, the new model M_i^{new} is then updated, ensuring that each client's model benefits from the collective insights of its neighbors. The learning rate α determines the extent to which the aggregated update is applied to the local model. The entire process of model updation in the P2P network is highlighted in Figure 5.1(b).

5.3.2 Similarity Measurement

In our FL framework, we compute the data similarity among clients in a privacy-preserving manner for protection against potential adversarial threats. The proposed approach calculates similarity scores $s(i, j)$ between client i and its neighbor j in the following manner:

Compute Features

Each node computes a number of features from its local dataset, capturing essential statistics such as mean, variance, skewness, kurtosis, and quantiles. These features serve as a summarized representation of the energy consumption patterns of different households. Once calculated, nodes share these features with their selected neighbors following Algorithm 5.

Similarity calculation

After receiving features from its neighbors, each client calculates the similarity scores $s(i, j)$ using the Blind RSA-based PSI Protocol [122]. This protocol ensures that two parties can find the intersection of their datasets without revealing any additional information about their individual sets. Let $\mathcal{S}_i$ denote the feature set obtained from the NILM data of client i, and $\mathcal{S}_j^{\text{recv}}$ denote the encrypted feature set received from a neighbor j.

Step 1: **Initialization:** Both parties agree on using RSA encryption, where the public key (N, e) is shared between them, and each maintains their own private key (N, d) securely.

Step 2: **Blinding:** Client i blinds its local feature set $\mathcal{S}_i$ as follows:

$$\text{Blinded}_{\mathcal{S}_i} = \{(x \cdot r^e) \mod N \mid x \in \mathcal{S}_i\} \tag{5.2}$$

Step 3: **Transferring:** The blinded set $\text{Blinded}_{\mathcal{S}_i}$ is transferred to neighbor j.

Step 4: **Intersection on Blinded Data:** Neighbor j computes the intersection $\mathcal{S}_{i,j}$ as follows:

$$\mathcal{S}_{i,j} = \text{Blind RSA-based PSI}(\text{Blinded}_{\mathcal{S}_i}, \mathcal{S}_j^{\text{recv}}) \tag{5.3}$$

Step 5: **Unblinding:** Client i unblinds the received set $\mathcal{S}_{i,j}$ to find the actual common elements:

$$\text{Unblinded}_{\mathcal{S}_{i,j}} = \{(x \cdot r^{-d}) \mod N \mid x \in \mathcal{S}_{i,j}\} \tag{5.4}$$

Step 6: **Calculate Similarity:** To determine the proportion of shared elements, we use the Jaccard similarity coefficient, which is defined as the size of the intersection divided by the size of the union of two sets [123]. The similarity $s(i, j)$ is calculated as follows:

$$s(i, j) = \frac{|\text{Unblinded}_{\mathcal{S}_{i,j}}|}{|\mathcal{S}_i| + |\mathcal{S}_j^{\text{recv}}| - |\text{Unblinded}_{\mathcal{S}_{i,j}}|} \tag{5.5}$$

This process is outlined in Algorithm 5. By adopting this approach, the framework manages to balance both the quality and diversity of the model updates, thereby enabling a more effective and robust FL process.

Algorithm 5 Calculate Similarity Using Private Set Intersection (PSI)

```
LocalSet ← ComputeSetFromFeatures(LocalData)
                          ▷ Share LocalSet for PSI-based similarity computation
for neighbor in Neighbors do
    $s(i, j)$ ← PSI(LocalSet, ReceivedSetFromNeighbor)
Return $s(i, j)$
```

5.4 Threat Model

Although learning a model on decentralized data eliminates the need for sharing it with a central server, it is not fully protected as the model updates could still reveal information about the user's private data. In this section, we outline our threat model.

5.4.1 Adversary's Knowledge

We assume that the adversary, being one of the nodes within the network, is aware of the model updates shared among nodes. The adversary also knows the network topology, i.e., who is connected to whom in the trust graph. Additionally, the adversary has knowledge of the encrypted features shared by nodes during the initial similarity calculation and the PSI algorithm being used, but it does not possess the private keys of other clients.

5.4.2 Privacy Attacks

While the space of possible attacks on FL is vast, we focus on a subset that holds particular relevance to our proposed approach as follows:

1. Model Poisoning Attacks: Poisoning attacks can manifest in two primary forms: targeted, where the adversary seeks to alter the model's behavior for specific inputs, and untargeted, where the aim is to degrade the model's overall performance rather than modifying the model's outcome for a specific input [124]. In the context of our P2P approach, while both types are of concern, we particularly focus on untargeted model poisoning attacks. The motivation for this is that untargeted attacks strive to disrupt the entire consensus of the P2P network, challenging the aggregate learning capability of the decentralized system. In such attacks, the attacker introduces noisy or incorrect data into their local model updates, with the hope that when these corrupted updates are aggregated with genuine ones, the resultant model's accuracy and integrity will decrease.

2. PSI Inversion: The adversary aims to reverse-engineer the PSI data to gain insights into the raw data sets that were intersected. Adversary tries to solve PSI^{-1}(blinded set) to reverse the blinding.

3. Membership Inference Attacks (MIAs): MIAs aim to ascertain whether a particular data point x was part of the training set used for a machine learning model [125]. This attack leverages the model's behavior, such as its predictions or confidence scores, to infer membership.

5.5 Performance Evaluation

5.5.1 Dataset and Preprocessing

For our experimentation, we utilized two real-world NILM datasets: the Reference Energy Disaggregation (REDD) dataset [118] and the UK Domestic Appliance-Level Electricity (UK-DALE) dataset [119]. These datasets are recognized as standard benchmarks in the field of NILM, offering detailed insights from various environments. They comprise both individual appliance and aggregate consumption data, making them suitable for both training and testing phases.

The REDD dataset consists of recordings from six distinct residential structures situated in the United States and spans an aggregate of 119 days. It encompasses both high-frequency and low-frequency measurements, offering a comprehensive view of energy consumption patterns. For our analysis, we leveraged its low-frequency data, specifically targeting four appliances (since these labels were available for all houses): the microwave (M), dishwasher (D), washer and dryer combo (W), and fridge (F). On the other hand, the UK-DALE dataset, released between 2013 and 2015, provides insights into power consumption patterns specific to the United Kingdom. This dataset encompasses data from five diverse residential structures. From UK-DALE, we employ its low-frequency readings and, while analyzing the aforementioned four appliances, we also add another appliance kettle (K) to our study list.

Similar to the the preprocessing techniques highlighted in BERT4NILM [98], we synchronize the timestamps between the primary channels and individual appliances. This allows us to resample data at six-second intervals and bridge any time gaps that are less than three minutes using a forward fill method. The configuration settings for our experiments are described in Table 5.1. We optimized four crucial hyperparameters for optimal performance: α, θ_{sim}, $\theta_{\text{acc\_high}}$ and $\theta_{\text{acc\_low}}$. After comprehensive testing, the optimal value for α was identified as 0.7, ensuring a balanced integration of external model characteristics while preserving the uniqueness of the local model. θ_{sim} sets the similarity threshold for accepting model updates from peers. Through experimentation, we determined the optimal value for θ_{sim} to be 0.5. Lastly, we found that setting $\theta_{\text{acc\_high}}$ to 0.8 and $\theta_{\text{acc\_low}}$ to 0.65 was optimal, providing a good balance between incorporating a diverse range of models and maintaining a high standard of model accuracy.

5.5.2 Baseline

To evaluate the performance of our proposed approach, we conducted a comparative analysis with other learning paradigms, while keeping the underlying neural network architecture intact:

- **Centrally-Trained Model**: This represents the conventional method of ML training where raw data from all households is aggregated and processed at a central hub. A single central model is trained and tested for all clients.

- **FL-Based NILM Model (with server)**: In this approach, each client employs its raw dataset to train a local model and then transmits model updates to the aggregation server. The server then aggregates these updates using the Federated Averaging (FedAvg) technique [39].

- **P2P Trained FL Model (without Server)**: In this approach, clients collaboratively train the model without relying on a central server.

Table 5.1: Experimental settings for the proposed approach.

Parameter	Value/Description
Datasets	REDD and UK-DALE
Number of Epochs	100
Learning Rate	Starts at 0.01; reduced by 10% every 10 epochs
Batch Size	128
No of Clients	20 to 100
Sampling time	6s
Optimization Algorithm	Adam optimizer
Model Architecture	BERT4NILM
Evaluation Metrics	Accuracy, Precision, Recall, MAE

5.5.3 Experiments

In this chapter, we use the sequence-to-sequence benchmark evaluation and adopt a training methodology similar to that of BERT4NILM[1]. This involves employing the Masked Language Model (MLM) approach from the pre-training process of BERT. In this method, the input sequence is subjected to random masking, where a predetermined proportion of the input elements is masked using a special token. The model then exclusively uses the outputs corresponding to these masked positions for loss computation, compelling it to learn from contextual information and predict the masked elements. This process significantly enhances the model's ability to capture critical patterns throughout the input sequence. The specific architectural parameters used for training are 2 transformer layers, 2 attention heads, and a maximum hidden size of 256, with weights initialized using truncated normal distributions.

To assess the model's generalization capabilities, the test data is drawn from completely unseen data of a house. The evaluation data is normalized using the mean and standard deviation derived from the training data and is input to the model without any masking, ensuring a thorough evaluation of the model's performance.

In our experimental evaluation, we sought to assess the performance of the energy disaggregation model under different scenarios:

Experiment 1- Centralized ML vs. Standard FL vs. P2P-FL

In the first scenario, we implemented the conventional approach of centralizing all data from NILM houses. A singular ML model was trained and tested on this combined dataset. In the second scenario, the NILM data is distributed proportionally amongst clients for local training and global model aggregation. Lastly, we implemented our proposed approach and determined the average results across all clients. Through this comparison, we aim to underscore the efficiency of each method by using accuracy as our evaluation metric.

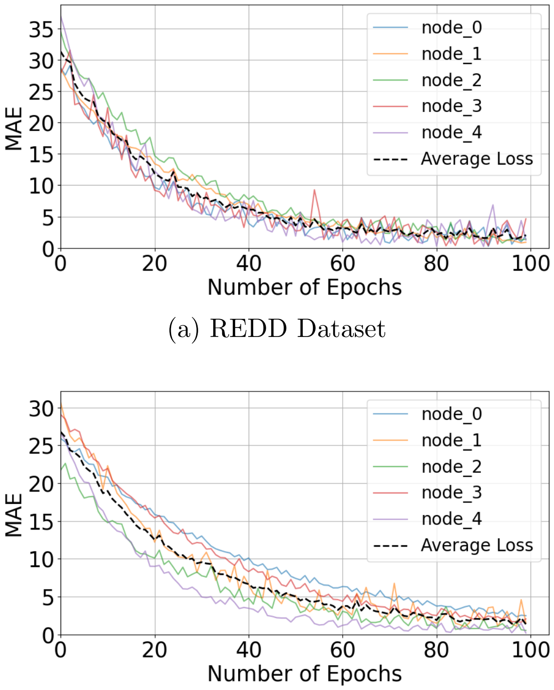
[1] `https://github.com/Yueeeeeeee/BERT4NILM`

Experiment 2- Comparing Standard FL with P2P-FL under different scenarios

We performed experiments using the Standard FL as well as our proposed P2P approach for 2 different scenarios: Firstly, we split the energy data of the houses uniformly amongst the clients. We then increased the number of clients from 20 to 100 and noted the performance of both approaches in each case. Secondly, we varied the data present with each client in a non-iid manner resembling the real-world NILM scenario to determine the influence of heterogeneity on performance. As the number of heterogeneity was increased from 20% to 100%, the amount of data available per house for each class decreased.

Experiment 3- Robustness to attacks

Next, we conducted experiments to assess the robustness and performance of our model in the case of adversarial attacks, specifically untargeted model poisoning attacks, membership inference attacks (MIAs), and PSI inversion. Each attack scenario is assessed for its success rate and impact on model accuracy.



