A DISSERTATION
SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN COMPUTER SCIENCE AND ENGINEERING

Optimizing Federated Learning for IoMT and Social Computing Based on Efficiency and Privacy Enhancements

by

Zhuotao Lian

March 2024

The thesis titled

Optimizing Federated Learning for IoMT and Social Computing Based on Efficiency and Privacy Enhancements

by

Zhuotao Lian

is reviewed and approved by:

Chief referee

Senior Associate Professor		Date
Chunhua Su	Su Chunhua	Feb 7, 2024
Professor		Date
Akihito Nakamura	A. Nakamura	Feb. 13, 2024
Senior Associate Professor		Date
Xin Zhu	Zhu Xin	Feb. 14, 2024
Senior Associate Professor		Date
Yasuyuki Kachi	Yasuyuki Kachi	Feb. 15, 2024

THE UNIVERSITY OF AIZU

March 2024

Contents

List of Figures

List of Tables

Abstract

In an era where machine learning technologies are increasingly critical for real-world applications, the quality of these models directly affects their success and utility. High-performing models often require the aggregation of large amounts of dispersed data for training. This centralized machine learning method leads to a key concern: the data collection process can potentially compromise privacy. Additionally, the transmission of this data incurs substantial communication costs and amplifies the risk of data breaches. Federated Learning (FL) emerges as a novel paradigm in this context, enabling collaborative machine learning while keeping the data local. This approach overcomes the limitations and risks associated with centralized machine learning. However, FL faces specific challenges, including significant communication expenses while transferring local updates and potential security risks. "Optimizing Federated Learning for IoMT and Social Computing Based on Efficiency and Privacy Enhancements" recognizes the innovative potential of FL and critically examines its challenges. It aims to pave the way for enhanced efficiency and strengthened privacy measures in Federated Learning applications, aligning with the evolving demands of the Internet of Medical Things (IoMT) and social computing.

The research journey begins with "Blockchain-Based Personalized Federated Learning for Internet of Medical Things". This work addresses the challenges posed by the rapid growth of artificial intelligence, blockchain technology, and edge computing services in the IoMT landscape. Traditional methods typically involve transferring patient data from endpoint devices to central servers for model training, posing significant risks to patient privacy. The heterogeneous nature of patient health conditions necessitates the development of customized healthcare solutions, which are unattainable through uniform models. To tackle these challenges, the study introduces a blockchain-based personalized federated learning system. Notably, the model is divided into base and personalization layers, with only the base layers being aggregated during the FL process. This innovative approach fulfills the need for personalized medical models. The system enables clients to participate in personalized model training without the need for direct data upload, thereby preserving privacy. The incorporation of blockchain technology adds an extra layer of decentralization to the FL process, thereby enhancing overall system security. The effectiveness of this approach is validated through simulations conducted on various datasets, demonstrating the system's robust performance.

The second key section of the dissertation introduces "DEEP-FEL: Decentralized, Efficient, and Privacy-Enhanced Federated Edge Learning for Healthcare Cyber-Physical Systems". The innovative DEEP-FEL framework, a crucial aspect of this research, revolutionizes the coordination among multiple healthcare institutions by applying decentralized FL. The foundation of this framework is an advanced hierarchical ring topology tailored for effective model aggregation. The ring's design is created to tackle a key bottleneck optimization issue, employing a heuristic algorithm that considers the varying

communication speeds among different medical institutions. This strategic ring design substantially reduces the amount of data exchanged, thus significantly boosting communication efficiency. Furthermore, DEEP-FEL employs an efficient parameter aggregation algorithm, which reduces the total amount of data transmitted by N nodes to only 2/N times that of traditional aggregation algorithms. Additionally, the system enhances data privacy across healthcare institutions by incorporating artificial noise into the transferred model.

Building upon our previous research in the medical field, the third part of our dissertation delves into another vital area: social computing, particularly under the COVID-19 pandemic and the advanced development of social media, where an abundance of information, including sensitive patient data, was widely shared. This situation steered my research towards social computing, with a specific focus on fake news detection. "FIND: Privacy-Enhanced Federated Learning for Intelligent Fake News Detection" recognizes the urgency of combating fake news, which poses significant risks to individuals and society. Notably, the study acknowledges that user interactions with news articles, such as browsing and commenting, can also reveal personal preferences and thus pose a threat to user privacy. Traditional machine learning methods for fake news detection often require collecting such sensitive user-side data, thereby increasing the risk of privacy leakage. FIND introduces an intelligent fake news detection system based on federated learning to mitigate this. The system trains a global model while keeping this sensitive data localized, aligning with the dissertation's overarching theme of enhancing communication efficiency and privacy. Additionally, FIND employs a sparsified update perturbation method to strengthen the system's resilience against privacy-compromising threats. The effectiveness of this approach is demonstrated through simulation experiments, showcasing its accuracy, security, and efficiency in the context of social computing.

This dissertation, through the integration of these individual yet interconnected studies, presents a thorough and detailed research of federated learning techniques under the theme "Optimizing Federated Learning for IoMT and Social Computing Based on Efficiency and Privacy Enhancements". Through three interconnected studies, the research introduces novel algorithms and system designs specifically tailored to elevate the efficiency and privacy aspects of federated learning. This comprehensive analysis and exploration mark a significant stride in advancing federated learning, particularly in its application within the dynamic realms of IoMT and social computing.

Chapter 1

Background

1.1 Introduction to Federated Learning

Federated Learning (FL) has emerged as a transformative paradigm in machine learning, gaining significant attention for its ability to address critical issues related to data privacy and communication efficiency. As depicted in Figure 1.1, FL operates on a distributed framework that fundamentally alters the traditional machine-learning pipeline. In a conventional centralized setting, data from various sources are aggregated on a central server for model training. However, this approach raises serious concerns about data privacy and incurs high communication costs.

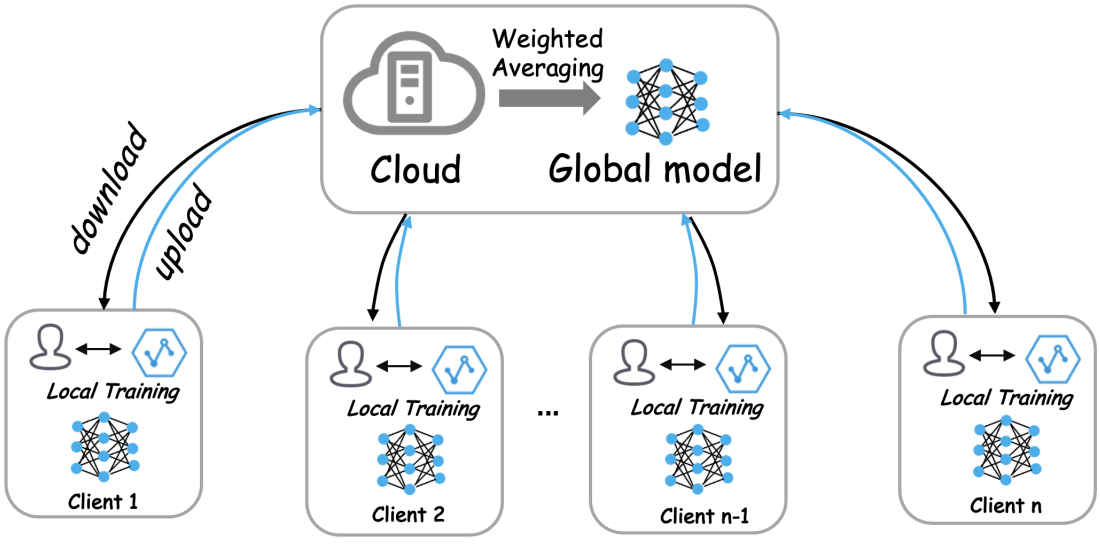


Figure 1.1: Federated Learning

In contrast, FL allows for a more privacy-preserving and communication-efficient methodology. Within this distributed framework, each participating device (commonly known as a node) retains its data locally. These nodes independently train local models based on their respective datasets. Once local training is complete, only the model parameters are sent to a central server for aggregation, thereby forming an updated global model. This global model is then disseminated back to the local nodes for subsequent rounds of local training and model refinement. This iterative process of local training and global aggregation continues until the global model achieves a predefined performance level or a designated number of training epochs is completed.

The distributed nature of FL is particularly advantageous for applications where data privacy is a paramount concern, such as healthcare and financial services. By keeping data localized on individual devices, FL minimizes the risks of data leakage and unauthorized access. Moreover, by transmitting only model parameters instead of raw data, FL significantly reduces the communication overhead, making it a more scalable solution for real-world applications.

This dissertation focuses on enhancing federated learning techniques for IoMT and social computing, as outlined in "Optimizing Federated Learning for IoMT and Social Computing Based on Efficiency and Privacy Enhancements". The aim is to deepen the understanding of this distributed approach, thereby boosting communication efficiency and data privacy. The research contributes to making Federated Learning more practical and robust across various application domains.

1.1.1 Horizontal vs. Vertical Federated Learning

As shown in Figure 1.2, Horizontal Federated Learning (HFL) involves multiple nodes that have different data samples but share the same feature space. This architecture is especially pertinent in healthcare applications where various institutions have diverse patient data but collect similar types of medical information, such as blood pressure, glucose levels, and medical history. In HFL, each institution can train a local model on its unique dataset and then contribute to a global model, thereby benefiting

from the collective intelligence without compromising data privacy.

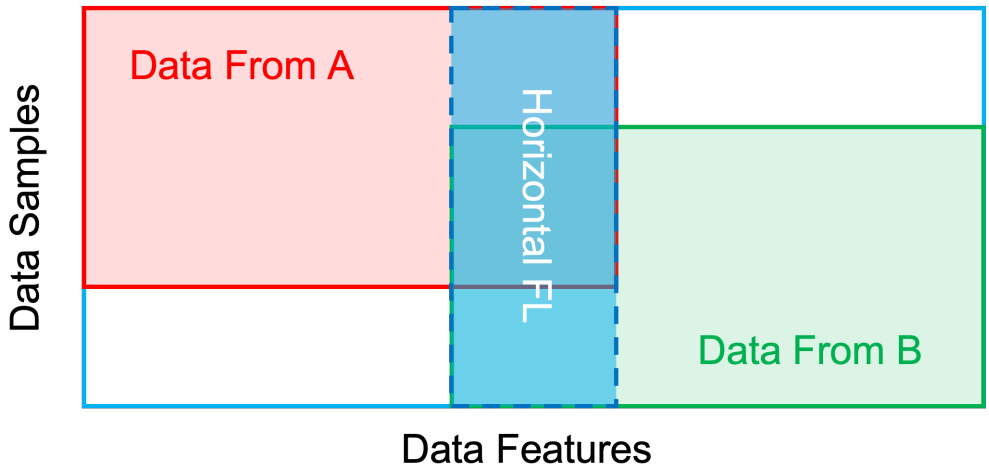


Figure 1.2: Horizontal Federated Learning

Conversely, Figure 1.3 illustrates the concept of Vertical Federated Learning (VFL), wherein different organizations possess the same data samples but different feature spaces. This is more suited for business collaborations, where, for example, a retail company and a credit card company might have information on the same set of customers but collect different types of data.

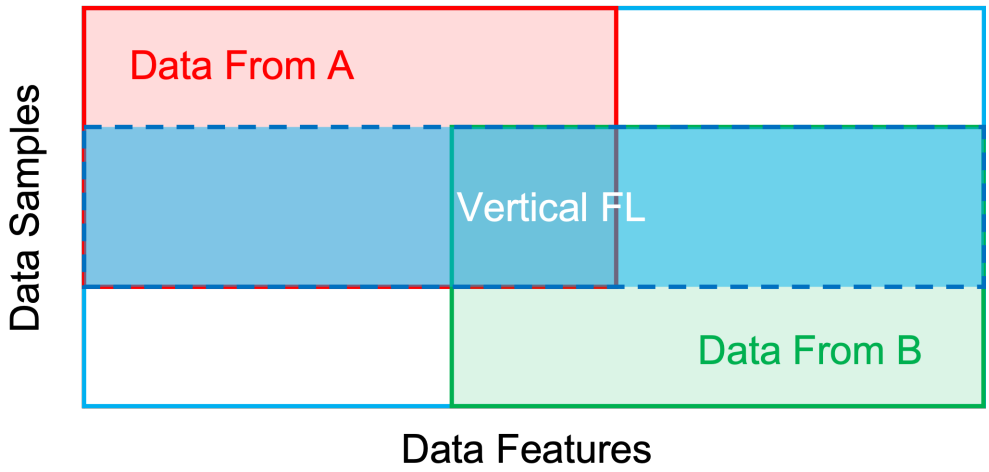


Figure 1.3: Vertical Federated Learning

It is noteworthy that the primary focus of this dissertation is on Horizontal Federated Learning. The healthcare-centric studies, such as DEEP-FEL and the Blockchain-Based Personalized Federated Learning for the Internet of Medical Things, employ HFL to address challenges in data privacy and communication efficiency. These works lever-

age the architecture of HFL to enable decentralized training across multiple healthcare institutions while ensuring robust privacy protections and optimized communication protocols.

1.1.2 Federated Learning Process

The Federated Learning process is an intricate orchestration of several key steps, as illustrated in Figure 1. These steps are foundational to the challenges and solutions discussed in this dissertation. The process begins with Initialization, where a global model is initialized and distributed to all participating nodes. This sets the stage for Local Training, the step where each node trains the model on its local dataset. It is at this juncture that the research presented in this dissertation introduces novel techniques for enhancing communication efficiency, particularly in the context of Horizontal Federated Learning.

Following Local Training, the next steps are Model Update and Aggregation. In Model Update, the locally trained models are sent back to a central server or a designated node for aggregation. This is a critical phase where the dissertation introduces optimized algorithms to minimize communication overhead and enhance privacy. For instance, the DEEP-FEL system employs a Ring-All-Reduce-based parameter update mechanism to significantly reduce the volume of data that needs to be communicated during this phase.

The Aggregation step is where the global model is updated with contributions from all nodes. This is another area where the dissertation makes significant contributions, particularly in ensuring that the aggregation process is both efficient and privacy-preserving. Techniques such as differential privacy and blockchain technology are integrated into this step to fortify the global model against potential privacy attacks.

By understanding these key steps in the Federated Learning process, one gains a comprehensive view of the challenges that this dissertation aims to address. Each of the works presented herein offers unique solutions to these challenges, thereby contributing to a more robust and efficient FL system.

1.1.3 Challenges and Opportunities: Setting the Stage for This Dissertation

Federated Learning, despite its transformative potential, is not without its challenges, which offer fertile ground for academic and practical contributions. These challenges are not merely theoretical constructs but have real-world implications, especially in sensitive sectors like healthcare and social networks. The primary challenges that this dissertation aims to address are twofold: communication efficiency and robust privacy protections.

In the realm of communication efficiency, the dissertation introduces novel techniques to minimize the overhead associated with transmitting model updates across nodes. For example, in the DEEP-FEL system, a heuristic algorithm is employed to optimize the ring topology based on the communication speeds between healthcare institutions, thereby reducing the data volume required for model aggregation.

On the privacy front, the dissertation explores various avenues to safeguard user data. In the healthcare sector, where patient data is highly sensitive, techniques like differential privacy and blockchain technology are employed to ensure robust privacy protections. In the context of social networks, where user interactions can reveal personal preferences and biases, the dissertation introduces perturbed sparsified model update methods to ensure that user data remains private during the model training process for fake news detection.

These challenges, along with their respective solutions, form the foundation of the studies presented in this dissertation. Each work contributes uniquely but synergistically to addressing these challenges, thereby enhancing the robustness and efficiency of Federated Learning systems, particularly in the context of Horizontal Federated Learning as illustrated in Figures 1.1 and 1.2.

1.1.4 Federated Learning Frameworks

This section is committed to an extensive exploration of Federated Learning (FL) frameworks, elucidating their widespread applications in both the industrial sector and academic research. This section covers the technical details and architectural characteristics of various frameworks and includes detailed introductions to their application scenarios and examples. Special emphasis is laid on how these simulation frameworks support the testing and development of innovative algorithms and their capabilities in handling complex datasets and simulating real-world conditions. Finally, this section will detail the specific FL framework employed in our research and the custom modifications made to it. This will encompass an in-depth analysis of the motivation behind choosing this framework, its application in our research, and the specific adjustments made to meet the research needs. We will explore how these modifications have enhanced the functionality of the framework, especially in terms of processing data privacy and improving computational efficiency.

One of the prominent federated learning frameworks is Google's TensorFlow Federated (TFF). Developed by Google Brain Team, TFF is an open-source framework specifically designed for machine learning and data processing on decentralized data [1]. TFF is known for its flexibility and scalability, making it suitable for a wide range of applications, from predictive modeling in healthcare to personalized content recommendation in streaming services. Its key feature is the ability to run federated computations on mobile devices while ensuring data privacy.

Another significant framework is IBM's Federated Learning. This framework is the brainchild of IBM Research and is tailored for collaborative machine learning without sharing sensitive data [2]. It is widely recognized for its robust security and privacy features, making it a go-to choice for industries like finance and healthcare. IBM's Federated Learning framework integrates advanced cryptographic techniques, such as differential privacy and secure multi-party computation, to safeguard data during the learning process.

NVIDIA's Clara Federated Learning (Clara FL) is another noteworthy framework in

this domain [3]. Designed by NVIDIA, Clara FL focuses on healthcare and biomedical applications. It allows institutions to collaborate on developing AI models without sharing patient data, thereby maintaining patient privacy. The framework is renowned for its high-performance computing capabilities, leveraging NVIDIA's GPU technology, and is instrumental in medical imaging and genomic sequencing applications.

Federated AI Technology Enabler (FATE): FATE, an open-source project initiated by WeBank's AI Group, is another framework extensively used in academic research for FL [4]. Designed to support various federated learning architectures and scenarios, FATE emphasizes data privacy and security. It enables researchers to conduct experiments in a privacy-preserving environment, which is crucial for sensitive data applications. FATE is commonly applied in finance for risk prediction models and in healthcare for collaborative disease prediction and analysis.

PyTorch Federated (PySyft): PyTorch Federated, developed by the open-source community and powered by PySyft library, is a popular choice among researchers for experimenting with FL [5]. This framework extends PyTorch to enable secure and private deep learning across distributed datasets. PySyft is particularly known for its user-friendly interface and support for advanced techniques like secure multi-party computation and differential privacy. Researchers use PySyft for a range of applications, from enhancing privacy in natural language processing models to developing secure, collaborative AI models in healthcare.

1.1.5 Framework Used in Our Research

In our study, the primary framework for Federated Learning (FL) is based on PyTorch Federated, with significant custom modifications to suit our specific research needs, particularly in the context of the Internet of Medical Things (IoMT). One notable customization is the elimination of the global server's role in our FL setup. Instead, we have integrated blockchain technology for parameter updating. This approach aligns with our designed decentralized parameter update algorithm, which is a key innovation in our research.

Additionally, we have developed a simplified version of a federated learning simulation using Keras. This simplified model was primarily utilized to simulate the impact of various parameter selection methods on the accuracy of the models. By focusing on parameter selection, we were able to omit the communication aspect in the simulations, allowing for more convenient and quicker preliminary results. This approach was particularly beneficial in streamlining the experimentation process, enabling us to focus on the accuracy and efficiency of the federated learning models under various conditions.

These customizations to the PyTorch Federated framework and the use of a Keras-based simulation model have been instrumental in addressing the unique challenges posed by IoMT applications. They allowed us to explore the intricacies of federated learning in a healthcare context, where data privacy and efficient model training are of paramount importance. Our approach demonstrates the flexibility of federated learning frameworks and showcases the potential for innovative adaptations to meet specific research objectives.

1.2 Fundamentals of Data Privacy and Communication Efficiency

This section lays the groundwork for understanding the two pivotal aspects that this dissertation focuses on: Data Privacy and Communication Efficiency in the context of Federated Learning.

1.2.1 Data Privacy in Federated Learning

Data privacy is of paramount importance in Federated Learning, particularly due to the decentralized nature of the data. Federated Learning systems are susceptible to various threats, such as model inversion and membership inference attacks, among others. To address these challenges, this dissertation studies two key technologies: Differential Privacy and Blockchain.

Differential Privacy

Differential Privacy, a cornerstone in privacy-preserving techniques, involves adding calibrated noise to data or query results. In the context of Federated Learning, Differential Privacy can be applied in two primary ways: Centralized Differential Privacy (CDP) and Local Differential Privacy (LDP).

Centralized Differential Privacy (CDP): This approach involves adding noise centrally to the aggregated model updates. This is usually done on the server side after collecting the updates from all participating nodes. While effective, it requires trust in the central server to maintain privacy.

Local Differential Privacy (LDP): More desirable in Federated Learning scenarios, LDP involves adding noise at the local device level. Here, noise is added at the local device level before any model updates are sent to the central server. This ensures that each participant's data remains private, even from the central server, thus offering a more robust privacy guarantee.

Blockchain Technology

Blockchain Technology serves as another pivotal element in enhancing data privacy within Federated Learning frameworks. Known for its secure and transparent transaction recording capabilities, blockchain technology brings an additional layer of trust to decentralized systems. In the context of Federated Learning, it offers the following advantages:

- Immutable Records: Once a transaction, such as a model update, is recorded on the blockchain, it becomes immutable. This ensures that data or model parameters cannot be tampered with, thereby enhancing data integrity.

- Consensus Mechanism: Blockchain operates on a consensus mechanism, typically Proof of Work (PoW) or Proof of Stake (PoS), requiring the agreement of all participating nodes to validate a transaction. This collective validation adds an extra layer of security and trust.

- Transparency and Auditability: The transparent nature of blockchain allows for easy auditability of transactions. This is crucial in Federated Learning where multiple parties are involved, and transparency is required to ensure that no malicious activities are taking place.

By integrating blockchain technology into Federated Learning, one can achieve a more secure and transparent environment, which is particularly beneficial in sensitive applications like healthcare and finance.

1.2.2 Communication Efficiency in Federated Learning

Communication efficiency stands as a cornerstone in the practical implementation of Federated Learning (FL). The communication overhead, which encompasses both the volume of data transferred and the frequency of these transfers, can significantly impede the speed of model training and place a strain on computational resources. This section elaborates on key techniques aimed at mitigating these challenges:

Random Selection Mechanism

In Federated Learning, the dynamic nature of the system poses additional challenges to communication efficiency. Nodes or clients can join or leave the network at any time, making it highly volatile. Random Selection Mechanisms address this issue by stochastically selecting a subset of available nodes for each training round. This approach not only reduces the communication overhead but also enhances the system's stability by accommodating the dynamic participation of nodes. Moreover, the randomness introduced through this mechanism can be beneficial for the model's generalization capabilities, making it more resilient to overfitting and better suited for real-world applications.

Model Compression Techniques

Model compression techniques offer a viable path to improve communication efficiency in Federated Learning by reducing the size of the model updates sent back to the central server. This is particularly important in resource-constrained environments such

as edge computing or Internet of Things (IoT) devices. Among various techniques, the following are commonly used:

- Quantization: This technique reduces the number of bits required to represent the model's parameters, thereby shrinking the model size without significantly compromising its performance.

- Pruning: This involves eliminating certain neurons or connections in the neural network that have minimal impact on the model's performance, thus reducing the model's complexity.

- Sparsification: A key focus in our research, sparsification further enhances model compression by converting a dense model into a sparse format. After sparsification, only the most significant parameters are retained, and the rest are set to zero. This allows for even more efficient communication as the sparse model can be further compressed before transmission.

By incorporating these model compression techniques, especially sparsification, Federated Learning systems can achieve substantial improvements in communication efficiency.

1.3 Overview of the Dissertation's Contributions

This dissertation, "Optimizing Federated Learning for IoMT and Social Computing Based on Efficiency and Privacy Enhancements", takes a comprehensive approach to addressing the challenges in Federated Learning (FL), particularly focusing on critical aspects of data privacy and communication efficiency. The contributions presented here delve deep into the realms of IoMT and social computing, offering extensive analysis and practical insights. The three main contributions are as follows:

1. In-Depth Analysis of Privacy in Federated Learning. The first key contribution of this dissertation is a comprehensive and detailed analysis of privacy enhancement

techniques within the federated learning environment. The study thoroughly explores commonly used data anonymization methods in the FL framework and how integrating blockchain technology can strengthen system security. This section not only discusses strategies to counter various threats such as inference attacks and data poisoning but also includes practical case studies on implementing these innovative techniques in IoMT and social computing scenarios. Furthermore, the paper examines the effectiveness of these technologies in protecting user privacy and enhancing data security, especially when handling sensitive medical data and personal information on social media platforms.

2. Focus on Optimizing Communication Efficiency in FL. The second major contribution is the in-depth optimization of communication efficiency in FL systems. Focusing specifically on optimizing parameter transmission in federated learning process, the dissertation introduces new algorithms and sparsified network structures for this purpose. These include the development of improved model sparsification algorithms and client selection algorithms, significantly reducing data transmission requirements and increasing the training efficiency of FL systems. The research also delves into the application of these methods in various FL environments, including scenarios with limited bandwidth and resources, and discusses how these technologies can accelerate data processing, leading to more efficient learning processes.

3. Cross-Domain Application and Evaluation. The third contribution encompasses extensive exploration and rigorous evaluation of FL applications across various domains, particularly highlighting the intersection of healthcare (IoMT) and social computing. This section conducts a thorough evaluation of optimized FL techniques in different contexts, demonstrating their adaptability and effectiveness in both IoMT and social computing environments. The dissertation provides detailed analyses of how federated learning tackles complex challenges in these areas, from sensitive handling of patient data in IoMT to effective identification and countering of misinformation in social networks. Through these case studies,

the paper showcases the diversity and flexibility of FL technology in practical applications, and its ability to address specific real-world problems.

Necessity of the Three Works

The necessity of the three works presented in this dissertation arises from the existing gaps and challenges in Federated Learning (FL). These works are not merely academic exercises but are crucial for the broader implementation and acceptance of FL in sensitive and critical domains.

1. Personalization in healthcare is increasingly becoming a requirement rather than an option. However, the need for personalized models often conflicts with data privacy concerns. The first work resolves this dilemma by employing blockchain technology, enabling personalized Federated Learning without compromising data privacy.

2. The healthcare sector, especially the Internet of Medical Things (IoMT), demands stringent data privacy and computational efficiency. Traditional centralized models are not only inefficient but also pose significant risks to patient privacy. The second work addresses these challenges by introducing a decentralized Federated Learning system tailored for healthcare applications.

3. The proliferation of fake news in social networks poses a significant societal risk. Traditional machine-learning approaches for fake news detection often require the collection of user data, which raises privacy concerns. The third work addresses this by employing Federated Learning and introducing additional security measures, making it a privacy-preserving solution for fake news detection.

Each work addresses a specific, pressing need in the application of Federated Learning, thereby filling existing gaps in the literature and practice. The importance of these works is multi-faceted. They contribute to enhancing the robustness and efficiency of FL systems, making them more viable for real-world applications. By specifically

addressing the challenges of data privacy and communication efficiency, these works pave the way for a more secure and efficient FL system.

1.4 Publications

The following first-authored papers have been published in major peer-reviewed journals and conferences during the doctoral program. The corresponding results of three major publications are presented in Chapters 3, 4, and 5.

Peer-Reviewed Major Journal papers

1. **Z. Lian**, W. Wang, Z. Han and C. Su, "Blockchain-Based Personalized Federated Learning for Internet of Medical Things," in IEEE Transactions on Sustainable Computing. (2023, SCI, IF: 3.9)

2. **Z. Lian** et al., "DEEP-FEL: Decentralized, Efficient and Privacy-Enhanced Federated Edge Learning for Healthcare Cyber-Physical Systems," in IEEE Transactions on Network Science and Engineering, vol. 9, no. 5, pp. 3558-3569, 1 Sept.-Oct. 2022. (2022, SCI, IF: 6.6)

3. **Z. Lian**, C. Zhang, C. Su, F. A. Dharejo, M. Almutiq and M. H. Memon, "FIND: Privacy-Enhanced Federated Learning for Intelligent Fake News Detection," in IEEE Transactions on Computational Social Systems. (2023, SCI, IF: 5)

4. **Z. Lian**, Q. Zeng, W. Wang, T. R. Gadekallu and C. Su, "Blockchain-Based Two-Stage Federated Learning With Non-IID Data in IoMT System," in IEEE Transactions on Computational Social Systems, vol. 10, no. 4, pp. 1701-1710, Aug. 2023, doi: 10.1109/TCSS.2022.3216802. (2023, SCI, IF: 5)

5. **Z. Lian**, Q. Zeng, W. Wang, D. Xu, W. Meng and C. Su, "Traffic Sign Recognition using Optimized Federated Learning in Internet of Vehicles," in IEEE Internet of Things Journal. (2023, SCI, IF: 10.6)

Peer-Reviewed Major Conference Papers

1. **Z. Lian**, C. Zhang, K. Nan and C. Su, "SPoiL: Sybil-Based Untargeted Data Poisoning Attacks in Federated Learning," NSS 2023 - Network and System Security, Kent, UK, 2023. Lecture Notes in Computer Science, vol 13983. Springer. (CORE (Computing Research and Education Association of Australasia) B)

2. **Z. Lian**, Q. Yang, Q. Zeng and C. Su, "WebFed: Cross-platform Federated Learning Framework Based on Web Browser with Local Differential Privacy," ICC 2022 - IEEE International Conference on Communications, Seoul, Korea, Republic of, 2022, pp. 2071-2076. (CORE B)

3. **Z. Lian**, Q. Zeng and C. Su, "Privacy-preserving Blockchain-based Global Data Sharing for Federated Learning with Non-IID Data," 2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW), Bologna, Italy, 2022, pp. 193-198.

Chapter 2

Thesis Structure

The structure of this dissertation is designed to provide a comprehensive exploration of the optimization of Federated Learning (FL) techniques while preserving data privacy and improving communication efficiency. The journey through this dissertation begins with a solid foundation in the "Background" chapter, where the fundamental concepts of FL are introduced, including its two primary architectures, horizontal and vertical FL, the essential FL processes such as Initialization, Local Training, Model Update, and Aggregation and commonly used frameworks. Furthermore, the "Challenges and Opportunities" chapter sets the stage for the three core works by highlighting the challenges faced by FL, including data privacy and communication efficiency, and their significance in real-world applications.

The subsequent chapters of this dissertation delve into the three seminal works that constitute its core contributions:

- Chapter 3:"DEEP-FEL: Decentralized, Efficient and Privacy-Enhanced Federated Edge Learning for Healthcare Cyber-Physical Systems" focuses on the application of Federated Learning in healthcare, particularly in the Internet of Medical Things (IoMT). This work introduces a decentralized Federated Learning system that enhances both data privacy and communication efficiency, making it particularly suitable for healthcare applications.

- Chapter 4: "Blockchain-Based Personalized Federated Learning for Internet of

Medical Things" addresses the need for personalized healthcare models in IoMT. This work employs blockchain technology to secure the Federated Learning process, allowing for personalized model training without compromising data privacy.

- Chapter 5: "FIND: Privacy-Enhanced Federated Learning for Intelligent Fake News Detection" tackles the issue of fake news detection in social networks. This work employs Federated Learning to train a global model for fake news detection while keeping user data localized. It also introduces a sparsified update perturbation method to further enhance system security.

The concluding chapter, "Conclusion and Future Directions", summarizes the key findings and contributions of the dissertation, emphasizing the advancements made in FL optimization. It also offers insights into potential future research directions in the field of FL, paving the way for further innovations in this dynamic domain.

Chapter 3

Blockchain-based Personalized Federated Learning for Internet of Medical Things

3.1 Introduction

In recent years, with the development of smart devices and the advancement of network technology, the Internet of Things (IoT) has attracted more and more attention from industry and academia. Because of the integration of sensing, computing, and communication capabilities, IoT has also demonstrated potential in healthcare and is expected to make more significant progress [6]. Specifically subdivided, the actual application of IoT in the healthcare field is called the Internet of Medical Things (IoMT) [7]. The applications of IoMT include monitoring, inspection, report analysis, data collection, calculation, etc [8].

For example, wearable devices can collect patient health data and transmit it to hospitals or medical institutions for health monitoring, disease diagnosis, and treatment [9]. As a user, patients can use the real-time tracking and monitoring brought by IoMT. As a data source, they can generate valuable medical data for further integrated research in the medical center. With the availability of massive amounts of data, general machine

learning (ML) models and deep learning (DL) techniques are being applied in many fields including healthcare [10]. Recently, increasingly machine learning techniques are being applied in the medical field by using the collected data for model training and prediction. Hence, IoMT has significantly improved the operation of healthcare services [11].