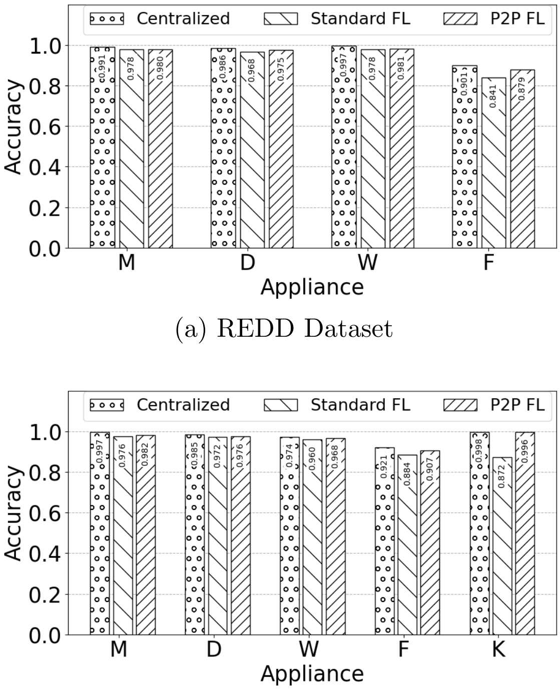
(a) REDD Dataset

(b) UK-DALE Dataset

Figure 5.2: Loss variation for 100 epochs.

5.6 Results and Analysis

The results presented in the following subsections demonstrate the effectiveness of our proposed P2P-FL approach. Figures 5.2a and 5.2b represent the Mean Absolute Errors (MAE) across 100 epochs for the two datasets, REDD and UK-DALE respectively. By analyzing the curves for individual nodes along with the average loss, we can draw the following inferences:



(a) REDD Dataset

(b) UK-DALE Dataset

Figure 5.3: Appliance accuracies for 100 epochs.

- The loss curves for all nodes (node_0 to node_4) consistently decrease as the number of epochs increases, indicating successful convergence of the training process.

- Some nodes (node_2 and node_3 in REDD; node_0 in UK-DALE) demonstrate slightly higher losses in the initial epochs, primarily due to data distribution discrepancies which the model initially struggles to learn, resulting in higher losses until it sufficiently adapts to these distinct patterns.

- The average loss, depicted by the dashed line, shows a steep decline in the early epochs, leveling off as it approaches 100 epochs. This indicates that, on average, the P2P-FL approach is effective in reducing the overall error across nodes.

- Small fluctuation in individual node losses can be observed in the latter epochs, hinting at potential noise or variability in the data.

5.6.1 Results from Experiment 1

Our findings from the initial experiment are shown in Figures 5.3a and 5.3b for datasets REDD and UK-DALE respectively. In most instances, P2P-FL not only outperforms Standard FL but also achieves accuracy that is nearly on par with Centralized ML. We note that in P2P-FL, every node has the ability to access data from other nodes. The selective acceptance of model updates significantly contributes to the improved performance of the model, especially in the presence of non-iid data. The few anomalies observed, such as the slightly diminished performance of P2P-FL for certain appliances (such as kettle), could be attributed to dataset-specific variations or the intrinsic challenges associated with P2P computations.

5.6.2 Results from Experiment 2

The results from our second experiment are depicted in Tables 5.2 and 5.3. With the increasing number of clients (Table 5.2), our proposed approach demonstrated better performance as compared to Standard FL, especially when the number of clients exceeded 60. For appliances such as the microwave and dishwasher, the accuracy remains high (above 0.8) for P2P-FL even with 100 clients. In contrast, the performance of the standard FL approach diminishes more rapidly as the number of clients increases. This is because, in P2P-FL, each client interacts directly with multiple other clients. This direct interaction exposes clients to a more diverse set of data samples from their peers, enriching their model updates and potentially leading to a more comprehensive global model. Moreover, with increased clients, there's a higher likelihood of noisy or even adversarial updates during the aggregation process. Since our approach is more robust to such issues, it reduces the effects of malicious or noisy peers.

From Table 5.3, we can see that although both methods experience a reduction in accuracy with increasing data heterogeneity, the decline is less pronounced in the P2P-FL approach. This is because, in P2P-FL, clients can learn from a diverse set of peers and benefit from a broader range of data representations. Moreover, in our approach, clients can choose which peers to interact with through predefined thresholds. This adaptability leads to more focused learning, improving the accuracy even further.

Table 5.2: Average accuracy scores for REDD.

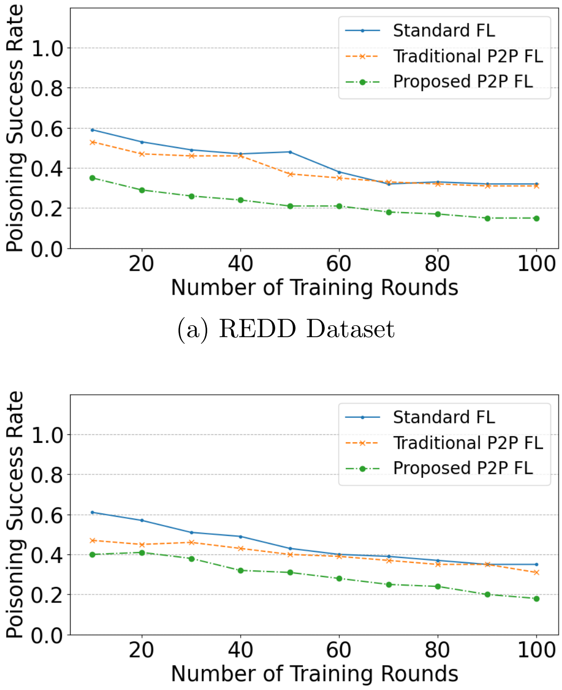
No of Clients	Microwave		Dishwasher		Washer		Fridge	
	FL	P2P-FL	FL	P2P-FL	FL	P2P-FL	FL	P2P-FL
20	0.955	0.968	0.997	0.995	0.961	0.988	0.866	0.867
40	0.856	0.892	0.925	0.934	0.896	0.921	0.845	0.851
60	0.824	0.852	0.891	0.901	0.878	0.902	0.736	0.847
80	0.795	0.836	0.889	0.892	0.840	0.894	0.701	0.821
100	0.743	0.801	0.751	0.856	0.821	0.882	0.693	0.801

5.6.3 Results from Experiment 3

The robustness of our model against untargeted model poisoning attacks is illustrated in Figures 5.4a and 5.4b. We can see that the P2P-FL approach is more effective in mitigating these attacks as compared to both the Standard FL and Traditional P2P-FL methods [126]. This can be attributed to its decentralized nature, combined with our enhanced security mechanism, which minimizes the influence of any single malicious actor. The strategic

Table 5.3: Average accuracy scores for UK-DALE.

Data Heterogeneity	Microwave		Dishwasher		Washer		Fridge		Kettle	
	FL	P2P-FL	FL	P2P-FL	FL	P2P-FL	FL	P2P-FL	FL	P2P-FL
20 %	0.923	0.944	0.913	0.925	0.941	0.954	0.871	0.895	0.931	0.945
40 %	0.792	0.810	0.789	0.890	0.787	0.789	0.757	0.772	0.802	0.821
60 %	0.753	0.784	0.731	0.765	0.763	0.774	0.721	0.741	0.753	0.781
80 %	0.621	0.661	0.549	0.604	0.623	0.624	0.572	0.612	0.601	0.631
100 %	0.576	0.612	0.512	0.582	0.591	0.592	0.543	0.566	0.571	0.582



(a) REDD Dataset

(b) UK-DALE Dataset

Figure 5.4: Comparison of the success rates for model poisoning attack with the increasing number of rounds.

selection process reduces the chances of incorporating malicious or poisoned updates into the global model, hence improving its resistance against poisoning attacks.

Figure 5.5 shows the effectiveness of the proposed P2P-FL approach against MIA with the increasing number of training rounds. For both datasets, we observe that the efficiency of attack diminishes as the number of training rounds increases. Starting from the initial rounds, the success rate of the MIAs is quite high as the model might be more vulnerable to such attacks, possibly due to its under-trained nature and the limited data it has seen. However, by the 100th training round, the success rate of the MIAs decreases to below 0.4 for both datasets. This drop implies that as the model becomes more refined and exposed to diverse data, its resilience to MIAs strengthens. Another reason for this descent is the unpredictability in the model aggregation process, which makes it more difficult for

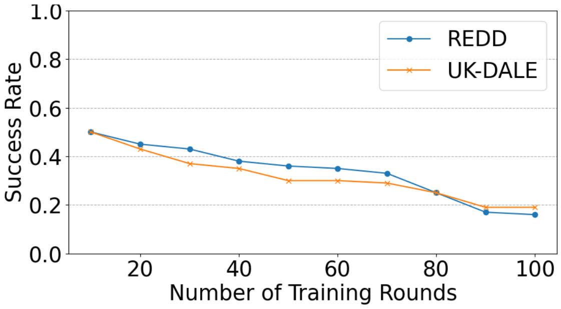


Figure 5.5: Success rates for MIA with the increasing number of rounds.

attackers to reverse-engineer the raw data elements contributing to each update. The PSI protocol utilizes the mathematical properties of RSA encryption for secure computations so it is computationally infeasible to attack. The use of the RSA-based PSI ensures that even if an adversary has access to the blinded set $Blinded_{S_i}$, they cannot infer the original set S_i without the private key d. Mathematically, given a blinded element B from set $Blinded_{S_i}$, the adversary would need to unblind it to get x. However, unblinding requires the operation $x \cdot r^{-d} \mod N$, which needs the private key d. Without d, the adversary cannot perform the unblinding and thus cannot determine x. Therefore, even after observing the intersection computation, an adversary cannot determine the actual elements that intersected.

Overall, across all experiments, our P2P-FL model consistently exhibited better performance, both in terms of accuracy and robustness. The decentralized nature proves advantageous in diverse settings, ranging from heterogeneous data distribution to adversarial environments. Moreover, the ability of P2P-FL to effectively handle non-iid data distributions makes it a suitable choice for real-world NILM applications.

5.7 Summary

In this chapter, we presented a novel approach to address different challenges of standard FL. By leveraging a P2P trust graph, we enable clients to interact and share knowledge without compromising data privacy. Despite the diversity of clients and potential presence of malicious actors in the network, the proposed framework ensures that the quality of model updates and accuracy is preserved. In the next chapter, we aim to integrate this approach with the concept of hierarchical blockchains introduced in Chapter 3 to develop a comprehensive framework for IoT data management.

Chapter 6

Integration of Privacy and Security in IoT Infrastructure

In this chapter, we address the integration of privacy and security within the IoT infrastructure, acknowledging the challenges posed by it. It explores the application of FL and blockchain to achieve a dual objective: enhancing sustainability through decentralized parameters optimization and security through immutable, distributed ledgers. This chapter presents an integration of P2P-FL which was discussed in Chapter 5 with the blockchain framework proposed in Chapter 3, aimed at enhancing the sustainability and privacy of blockchain networks used in IoT applications.

6.1 Sustaiable and Secure IoT

Majorly, every node in the blockchain network independently validates transactions and maintains a complete copy of the ledger, leading to considerable computational overhead and energy expenditure. As the volume of transactions increases, so does the latency [127], further straining the network's capacity to process transactions efficiently. Therefore, in the face of global sustainability goals, the burgeoning energy footprint of blockchain networks cannot be overlooked [128].

With the integration of blockchain and IoT, there is more emphasis on the critical need for efficient energy utilization [129]. As IoT devices proliferate, the volume of transactions requiring validation and ledger updates in a blockchain network increases exponentially. This surge in activity can lead to bottlenecks, particularly in traditional blockchain models further increasing the issues of energy consumption and latency.

In blockchain, the original consensus algorithm — Nakamoto's protocol [130], also known as Proof-of-Work (PoW) holds a foundational place. It was the first to achieve consensus in a permission-less environment since Bitcoin's invention in 2009. Despite being a trusted and secure public consensus algorithm, PoW is recognized as a computationally intensive process. For instance, the energy required for Bitcoin mining is substantial enough to surpass the entire power consumption of Switzerland [131]. Therefore, Bitcoin's carbon footprint is non-negligible, with estimates suggesting an annual emission of approximately 33.5 MtCO2e as of May 2018 [132]. The underlying cause of PoW's excessive energy consumption is the intention to render attacks on the network prohibitively expensive. Alternatively, Proof of Stake (PoS) [133] emerged as a more energy-efficient consensus alternative, where validators prove ownership of a certain stake to vote on new blocks rather than performing extensive computation. This method, however, is susceptible to the ‘Nothing-at-Stake’ problem, where validators may support multiple blockchain forks to maximize reward potential [134].