However, studies have shown that aggregated sensitive medical data are exposed to security risks [12]. In addition, health-related data in IoT devices is so closely tied to people's privacy that casual sharing or aggregation seems impractical [13]. Federated learning (FL) was proposed to take into account the processing of big data and protect the privacy of clients [14]. It enables local training on devices and the update of the model is achieved by global parameter aggregation, avoiding the privacy leakage of the raw data. Despite this, most of the existing FL research aims to train the same global model. However, in the healthcare domain, this is not necessarily optimal.

In the medical field, the data are often non-independently and identically distributed (non-IID). Patients have different physical characteristics, such as age and gender, which can make the data collected by IoMT devices vary greatly, even for patients diagnosed with the same disease. To illustrate with a concrete example, an adult's activity might have very few falls and mostly walking or standing data samples. However, data on falls in young children may be relatively plentiful. This uneven distribution can significantly reduce the efficiency and accuracy of learning when training a centralized model [6].

Training on non-IID data is a common problem in the field of FL, and most of the existing work is to improve the training effect or prediction accuracy of models through improvements at the algorithm and architecture levels. In other words, the disparate impact of non-IID data can be weakened [15–18]. However, the needs of real situations are often different. Personalization of global models is necessary to address the problem of non-IID data [19], especially for certain domains where personalized models can provide better customization.

In addition, traditional FL often relies on a centralized parameter server for parameter aggregation and model updates. However, this centralized architecture has corre-

sponding drawbacks, such as the single point of failure problem and security issues [20]. Fortunately, the emergence of blockchain offers a good solution for decentralized FL. The blockchain is a distributed ledger that comprises a series of data blocks in sequential order. All the participants maintain and monitor the transaction data recorded in the blockchain. Through the integration of blockchain and FL, the parameter aggregation and training task coordination processes can be decentralized and performed in a secure manner [21]. For example, Lu et al. [22] combined an asynchronous FL scheme with a hybrid blockchain to strengthen secure and decentralized data sharing for the Internet of Vehicles (IoV). Moreover, the node selection in the blockchain is optimized by Deep Reinforcement Learning (DRL), which improves the efficiency of the training stage. Considering the omitted poisoning attacks in the FL, Qu et al. [23] proposed a novel blockchain-based FL framework to achieve a balance between privacy, security, and efficiency issues in the fog computing environment. FL-Block enables mobile devices to exchange local model updates with a global learning model among the blockchain. However, few research concentrates on the blockchain-based personalized FL for IoMT in the current stage.

In order to cope with all the above problems, we propose blockchain-based personalized FL for the IoMT scenario. First, to address the data privacy issue, we adopt the FL paradigm that enables IoMT devices to collaboratively train a global model in a form that protects data security. Second, to address the non-IID problem, we achieve a personalized model with better results by globally updating the trained base layers of the model and keeping the personalization layers local. Finally, to further enhance privacy protection, avoid the single point of failure, etc., we use blockchain technology to perform global updates in FL with a decentralized architecture.

In summary, our main contributions are as follows:

- To address the security and privacy concerns associated with medical data in the IoMT scenario, we propose a blockchain-based personalized federated learning system that enables the global model to be trained collaboratively among distributed devices while keeping private patient data locally.

- We adopt the model partitioning approach to divide the model into base layers which will be aggregated and updated globally, and personalization layers which capture personalized features and do not participate in global aggregation. This enables us to achieve personalized models with better results on heterogeneous medical data than a one-size-fits-all global model. Additionally, we deploy the FL parameter updates on the blockchain to further strengthen privacy protection and avoid the potential single point of failure.

- We conduct exhaustive experiments to demonstrate the superiority of our system in meeting the personalized medical needs of individual patients, as compared to conventional FL methods.

The structure of this article is as follows. Section 2 provides the background knowledge of our proposed method. In Section 3, we present the system design of our blockchain-based personalized FL. Section 4 discusses potential attacks on our framework and corresponding countermeasures. Section 5 presents detailed simulation experiments to validate our system. Finally, in Section 6, we conclude this article and discuss future improvements.

3.2 Related Work

3.2.1 Internet of Medical Things

Compared to traditional medical devices, IoMT not only has its reliability and security but also has the versatility, scalability, and dynamism of IoT [24]. IoMT considers a network of interconnected medical devices and people to further provide better care and a higher quality of life for patients by combining IoT technology and healthcare [25]. IoMT has been widely researched and applied in remote health monitoring, fitness programs, smart hospital, etc. Specifically, IoMT can be used for remote monitoring of patient's heart rate, as in [26], they propose a mobile system for ECG detection in cardiac patients using wireless body sensors and commercial off-the-shelf sensors.

To deal with security and privacy issues in the physical layer, Wang et al. [27] combined emerging Blockchain and PUF to propose a lightweight authentication protocol for IoMT. According to Elliptic Curve Digital Signature Algorithm (ECDSA), the authors [28] suggested an efficient and massive batch verification scheme where group testing technology is also imported. A novel model called XSRU-IoMT was proposed in [29], which can quickly and accurately detect adversarial vectors which exist in IoMT networks. To raise the speed of the training process, the authors applied skip connections to bidirectional simple recurrent units (SRU). Almogren et al. [30] introduced a fuzzy-based trust management mechanism to mitigate potential Sybil attacks for Internet of Medical Things (IoMT), which provides a trust management system for eHealth users. Nguyen et al. [31] integrated mobile edge computing (MEC) and blockchain to construct distributed hospital networks that enable secure and efficient data offloading and data sharing.

3.2.2 Federated Learning

Federated learning (FL), a novel distributed machine learning approach, enables a global model training by sharing only the results of local training without collecting users' private data. Typically, it is a centralized architecture containing a parameter server as the central node to perform operations such as parameter aggregation. In a general scenario, a portion of clients are selected as participants in each training round without considering the communication cost, and partial customers participate in each round [32]. Due to its privacy-preserving properties, FL has been studied and used in many fields. For example, in the field of finance, FL is applied to train a shared fraud detection model with class-imbalanced private data from different financial institutions [33]. FL is considered as a natural fit with the open banking data market, and the challenges and corresponding solutions are explored in depth in [34]. Also in the medical field, it is attracting more and more attention for some healthcare applications [35–37]. FL-based healthcare is popular to protect the local data of patients while fully exploiting the value of the data to provide better healthcare services to them. As

FL attracts many health data clients for computation and model aggregation, its training quality, such as accuracy, will be significantly improved, which may be difficult to achieve by centralized machine learning approaches with fewer data [38].

3.2.3 Personalized Federated Learning

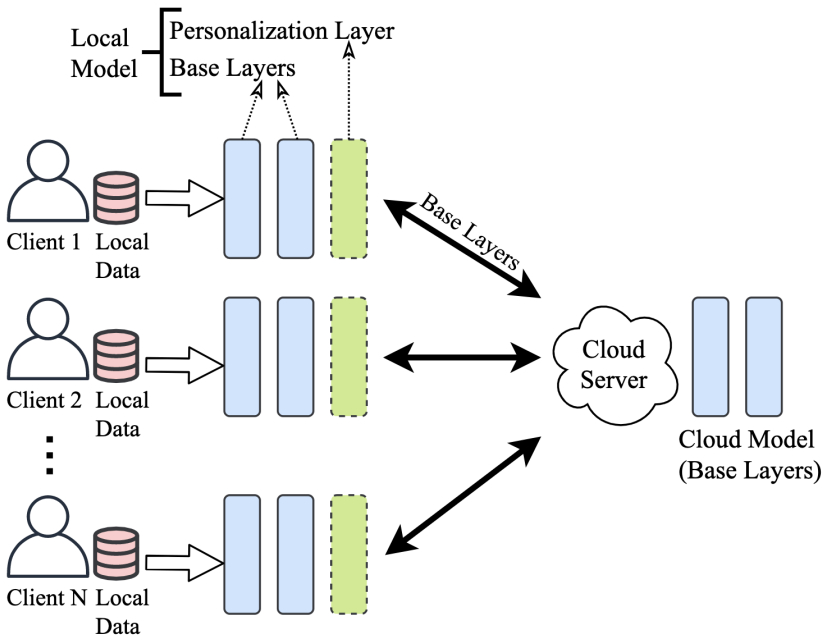


Figure 3.1: FedPer

Data heterogeneity is one of the main challenges in the field of FL, which is the non-IID training problem we mentioned earlier. It can make the global model perform poorly on a single client, and even affect the joining of new clients in FL [39]. It can be broadly divided into two scenarios, i.e., weakening the impact of non-IID to achieve better global uniformity, and catering to personalization and training personalized models to provide better individual services. For the purpose of this article, we will focus on the second scenario. Most personalization FL techniques involve two basic steps: first, clients collaborate to train a global model and then train their personalized models with local data individually [40]. Some existing solutions to this problem include federated transfer learning [41], Federated Meta-learning [42], Personalization Layers [43], etc.

Among them, the FedPer method proposed in [43] has attracted our attention for its easy-to-extend and easy-to-understand advantages. It is essentially based on the idea

of transfer learning, which can be used as a common method to realize personalized FL. It divides the model into base layers and personalization layers, where the base layers are collaboratively trained through federated learning, and the personalization layers only participate in local training rather than global aggregation, so as to realize personalized needs and reduce communication overhead. Inspired by this work, we make some improvements in terms of security and propose a personalized FL system based on blockchain.

3.2.4 Blockchain based Federated Learning

As a promising technology, blockchain has the advantages of decentralization, reducing network congestion, ensuring the security of data transmission and sharing, and so on [44]. By virtue of its distributed nature, is seen as a secure and decentralized potential footstone on the Internet of Things [45]. Since a centralized server always exists for model aggregation and issues in the FL training process, the security and privacy issues cannot be well avoided. Especially, the single point of failure will lead to the system collapse, so the emerging blockchain technique is introduced to decentralize the server. Lu et al. [46] firstly designed a privacy-preserving data sharing scheme that combines FL with blockchain. Without the centralized server, the potential attacks can be alleviated and the utilization of computing resources can be improved by client coordination. Furthermore, given the malicious clients or central servers which may reveal user privacy in the global model, Li et al. [47] devised an innovative committee consensus mechanism for the blockchain, where the dishonest nodes are removed from the training process. Kang et al. [48] proposed a reputation-based worker selection mechanism for the FL. The reputation of workers is calculated according to a multi-weight subjective logic model and consortium blockchain is imported to manage ultimate reputation in order to prevent unknown attackers. To achieve a considerable balance between efficiency and security, Qu et al. [23] proposed a novel FL called FLblock, which allows the device to exchange local updates in the vicinity of end devices in the blockchain. Although there are many blockchain-based FL schemes available,

it can be challenging to find a privacy-preserving framework that is specifically designed for the IoMT scenario. Furthermore, most existing approaches require clients to upload complete model parameters, which can consume a significant amount of communication and storage resources. However, in this article, we propose a more efficient approach that only uploads and aggregates the base layers, improving both efficiency and model performance under personalized needs.

3.3 System Design

In this section, we introduce the system design and comprehensive steps for our proposed framework. We summarize the symbols used below in the table 3.1.

Table 3.1: Variables and Symbols

L_b	Number of base layers.
L_p	Number of personalization layers.
L	The total number of model layers.
$\boldsymbol{w}_{i,b}^t$	Weight tensors of base layers of client i in the t-th round.
$\boldsymbol{w}_{i,p}^t$	Weight tensors of personalization layers of client i in the t-th round.
L	The total number of model layers.
$\boldsymbol{w}_i(t)$	Local model of client i at epoch t.
$\boldsymbol{w}_i'(t)$	The selected part of model of client i at epoch t.
η	Learning rate.
T	Number of training epochs.
∇F_i	The gradient of the loss function with respect to the model parameters of client i.

In the system design of this paper, we consider that there are N patients involved in the FL process. The patients have corresponding IoMT devices to monitor them and collect data. We consider that all these devices have computational and communication capabilities, i.e., they can support machine learning tasks. The system workflow can be divided into three steps, as shown in Figure 5.2, namely Task Assignment, Local Training (i.e., Base + Personalization Layers), and Blockchain-based Global Aggregation (i.e., Base Layers), where the first step is executed once before training initialization,

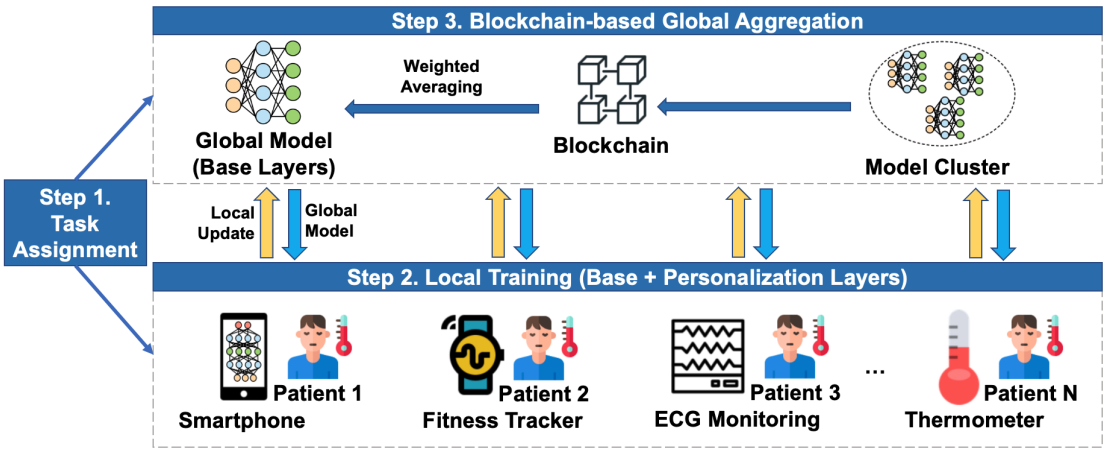


Figure 3.2: System design

the second and third steps loop until the specified number of epochs is reached or until the global model achieves the target accuracy.

3.3.1 Task Assignment

The first step to be taken is the training tasks assignment. In FL, individual clients coordinate to train a global model. Then before starting, the initialization model for training, such as a convolutional neural network (CNN) model, is first specified and the devices involved in the training are identified. Since we use a model partitioning training method to achieve personalization requirements, we also need to specify the model partitioning method before training. For N patients, we consider model partitioning to be uniform for each individual. Denote the number of base and personalization layers by L_b and L_p. In this case, both the base and personalization layers are trained in the second part of the system workflow, but only the base layers are involved in the global aggregation, i.e., the third step. Here we adopt a more general setting compared to [43], i.e., L_b and L_p take non-negative values and both satisfy that:

$$L_b + L_p = L, \tag{3.1}$$

L denotes the total number of model layers. When $L_b = 0$, all layers belong to the personalization layers, which are only involved in local training. So there is no global

Algorithm 1 Local Training

Initialize $\boldsymbol{w}_{i,b}^0$,$\boldsymbol{w}_{i,p}^0$ with $\boldsymbol{w}_b^0$ and $\boldsymbol{w}_p^0$;
Initialize parameter η, M, T;
for epoch $t = 1, 2, ..., T$ **do**
    Select M clients randomly;
    for each client $i = 1, 2, ..., M$ **do**
        $(\boldsymbol{w}_{i,b}^t, \boldsymbol{w}_{i,p}^t) = (\boldsymbol{w}_{i,b}^{t-1}, \boldsymbol{w}_{i,p}^{t-1}) - \eta \nabla F_i((\boldsymbol{w}_{i,b}^{t-1}, \boldsymbol{w}_{i,p}^{t-1}))$;
        Upload $\boldsymbol{w}_{i,b}^t$ to blockchain for aggregation;
    end for
end for

aggregation phase at this time, which is equivalent to conventional machine learning on a single client. Correspondingly, when $L_p = 0$, there are no personalization layers equivalent to conventional federated learning.

In addition, since a blockchain-based global update is used in our system, the corresponding consensus algorithm also needs to be specified. The consensus protocol of our consortium blockchain is Proof of Stake (PoS) combined with the reputation-based mechanism. The miners compete for the leader who is responsible for the model aggregation and the next task assignment.

3.3.2 Local Training (Base + Personalization Layers)

After specifying the model partitioning method, each IoMT device is first trained locally. Note that for ease of representation, in the following, client i also refers to the IoMT device of patient i. We use the $\boldsymbol{w}_{i,b}^t$ to denote the weight tensors of the base layers of client i in the t-th round. Note that the base layers consist of layers with different dimensions. Similarly, we denote the personalization layer of client i in round t as $\boldsymbol{w}_{i,p}^t$. The complete model of client i in round t is denoted by $(\boldsymbol{w}_{i,b}^t, \boldsymbol{w}_{i,p}^t)$.

Then we give the corresponding algorithm description. First, in the initialization phase, the training parameters and local models are determined and initialized for all clients. At the same time, the division method of the model is specified and base layers and personalization layers are clarified. During the training phase, each client uses local private data to update model parameters via gradient descent. The point is that the client only uploads base layers of the model to blockchain for the secure global aggregation,

Algorithm 2 Blockchain-based Global Aggregation

Input: $\boldsymbol{w}_{i,b}^t$
Output: $\boldsymbol{w}_g^t$
for epoch $t = 1, 2, ..., T$ **do**
    Select the miner with $Max(coinage)$;
    Receive $\boldsymbol{w}_{i,b}^t$ from M clients;
    $\boldsymbol{w}_g^t = \sum_{i=1}^{M}(n_i / \sum_{i=1}^{M} n_i)\boldsymbol{w}_{i,b}^t$
    Publish global update w_g^t;
end for

while the personalization layers are maintained locally.

3.3.3 Blockchain-based Global Aggregation

Firstly, users upload local training models to the miners. Then miners check the validity of signatures attached to the uploaded models. If the signature is legitimate, the updates are put into the transaction. The miners also need to compete for the winner which is determined by the token-owning amount, where the selection process can be presented as $Max(coinage)$. Note that $coinage$ equals the total number of owning coins plus accounting days. The selected winner should aggregate all local models to form a global model that is considered the task for the next turn. After each epoch, the submitted models from miners are evaluated to generate a reputation credit, which becomes a reference for token distribution. The evaluation criteria is illustrated as $r = \frac{\frac{1}{n}\sum_{1}^{i=n} w_{i,b}^t}{w_g^t}$, where r refers to the calculated reputation. The detailed procedures are illustrated in Algorithm. 10

3.4 Security Analysis

In this section, we discuss the potential attacks that our framework can defend.

1. **Single-point-of-failure attack:** In traditional FL, a centralized server always exists. Once this server is invaded, the whole system will be under control by the attacker. In our proposed blockchain-based FL framework, the single server has been replaced by a series of lightweight nodes which defend against the possible

occurrence of single-point-of-failure attacks. In addition, the widely used proof-of-work (PoW) consensus in the blockchain cannot defend the 51% attack, which means the adversary takes control of the majority of computation power in the blockchain system. Given the 51% attack brought by PoW, we utilize the PoS consensus adopted by Ethereum. The block generation authority is determined by coinage rather than the computation power. Our designed blockchain-based FL framework prevents single-point-of-failure attacks in a further step.

2. **Poisoning attack:** Poisoning attack occurs when multiple parties train a model and attackers amid them maliciously modify their own samples involved in the uploaded model in order to interfere with the global model of the FL. In our proposed blockchain-based FL framework, the introduced reputation-based miner selection mechanism can eliminate this potential risk. If the accuracy of the submitted model deviates from the final global model excessively, the corresponding miners only get a few token rewards from the system, which decreases the success rate in the next round. This incentive mechanism mitigates poisoning attack to some degree.

3. **Data leakage attack:** The previous research has reported that the adversary can infer the privacy data from the issued global model. However, in our framework, we utilize the consortium blockchain to block all unauthorized users in advance. If the signature of the uploaded model is against the records in the database, the contribution of users will not be counted. Therefore, the data leakage issue in our model can be alleviated.

3.5 Simulation Experiments

In this section, we perform detailed simulation experiments to validate our system. In the following, we give the specific experimental setup and comprehensively analyze the results.

3.5.1 Setting

The simulation experiments are executed on Ubuntu 18.04 with an Nvidia GTX 3070 GPU. In our experiments, we use stochastic gradient descent to perform parameter updates for our clients. In our setup, 30 clients are involved in the training. In the following experiments, we test the performance of the system under different datasets, different data heterogeneity, and the different number of personalization layers scenarios, respectively. For the experiments, we derive the results after 100 rounds of training, where one round means one global update. The parameters of the simulation experiments are given in Table 5.2.

Table 3.2: Simulation Parameters

Parameters	Values
System	Ubuntu 18.04
GPU	Nvidia GTX 3070 GPU
Number of Clients	30
Number of Training Rounds	100
Non-IID Level (α)	0.2, 0.5, 0.8
Number of Personalization Layers (ϵ)	0, 1, 2, 3
Training Model	ResNet-34

Dataset

- **Fashion-MNIST.** Fashion-MNIST is proposed as a replacement for the widely known MNIST handwritten digital image dataset because the latter is too easy to train, and traditional machine learning algorithms are easily able to achieve 97% test accuracy [49]. Like MNIST, it includes 60,000 images, 50,000 for the training set, and the remaining 10,000 for the test set. Each image is a 28x28 grayscale image.

- **CIFAR-10.** The CIFAR-10 dataset contains a total of 60,000 32x32 color images. It includes 10 categories, where each category has 6000 samples. It is generally divided into a training set (i.e., 50,000 images) and a test set (i.e., 10,000 images) [50].

Model

In this paper, we choose to use the ResNet-34 model, which is the popular and advanced convolutional neural network model for image classification. It is different from traditional neural networks in the sense that it takes residuals from each layer and uses them in the subsequent connected layers. The model network contains a total of 5 convolution groups, and each convolution group contains one or more basic convolution calculation processes. Each convolution group contains 1 downsampling operation to reduce the size of the feature map by half. Downsampling is achieved in the following two ways: (1)Maximum pooling: the step size is 2, only used for the second convolution group (Conv2_x); (2)Convolution: the step size is 2, used for 4 convolution groups except for the second convolution group. In addition, ResNet networks are built from basic blocks. Basic block as shown in 3.3 is used to build ResNet-34 and ResNet-18. For networks such as ResNet-50, a different "bottleneck" building block is used to build them [51]. But in any case, they can be considered to be stacked with the basic blocks. This also gives us a good basis for our model partitioning. Hence, we consider the basic blocks as in figure 3.3 as the unit for dividing the basic and personalization layers of the model. Figure 3.4 also shows the code details of the blocks used to build the model.

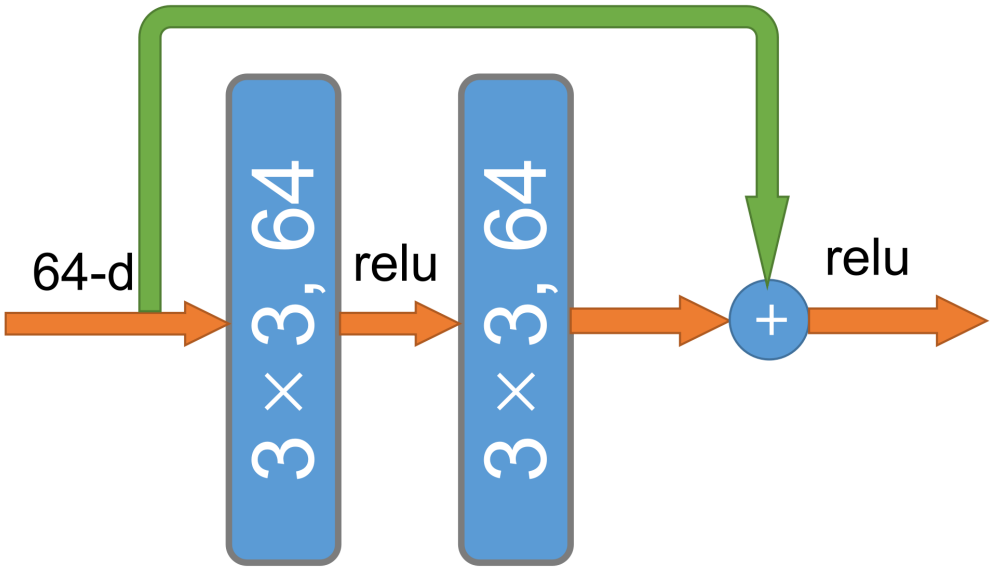


Figure 3.3: Building block of ResNet-34.

```
(2): BasicBlock(
  (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu): ReLU(inplace=True)
  (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
```

Figure 3.4: Code of building block.

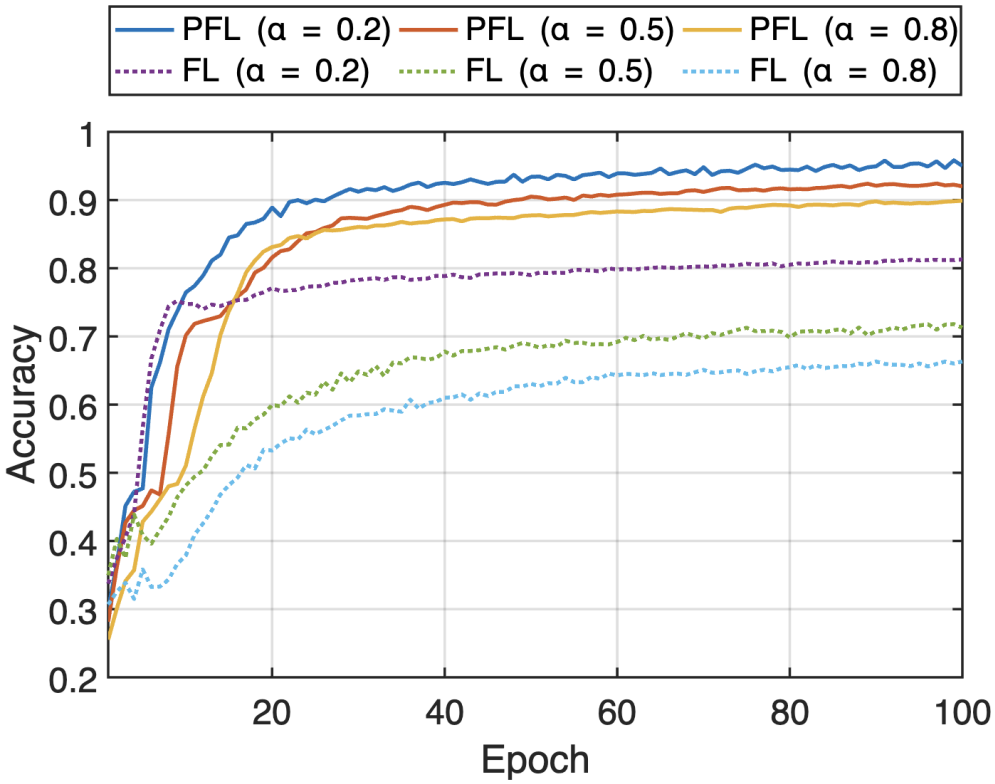


Figure 3.5: Accuracy versus α on Fashion-MNIST.

3.5.2 Training Versus Different Non-IID Level

To describe data heterogeneity, we introduce the concept of Non-IID level and denote it by α. When $\alpha = 0.4$, it means that 40 percent of the client's local data belong to the same class. When $\alpha = 0$, it means that the data are independently and identically distributed. We test on Fashion-MNIST and CIFAR-10 datasets with 30 clients. Note that epoch refers to the number of global model updates. In our experiments, we set the training to terminate when it reaches 100 epochs, which means that the global model (i.e., Base Layers) is updated 100 times. Note that since each client only uploads the base layers of its local model to the blockchain, when we call the global model we refer to the weighted-averaged result of the base layers uploaded.

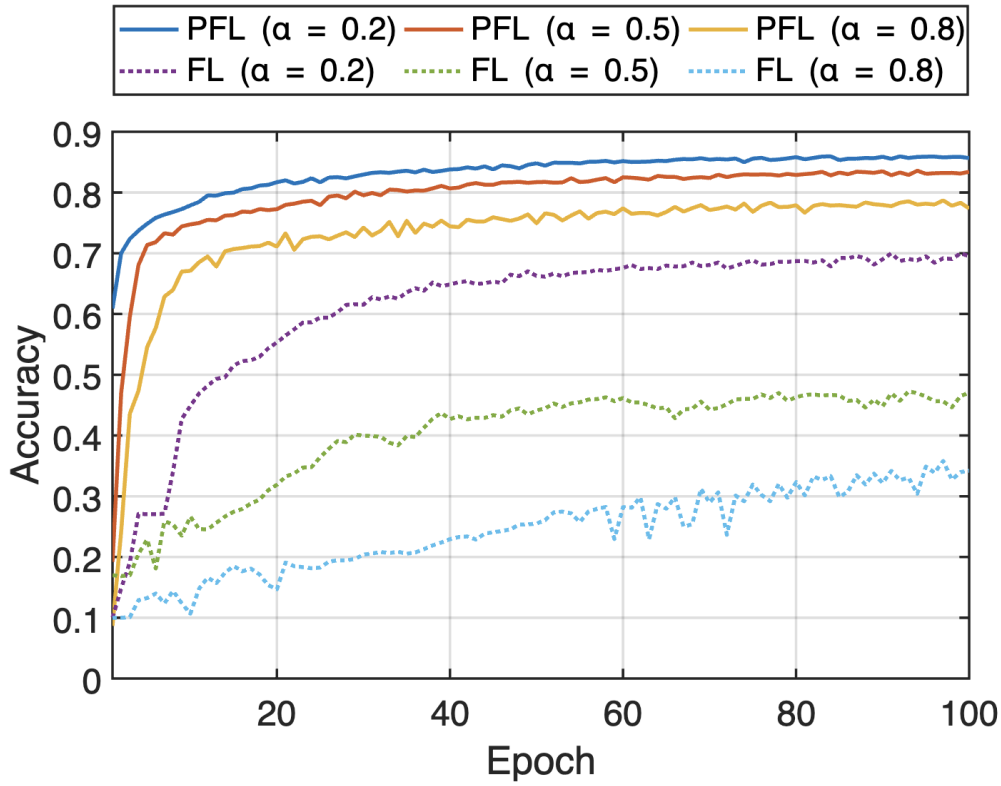


Figure 3.6: Accuracy versus α on CIFAR-10.

For ease of presentation, we denote personalized FL as PFL. Moreover, we will use FL with FedAvg algorithm as the benchmark for comparison. In this subsection, we explore the effect of data heterogeneity on training performance, and the personalization setting is the same that the last block and the final fully connected layer are considered as the personalization layers and will not take part in the global aggregation.

For traditional FL-related studies, accuracy refers to the performance of the global model. However, in our study, we explore FL under personalization requirements. Hence, to measure the effectiveness of our system, we refer $Accuracy$ to the averaged test accuracy on local models of each client. That is:

$$Accuracy = \frac{\sum_{i=1}^{M} Accuracy(i)}{M}, \tag{3.2}$$

where M refers to the total number of clients participating in the training, and $Accuracy(i)$ denotes the test accuracy on client i's local model.

The effect of data heterogeneity on accuracy is shown in Figure 3.5 and 3.6. We

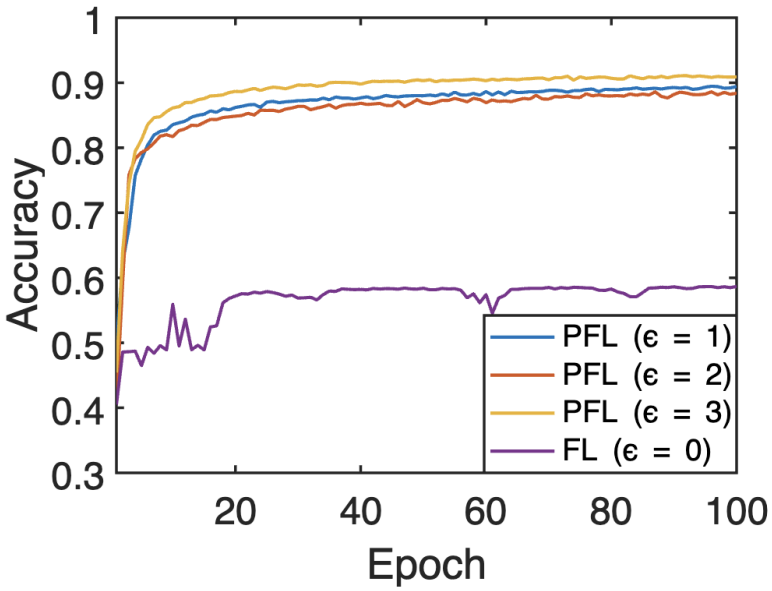


Figure 3.7: Accuracy versus ϵ on Fashion-MNIST.

conduct experiments on three different Non-IID scenarios, namely $\alpha = \{0.2, 0.5, 0.8\}$. It is clearly seen that PFL is better than FL in all settings, and we analyze that this is mainly because the traditional FL aims to train a uniform model by weighted averaging, which cannot capture the personalization features of clients well. In contrast, PFL is able to learn more personalization features of clients by dividing the model into base and personalization layers. The personalization layers will be maintained and updated locally with clients' own data, so it achieves better results. Specifically, on Fashion-MNIST dataset, the accuracy of PFL compared to FL improves by about 0.15, 0.2, and 0.23 in the three cases, respectively. The difference is even more pronounced on CIFAR-10, which improves by 0.16, 0.36, and 0.43, respectively. It can be seen that the base and personalization layers-based FL can dramatically improve the training effect in non-IID scenarios. Moreover, the performance degradation of PFL is quite insignificant compared to FL as α increases. As Figure 3.5 illustrates, for the conventional FL, the difference in accuracy between $\alpha = 0.2$ and $\alpha = 0.8$ is about 0.15 while it is about 0.06 for PFL. This is because PFL learns to personalize the model, and personalization is its goal. FL, on the other hand, seeks a homogenized model, which obviously cannot cope well with data heterogeneity, and thus the performance degradation is particularly obvious as α increases.