While centralized traditional systems strive to reduce carbon emissions and provide secure data management, they struggle to match the characteristics of decentralized blockchain [135], which values transparency and openness. This necessitates a move towards decentralized and adaptive solutions that can optimize network parameters in real time, ensuring efficiency and sustainability without compromising on performance and privacy. The concept of Federated Learning (FL) emerges as a beacon of innovation to address these pressing issues [136]. FL ensures privacy by allowing nodes to contribute to a global model while keeping their individual data localized. This privacy-centric approach is vital for maintaining the confidentiality and security of information, which is pertinent in sectors especially with stringent data protection regulations. The ability of FL to maintain privacy while facilitating collective intelligence makes it an ideal candidate for optimizing blockchain networks in a manner that is adaptable to dynamic conditions and unique experiences of individual nodes [137]. However, centralized FL has some inherent vulnerabilities, such as the risk of a single point of failure and the reliance on the trustworthiness of the central server. These limitations can hinder the scalability and robustness of the system, and raise concerns about data privacy and security. To overcome these challenges, Peer-to-Peer Federated Learning (P2P-FL) [126] emerges as a more resilient, distributed, and decentralized alternative. In a P2P-FL system, nodes or clients directly communicate and collaborate with each other to train a shared model. Each node in P2P-FL contributes to the learning process based on its local data and computational resources, and the learning updates are shared among peers without passing through a central entity.

The integration of P2P-FL and blockchain technology can also provide a significant contribution in designing energy trading frameworks, offering a robust solution for managing and optimizing energy resources in a decentralized and secure manner. This approach not only streamlines energy distribution but also empowers consumers and producers, paving the way for more adaptive and resilient energy networks.

Therefore, in this chapter, we introduce an approach to tackle these challenges: integrating P2P-FL with blockchain network for sustainability. This approach aims to transform each node within the blockchain network from being a passive participant to an active agent in the network optimization process. The nodes collaboratively engage in a decentralized learning process, sharing insights and updates to create a global model capable of determining the optimal network parameters. Our federated system therefore not only learns from the network but also informs it, creating a responsive and self-optimizing blockchain ecosystem. A general framework of our designed approach is represented in Figure 6.1. The major contributions of this chapter are described as follows:

- We propose a distributed framework for sustainable blockchain by integrating it with P2P-FL to determine optimal blockchain parameters for energy conservation without

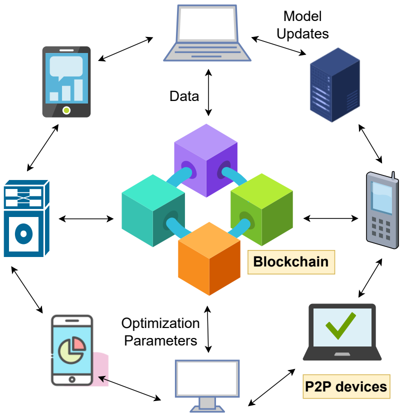


Figure 6.1: A Conceptual Representation of the Blockchain-FL Network Architecture.

sacrificing data privacy. We then utilize a load-balancing strategy to distribute data amongst multiple blockchains according to these optimal parameters.

- We present our problem mathematically using a non-cooperative game-theoretic model. This approach ensures that each node, while optimizing its own performance, also contributes positively to the collective efficiency and sustainability of the blockchain network.
- We also illustrate a practical case study in renewable energy trading, demonstrating how blockchain can be effectively used in green energy sectors. By tokenizing renewable energy and facilitating peer-to-peer transactions on a decentralized ledger, this framework can contribute to the development of more sustainable and efficient energy markets.
- Lastly, we validate the proposed framework through experimentation and results. The experimental results indicate a substantial decrease in energy consumption for blockchain networks along with a reduction in the average execution times.

6.2 Preliminaries

6.2.1 Sustainability in Blockchain

Blockchain technology fundamentally operates on a chain of blocks, each containing transaction data. These blocks, denoted as $B_1, B_2, \ldots, B_n$, form the backbone of the blockchain ledger. A crucial aspect of blockchain is its consensus mechanism, particularly Proof of Work (PoW) in many traditional blockchains. PoW involves solving complex cryptographic puzzles, a process that ensures security and integrity but is also energy-intensive. The mathematical representation of PoW is finding a nonce value n such that the hash of the block's content concatenated with n meets a specific condition, typically involving leading zeroes:

$$H(B_i \| n) < D, \tag{6.1}$$

where H represents a cryptographic hash function, and D is the difficulty target, adjusted dynamically to maintain consistent block generation times.

In the context of blockchain, sustainability primarily concerns the energy efficiency of the network. If E denotes the network's total energy consumption and N_T the total number of processed transactions, the energy efficiency η of the network is expressed as:

$$\eta = \frac{E}{N_T}. \tag{6.2}$$

6.2.2 FL and FedAvg Algorithm

FL enables a distributed approach where each node in the network, denoted as n_i, trains models on its local data D_i and computes model updates ΔM_i. These local updates are then aggregated to form a global model M_G. In centralized FL, this aggregation is typically done using the FedAvg algorithm [39], which mathematically can be represented as:

$$M_G^{(t+1)} = M_G^{(t)} + \frac{1}{N}\sum_{i=1}^{N} \Delta M_i, \tag{6.3}$$

where $M_G^{(t)}$ is the global model at iteration t, and N is the total number of nodes participating in the training. The FedAvg algorithm is crucial for FL as it allows for the synthesis of a robust global model from diverse local datasets while preserving data privacy and reducing central data storage requirements.

6.2.3 Peer-to-Peer Federated Learning (P2P-FL) with Gossip Learning

In P2P-FL, the concept of FL is extended to a decentralized architecture where nodes directly share model updates with each other, eliminating the need for a central server. This approach enhances privacy and further reduces the reliance on a central data repository. One effective method for implementing P2P-FL is through gossip learning protocols [126]. In this model, each node n_i updates its model M_i by incorporating information from a subset of peer nodes, referred to as $\mathcal{N}_i$. The update rule in a gossip learning framework can be expressed as:

$$M_i^{(t+1)} = M_i^{(t)} + \alpha \sum_{j \in \mathcal{N}_i} w_{ij}(M_j^{(t)} - M_i^{(t)}), \tag{6.4}$$

where w_{ij} is the weighting factor assigned to the model update from node j, and α is the learning rate.

We use gossip learning in our approach since it aligns with the decentralized nature of blockchain. It allows robust model updates by leveraging random and stochastic peer selection for information exchange. This ensures a wide dissemination of knowledge across the network as well as optimizes the learning process by distributing computational load. Consequently, it leads to more energy-efficient operations, significantly contributing to the sustainability of the blockchain network.

6.3 Proposed Approach

In this section, we propose a system model that uses P2P-FL to enhance the sustainability and efficiency of blockchain networks. This decentralized approach functions without a central authority in a collaborative learning environment, mitigating single points of failure and bottlenecks. The detailed framework of our proposed approach as shown in Figure 6.2 is described as follows:

6.3.1 Blockchain Network and Nodes

The system consists of a network of blockchain nodes that are responsible for processing transactions and maintaining the integrity of the blockchain. Each node is an independent entity that contributes to the overall decision-making process of the network by participating in consensus and validating transactions. These nodes are not only executors but also learners that continuously collect data and determine the optimal network's performance parameters. We consider a blockchain network $\mathcal{B}$ composed of N peer nodes $\{n_1, n_2, \ldots, n_N\}$, each participating in consensus and transaction verification. These nodes collectively work towards optimizing a set of performance parameters Θ indicative of the network's sustainability. At each node n_i, local data D_i is collected which includes:

- T_{p_i}: Transaction processing time.
- T_{bg_i}: Block generation time.
- T_{bp_i}: Block propagation time.
- T_{l_i}: Latency in the network.
- E_i: Energy consumed.

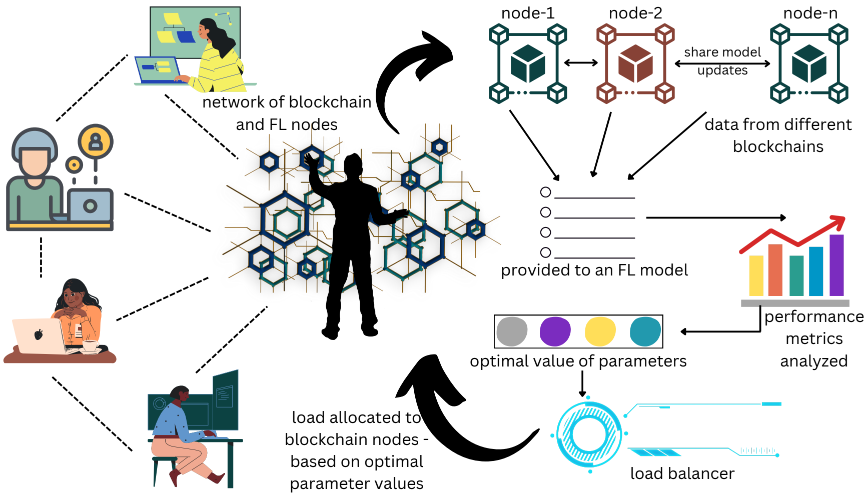

node-1
node-2
node-n
share model updates
network of blockchain and FL nodes
data from different blockchains
provided to an FL model
performance metrics analyzed
optimal value of parameters
load allocated to blockchain nodes - based on optimal parameter values
load balancer

Figure 6.2: Framework of our Proposed Approach with Blockchain and P2P Nodes.

6.3.2 Local Model Training

Each blockchain node also trains a local predictive model that uses the data collected by the node to make predictions about the optimal parameters for processing. This predictive capability is used for anticipating and mitigating potential inefficiencies in blockchain's functioning. Each node strives to refine its model so that the predicted parameters minimize any discrepancy from the ideal state of the network. Each node n_i uses a local predictive model M_i, parameterized by θ_i, to estimate the optimal set of network parameters $\Theta_i^{opt}(t)$:

$$M_i(D_i(t); \theta_i) \rightarrow \Theta_i^{opt}(t) \tag{6.5}$$

The objective function for optimizing the blockchain parameters is defined as:

$$F(\Theta_i^{opt}) = \sum_{i=1}^{N} (\alpha \cdot T_{p_i} + \beta \cdot T_{bg_i} + \gamma \cdot T_{bp_i} + \delta \cdot T_{l_i}) + \lambda \left(\alpha^2 + \beta^2 + \gamma^2 + \delta^2\right) \tag{6.6}$$

Where:

- $F(\Theta_i^{opt})$ is the objective function to be minimized.
- Θ_i^{opt} represents the set of network performance parameters for node i.
- $\alpha, \beta, \gamma, \delta$ are the weights corresponding to each parameter in the objective function.
- λ is the regularization parameter to prevent overfitting by penalizing the magnitude of the weights.

The loss function F_i at each node aims to minimize the difference between the current and the optimal network parameters:

$$F_i(\theta_i) = ||\Theta - \Theta_i^{opt}(t)||^2 \tag{6.7}$$

6.3.3 Peer-to-Peer Model Synchronization

For sharing model updates amongst peers, we use the basic Gossip protocol [126] that works as follows:

- Each node in the network holds a local dataset and participates in the model training by sending model updates to its peers.
- At regular intervals, each node engages in a communication process with other nodes in the network.
- When a node receives a model from another node, it integrates this external information into its own model.
- The model parameters are updated through a gradient descent process, incorporating a learning rate and regularization.

- The nodes operate asynchronously since they don't wait for each other to complete a round of updates. The process does not rely on any central control, maintaining the decentralized nature of the network.

- Nodes use a peer sampling service to randomly select peers for exchanging model parameters. Over time, as models are exchanged and updated across the network, they collectively evolve to approximate a model that would have been trained on the entire dataset centrally.

We use gossip-based communication since it can withstand network partitions because it does not require every node to receive updates simultaneously. The synchronization of models across the network can be formalized as follows:

$$\theta_i^{sync}(t+1) = \sum_{j \in \mathcal{N}_i} w_{ij} \theta_j(t) \tag{6.8}$$

where $\mathcal{N}_i$ represents the neighboring nodes of n_i and w_{ij} are the weighting factors based on the reliability and contribution of each neighboring node's model.