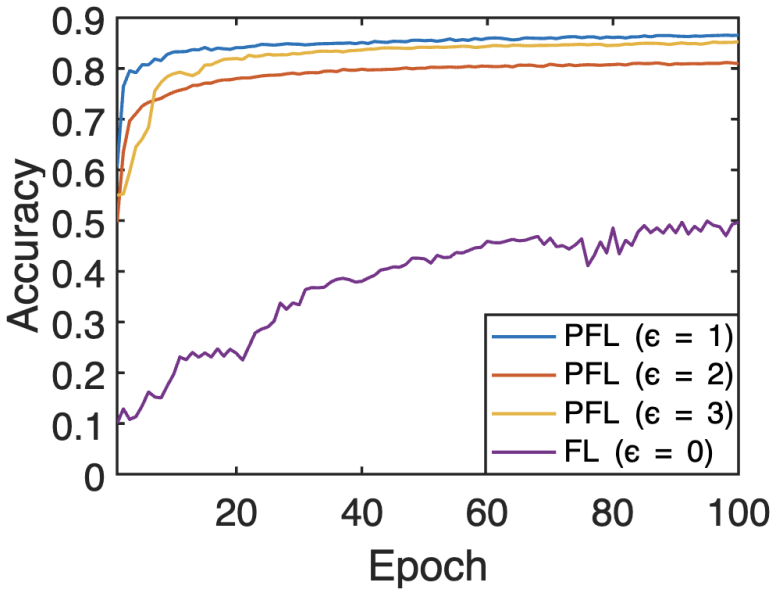


Figure 3.8: Accuracy versus ϵ on CIFAR-10.

3.5.3 Training Under Different Model Partition

As we mentioned above, ResNet-34 is superimposed by basic blocks, and here we divide the model in terms of each block. We introduce the parameter ϵ to indicate the number of personalization layers and thus describe the division of the model. For example, when $\epsilon = 4$, it means that the last 3 basic blocks and the final fully connected layer of the model belong to the personalization layers.

To explore the effect of model division on the results, we unify the non-IID level to $\alpha = 0.6$. For the settings of personalization and base layers, we give four different settings, i.e., $\epsilon = \{0, 1, 2, 3\}$. Note that with $\epsilon = 0$, there are no personalization layers at this point, so we denote it as FL. The results are shown in Figure 3.7 and 3.8.

From these figures, it is easy to see that the performance of personalized federated learning is very superior compared to FL. Regardless of the number of personalization layers, the model's performance of PFL is much better than that of FL. In other words, the model performs much better when ϵ does not take 0 than when it equals to 0. And when $\epsilon \neq 0$, the performance varies on different datasets. On Fashion-MNIST, the highest accuracy is achieved when $\epsilon = 3$, followed by taking the values 1 and 2. While on CIFAR-10, the accuracy at $\epsilon = 1$ is higher than when it is equal to 3 and 2. Therefore, we can conclude that personalization layers are necessary and can have

a huge improvement in the training effect. However, the performance improvement is most obvious from no personalization layers to personalization layers, and it is difficult to specify how to divide the personalization layer to achieve the best results. Just using the last fully connected layer as a personalization layer can still bring huge performance gains.

3.6 Conclusion and Future Work

In this paper, we propose a blockchain-based personalized FL approach for IoMT scenarios. We implement the need for patient personalization services in the IoMT context by dividing the model into personalization layers and base layers, and stipulating that the personalization layers do not participate in global updates. And the patient's privacy is protected by applying the FL approach. We also incorporate blockchain technology for the global model updating to enhance privacy protection. Finally, we conduct simulation experiments to explore the impact of data heterogeneity and different personalization layers partitioning methods on performance, and the results show that our approach has superior improvement over traditional federated learning with FedAvg.

In future work, we will mainly focus on improving two aspects. The first is to apply technologies such as differential privacy to prevent the model from being collected and analyzed by honest but curious participants [14], to resist inference attacks [52], model inversion attacks [53], and so on. On the other hand, we will work on applying redactable blockchain in the field of FL.

Chapter 4

DEEP-FEL: Decentralized, Efficient and Privacy-Enhanced Federated Edge Learning for Healthcare Cyber-Physical Systems

4.1 Introduction

Through physical and network world integration, the artificial intelligence (AI)-driven healthcare cyber physical systems (CPSs) assure the smooth execution of the medical process by using corresponding equipment to monitor and collect data from patients [54]. As the digital society of the future evolves, data is already seen to exist as a valid virtual asset that can be shared and used by people [55]. Simultaneously, the rapid development of learning and hardware technologies [56] drive a significant revolution in the domain of healthcare data analytic in recent years [57]. For example, as the increased interest of wearable devices and other Internet of Medical Things (IoMT) for healthcare services, particularly in health, well-being, disease prevention, and fitness, as well as the paradigm shift toward healthcare that is personalized and controlled by individuals [58].

While the conventional deep learning-based applications used in medical institutions always need collect raw data from different hospitals or individuals to a centre server before training, the personal sensitive information leakage and high transmission overhead cannot be avoided. Furthermore, the healthcare data often stem from relatively clinical institutions, and then might encounter unquantifiable bias due to the heterogeneity [59]. Medical institutions are unwilling to share their personalized, highly sensitive information, and even model information which could be inferred from federated training outputs. Therefore, it is of importance to apply privacy enhancement technology to strengthening the privacy protection between different institutions.

Scalable and viable solutions are necessary to protect digital assets such as proprietary data in such an expanding digital environment [60]. In order to take full advantage of this sensitive data to provide better healthcare, putting in place the demanded security practices is quite crucial [61]. Therefore, Federated Learning (FL) as a privacy-enhancing distributed machine learning technique can be used to address the above challenges. Considering data sensitivity and fragmentation, FL enables clients to co-train a shared global model locally without transferring raw data [62, 63].

To achieve high performance and efficiency of FL, Liu *et al.* [64] reduced the rounds of communication process among clients who conduct multiple local updates before aggregation by Federated Stochastic Block Coordinated Descent. Then, it is theoretically analyzed concerning the impact of the number of local updates. While these existing efficient FL frameworks are parameter server-based that is susceptible to the white-box attack (e.g., membership inference attack), which may result in the client-level privacy leakage [65]. Furthermore, since these frameworks are concentrated on data governance problem alleviation, the potential single-point failure and high communication overhead issues cannot be well addressed.

To address the above challenges, we aim to develop a system that owns the following features:

- **Decentralized:** We would like to use a decentralized structure to avoid the single points of failure. At the same time, the role of each participant is fair, especially

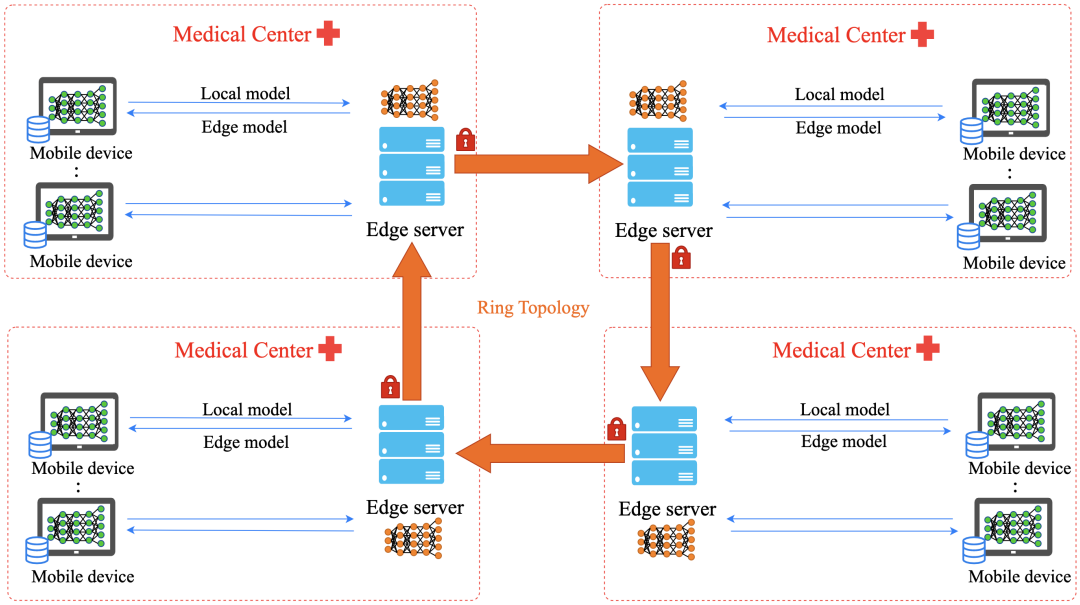


Figure 4.1: DEEP-FEL distributed training with four-party collaboration.

In Figure 4.1 we show an example including four medical centers. The edge server is responsible for parameter transmission (disturbed model data), and the mobile devices correspond to the IoMT devices in these institutions, which can collect and store patient data and perform certain computations.

for cooperation between privacy-sensitive medical institutions.

- **Efficient:** Since the communication capabilities of mobile devices are uneven, fast communication can be carried out in a local area network or small area. Therefore, in order to improve communication efficiency, direct communication between mobile devices across medical centres should be forbidden, and the edge servers with strong communication abilities should be used to aggregate information in medical institutions before mutual communication with servers in other medical institutions.

- **Privacy-enhanced:** Medical privacy data should not be shared and should be fully utilized. So federated learning is a viable solution. At the same time, federated learning itself also faces some security threats, so further privacy enhancement is also a design direction we need to consider.

Hence, to achieve the above-mentioned goals, we aim to develop a privacy-enhanced efficient decentralized mobile healthcare system with differential privacy. Figure 4.1

gives a general example of the four-party collaboration. Since each edge server and its subordinate mobile devices belong to the same medical institution, four medical institutions form a ring topology. Note that each institution can be regarded as a node, and two adjacent nodes on the ring can communicate with each other. First, the mobile devices of each medical institution conduct local training based on the collected patient information. After the training task completion, the model weights will be transmitted to the server in the medical institution for preliminary aggregation. Then the parameter aggregation between medical institutions is carried out based on the ring topology through the interaction between the servers. Note that the server should firstly add artificial noise to the model parameters for privacy protection enhancement and then communicate with the adjacent medical institutions on the topological structure. Finally, a round of federated learning is completed. The same training process will continue for several rounds until the ideal set accuracy or training duration is reached. More details will be illustrated in Section 4.3.

The main contributions of this paper are listed as follows:

(1) In our system, we first adopt a hierarchical architecture based on ring topology. This architecture utilizes communication and storage capabilities of edge servers to aggregate model parameters and communicate with other institutions. Moreover, the mobile devices can quickly communicate with the edge server inside the same institution, further improving the overall efficiency of the system.

(2) We consider the communication efficiency between servers within different medical institutions, and present the detailed processes to construct the ring topology as an optimization problem, which is then solved efficient heuristics solution to improve the communication bottleneck.

(3) Privacy-enhancing and efficient global parameter aggregation algorithms are designed for our system. Compared with the conventional methods, it reduces communication cost to improves the system efficiency. Besides, it also enhances privacy protection by adding artificial noise.

(4) We conduct thorough and detailed experiments on three medical datasets. The experiment performance proves the superiority of our system.

4.2 Preliminaries and related work

4.2.1 Centralized Federated Learning

In general, centralized federated learning requires participants to collaboratively achieve a joint ML model under the orchestration of a center parameter server. In this scenario, the parameter server can easily aggregate the parameters and guarantee convergence with synchronous or asynchronous parameter aggregation protocols (e.g., ADMM, SSGD, AdaDelay). For example, in [66], the authors exploit to make the clients perform online learning with continuous streaming local data, and the parameter server aggregates the learning parameters in the proposed federated learning framework which asynchronously updates the model.

However, the FL with a centralized parameter server could suffer from some security and stability concerns. Since the parameter server might be an adversary who can receive updates from each participant over time, and then analyze the private information of each participant [67,68]. In addition, it also faces some issues about single-point failure and high communication overhead and limited network bandwidth [69].

4.2.2 Decentralized Federated Learning

In recent years, a series of work about decentralized FL have been done to improve the problems caused by centralized structure. In [70], the authors considered a peer-to-peer FL structure in which the participants iterate and aggregate the beliefs of their one-hop neighbors to generate a global model. Similarly, peer-to-peer FL towards medical applications has been proposed to address the problem that all clients need to agree on one trusted central party whose failure would disrupt the training process [71]. Furthermore, to sidestep the limitation of connectivity between IoT devices, the authors [72] proposed a peer-to-peer learning structure as a solution, which does not require a central

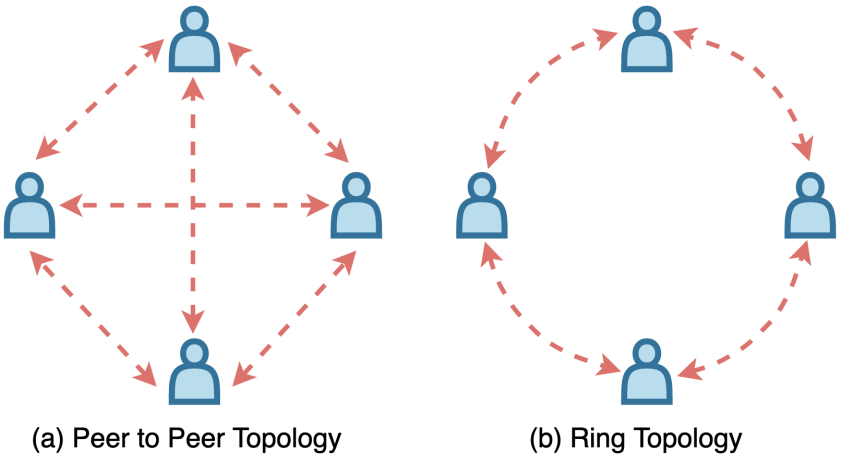


Figure 4.2: A comparison of decentralized federated learning topologies with four clients.

node to orchestrate the model training.

As usual, decentralized architecture can be divided into two types (i.e., peer to peer and ring topology), which are shown in Figure 4.2. The peer-to-peer architecture is characterized by high flexibility, and everyone can freely communicate mutually. Although some current works have utilized this topology to construct a decentralized FL framework [71,73,74], some obvious shortcomings still exist. For example, in the context of mobile devices, when users send excessive messages to the same user, communication bottlenecks may be aggravated. In addition, mobile devices are mostly supported by batteries and the end-to-end structure requires frequent communication, so it may be difficult for mobile devices to participate in training stably and continuously.

On the other hand, the ring architecture is also used in decentralized federated learning [67], which can reduce communication overhead and improve system bandwidth and stability compared with traditional decentralized algorithms. In the ring topology, only two adjacent nodes need to communicate, so we can further improve the efficiency of the system by optimizing the ring construction. In this paper, we consider each decentralized endpoint to be a server in a medical institution. Therefore, compared to mobile devices, our endpoints have higher stability and stronger communication capabilities, which also theoretically reduces topology reconfiguration and inefficiencies due to ring topology instability. In addition, through the parameter update algorithm

we designed, the characteristics of the ring topology are cleverly combined, and the transmission of the communication data volume is greatly reduced.