Algorithm 6 P2P Federated Learning for Blockchain Optimization

```
Initialize network with $N$ nodes, each with initial parameters $\theta_i(0)$
Define sustainability goals $S$
for each global epoch $t = 1, 2, \ldots, T$ do
    for each node $n_i \in \mathcal{B}$ in parallel do
        Collect local data $D_i(t)$
        Train local model $M_i$ with $D_i(t)$ to obtain $\Theta_i^{opt}(t)$
        $M_i(D_i(t); \theta_i) \rightarrow \Theta_i^{opt}(t)$
        Exchange model parameters with peer nodes $\mathcal{N}_i$
        $\theta_i^{sync}(t+1) \leftarrow \sum_{j \in \mathcal{N}_i} w_{ij} \theta_j(t)$
    Aggregate synchronized parameters and distribute the load
    Monitor and adjust to ensure alignment with sustainability goals $S$
    $\mathcal{M}(\Theta(t+1)) \rightarrow S(t+1)$
```

6.3.4 Load Distribution to Different Blockchains

After local model training and updation, we use a strategy for dynamic load distribution across multiple blockchain networks. Since adjusting the parameters of a single blockchain dynamically to optimize performance is not always feasible, this approach seeks to allocate data loads to various blockchains based on their operational efficiency.

The core of this strategy is an optimization-based load allocation mechanism. Each blockchain in our ecosystem is evaluated based on the chosen set of performance parameters (Θ). Building upon the foundational concept of HierChain [138], we use a load-balancing mechanism that aligns with our previously proposed method of distributing loads across multiple blockchains. The algorithm evaluates the current state of each blockchain in terms of the optimized parameters and selects the one that offers the best performance with the lowest energy consumption. This decision is made based on real-time data and predictive analytics, ensuring that the load is always directed to the most efficient blockchain. This load distribution strategy offers several benefits:

- Energy Efficiency: By routing loads to the most energy-efficient blockchain, the overall energy consumption of the network is reduced.
- Optimized Performance: Each blockchain can operate under optimal conditions, enhancing the network's overall performance.
- Scalability: This approach allows the network to scale more effectively by distributing loads across multiple blockchains.

6.3.5 Sustainability Metrics and Monitoring

The network performance and sustainability are continuously monitored using a function $\mathcal{M}$, which assesses whether the current parameters align with the predefined sustainability goals S:

$$\mathcal{M}(\Theta(t+1)) \to S(t+1) \tag{6.9}$$

The steps in the working of our proposed approach are presented in Algorithm 6. The energy consumption E of a blockchain can be expressed as a function of the chosen parameters, along with other system parameters:

$$E = (T_{p_i} + T_{bg_i} + T_{bp_i} + T_{l_i}) \cdot P \cdot E_{\text{rate}} + E_{\text{fixed}} \tag{6.10}$$

where:

- E is the total energy consumption.
- P is the power usage per unit time for mining and maintaining the network.
- E_{rate} is the energy consumption rate per unit of computational work.
- E_{fixed} is the fixed energy cost.

6.4 Case Study: Renewable Energy Trading on a Blockchain Network

This case study explores the implementation of our P2P-FL model in a blockchain network specifically designed for trading renewable energy credits or tokens. The primary objective is to harness the decentralized and secure nature of blockchain technology to create an efficient marketplace for energy transactions.

6.4.1 Description

In the renewable energy sector, there is an urgent need for innovative systems that can efficiently and sustainably manage the trading of energy credits or tokens. Renewable energy trading on a blockchain network represents the generation of energy from renewable sources, including solar panels, wind turbines, and hydroelectric plants. This green energy is then tokenized, converting each unit of energy, typically measured in kilowatt-hours, into a digital token which is recorded on the blockchain.

The fundamental concept of this framework allows individuals, businesses, and communities that generate renewable energy to participate in a decentralized marketplace. Through this marketplace, they can directly sell their excess energy to other consumers and organizations, effectively transforming participants into prosumers — individuals who simultaneously produce and consume energy. The renewable energy trading on blockchain network [139] includes the following key characteristics and elements:

- Tokenization of Energy: The process of converting renewable energy production into digital tokens is a critical part of this system. Each token represents a specific quantity of energy generated from sustainable sources. This tokenization enables fractional ownership, allowing individuals to buy, sell, and trade energy in smaller, more accessible units.
- Blockchain Ledger: The blockchain ledger is another primary component of the system, providing a transparent and immutable record of all energy transactions. Every token transfer, purchase, and sale is securely recorded on the blockchain, ensuring transparency and traceability.
- Peer-to-Peer Transactions: A defining feature of this system is the direct peer-to-peer transactions between energy producers and consumers. Participants can negotiate terms, set prices, and trade energy tokens without the need for intermediaries, such as traditional energy suppliers.
- Environmental Impact: By incentivizing the use of renewable energy sources and promoting energy efficiency, this system contributes to reducing greenhouse gas emissions and combating climate change. Users can monitor and verify the origin of their energy, ensuring it comes from sustainable sources.

However, there are major challenges in combining this technology with current energy systems such as complying with regulations, maintaining data privacy, and making sure that the blockchain can handle a high volume of transactions.

6.4.2 Distributed Approach

The decentralized and secure characteristics of blockchain when combined with the cooperative learning of P2P-FL can help establish an efficient and sustainable marketplace for renewable energy transactions. The major objectives of this framework are as follows:

- Optimize Transaction Processing: Utilizing distributed learning to enhance the speed and reliability of transactions in the energy trading platform, thereby reducing processing times and costs.

- Minimize Energy Consumption: Implementing P2P-FL to lower the overall energy consumption of the blockchain network, aligning the technology with the sustainability goals of the renewable energy market.
- Improve Trading Efficiency: Streamlining the trading process to make renewable energy more accessible and cost-effective for participants, thereby encouraging the adoption of sustainable energy sources.
- Data Privacy: Utilising a privacy-preserving approach while sharing energy data to preserve the confidentiality of private user data.

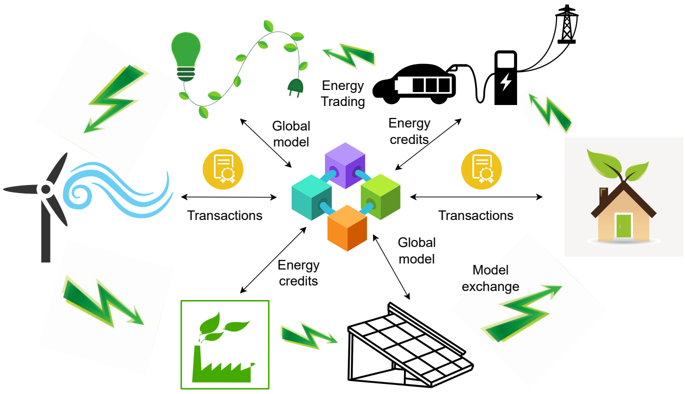


Figure 6.3: Sustainable Energy Trading using Distributed FL and Blockchain.

Considering these objectives, a graphical description of the energy trading model is shown below in Figure 6.3. Each participant in the network, from individual renewable energy producers to large-scale consumers and energy companies, operates as a node on the blockchain. These nodes can execute transactions, involving the buying, selling, or trading of energy credits, represented as digital tokens on the blockchain. In the context of energy trading, where transactions are numerous, the proposed P2P-FL system intelligently balances the load to maintain a balance between capacity and speed, thereby optimizing network throughput and efficiency.

A key benefit of this approach is the focus on the energy efficiency of the blockchain network itself. In a sector centered around sustainability, it is essential that the blockchain's operational energy consumption is minimized. Moreover, the network's ability to dynamically adjust and respond to changing transaction patterns and loads makes it exceptionally adaptive. This responsiveness ensures optimal performance even as energy production and consumption patterns fluctuate which is a common scenario in renewable energy markets.

6.5 Mathematical Formulation

In this section, we present the mathematical foundation of our problem, outlining the strategic interactions among nodes and providing a theoretical basis for the proposed optimization technique. We present a formal analysis through game theory to align each node's individual goals with the overall network's objectives. A non-cooperative game-theoretic model is introduced in the following subsections for optimization of blockchain parameters in a P2P-FL setting.

6.5.1 Problem Definition

We describe our problem in the form of a game where the blockchain network is modeled as a system of interacting nodes. Each node seeks to optimize a set of controllable parameters to maximize its own utility while maintaining the global network performance. The network optimization game is represented by a tuple $\mathcal{G} = (\mathcal{N}, \{\Theta_i\}, \{U_i\})$, where:

- $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of players (nodes).
- Θ_i is the set of strategies for player i, corresponding to the parameters each node will adjust.
- $U_i : \Theta_1 \times \Theta_2 \times \ldots \times \Theta_N \rightarrow \mathbb{R}$ is the utility function for player i, which depends on the collective strategy profile.

The strategy space for each player comprises the tunable parameters of the blockchain protocol, such as transaction processing time, network latency, and other communication delays. These parameters can be represented as a vector for each player:

$$\Theta_i = (T_{p_i}, T_{bg_i}, T_{bp_i}, T_{l_i}, E_i) \tag{6.11}$$

Strategic adjustments to key parameters can significantly enhance the network's throughput and responsiveness. A faster T_p accelerates transaction handling, while shorter T_{bg_i} and T_{bp_i} lead to quicker block creation and synchronization across the network, respectively. A balance between rapid processing and energy efficiency is essential to maintain a robust, effective, and eco-friendly blockchain infrastructure, aligning with both immediate performance goals and long-term sustainability objectives.

6.5.2 Utility Function

The utility function for each node captures its payoff, which is inversely related to the cost associated with the chosen strategy. It includes the node's operational costs, the cost of non-compliance with the optimal state, and penalties for divergent behavior from the network consensus.

$$U_i(\Theta_i, \Theta_{-i}) = -\left(F_i(\Theta_i) + \lambda \sum_{j \in \mathcal{N}_i} d(\theta_i, \theta_j) \right) \tag{6.12}$$

where Θ_{-i} represents the strategies of all players except i, F_i is the cost function related to the node's performance, λ is a regularization parameter, and $d(\theta_i, \theta_j)$ is a measure of the discrepancy between the strategies of node i and the strategies of its peers.

6.5.3 Nash Equilibrium

In game formulation, Nash Equilibrium is a state where no node has an incentive to unilaterally change its strategy given the strategies of the others. This concept is fundamental in non-cooperative games as it represents a stable state from which no player benefits by deviating. A strategy profile $\Theta^* = (\Theta_1^*, \Theta_2^*, \ldots, \Theta_N^*)$ constitutes a Nash equilibrium if no player can unilaterally deviate to improve their utility:

$$U_i(\Theta_i^*, \Theta_{-i}^*) \geq U_i(\Theta_i, \Theta_{-i}^*), \forall i \in \mathcal{N}, \forall \Theta_i \in \Theta_i \tag{6.13}$$

6.5.4 Learning Dynamics

Learning dynamics describe how the nodes adapt their strategies over time in response to the observed utility. This process is modeled as a gradient ascent on the utility function, where each node incrementally adjusts its parameters in the direction that is locally optimal.

$$\theta_i(t+1) = \theta_i(t) + \eta \nabla_{\theta_i} U_i(\Theta_i(t), \Theta_{-i}(t)) \tag{6.14}$$

where η is the learning rate and it determines the size of the adjustments. The gradient $\nabla_{\theta_i} U_i$ points in the direction of the greatest increase of utility for node i, given the current strategy profile.

Therefore, in this game, each node (player) seeks to optimize its utility, which is a function of both its individual performance and the degree to which it aligns with its peers. This model can help analyze and predict the behavior of nodes within the network as they interact and learn from one another.

Table 6.1: Testbed Configuration for P2P FL Blockchain Simulation

Testbed Component	**Specification**
Simulation Platform	X-BlockSim Simulator
Node Implementation	P2P-FL Local, Gateway, and Ethereum Nodes
Consensus Algorithm	PoW, PoS
Number of Nodes	100
Local Model Specs	3-layer Neural Network
Local Batch Size	64
Learning Rate	0.01
Optimizer	Adam
P2P-FL Algorithm	Gossip Learning
Sustainability Metrics Evaluated	Energy Efficiency, Latency
Server Specifications	8 vCPU, 32 GB RAM, 512 GB SSD
Operating System	Ubuntu 20.04 LTS
Programming Language	Python (for both Simulation and FL)

6.6 Simulation Setup

To evaluate the effectiveness and performance of our proposed framework, we used a blockchain network simulator which is an improved version of the BlockSim [140] simulator. It allows us to simulate the deployment of different blockchain models with varying parameters. We have used this simulator for analyzing and benchmarking the different metrics of blockchain that are critical in the performance measurement of our framework. The parameters evaluated are block generation time, latency of the network for block, block propagation delay, and transaction processing time. Initially, we configure a network that connects multiple blockchains through gateways. It is structured into several layers, including the network layer, consensus layer, and application layer. Blockchain nodes perform diverse set of roles within the network layer. They serve as regular users submitting transactions, miners responsible for generating events upon mining a block and adding them to the queue, and gateways responsible for managing transactions from connected nodes. The number of nodes used are calculated using the formula $N = G_n + (G_n \cdot D_n)$, where G_n represents the number of gateway nodes in the network and D_n is the number of devices per gateway in the network. The middle layer includes the consensus algorithm, block generation, and fork resolution. The transactions that are carried out in the network layer are finalized in this consensus layer. Our simulation setup uses several consensus mechanisms (for different blockchains) for finalizing the blocks. The final layer is the application layer, responsible for aggregating all blockchain parameters' statistics. We use a stochastic simulation model with 10 device nodes per gateway in the network and 10 gateway nodes within the network to replicate a small-to-medium sized distributed blockchain. Table 6.1 presents the simulation parameters used in our experiments. Each node in the simulation is characterized by unique computational and network capabilities reflective of a typical Ethereum node. The nodes are heterogeneous in terms of hardware configurations and geographical distribution.

Blockchain Simulation Parameters:

We have configured the simulator to reflect the current Ethereum network characteristics:

- Block Time: Block time is considered as the generation time of the whole block that includes the difference between block creation event and block commit event.
- Transaction Processing Time (TPS): We consider transaction processing time ass the time taken for all the transactions to be verified and added to the block.
- Propagation Delay: It is the overall time required for a block to be propagated across the whole network after its generation.
- Network Latency: The network latency encompasses propagation delay, validation time, and network congestion time.
- Network Adaptation Mechanism: The network's adaptation is event-driven, triggered by the completion of each learning round or upon the occurrence of

significant network events, such as sudden spikes in transaction volume or substantial changes in node connectivity.

FL Parameters:
For the P2P-FL setup, a simple neural network with two hidden layers is chosen for local predictions, sufficient to capture the relationship between the input features and the output parameters. To facilitate effective learning, an initial learning rate of 0.01 was set, which was adaptively modified using an exponential decay rate of 0.96 every 5 epochs. For maintaining the integrity and accuracy of the models, a separate validation dataset comprising 20% of the local data is used. Each node processed data in batches of 32, a size that was found to be optimal given the average computational capabilities of the nodes in our network. A gossip-based protocol is employed for model parameter exchange, where each node randomly selects peers for synchronization and aggregation.

Load Distribution Metrics:
In order to compare our load distribution strategy, we have compared our proposed approach with three load distribution techniques [141]:

- *Random Allocation-* This method involves distributing the load or tasks randomly across all available resources. The primary advantage of this approach is its low time complexity and the fact that it doesn't require tracking the state or performance of resources. However, its randomness does not guarantee an even distribution of load, especially in situations where resources vary in capacity or performance.

- *Round Robin-* Round robin is a systematic, cyclic approach where tasks are allocated to resources in a sequential, rotating order. It ensures that every resource is utilized in a fair and predictable manner, making it suitable for scenarios where each task is of similar nature and requires roughly equal processing time.

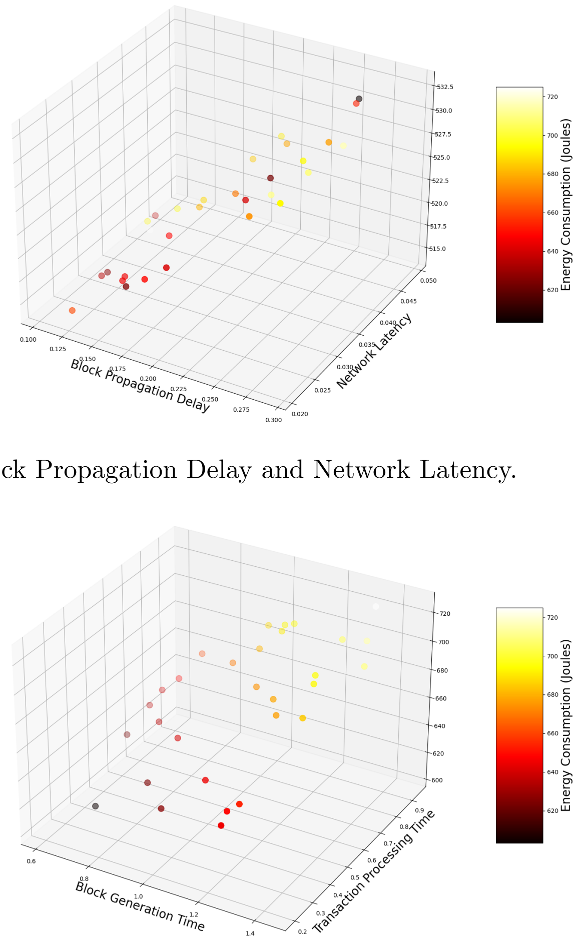
- *Least Loaded-* The least loaded technique involves assigning tasks to the resource that is currently underutilized or has the least load at the moment of assignment. This strategy requires real-time monitoring of resource utilization to determine which one is the least loaded. It aims at balancing the load by preventing certain resources from being overburdened while others are idle. The main challenge with the Least Loaded approach is the overhead involved in continuously monitoring the load on each resource, which can become complex in large-scale systems.

Sustainability Metrics and Evaluation:
The following sustainability metrics are considered in the simulation:

- Energy Efficiency: Measured in Joules reflecting the amount of energy consumed per day on the blockchain network. The given equation (6.10) is utilized to calculate the % of energy consumption.

- Resource Utilization: Assessed through the execution time and load distribution during transaction processing and execution.

To ensure statistical significance, each simulation run is repeated 30 times. Results are subjected to statistical analysis techniques, including variance analysis and hypothesis testing, to determine the reliability and validity of the observed improvements in sustainability metrics.



(a) Block Propagation Delay and Network Latency.

(b) Block Generation Time and Transaction Processing Time.

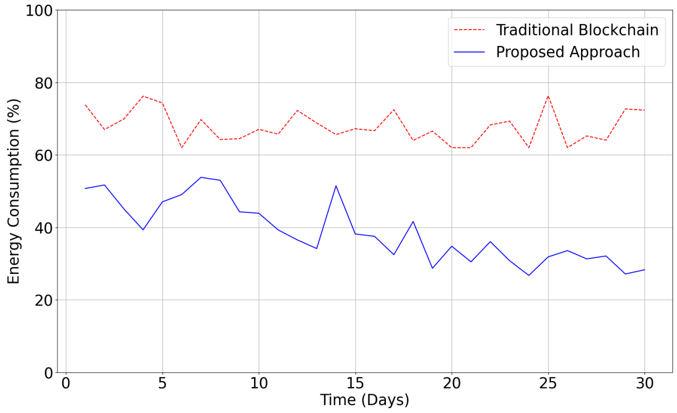
Figure 6.4: Average Energy Consumption (Joules).

6.7 Results and Analysis

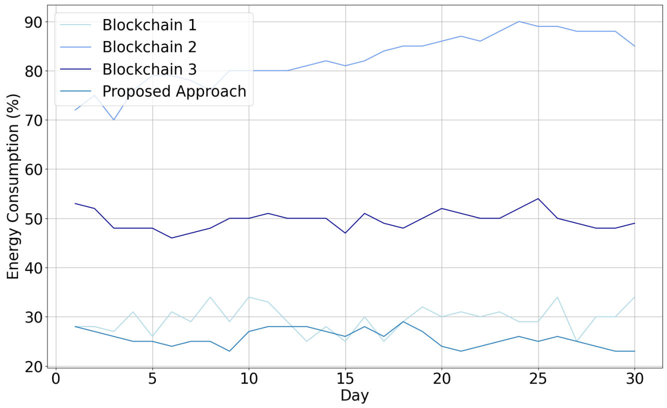
This section presents the results of our simulation and the efficacy of our proposed approach in improving the sustainability and operational efficiency of blockchain networks.

6.7.1 Energy Consumption Analysis

Our first objective was to understand the relationship between energy consumption and various network parameters. We examined the energy consumption as a function of block propagation delay and network latency. The results, as visualized in Figure 6.4 depicts a direct correlation between these variables and energy consumption. As shown in Figure 6.4a, on an increase in the block propagation delay from 0.1 to 0.5 seconds, an incremental trend in energy consumption is observed, indicating that longer propagation delays could significantly impact the overall energy profile of a blockchain network. Similarly, increases in network latency also led to a rise in energy usage, though at a less steep gradient than block propagation delay, showing that its impact is comparatively less pronounced. Figure 6.4b depicts energy consumption variation with block generation time and transaction processing time. Here, the energy consumption exhibited a more complex relationship with the parameters. The plot revealed a region where both increased block generation time and transaction processing time contributed to higher energy consumption, with values reaching up to 720 Joules. Moreover, as transaction processing times increased beyond a certain threshold, energy consumption started to plateau, indicating the presence of an optimal processing time that minimizes energy usage without compromising performance.



(a) Comparison of Energy Consumption.

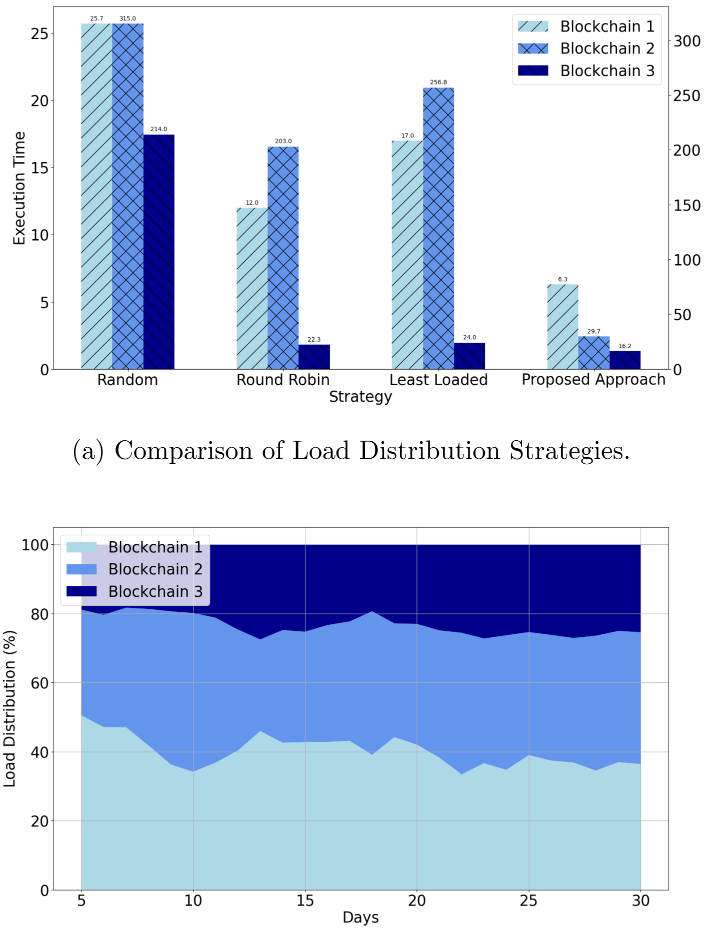


(b) Energy Distribution across Blockchains.

Figure 6.5: Energy Consumption across different Blockchains.

As shown in Figure 6.5a, the energy consumption analysis over time demonstrates a consistently lower energy footprint for our approach as compared to the traditional blockchain model. Throughout the 30-day period, the traditional model's energy consumption fluctuated between 60-80%, while the P2P-FL model sustained a markedly lower range of 40-60%. This contrast underlines our framework's energy efficiency, proving its potential for sustainable blockchain operations.