4.2.3 Hierarchical Federated Learning

The breakthrough of theory and techniques in edge computing dramatically strengthens the computing capacity of the cloud and improves the quality of mobile services. The authors introduced an edge cloud architecture to overcome the latency caused by centralized or remote cloud-based IoT data processing [75]. Similarly, the IHSF [76] approach improved hybrid fog computing IoT systems based on software-defined networks to enhance the performance of communication based on edge computing models. The authors [77–79] exploited to combine edge computing with federated learning in some fields (e.g., Internet of Things, social network), which results in high performance on convergence and efficient training. In the hierarchical FL scenario, the authors migrated the learning model parameters from clients to multiple edge nodes, without transferring them to a centralized parameter server. However, it is tough to leverage the powerful computation of cloud-sole with edge computing. Hence, in [80], the authors exploited to integrate the sufficient storage of cloud server and efficient communication of edge server into a three-tier client-edge-cloud federated learning framework to balance the trade-off between communication and computation, by which multiple edge servers can execute partial model aggregation tasks.

4.2.4 Differential Privacy in Federated Learning

Differential privacy (DP), as a strictly theoretical privacy-preserving technique, can maximize the data utility while guaranteeing an effective privacy level [81]. In [82], the authors pointed out that in federated learning, gradient updates transmitted between servers and participants had been shown to be potentially recoverable by attackers. And the features of DP make it effective in federated learning against threats such as the above, thus enhancing privacy protection. In [83], the authors proposed a differential privacy mechanism-based federated learning framework that not only makes the clients

collaboratively learn a shared model without raw data transferring during the training process but also protect the parameters from potential privacy attacks. In [84], the authors exploited to address the problem that local dataset might be leaked by analyzing the shared model and thus proposed a differential privacy-based algorithm for clients in federated optimization to conceal clients' information, bridging the gap between privacy-preserving and model accuracy. Further, unlike DP, the normal local differential privacy is developed to protect the data privacy during the collection. Hence, the authors [85] investigated to provide the local differential privacy for the learning parameters in the FL of large-scale DNNs through multiple individual clients' local datasets for privacy-preserving. Although in fields such as healthcare, there are some works combined with federated learning, which have been considered to protect privacy, but due to the security threats mentioned above, it is necessary to further utilize differential privacy to enhance privacy protection.

4.2.5 Healthcare Cyber Physical Systems

Cyber-physical systems (CPS), which closely interweave software and physical components, can be applied to numerous fields including healthcare, environmental protection [86]. In healthcare applications, one critical goal of the CPS model is to ensure data security [87]. The development of computing abilities for mobile devices makes this process more feasible and innovative in monitoring and delivery of healthcare information, namely, mobile healthcare [88]. For example, the authors proposed a voice pathology detection system on the mobile healthcare framework with deep learning technologies, in which voices are captured using smart mobile devices [89]. Meanwhile, in [90], the authors proposed an intelligent m-healthcare system based on IoT technology for offline human activity classification and robust and precise human activity recognition by using data mining techniques. However, healthcare data are usually fragmented and private sensitive, which makes it difficult to share across hospitals. Hence, federated learning as a collaborative learning method needs the clients to train a shared global model under a parameter server with all the sensitive data in local insti-

tutions for privacy preservation.

4.3 DEEP-FEL System

We consider that there are an edge server and several medical devices within each medical institutions, whose total number is n. Similar assumptions have been mentioned in many articles, mainly considering the computing and communication capabilities of edge servers [91–93]. Since the main contribution of this paper is the algorithm and architecture level, the specific settings for edge servers will not be expanded here. Different medical institutions form a decentralized ring topology for global model aggregation. In order to avoid confusion, we point out that the edge server is equivalent to the parameter aggregator in each institution, and it can participate in multi-party collaboration on behalf of the medical institution. Therefore, in the following, the nodes in the ring topology also mean edge server of each institution, which transmit end data to logically next server and receive it from the previous one. The devices in each institution perform local training based on the collected data from patients. The edge server in each medical institution acts as a parameter server for medical device training result collection and edge model generation.

The overall workflow is briefly given as follows:

- **Step 1:** The devices in each medical institution use the collected patient data for local training.
- **Step 2:** After the local training, the medical devices upload the training results to the edge server in the corresponding institution.
- **Step 3:** Each edge server first aggregates the received information and generates an edge model which corresponds to the representing medical institution.
- **Step 4:** Edge servers add artificial noise to the model weights for privacy enhancement.

- **Step 5:** Edge servers perform the global aggregation process based on the formed ring topology to generate a new global model. And then, they will distribute the new model to medical devices to update their local models. If the model has reached the expected accuracy or the specified number of training epochs has been performed, the whole process ends, otherwise, skip to step 1.

The following content is divided into four parts (i.e., ring topology, local training, privacy enhancement, and ring-based aggregation). We illustrate the four parts and introduce the corresponding algorithms in detail.

4.3.1 Ring Topology

As Figure 4.1 illustrates, each medical institution acts as the node on a ring topology. Since the edge server in the medical institution participates in the global parameter update, we also regard the edge server as a node on the ring topology. Although the ring topology in FL has been proposed in [67], our method has more advantages compared with their work:

1. Wang et al. [67] directly use mobile devices as nodes in the ring topology. In actual federated learning, the number of mobile devices may be hundreds or thousands. For such a large-scale deployment, it will be tough to communicate under the ring topology due to the heterogeneity in hardware, network condition, and so on. In our scheme, we use the hierarchical structure to improve system efficiency, which allows edge servers with stronger computing and communication capabilities to act as nodes in the ring topology.

2. Wang et al. [67] form a ring is to use the node identity to determine the location of each node through a hash function. Although the theory is feasible, it would be very unreasonable considering the heterogeneity of the equipment of federated learning and the difference in communication capabilities. We consider the differences in communication between each node and propose a heuristic method to organize them into a ring more reasonably.

3. During update process, each node will transmit the complete model to other nodes. The total amount of data transmitted is $N(N-1)M$, where N represents the number of nodes and M represents the size of the model, and we reduce it to $\frac{2}{N}(N-1)M$ in our proposal.

Then, we will first illustrate the optimization problem in ring construction and propose the corresponding solution.

Formulation

We assume that there are a total of n servers that form a ring topology to confirm the node which communicates in the process of parameter aggregation. We assume that the time cost for server i to j to transmit data is c_{ij}. Thus, we have:

$$c_{ij} = \frac{s_{ij}}{b_{ij}}, i, j \in [1, n] \tag{4.1}$$

where s_{ij} indicates the size of data transferred from server i to j and b_{ij} illustrates the bandwidth rate from server i to j.

For better understanding, we illustrate the above-mentioned process in the form a graph. Consider a complete graph $G = (V, E)$, $V = \{v_1, v_2, ..., v_n\}$ represents the set of all nodes, $E = \{e_{ij} : i, j \in [1, n]\}$ is the edge between nodes set. Each edge e_{ij} owns its corresponding weight (i.e., c_{ij}), which represents the communication overhead from node i to j. Then, we define the problem is to find a Hamiltonian cycle, and minimize the maximum weight of the edge.

In the transmission process, the communication between each two nodes is carried out in parallel due to mutual independence. Therefore, the time cost of each round depends on the slowest pair of nodes (i.e., the edge with the largest weight in the above formulation). Note that each node has transmitted information to the next node and received the message of the previous node. Hence, our optimization goal is to achieve the largest weight as small as possible.

Actually, our formalized problem is a variant of the famous traveling salesman problem (TSP) called the asymmetric bottleneck traveling salesman problem (BTSP), which

aims to find a tour $a_1, a_2...a_n$ that has the minimum tour value. It can be defined as $max\{c_{a_i,a_{i+1}}, i \in [1, n-1]\}$ [94]. Asymmetric means that the cost between node a to b and b to a is different $c_{ij} \neq c_{ji}$, which is closer to the realistic situation of our problem. Our ring construction problem is abstracted to BTSP and NP-hard [95]. In the following part, we will introduce the heuristic Lin-Kernighan-Helsgaun (LKH) solution.

Heuristic Solution

It is well known that the BTSP problem could be handled by a TSP solver [95]. The Lin-Kernighan-Helsgaun (LKH) algorithm [96] is generally considered as an effective heuristic for solving TSP, and the LKH-based BTSP solver called BLKH has already been proposed in [97], which shows that the BTSP below a million vertices level can be concluded to the optimum in a reasonable time by using LKH as a black box. We briefly describe BLKH in Algorithm 3 based on [97].

Note that the $BBSSPA$, $BSCSSP$ and BAP functions in line 4 of Algorithm 3 are the methods to improve the lower bound by solving Bottleneck Biconnected Spanning Subgraph problem, Bottleneck Strongly Connected Spanning Subgraph problem, and Bottleneck Assignment Problem relatively, which can be found in [98]. And the $solveByLKH$ is the LKH solver for standard TSP instance.

4.3.2 Local Training

Procedure on Device

Prior to the local calculation process, the united initial model is issued to each mobile device within the medical center. Each device trains on the initial model based on the patient data collected by itself. The local training algorithm is shown in Algorithm 4. After the training, each mobile device will send the training results to the server of the medical center for edge parameter aggregation.

Upon receiving the latest edge model (after the global update), the mobile device will use the local data for the next round of calculations. This process is repeated until the set number of rounds or the accuracy value of the model has been reached.

Algorithm 3 BLKH

Input: Cost matrix c
Output: Bottleneck b and Tour t
Initialize LowerBound l by computing 2-Max bound
$l = Max\{l, BBSSPA(l), BSCSSP(l), BAP(l)\}$
$Low = l$
$b = High = solveByLKH(Low, MAX\_INT)$
while $Low < High$ and $b \neq l$ **do**
    $temp\_b, t = solveByLKH(Low, High)$
    if $temp\_b < b$ **then**
        $b = high = temp\_b$
        if $High \leq Low$ **then**
            $Low = l + \frac{High - l}{2}$
        end if
    else
        $Low = Low + \frac{High - Low + 1}{2}$
    end if
end while
return b, t

Procedure on Server

When the edge server collects the updates from the mobile devices in the institution, it will perform a weighted average to generate an edge model that represents the medical institution. The edge model will participate in the subsequent global update based on the ring topology. After the update is completed, the server will distribute the latest global model to each medical device for next round training.

4.3.3 Privacy Enhancement

Differential privacy (DP) is a common privacy-preserving technique in deep learning, as in [99], where a Gaussian noise satisfying differential privacy is added to the stochastic gradient descent process to protect privacy issues during training. The main idea of DP is to achieve, for two neighboring datasets, that the deletion or modification of a tuple did not affect the output of the query function.

Local differential privacy (LDP), as a variant of DP, considers that changes between any two tuples do not affect the output of the query function. In the LDP setting, users only send the perturbed report to the server rather than original data to protect it from inference attacks. The following is the definition of ϵ-LDP.

Algorithm 4 Local Training

Input: Edge model $\boldsymbol{\omega}_i$
Output: New edge model $\boldsymbol{\omega}_i$
for each medical institution $i = 1, ..., n$ in parallel **do**
    Randomly select m devices to train
    for each device $j = 1, ..., m$ in parallel **do**
        $\boldsymbol{\omega}_{ij} \leftarrow \boldsymbol{\omega}_i$
        $\boldsymbol{\omega}_{ij} \leftarrow \boldsymbol{\omega}_{ij} - \eta \nabla F_{ij}(\boldsymbol{\omega}_{ij})$
        Send local model $\boldsymbol{\omega}_{ij}$ to edge server i
    end for
    for each server $i = 1, ..., n$ in parallel **do**
        Receive all updates from devices in institution
        $\boldsymbol{\omega}_i \leftarrow \sum_{j=1}^{m} \frac{\boldsymbol{\omega}_{ij}|D_{ij}|}{\sum_{j=1}^{m}|D_{ij}|}$
    end for
end for
return $\boldsymbol{\omega}_i$

Algorithm 5 Data Perturbation

Input: $\boldsymbol{\omega}_i$, Laplace function $l(x)$
Output: Perturbed edge model $\widetilde{\boldsymbol{\omega}_i}$
for each institution $i = 1, ..., n$ in parallel **do**
    for each element $x \in \boldsymbol{\omega}_i$ **do**
        Replace x with $\widetilde{x} = x + Lap(\frac{\Delta s}{\epsilon})$
        Then, we get $\widetilde{\boldsymbol{\omega}_i} \leftarrow \boldsymbol{\omega}_i$
    end for
end for
return $\widetilde{\boldsymbol{\omega}_i}$

Definition 1 *(ϵ-LDP). Since $\epsilon > 0$, a random algorithm l satisfies ϵ-local differential privacy, if and only if any inputs x_1, x_2 in the finite field of possible values for user data x, for any output y, respectively. Hence, we have:*

$$\frac{Pr[l(x_1) = y]}{Pr[l(x_2) = y]} \leq e^{\epsilon}. \tag{4.2}$$

According to the equation (5.4), a lower privacy budget ϵ results in a higher plausibility of the distribution, which can obtain better privacy guarantees. In the distributed setting of federated learning, LDP is ideally suited for client-level privacy protection. To enhance the privacy protection of local updates within medical institutions, we adopted LDP to add artificial noise on the edge model before the global aggregation process.

As Algorithm 5 illustrates, the weights will be perturbed on edge servers through the Laplace mechanism before the global aggregation. In line 4 of Algorithm 5, x represents the value of the element in $\boldsymbol{\omega}_i$. In line 5, $Lap(\cdot)$ indicates the Laplace distribution,

$(\frac{\Delta s}{\epsilon})$ is the distribution scale parameter, and Δs is the local sensitivity that denotes the difference between the max and min of e. The goal of this mechanism is to adopt the Laplace distribution to achieve ϵ-LDP with the exact query result.

Each medical device uses its local data to train the local model and then uploads it to the edge server. The servers aggregate and average the uploaded weights from the devices, which preserve weights privacy with the data perturbation algorithm and then take part in the ring-based global aggregation to update the global model. We add artificial noise to prevent leakage of information about the edge model in each institution, so the honest but curious participant will hardly be able to infer the privacy characteristics of the medical institutions.

4.3.4 Ring-based Aggregation

The global update process is based on the ring topology determined in subsection 4.3.1 and we design our RingAVG algorithm which can greatly reduce the amount of data transmitted compared with traditional methods based on Ring-all-reduce, which was proposed by Baidu in 2017 [100]. Their research aims to reduce the communication overhead between different GPUs, allowing them to spend more time on model calculations. Although their design is under the traditional machine learning paradigm, the distributed thinking has inspired this research.

Then, we are going to explain the RingAVG algorithm in detail through both the figure and algorithm description. As Figure 4.3 illustrates, we give an example of four-party collaboration. Since it is a ring topology, in order to facilitate uniform expression with mathematical formulas, we make its subscripts start from zero. Similarly, part of the subscripts after the model is evenly divided start from zero.

Before starting, each server will calculate the weighted value of its edge model after perturbation to facilitate subsequent aggregation operations. For example, when N parties participate, a total of $2(N-1)$ rounds of transmission are required. In the figure, $N = 4$, so a total of 6 rounds of transmission are required. And it can be divided into two stages, scatter-reduce, and all-together stage. The distinction between these

two stages is whether all parties get part of the final model. Note that in the following we describe the example in the figure ($N = 4$). The server 0 we mentioned refers to the server with subscript 0 in the ring topology.

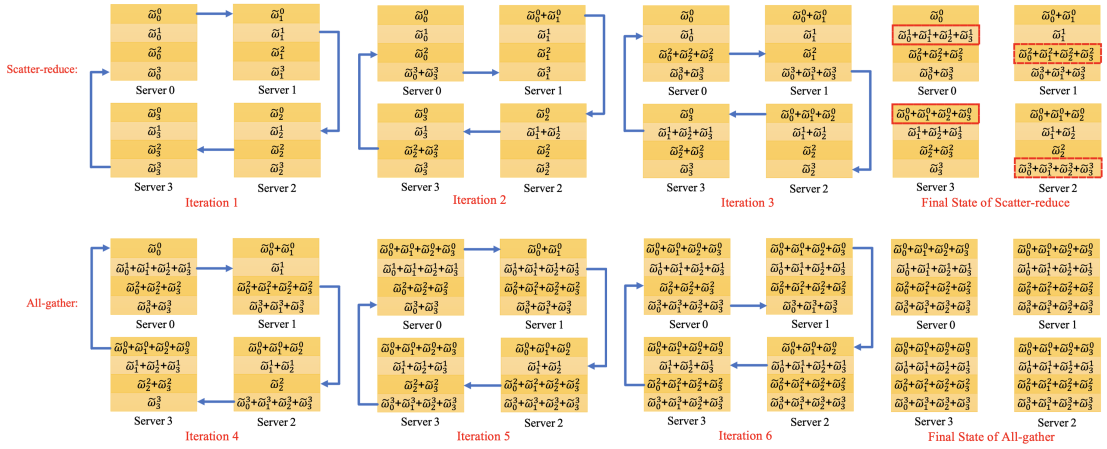


Figure 4.3: Ring-all-reduce. To simplify the representation, we have omitted the initialization process in the figure, that is, before the algorithm starts, each server will multiply its edge model parameters by its weight as the initial input of the global aggregation. Therefore, the final result can be obtained through multiple accumulations instead of weighted averaging.