Figure 6.5b demonstrates the comparative energy consumption percentages of the three distinct blockchain configurations (with varying operational parameters): Blockchain 1 (private), Blockchain 2 (public), and Blockchain 3 (consortium) over a 30-day period. Blockchain 1 exhibits the highest energy consumption, with less fluctuation, indicating a steady demand on computational resources, due to the inherent requirements of maintaining a public ledger. Blockchain 2 and Blockchain 3 show intermediate levels of energy usage with moderate variability, reflecting their hybrid structures which balance the openness of public blockchains with the controlled access of private networks. However, our proposed approach consistently maintains the lowest consumption with a reduction in energy usage by up to 50% or more, signifying a substantial improvement in the energy footprint of blockchain operations.



(a) Comparison of Load Distribution Strategies.

(b) Load Distribution across Blockchains.

Figure 6.6: Load Distribution Performance across different Blockchains.

6.7.2 Load Distribution Across Blockchains

Figure 6.6a provides the comparison of execution times for a particular load on the three blockchain types under various load distribution strategies. It is evident that the proposed P2P-FL approach leads to significantly lower execution times across all blockchain types when compared to other strategies. In the case of random allocation, the execution times are considerably erratic, reflecting a lack of systematic resource allocation which can lead to inefficiencies and bottlenecks. The round-robin and least-loaded strategies offer some improvement by promoting a more equitable distribution of tasks. However, they still do not fully optimize the execution times. Our approach results in the shortest execution times, indicative of its intelligent routing mechanism that efficiently allocates tasks based on real-time network conditions and the operational capacities of each blockchain.

The load distribution over time for our proposed approach is depicted in Figure 6.6b. Across the span of 30 days, Blockchain 1 shouldered the majority of the load initially, but as the days progressed, the model intelligently redistributed the load to Blockchain 2 and Blockchain 3. This dynamic load balancing act underscores the adaptability of our approach to varying network conditions and demands, thus optimizing the overall network efficiency and energy consumption.

6.7.3 Model Training and Predictive Performance

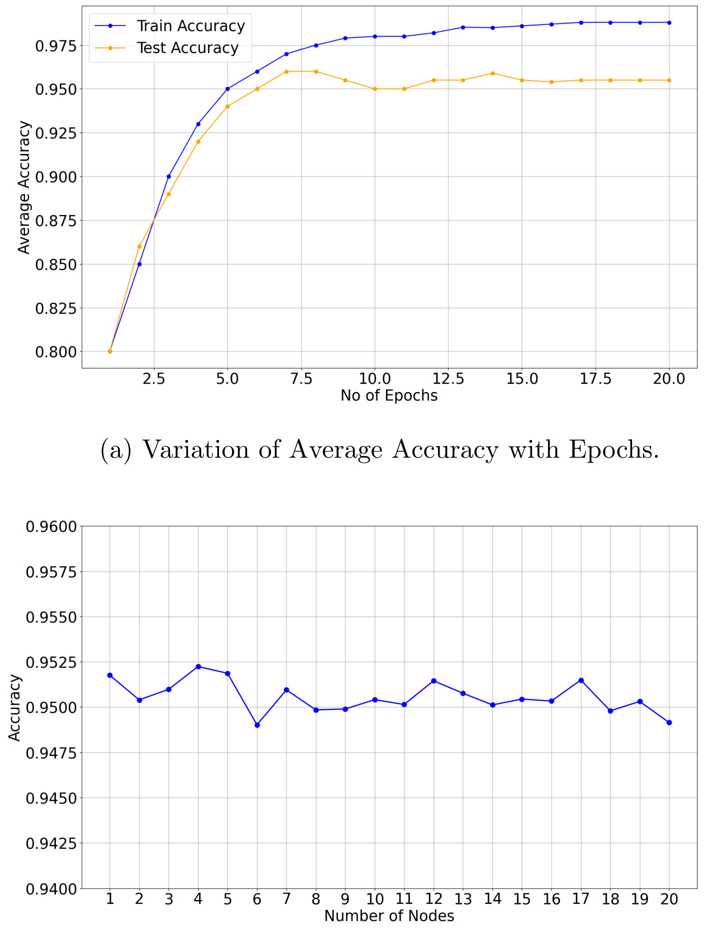
The evaluation of model training and predictive performance of our model is highlighted in Figure 6.7a. The average training accuracy started high and plateaued at around 97.5%, indicating a quick convergence to high-performance levels. The test accuracy closely followed, stabilizing at 92.5%. Figure 6.7b illustrates the individual accuracies of 20 different nodes for 15 epochs. The accuracies vary slightly from node to node, with a mean accuracy hovering around 0.95. This indicates that the P2P-FL framework is effective in synchronizing the learning process across multiple nodes. The model's ability to learn collaboratively without a centralized authority and still reach an accord on the model's accuracy works well with blockchain decentralization and security aspects.

In summary, our results and analysis validate our approach in enhancing the blockchain technology and addressing critical sustainability challenges. The model's decentralized learning mechanism, coupled with its dynamic load distribution, not only reduces energy consumption but also maintains high levels of performance and scalability.

6.8 Security and Scalability Aspect of Proposed Approach

6.8.1 Security Considerations

The P2P-FL model inherently enhances security through its distributed architecture, which eliminates single points of failure and distributes trust among multiple nodes. By keeping the training data local to each node and only sharing model updates, P2P-FL



(a) Variation of Average Accuracy with Epochs.

(b) Variation of Accuracy with No of Nodes.

Figure 6.7: P2P Model Training Accuracy with varying number of Epochs and Nodes.

inherently protects the privacy of the data. This characteristic is crucial in preventing data breaches and leaks. The distributed nature of the P2P-FL also contributes to its robustness against several adversarial attacks that can arise due to the presence of a trusted server. Any attempt to compromise the model would require a substantial portion of the nodes to be compromised simultaneously, which is considerably more difficult to achieve than attacking a centralized system. While the proposed approach brings substantial improvements in the security of blockchain networks, it is important to recognize that it does not currently offer explicit protections against more sophisticated attacks such as model poisoning and model inference [142]. However, techniques such as robust aggregation algorithms, anomaly detection, and secure multi-party computation can be incorporated into the model for mitigating such attacks [124].

6.8.2 Scalability Aspects

Scalability is a critical aspect of blockchain technology, particularly as networks grow and transaction volumes increase. The proposed approach addresses scalability through several mechanisms.

- Dynamic Load Distribution: One of the most significant features of our framework is its ability to dynamically distribute the load across multiple blockchains, including public, private, and consortium chains. This not only enhances the system's ability to scale but also optimizes resource utilization across the network, paving the way for handling larger datasets and more complex computations without a proportional increase in energy consumption or processing time.
- Decentralized Learning and Decision Making: The decentralized learning aspect of P2P-FL facilitates scalability by distributing the computational load required for model training across numerous nodes. This distribution ensures that as the network grows, new nodes contribute additional computational resources, enabling the system to scale naturally. Moreover, decision-making processes that are critical to the network's operation, such as consensus mechanisms and transaction validations, benefit from the collective intelligence of the entire network, leading to more robust and scalable solutions. Finally, since the execution time for our approach is less as compared to other techniques, the network can process transactions and blocks more swiftly, thus increasing throughput without necessitating a proportional increase in resources.

6.9 Summary

Blockchain's potential to contribute to a sustainable future lies in its ability to enable transparent, efficient, and secure transactions across various sectors of IoT. From energy trading to supply chain management, the technology offers an unprecedented opportunity to track, verify, and incentivize sustainable practices. However, the traditional models of blockchain operations, particularly those associated with cryptocurrencies, have been critiqued for their high energy consumption and environmental impact. This chapter demonstrated the integration of P2P-FL with blockchain technology, offering a novel solution to enhance the sustainability and efficiency of blockchain networks. Our approach uniquely combines decentralized learning mechanisms with the inherent security and transparency of blockchain, addressing critical challenges such as scalability, energy consumption, and latency.

Chapter 7

Conclusion

7.1 Summary

In this concluding chapter of the thesis, we summarize our work and describe how we addressed critical challenges and research questions in IoT applications. As more devices connect to the internet, from refrigerators to traffic lights, ensuring their safety and the privacy of the data they handle is crucial. Our research identified key vulnerabilities within the existing methods and the limitations of traditional security frameworks in addressing the decentralized, heterogeneous, and dynamic nature of IoT networks. One major concern is how to keep all these devices and their data safe from hackers and other cyber threats. Another concern is privacy; as these devices collect and share a lot of personal information, so we need to ensure this data doesn't fall into the wrong hands. These vulnerabilities necessitated a robust solution to protect the integrity of data in IoT networks. Therefore, this thesis proposed an innovative integration of blockchain technology and FL to address the dual challenges of ensuring data integrity and confidentiality while maintaining the privacy of user information.

Initially, we discussed the introduction of a blockchain-based architecture for IoT applications. By addressing scalability issues pertinent to blockchains, we demonstrated that blockchain could offer a secure and distributed ledger system capable of authenticating data, authorizing access, and ensuring the integrity of device communications. Further, the utilization of FL was adopted as a solution for preserving user privacy in IoT. By facilitating the local processing of data on IoT devices, FL ensures that sensitive information remains within its origin, thereby minimizing the risk of data breaches. Moreover, our exploration into the optimization of FL models for IoT devices highlighted the potential for achieving high levels of machine learning performance without compromising user privacy. Finally, the thesis addresses the problem of sustainability in blockchain-based frameworks for IoT. Since blockchains consume a lot of energy, it is not computationally efficient for resource-constrained IoT devices. Therefore, our research journey concluded with the development of a sustainable framework that demonstrates the practical viability of blockchain-FL in the IoT context. The prototype also served as a platform for identifying operational challenges and limitations, offering valuable insights that informed our recommendations for future research and development.

7.2 Future Directions

In the future, our work can be extended in several directions. Addressing the challenges due to cyber threats will require continuous advancements in security technologies and strategies. Future work should focus on developing adaptive and resilient security solutions that can preempt and counter emerging cyber threats. Additionally, the exploration of energy-efficient blockchain implementations and the ethical considerations surrounding IoT data collection and processing will be essential in ensuring the sustainable and responsible development of IoT technologies. The following areas represent some directions for future work:

- **Enhancing Blockchain Scalability for IoT:** Future research should focus on further overcoming the scalability challenges of blockchain in IoT applications. Exploring new consensus mechanisms or blockchain architectures designed specifically for the IoT could improve transaction speeds and data processing capabilities, making blockchain a more viable solution for large-scale IoT networks.
- **Advanced Privacy-Preserving Mechanisms in FL:** There is a critical need for the development of more advanced privacy-preserving mechanisms within the FL framework. Future studies could investigate the use of homomorphic encryption, secure multi-party computation, or other cryptographic techniques to enhance privacy guarantees without compromising the utility of the trained models.
- **Optimization of Decentralized FL Architectures:** Further research is required to optimize decentralized FL systems, particularly in designing efficient algorithms and protocols that reduce communication overhead and latency. This could involve the exploration of edge computing to facilitate faster model updates and aggregation in a distributed learning environment.
- **Cross-Domain Applicability of Integrated Blockchain-FL Solutions:** Examining the applicability and effectiveness of the proposed blockchain-FL framework across different IoT domains, such as healthcare, smart cities, and industrial IoT, would provide valuable insights into its versatility and adaptability to various use cases and requirements.
- **Regulatory and Ethical Considerations:** Finally, maintaining regulatory compliance and ethical considerations surrounding IoT data collection and processing will be essential. Future work should explore frameworks that align technological innovations with legal and ethical standards to ensure the responsible and legitimate use of IoT technologies.

7.2.1 Extending Solutions to other IoT Applications

Building upon the solutions and frameworks proposed in this thesis, blockchain technology and FL can be adapted to enhance security and privacy across a wide array of IoT-based applications. The following outlines how these innovations can be extended to different sectors:

1. Smart Cities: The blockchain architecture developed for secure and scalable data management in IoT can be extended to smart city infrastructures. By utilizing smart contracts for automated transaction validation and decentralized data storage, urban IoT systems such as traffic management, public safety, and utility services can benefit from enhanced security and transparency. FL can be employed to optimize urban IoT systems by processing data locally on edge devices such as traffic cameras and sensors, thus preserving privacy. Customized FL models can predict traffic patterns, manage energy distribution, and monitor environmental conditions without centralizing sensitive data.