Scatter-reduce

At the beginning of the algorithm, the server divides the model into four parts equally. In the first round of transmission, server 0 transmits part 0 of the model to server 1 and receives part 3 of the model sent by server 3. The rest of the servers can be inferred by analogy. When server 0 receives the third part of the model, it adds it to the third part of the model it maintains. Then after the first round, each server has a model part that is aggregated from both sides. Note that since we weighted the models in advance, the models from the other servers and the local model are simply added to get the updated model of this round. In the second round, server 0 sends the third part of the model (i.e., the most recently updated part) to server 1 and receives the second part of the model from server 3. Server 0 adds it to the second part of the model, thus maintaining the second part of the model obtained by the three-point aggregation. And so on, after the second round, each server has a model from the three-way aggregation. The state after three rounds is represented by the last subgraph in the first row, as shown

Algorithm 6 RingAVG

Input: $\boldsymbol{\omega}_j$, N, r, D_j
Output: $\boldsymbol{\omega}_j$
for each server $j = 0, 1, ..., n-1$ in parallel **do**
    $\boldsymbol{\omega}_j \leftarrow \frac{|D_j|}{|D_N|}\boldsymbol{\omega}_j$
end for
[Scatter-reduce]
for round $r = 1, 2, ..., N-1$ **do**
    for each server $j = 0, 1, ..., n-1$ in parallel **do**
        $pre = (i - 1 + N)\%N$
        $next = (i + 1)\%N$
        $part(i, r) = (i - (r - 1) + N)\%N$
        Model Transmission:
        send $\boldsymbol{\omega}_j^{part(j,r)}$ to edge$_{next}$
        receive $\boldsymbol{\omega}_{pre}^{part(pre,r)}$ from edge$_{pre}$
        Update Model:
        $\boldsymbol{\omega}_j^{part(pre,r)} \leftarrow \boldsymbol{\omega}_j^{part(pre,r)} + \boldsymbol{\omega}_{pre}^{part(pre,r)}$
    end for
end for
[All-gather]
for round $r = N, N+1.N+2, ..., 2N-2$ **do**
    for each server $j = 0, 1, ..., n-1$ in parallel **do**
        send $\boldsymbol{\omega}_j^{part(i,r)}$ to edge$_{next}$
        receive $\boldsymbol{\omega}_{pre}^{part(pre,r)}$ from edge$_{pre}$
        replace $\boldsymbol{\omega}_j^{part(pre,r)}$ with $\boldsymbol{\omega}_{pre}^{part(pre,r)}$
    end for
end for

in figure/refFIG:ringallreduce.

All-gather

Before this phase begins, the quartet obtains a portion of the final required model maintained by each server. During the rest of the phase, each server in turn sends a part of the final model to the next adjacent server and receives updates from the previous server. After receiving this model, each server directly replaces the corresponding part maintained by itself. After three rounds, each server's model is completely updated and the new global model is obtained.

The description of the algorithm is slightly more complex considering the general representation of the ring subscript. For better understanding, we illustrate the process of the algorithm using representative figures.

RingAVG Algorithm

Figure 4.3 illustrates the detailed process of this algorithm. First of all, the edge servers form a ring topology. For ease of presentation, the subscripts here start from 0, indicating their position on the ring. Assume there are N servers, then the subscript is from 0 to $N-1$. pre represents the previous node of the current node. If the subscript of the current node is 0, then the subscript of the previous node is $pre = (i-1+N) = N-1$. $next$ represents the next node of the current node. If the current node is 1, then the next node is $next = (i+1)\%N = 2$. $part(i,r)$ represents the subscript of the model part to be sent by the server i in the r round. Note that in the $All\text{-}gather$ stage, we use the subscript without repeating its definition.

4.4 Evaluation

In this section, we first perform experiments to verify the heuristic BLKH solution compared with greedy and random solutions on the ring construction problem. For the instances, we choose ten commonly used large-scale asymmetric instances in BTSP. Then, we apply BLKH into our system and carry out simulation experiments that prove the superiority of DEEP-FEL.

4.4.1 Experiments on Ring Construction Problem

Settings

First, we will explain our input. According to our problem, a two-dimensional Cost matrix is actually formed, as shown in the figure below. Among them, $Cost[i,j]$ represents the cost of transferring from i to j. Because we cannot pass it to ourselves, we set $cost[i,i]$ to a large number. Then it is flattened into a one-dimensional matrix as input, and the number of nodes is also input, which is convenient for algorithm processing.

The greedy algorithm strategy here is to start with the first node and select the next

Table 4.1: Cost Matrix

Node	**1**	**2**	**3**	**...**	**n-1**	**n**
1	$+\infty$	$c_{[1,2]}$	$c_{[1,3]}$	$\cdots$	$c_{[1,n-1]}$	$c_{[1,n]}$
2	$c_{[2,1]}$	$+\infty$	$c_{[2,3]}$	$\cdots$	$c_{[2,n-1]}$	$c_{[2,n]}$
3	$c_{[3,1]}$	$c_{[3,2]}$	$+\infty$	$\cdots$	$c_{[3,n-1]}$	$c_{[3,n]}$
...	...	...	...	...	...	...
n-1	$c_{[n-1,1]}$	$c_{[n-1,2]}$	$c_{[n-1,3]}$	$\cdots$	$+\infty$	$c_{[n-1,n]}$
n	$c_{[n,1]}$	$c_{[n,2]}$	$c_{[n,3]}$	$\cdots$	$c_{[n,n-1]}$	$+\infty$

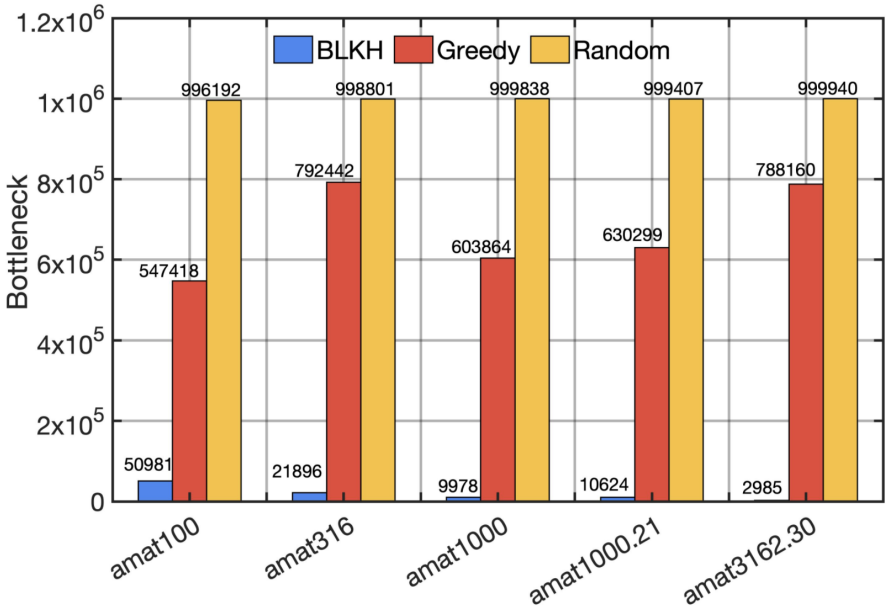


Figure 4.4: Performance comparison among BLKH, Greedy, and Random

node with the smallest cost in turn. The random strategy starts from the first node and randomly selects the next node. Examples for testing are randomly generated. For each algorithm, we run ten times to get the best result.

Results

We choose five instances to test our heuristic solution, that is, amat100, amat316, amat1000, amat1000.21 and amat3162.20. In which there are 100, 316, 1000, 1000, and 3162 nodes relatively. As Figure 4.4 shows, we can see that the performance of BLKH far surpasses ordinary greedy or random methods in the above several examples. Of course, because the examples we take are widely used, in order to test the difference of the algorithm, there will be targeted settings in the generation of values. Therefore, the

difference between the three is more obvious. In our experiments, we also noticed that although BLKH's algorithm execution time is the longest, in the case of 1000 nodes, the execution time is also within one second, so we think it can be actually used.

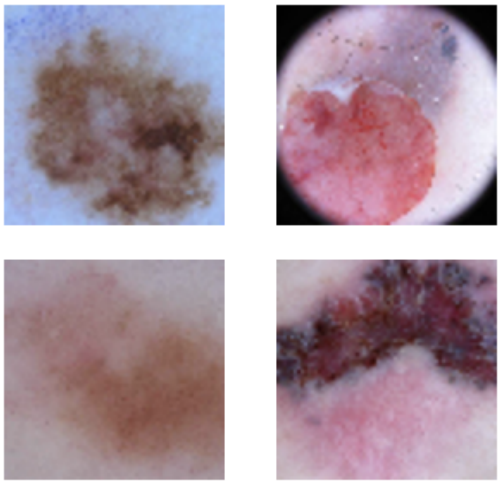
4.4.2 Experiments on DEEP-FEL

Dataset

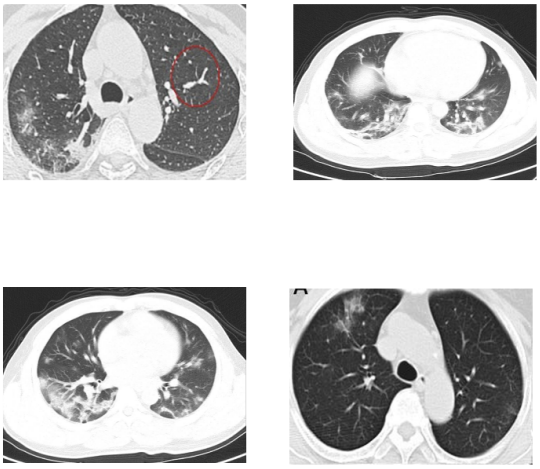
We first describe the datasets (e.g., COVID-19 CT Scans, Eye Disease [101], and Skin Cancer [102]) for our evaluation. CT scans store raw voxel intensity in Hounsfield units (HU) which range from -1024 to above 2000 in this dataset. Above 400 are bones with different radio intensity, so it is used as a higher bound. A threshold between -1000 and 400 is commonly used to normalize CT scans. In our experiment, we leverage COVID-CT data [103] as the medical data set which consists of 651 training samples and 188 testing samples.The CT images each have a dimension of 311×224 pixels and the depth size varying from about 50 to 400 which store raw voxel intensity in HU. Meanwhile, Skin Cancer data set are also used to detect whether the presence of melanoma in images of malignant and benign moles taken. In addition, Eye Disease data set is also used for the classification task. Figure 4.5 shows an instance of these medical dataset. The training fraction is set as 0.8 that means 80% of each type of medical dataset is used to training process and the remained part is used to testing process.

Settings

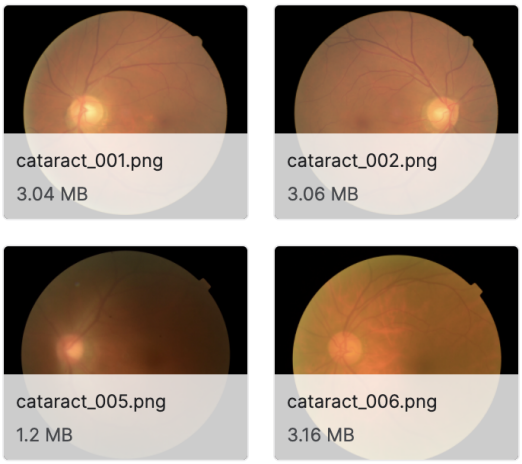
The simulation algorithms are executed on Ubuntu 18.04 with 4 Nvidia GTX 3070 GPUs. Considering the datasets in our evaluation, we select a deep learning network, 3D convolutional neural network (CNN), as the shared training model, which takes as input a 3D volume or a sequence of 2D frames (e.g. slices in a CT scan). It has shown great success for the utilization of volumetric (i.e., spatially 3D) convolutions in video analysis [104] since time can be viewed as the third dimension.



(a) Skin Cancer



(b) COVID-19 CT



(c) Eye Disease

Figure 4.5: Examples of medical dataset (preprocessed)

Models

In this subsection, we will show the steps needed to build a 3D convolutional neural network (CNN) to predict the presence of viral pneumonia in computer tomography (CT) scans. And the specific details of the deep learning model are shown in Table 4.2. The model has a 17 layers 3D CNN which comprises four 3D convolutional (CONV) layers with two layers consisting of 64 filters followed by 128 and 256 filters all with a kernel size of $3 \times 3 \times 3$.

4.4.3 Results and Analysis

Experiments on Different Machine Learning Paradigms

We first test on the Skin Cancer dataset. Our comparison objects are local training on a single device without cooperation, distributed federated learning, and our proposed DEEP-FEL. The lack of cooperation means that the device can only use its local data to train the model. In order to highlight the superiority of our system, the distributed FL here is based on the ring topology with the conventional aggregation algorithm as mentioned in Section III A. The main difference compared with our proposed system is the global aggregation algorithm. We simulate four medical institutions, each with 5 devices participating in the training. For privacy enhancement, the addition of noise will

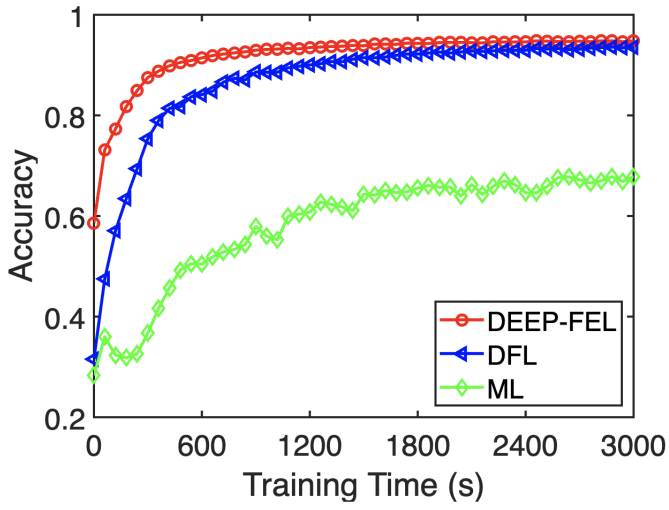


Figure 4.6: Accuracy versus training time on different systems.

affect the model accuracy. In order to highlight the experimental point on efficiency, we set a larger value for ϵ, here it is 10.

Considering the ring construction problem, in the conventional ring decentralized federated learning, we apply the greedy heuristic solver. We consider that when a pair of nodes corresponding to the bottleneck of the ring has been determined, the communication delay mainly depends on the amount of data transmitted. The specific communication settings are similar to those we mentioned earlier. First, a communication cost matrix ranging from 1 to 10 is randomly generated, such as Table 4.1, which represents the communication time cost of transmitting a complete model data from one certain node to the other nodes. Then the greedy and the BLKH heuristic algorithms are respectively applied to obtain the system communication bottleneck, and then the experimental figures are drawn.

Please note that for the sake of simplicity, in the following we will abbreviate the single device training as ML, and the decentralized ring-based FL as DFL. We test it for about 3000 seconds, and the result is shown in Figure 4.6.