2. Agricultural Technology: The HierChain framework for data storage and sharing can be applied to agricultural IoT systems. By classifying agricultural data based on sensitivity and storage needs, and utilizing fog nodes for data preprocessing, the integrity and privacy of farm data can be maintained. Moreover, FL can be applied to agricultural IoT to analyze soil health, predict crop yields, and manage resources efficiently while ensuring that individual farm data remains confidential.

3. Industrial Automation: The blockchain security architecture proposed for IoT can secure industrial IoT (IIoT) systems by ensuring the integrity and traceability of machine-to-machine communications and operational data. This framework can be crucial for compliance and monitoring in industrial settings. It can be combined with the P2P-FL approach to predict machine failures and optimize maintenance schedules. By employing privacy-preserving aggregation strategies, industrial data can be analyzed without compromising proprietary information.

4. Healthcare IoT: The FL framework can be adapted to healthcare IoT for predictive analytics in patient monitoring devices. This approach ensures that sensitive health data remains on local devices, thus complying with stringent healthcare privacy regulations while providing valuable insights.

5. Energy Sector: The proposed blockchain solutions can be used to create transparent and secure transactions within energy trading platforms. By implementing sidechains and offline storage, the blockchain can handle the large volumes of data generated by energy IoT devices efficiently.

The outlined future research directions promise not only to extend the contributions of this work but also to inspire further innovations in the field. As we continue to advance in technology and knowledge, our collective efforts will undoubtedly lead to the development of more resilient, efficient, and user-centric IoT solutions.

References

[1] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, “Internet of things security: A survey,” *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017.

[2] A. Wang, R. Liang, X. Liu, Y. Zhang, K. Chen, and J. Li, “An inside look at IoT malware,” in *International Conference on Industrial IoT Technologies and Applications*, pp. 176–186, Springer, 2017.

[3] H. Kim and E. A. Lee, “Authentication and authorization for the internet of things,” *IT Professional*, vol. 19, no. 5, pp. 27–33, 2017.

[4] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, “DDoS in the IoT: Mirai and other botnets,” *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[5] T. Heer, O. Garcia-Morchon, R. Hummen, S. L. Keoh, S. S. Kumar, and K. Wehrle, “Security challenges in the ip-based internet of things,” *Wireless Personal Communications*, vol. 61, pp. 527–542, 2011.

[6] A. Singla, A. Mudgeri, I. Papapanagiotou, and A. Yavuz, “Fast and scalable authentication for vehicular internet of things,” in *Proceedings of the 16th annual information security symposium*, pp. 1–1, 2015.

[7] A. A. Yavuz, A. Mudgerikar, A. Singla, I. Papapanagiotou, and E. Bertino, “Real-time digital signatures for time-critical networks,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2627–2639, 2017.

[8] D. Christin, A. Reinhardt, P. S. Mogre, R. Steinmetz, *et al.*, “Wireless sensor networks and the internet of things: selected challenges,” *Proceedings of the 8th GI/ITG KuVS Fachgespräch Drahtlose sensornetze*, pp. 31–34, 2009.

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

[10] C. M. Medaglia and A. Serbanati, “An overview of privacy and security issues in the internet of things,” in *The Internet of Things: 20 th Tyrrhenian Workshop on Digital Communications*, pp. 389–395, Springer, 2010.

[11] A. Karale, “The challenges of iot addressing security, ethics, privacy, and laws,” *Internet of Things*, vol. 15, p. 100420, 2021.

[12] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, “Internet of things: Vision, applications and research challenges,” *Ad hoc networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[13] S. Alam, M. M. Chowdhury, and J. Noll, “Interoperability of security-enabled internet of things,” *Wireless Personal Communications*, vol. 61, pp. 567–586, 2011.

[14] C. P. Mayer, “Security and privacy challenges in the internet of things,” *Electronic Communications of the EASST*, vol. 17, 2009.

[15] J. Mineraud, O. Mazhelis, X. Su, and S. Tarkoma, “A gap analysis of internet-of-things platforms,” *Computer Communications*, vol. 89, pp. 5–16, 2016.

[16] S. Babar, A. Stango, N. Prasad, J. Sen, and R. Prasad, “Proposed embedded security framework for internet of things (iot),” in *2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE)*, pp. 1–5, IEEE, 2011.

[17] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, “Security and privacy for cloud-based iot: Challenges,” *IEEE Communications Magazine*, vol. 55, no. 1, pp. 26–33, 2017.

[18] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, “Security and privacy challenges in industrial internet of things,” in *Proceedings of the 52nd annual design automation conference*, pp. 1–6, 2015.

[19] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[20] A. Fongen, “Identity management and integrity protection in the internet of things,” in *2012 third international conference on emerging security technologies*, pp. 111–114, IEEE, 2012.

[21] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, “Denial-of-service detection in 6lowpan based internet of things,” in *2013 IEEE 9th international conference on wireless and mobile computing, networking and communications (WiMob)*, pp. 600–607, IEEE, 2013.

[22] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[23] Z. Sheng, S. Yang, Y. Yu, A. V. Vasilakos, J. A. McCann, and K. K. Leung, “A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities,” *IEEE wireless communications*, vol. 20, no. 6, pp. 91–98, 2013.

[24] S. Misra, P. V. Krishna, H. Agarwal, A. Saxena, and M. S. Obaidat, “A learning automata based solution for preventing distributed denial of service in internet of things,” in *2011 international conference on internet of things and 4th international conference on cyber, physical and social computing*, pp. 114–122, IEEE, 2011.

[25] H. Suo, J. Wan, C. Zou, and J. Liu, “Security in the internet of things: a review,” in *2012 international conference on computer science and electronics engineering*, vol. 3, pp. 648–651, IEEE, 2012.

[26] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in internet of things: The road ahead,” *Computer networks*, vol. 76, pp. 146–164, 2015.

[27] R. Roman, P. Najera, and J. Lopez, “Securing the internet of things,” *Computer*, vol. 44, no. 9, pp. 51–58, 2011.

[28] R. H. Weber, “Internet of things: Privacy issues revisited,” *Computer Law & Security Review*, vol. 31, no. 5, pp. 618–627, 2015.

[29] R. Roman, J. Zhou, and J. Lopez, “On the features and challenges of security and privacy in distributed internet of things,” *Computer networks*, vol. 57, no. 10, pp. 2266–2279, 2013.

[30] S. R. Peppet, “Regulating the internet of things: first steps toward managing discrimination, privacy, security and consent,” *Tex. L. Rev.*, vol. 93, p. 85, 2014.

[31] E. R. Naru, H. Saini, and M. Sharma, “A recent review on lightweight cryptography in iot,” in *2017 international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC)*, pp. 887–890, IEEE, 2017.

[32] V. A. Thakor, M. A. Razzaque, and M. R. Khandaker, “Lightweight cryptography algorithms for resource-constrained iot devices: A review, comparison and research opportunities,” *IEEE Access*, vol. 9, pp. 28177–28193, 2021.

[33] D. Dragomir, L. Gheorghe, S. Costea, and A. Radovici, “A survey on secure communication protocols for iot systems,” in *2016 international workshop on Secure Internet of Things (SIoT)*, pp. 47–62, IEEE, 2016.

[34] I. L. B. M. Paris, M. H. Habaebi, and A. M. Zyoud, “Implementation of ssl/tls security with mqtt protocol in iot environment,” *Wireless Personal Communications*, vol. 132, no. 1, pp. 163–182, 2023.

[35] D. Fakhri and K. Mutijarsa, “Secure iot communication using blockchain technology,” in *2018 international symposium on electronics and smart devices (ISESD)*, pp. 1–6, IEEE, 2018.

[36] M. F. Elrawy, A. I. Awad, and H. F. Hamed, “Intrusion detection systems for iot-based smart environments: a survey,” *Journal of Cloud Computing*, vol. 7, no. 1, pp. 1–20, 2018.

[37] J. Du, C. Jiang, E. Gelenbe, L. Xu, J. Li, and Y. Ren, “Distributed data privacy preservation in iot applications,” *IEEE Wireless Communications*, vol. 25, no. 6, pp. 68–76, 2018.

[38] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, “Federated learning with differential privacy: Algorithms and performance analysis,” *IEEE transactions on information forensics and security*, vol. 15, pp. 3454–3469, 2020.

[39] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[40] J. E. Beasley and P. C. Chu, “A genetic algorithm for the set covering problem,” *European journal of operational research*, vol. 94, no. 2, pp. 392–404, 1996.

[41] A. D. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, “A decentralized privacy-preserving healthcare blockchain for IoT,” *Sensors*, vol. 19, no. 2, p. 326, 2019.

[42] O. Attia, I. Khoufi, A. Laouiti, and C. Adjih, “An IoT-blockchain architecture based on hyperledger framework for health care monitoring application,” in *NTMS 10th IFIP International Conference on New Technologies, Mobility and Security*, pp. 1–5, IEEE Computer Society, 2019.

[43] P. P. Ray, D. Dash, K. Salah, and N. Kumar, “Blockchain for IoT-based healthcare: background, consensus, platforms, and use cases,” *IEEE Systems Journal*, vol. 15, no. 1, pp. 85–94, 2020.

[44] R. Jayaraman, K. Salah, and N. King, “Improving opportunities in healthcare supply chain processes via the Internet of Things and blockchain technology,” *International Journal of Healthcare Information Systems and Informatics*, vol. 14, no. 2, pp. 49–65, 2019.

[45] A. Celesti, A. Ruggeri, M. Fazio, A. Galletta, M. Villari, and A. Romano, “Blockchain-based healthcare workflow for tele-medical laboratory in federated hospital IoT clouds,” *Sensors*, vol. 20, no. 9, p. 2590, 2020.

[46] S. Biswas, K. Sharif, F. Li, B. Nour, and Y. Wang, “A scalable blockchain framework for secure transactions in IoT,” *Internet of Things Journal*, vol. 6, no. 3, pp. 4650–4659, 2018.

[47] R. Akkaoui, X. Hei, and W. Cheng, “Edgemedichain: A hybrid edge blockchain-based framework for health data exchange,” *IEEE Access*, vol. 8, pp. 113467–113486, 2020.

[48] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, “LSB: A lightweight scalable blockchain for IoT security and anonymity,” *Journal of Parallel and Distributed Computing*, vol. 134, pp. 180–197, 2019.

[49] A. R. Shahid, N. Pissinou, C. Staier, and R. Kwan, "Sensor-chain: a lightweight scalable blockchain framework for Internet of Things," in *International Conference on Internet of Things and IEEE Green Computing and Communications and IEEE Cyber, Physical and Social Computing and IEEE Smart Data*, pp. 1154–1161, IEEE, 2019.

[50] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.

[51] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," tech. rep., Manubot, 2008.

[52] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Annual International Cryptology Conference*, pp. 357–388, Springer, 2017.

[53] M. Castro, B. Liskov, *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, pp. 173–186, 1999.

[54] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments." [Online]. Available: `https://www.bitcoinlightning.com/`, 2016.

[55] "Raiden network." [Online]. Available: `https://raiden.network/`.

[56] B. Mbarek, N. Jabeur, T. Pitner, *et al.*, "MBS: Multilevel blockchain system for IoT," *Personal and Ubiquitous Computing*, pp. 1–8, 2019.

[57] "RSK." [Online]. Available: `https://www.rsk.co/`.

[58] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[59] P. Angin, M. B. Mert, O. Mete, A. Ramazanli, K. Sarica, and B. Gungoren, "A blockchain-based decentralized security architecture for IoT," in *International Conference on Internet of Things*, pp. 3–18, Springer, 2018.

[60] A. Z. Ourad, B. Belgacem, and K. Salah, "Using blockchain for IoT access control and authentication management," in *International Conference on Internet of Things*, pp. 150–164, Springer, 2018.

[61] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of trust: A decentralized blockchain-based authentication system for IoT," *Computers & Security*, vol. 78, pp. 126–142, 2018.

[62] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "LSB: A lightweight scalable blockchain for IoT security and anonymity," *Journal of Parallel and Distributed Computing*, vol. 134, pp. 180–197, 2019.

[63] O. Novo, "Scalable access management in IoT using blockchain: a performance evaluation," *IEEE Internet of Things Journal*, 2018.

[64] S. Ding, J. Cao, C. Li, K. Fan, and H. Li, "A novel attribute-based access control scheme using blockchain for IoT," *IEEE Access*, vol. 7, pp. 38431–38441, 2019.

[65] H. Wang and J. Zhang, "Blockchain based data integrity verification for large-scale IoT data," *IEEE Access*, vol. 7, pp. 164996–165006, 2019.

[66] M. Ma, G. Shi, and F. Li, "Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the IoT scenario," *IEEE Access*, vol. 7, pp. 34045–34059, 2019.

[67] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning," *Ieee Network*, vol. 33, no. 5, pp. 156–165, 2019.

[68] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *ICC 2019-2019 IEEE international conference on communications (ICC)*, pp. 1–7, IEEE, 2019.

[69] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, 2020.

[70] D. Conway-Jones, T. Tuor, S. Wang, and K. K. Leung, "Demonstration of federated learning in a resource-constrained networked environment," in *2019 IEEE international conference on smart computing (SMARTCOMP)*, pp. 484–486, IEEE, 2019.

[71] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2020.

[72] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE journal on selected areas in communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[73] A. Feraudo, P. Yadav, V. Safronov, D. A. Popescu, R. Mortier, S. Wang, P. Bellavista, and J. Crowcroft, "Colearn: Enabling federated learning in mud-compliant iot edge networks," in *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, pp. 25–30, 2020.

[74] Y. Qu, L. Gao, T. H. Luan, Y. Xiang, S. Yu, B. Li, and G. Zheng, "Decentralized privacy using blockchain-enabled federated learning in fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5171–5183, 2020.

[75] S. Savazzi, M. Nicoli, and V. Rampa, "Federated learning with cooperating devices: A consensus approach for massive iot networks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4641–4654, 2020.

[76] J. Wang, S. Wang, R.-R. Chen, and M. Ji, "Local averaging helps: Hierarchical federated learning and convergence analysis," *arXiv preprint arXiv:2010.12998*, vol. 2, 2020.

[77] Z. Zhao, C. Feng, H. H. Yang, and X. Luo, "Federated-learning-enabled intelligent fog radio access networks: Fundamental theory, key techniques, and future trends," *IEEE wireless communications*, vol. 27, no. 2, pp. 22–28, 2020.

[78] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *ICC 2020-2020 IEEE international conference on communications (ICC)*, pp. 1–6, IEEE, 2020.

[79] S. Pal, A. Ghosh, and V. Sethi, "Vehicle air pollution monitoring using iots," in *Proceedings of the* 16^{th} *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 400–401, ACM, 2018.

[80] G. Srivastava, R. M. Parizi, and A. Dehghantanha, "The future of blockchain technology in healthcare internet of things security," *Blockchain cybersecurity, trust and privacy*, pp. 161–184, 2020.

[81] V. Agarwal and S. Pal, "Blockchain meets IoT: a scalable architecture for security and maintenance," in *17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 53–61, IEEE, 2020.

[82] A. Hafid, A. S. Hafid, and M. Samih, "Scaling blockchains: A comprehensive survey," *IEEE Access*, vol. 8, pp. 125244–125262, 2020.

[83] M. Shuaib, S. Alam, M. S. Alam, and M. S. Nasir, "Compliance with HIPAA and GDPR in blockchain-based electronic health record," *Materials Today: Proceedings*, 2021.

[84] M. Shamila, K. Vinuthna, and A. K. Tyagi, "A review on several critical issues and challenges in IoT based e-healthcare system," in *International Conference on Intelligent Computing and Control Systems (ICCS)*, pp. 1036–1043, IEEE, 2019.

[85] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," *Business & Information Systems Engineering*, vol. 59, pp. 183–187, 2017.

[86] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[87] H. Khan, A. Srivastav, and A. K. Mishra, "Use of classification algorithms in health care," in *Big Data Analytics and Intelligence: A Perspective for Health Care*, pp. 31–54, Emerald Publishing Limited, 2020.

[88] Y. Zhao, J. Zhao, J. Kang, Z. Zhang, D. Niyato, S. Shi, and K.-Y. Lam, "A blockchain-based approach for saving and tracking differential-privacy cost," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8865–8882, 2021.

[89] A. Bozkurt and H. Ucar, "Blockchain technology as a bridging infrastructure among formal, non-formal, and informal learning processes," in *Research Anthology on Adult Education and the Development of Lifelong Learners*, pp. 959–970, IGI Global, 2021.

[90] E. Strehle, "Public versus private blockchains," tech. rep., BRL working paper, Blockchain Research Lab, 2020.

[91] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2567–2572, IEEE, 2017.

[92] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[93] A. Shostack, *Threat modeling: Designing for security.* John Wiley & Sons, 2014.

[94] O. Ardakanian, *Advances in Distribution System Monitoring*, pp. 13–16. Cham: Springer International Publishing, 2020.

[95] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[96] C. Fischer, "Feedback on household electricity consumption: a tool for saving energy?," *Energy efficiency*, vol. 1, no. 1, pp. 79–104, 2008.

[97] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, 2011.

[98] Z. Yue, C. R. Witzig, D. Jorde, and H.-A. Jacobsen, "Bert4nilm: A bidirectional transformer model for non-intrusive load monitoring," in *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring*, pp. 89–93, 2020.

[99] Y. Keneshloo, T. Shi, N. Ramakrishnan, and C. K. Reddy, "Deep reinforcement learning for sequence-to-sequence models," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 7, pp. 2469–2489, 2019.

[100] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[101] S. Sykiotis, M. Kaselimi, A. Doulamis, and N. Doulamis, "Electricity: An efficient transformer for non-intrusive load monitoring," *Sensors*, vol. 22, no. 8, p. 2926, 2022.

[102] Y. Himeur, A. Alsalemi, F. Bensaali, A. Amira, and A. Al-Kababji, "Recent trends of smart nonintrusive load monitoring in buildings: A review, open challenges, and future directions," *International Journal of Intelligent Systems*, vol. 37, no. 10, pp. 7124–7179, 2022.

[103] H. Wang, C. Si, G. Liu, J. Zhao, F. Wen, and Y. Xue, "Fed-NILM: A federated learning-based non-intrusive load monitoring method for privacy-protection," *Energy Conversion and Economics*, vol. 3, no. 2, pp. 51–60, 2022.

[104] Y. Zhang, G. Tang, Q. Huang, Y. Wang, K. Wu, K. Yu, and X. Shao, "Fednilm: Applying federated learning to NILM applications at the edge," *IEEE Transactions on Green Communications and Networking*, 2022.

[105] H. Wang, C. Si, and J. Zhao, "A federated learning framework for non-intrusive load monitoring," *arXiv preprint arXiv:2104.01618*, 2021.

[106] H. Pötter, S. Lee, and D. Mossé, "Towards privacy-preserving framework for non-intrusive load monitoring," in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, pp. 259–263, 2021.

[107] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948, PMLR, 2020.

[108] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning*, pp. 634–643, PMLR, 2019.

[109] H. Hu, Z. Salcic, L. Sun, G. Dobbie, and X. Zhang, "Source inference attacks in federated learning," in *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 1102–1107, IEEE, 2021.

[110] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[111] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-IID data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, 2021.

[112] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-IID data silos: An experimental study," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 965–978, IEEE, 2022.

[113] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE signal processing magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[114] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135, PMLR, 06–11 Aug 2017.

[115] Y. Wang, A. Pandharipande, and P. Fuhrmann, "Energy data analytics for nonintrusive lighting asset monitoring and energy disaggregation," *IEEE Sensors Journal*, vol. 18, no. 7, pp. 2934–2943, 2018.

[116] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[117] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, 2018.

[118] J. Z. Kolter and M. J. Johnson, "REDD: A public data set for energy disaggregation research," in *Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*, vol. 25, pp. 59–62, Citeseer, 2011.

[119] J. Kelly and W. Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes," *Scientific data*, vol. 2, no. 1, pp. 1–14, 2015.

[120] L. Pereira and N. Nunes, "Performance evaluation in non-intrusive load monitoring: datasets, metrics, and tools—a review," *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, vol. 8, no. 6, p. e1265, 2018.

[121] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "Braintorrent: A peer-to-peer environment for decentralized federated learning," *arXiv preprint arXiv:1905.06731*, 2019.

[122] E. De Cristofaro and G. Tsudik, "Practical private set intersection protocols with linear complexity," in *International Conference on Financial Cryptography and Data Security*, pp. 143–159, Springer, 2010.

[123] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, "Using of jaccard coefficient for keywords similarity," in *Proceedings of the international multiconference of engineers and computer scientists*, vol. 1, pp. 380–384, 2013.

[124] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to {Byzantine-Robust} federated learning," in *29th USENIX security symposium (USENIX Security 20)*, pp. 1605–1622, 2020.

[125] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized

and federated learning," in *2019 IEEE symposium on security and privacy (SP)*, pp. 739–753, IEEE, 2019.

[126] I. Hegedűs, G. Danner, and M. Jelasity, "Gossip learning as a decentralized alternative to federated learning," in *Distributed Applications and Interoperable Systems: 19th IFIP WG 6.1 International Conference, DAIS 2019, Held as Part of the 14th International Federated Conference on Distributed Computing Techniques, DisCoTec 2019, Kongens Lyngby, Denmark, June 17–21, 2019, Proceedings 19*, pp. 74–90, Springer, 2019.

[127] H. Gamage, H. Weerasinghe, and N. Dias, "A survey on blockchain technology concepts, applications, and issues," *SN Computer Science*, vol. 1, pp. 1–15, 2020.

[128] C. Schinckus, "The good, the bad and the ugly: An overview of the sustainability of blockchain technology," *Energy Research & Social Science*, vol. 69, p. 101614, 2020.

[129] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.

[130] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized business review*, 2008.

[131] S. Köhler and M. Pizzol, "Life cycle assessment of bitcoin mining," *Environmental science & technology*, vol. 53, no. 23, pp. 13598–13606, 2019.

[132] C. Mora, R. L. Rollins, K. Taladay, M. B. Kantar, M. K. Chock, M. Shimada, and E. C. Franklin, "Bitcoin emissions alone could push global warming above 2 c," *Nature Climate Change*, vol. 8, no. 11, pp. 931–933, 2018.

[133] F. Saleh, "Blockchain without waste: Proof-of-stake," *The Review of financial studies*, vol. 34, no. 3, pp. 1156–1190, 2021.

[134] A. Baliga, "Understanding blockchain consensus models," *Persistent*, vol. 4, no. 1, p. 14, 2017.

[135] N. Truong, G. M. Lee, K. Sun, F. Guitton, and Y. Guo, "A blockchain-based trust system for decentralised applications: When trustless needs trust," *Future Generation Computer Systems*, vol. 124, pp. 68–79, 2021.

[136] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[137] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4734–4746, 2020.

[138] V. Agarwal and S. Pal, "Hierchain: A hierarchical blockchain-based data management system for smart healthcare," *IEEE Internet of Things Journal*, 2023.

[139] M. U. Gurmani, T. Sultana, A. Ghaffar, M. Azeem, Z. Abubaker, H. Farooq, and N. Javaid, “Energy trading between prosumer and consumer in p2p network using blockchain,” in *Advances on P2P, Parallel, Grid, Cloud and Internet Computing: Proceedings of the 14th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2019) 14*, pp. 875–886, Springer, 2020.

[140] C. Faria and M. Correia, “Blocksim: Blockchain simulator,” in *2019 IEEE International Conference on Blockchain(Blockchain)*, pp. 439–446, 2019.

[141] M. J. Peixoto and A. Azim, “A collaborative and distributed task management system for real-time systems,” in *2023 IEEE 26th International Symposium on Real-Time Distributed Computing (ISORC)*, pp. 117–125, IEEE, 2023.

[142] M. S. Jere, T. Farnan, and F. Koushanfar, “A taxonomy of attacks on federated learning,” *IEEE Security & Privacy*, vol. 19, no. 2, pp. 20–28, 2020.