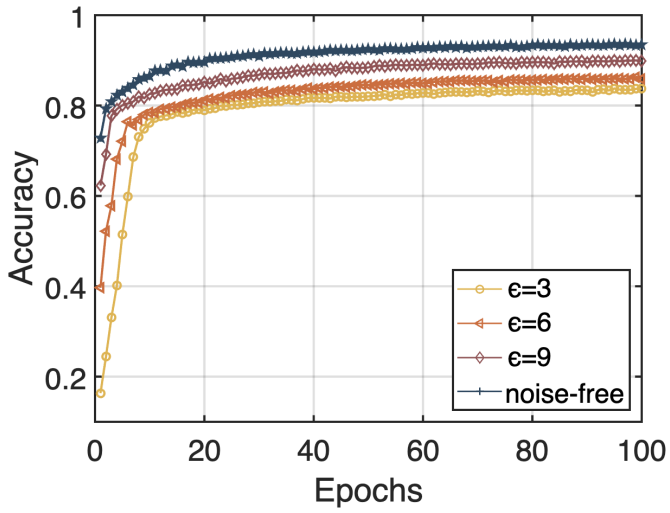
It can prove the superiority of our system in two ways. The first is the accuracy of the model. It can be visualized from the figure that the accuracy of ML is the lowest. This is because the amount of data collected by a single device is limited and it is not possible to train a generic model based on this data alone. This also highlights the value of our federated learning system. In terms of accuracy, DFL and DEEP-FEL are about 46% more accurate than ML. On the other hand, DEEP-FEL is also better than DFL in terms of time cost. When the model accuracy of both systems reaches 0.87, DFL takes about 1020 seconds, while our DEEP-FEL only takes about 300 seconds, which is about one-third of the time overhead of DFL. This reflects the communication efficiency of our system. There are two reasons for this. First, we apply the BLKH heuristic algorithm to construct the ring topology while considering the difference in communication capability between nodes, which improves the efficiency of our system. On the other hand, our RingAVG algorithm greatly reduces the total amount of data transmitted, which in turn saves the communication overhead of model aggregation.

Table 4.2: Model Parameters

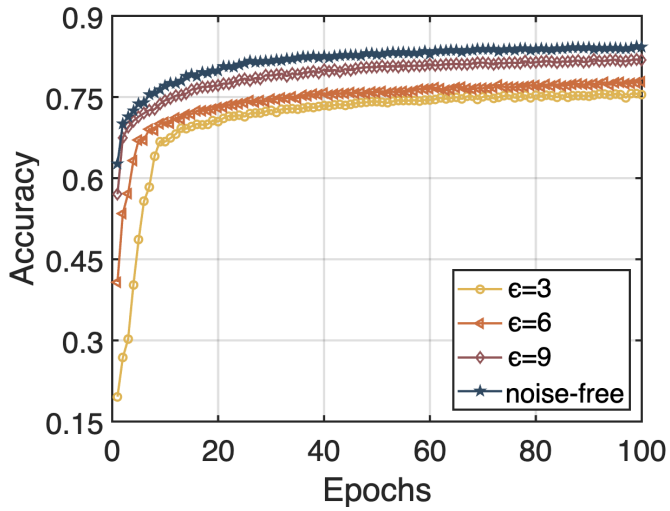
Layer Name	Output Shape	# Parameters	Kernel Size
Conv3D	$[N, 126, 126, 62, 64]$	1972	$[3 \times 3] \times 64$
MaxPooling3D	$[N, 63, 63, 31, 64]$	0	$[3 \times 3]$
BatchNo	$[N, 63, 63, 31, 64]$	256	$[3 \times 3] \times 64$
Conv3D	$[N, 61, 61, 29, 64]$	110656	$[3 \times 3] \times 64$
MaxPooling3	$[N, 30, 30, 14, 64]$	0	$[3 \times 3]$
Batch	$[N, 30, 30, 14, 64]$	256	$[3 \times 3] \times 64$
Conv3D	$[N, 28, 28, 12, 128]$	221312	$[3 \times 3] \times 128$
MaxPooling3	$[N, 14, 14, 6, 128]$	0	$[3 \times 3]$
Batch	$[N, 14, 14, 6, 128]$	512	$[3 \times 3] \times 128$
Conv3D	$[N, 12, 12, 4, 256]$	884992	$[3 \times 3] \times 256$
MaxPooling3	$[N, 6, 6, 2, 256]$	0	$[3 \times 3]$
Batch	$[N, 6, 6, 2, 256]$	1024	$[3 \times 3] \times 256$
G1	$[N, 256]$	1024	None
Dense	$[N, 512]$	131584	None
Dropout	$[N, 512]$	0	None
Dense	$[N, 1]$	513	None

Table 4.3: The Size of Transferred Data Per Node Per Round

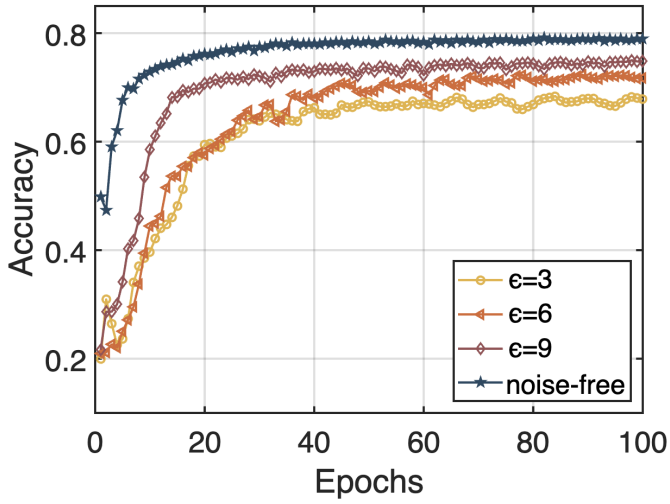
Neural Network	Transferred data size (M)	
	DFL (N=10)	DEEP-FEL (N=10)
$Alexnet$	61.1	12.22
$Densenet121$	7.98	1.596
$Resnet-18$	11.69	2.338
$Resnet-34$	21.8	4.36
$Resnet-50$	25.56	5.112
$VGG-13$	133.05	26.61
$VGG-16$	138.36	27.672
$VGG-19$	143.67	28.734



(a) Skin Cancer dataset

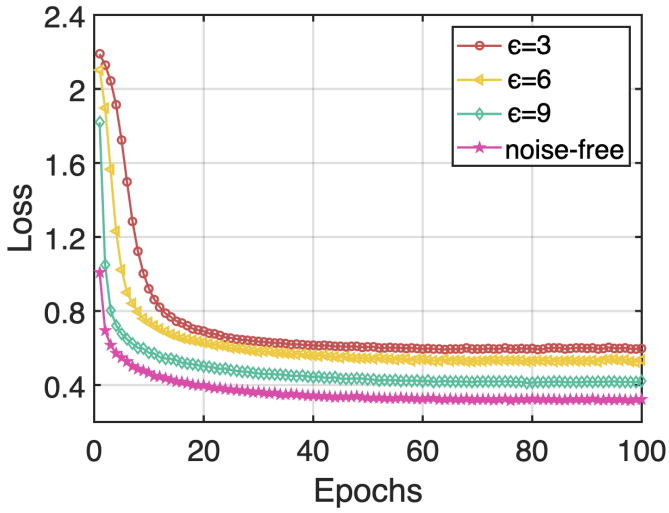


(b) COVID-19 CT dataset

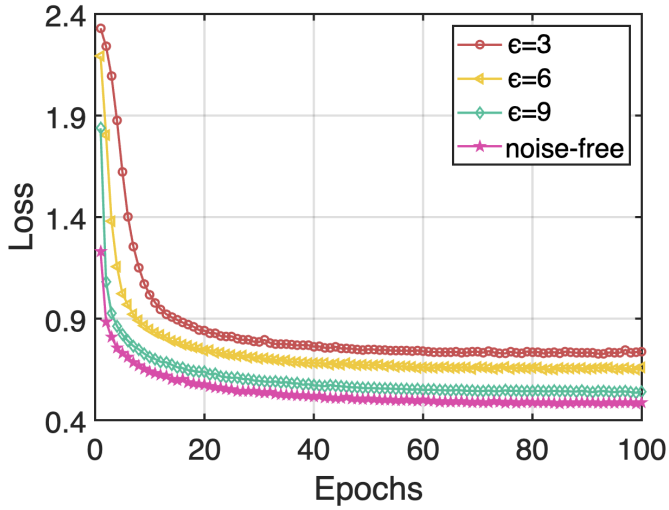


(c) Eye Disease dataset

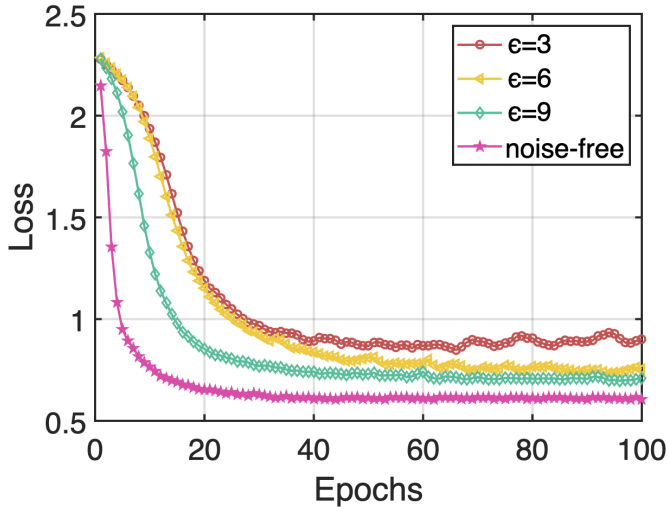
Figure 4.7: Accuracy versus privacy budget ϵ on different datasets.



(a) Skin Cancer dataset



(b) COVID-19 CT Scan dataset



(c) Eye Disease dataset

Figure 4.8: Training loss versus privacy budget ϵ on different datasets.

Moreover, we plot the table 4.3 to demonstrate the superiority of our algorithm in reducing the communication overhead under common models. Here we take $N = 10$ as an example. According to the description of the algorithm in the system design section, the amount of model data transmitted per node per round (one global model update) is only $2/N$ of the original amount of data compared to the traditional update algorithm based on the ring topology. Therefore, the larger N is, the more obvious is the advantage of our algorithm. And as long as $N \leq 2$, our algorithm does not have to be inferior to the traditional update algorithm.

Experiments of DEEP-FEL on Three Medical Datasets

In this section, we evaluate the performance of DEEP-FEL on three medical datasets with changes in privacy budget ϵ, considering the LDP mechanism that we applied to enhance privacy protection.

We select values 3,6 and 9 of ϵ which indicates the degree of privacy protection to test the impact of adding artificial noise on the training process. And the baseline is noise-free, which means that no noise is added. We visualized the changes in accuracy and loss on the three datasets as Figure 4.7 and Figure 4.8 illustrate.

Through Figure 4.7, we can intuitively see that with the decrease of ϵ, the greater the degree of privacy protection, the greater the impact on the training effect. For example, as shown in Figure 4.7(c), when no noise is added, the accuracy of convergence is about 0.8, and when $\epsilon = 3$, the accuracy is only about 0.67. When ϵ increases to 6 and 9, the accuracy increases to 0.72 to 0.75 accordingly.

Combining with Algorithm 5 in the article, it can also be seen that ϵ is in the denominator position of the disturbance term, so the smaller it is, the greater the noise added and the greater the impact on accuracy. The trend is similar on the other two datasets. For Figure 4.8, it is the training loss value we plotted accordingly. Similarly, when the ϵ is smaller, the loss value is greater, and even fluctuates to a certain extent.

Although ϵ that is too small will affect the training results, in actual situations, we can minimize epsilon while ensuring a certain accuracy. Thereby choosing a suitable

privacy budget value that can protect privacy well without significantly affecting the training.

4.5 Conclusion

In this paper, to protect the data security in the increasingly popular healthcare cyber physical systems, we proposed DEEP-FEL, a decentralized, efficient and privacy-enhanced federated edge learning system. In this framework, we applied federated learning to protect medical devices' local data, and also incorporated edge computing to improve the efficiency of global model update process. In order to enhance privacy, local differential privacy technology is also utilized. In addition, for the decentralized ring architecture, we designed an efficient parameter aggregation algorithm, and constructed the ring topology communication bottleneck as an optimization problem, which was solved by the efficient BLKH heuristic later. Finally, compared with existing works, evaluation performance on Skin Cancer, COVID-19 CT Scan and Eye Disease datasets demonstrates that our system achieved outstanding performances in terms of communication efficiency and privacy protection. Future work we may improve further the communication efficiency of this system in more complex and asynchronous collaborative applications using the software-defined network.

Chapter 5

FIND: Privacy-enhanced Federated Learning for Intelligent Fake News Detection

5.1 Introduction

Fake news is used to describe various false information. With social media applications dominating the top of significant software stores, a large amount of information, whether true or false, has been broadcast on social networks. Social media is a double-edged sword. On the one hand, its low cost, convenience, and rapid information dissemination allow people to quickly gain knowledge about the world. On the other hand, malicious, harmful, and misleading fake news can severely impact people and society [105].

A classic example is the 2016 U.S. presidential election, where many American citizens were concerned about the impact of fake news during the election, spread primarily through social media. In [106], the authors discussed the economics of fake news, where 14% of Americans had regarded social media as their most important source of election information according to the survey. People are more likely to believe stories that favor their preferred presidential candidate, so fake news on social media indirectly

affects the election's outcome. Fake news also poses a particular concern during national conflicts, where it can exacerbate the situation. [107] examined the influence of violently inflammatory fake digital images on social media platforms during the recent crises in Russia and Ukraine, highlighting the substantial damage caused by the dissemination of such fake images. Moreover, the global COVID-19 pandemic has resulted in an excessive sharing of information related to viruses, diseases, treatments, vaccines, and lockdowns [108]. Often, this information is widely circulated without proper verification, leading to panic and poor decision-making. In the medical field, the collection of accurate and relevant data and information plays a critical role in improving patient quality of life and enabling correct diagnoses [109].

Due to the negative impact of fake news, propagation detection, and prevention have enormous positive implications. Artificial intelligence and machine learning technologies can improve user experience with their precise predictive capabilities [110], and it has found applications across various domains in computer science to address a wide range of challenges and problems [111]. Various machine learning techniques have shown great potential and usability in fake news detection.

Generally, these methods are centralized, meaning that the model is trained centrally after gathering enough data on the server side. Then the trained model is used for fake news detection. However, this architecture ignores a critical issue–privacy protection. The process of collecting personal information often poses a threat to the user's privacy. Collecting users' data for training a fake news detection model may involve analyzing their browsing history, social media activities, and personal preferences. This process could result in the creation of detailed user profiles that contain sensitive information, such as political affiliations, religious beliefs, or personal interests. If these profiles are not adequately protected, they could be vulnerable to misuse, targeted advertising, or even malicious targeting.

To cope with this problem, we can apply a distributed privacy-enhanced machine learning approach to fake news detection. Federated learning (FL) [112] is a distributed machine learning paradigm that enables users to train a global model collaboratively

without direct user data sharing. Moreover, the final model accuracy of FL is close to that of centralized training methods. In FL, the training process can be summarized into two phases: local training and central aggregation. In the local training phase, each user trains a local model using its local data and sends the results to the parameter server. Then, in the aggregation phase, the server generates a new global model by weighted average and sends it to the users. The above process iterates continuously until the training goal is reached (i.e., it usually refers to the number of training rounds or the convergence criteria).

Although FL has alleviated the raw data leakage risk, many studies still demonstrate that attackers can compromise users' privacy through intercepted data (e.g., gradients or model parameters). In [113], the authors performed a model inversion attack by exploiting the model parameters shared in FL and successfully stealing critical information. Model inversion attack aims to construct an inversion model by learning the inputs and outputs of the target model, thereby stealing private training data [114]. Membership inference attacks can also pose significant privacy concerns. Given a data record and an FL model, the attacker aims to determine whether the data is in the model's training dataset or not [115]. Differential privacy techniques can effectively prevent the above-mentioned attacks and attract plenty of attention [116]. In simple terms, it aims to allow analyzing the dataset and its related statistics such as mode, median, and mean without personal information leakage by adding artificial noise to the model parameters [117].

This paper focuses on designing an FL system for fake news detection. Each user only needs to train locally and upload their updates to train a high-accuracy detection model collaboratively. At the same time, considering the threats faced by the FL system, we apply a local differential privacy mechanism by adding noise into the sparsified model to achieve privacy enhancement and save communication overhead while ensuring the high performance of the detection system.

The contributions relative to this paper can be summarized as follows:

- Given the increasingly serious problem of fake news and the privacy of users, we propose FIND, a fake news detection system based on FL, which can train a high-

accuracy detection model without gathering the user's local data. To the best of our knowledge, this is the first work to comprehensively apply FL for fake news detection.

- In order to strengthen the privacy protection in FL, we apply differential privacy technology with sparsified model representation, which not only achieves privacy enhancement but also reduces communication overhead.

- We conducted experiments on the widely used Kaggle fake news dataset to investigate the impact of different machine learning paradigms, varying local data volumes, and different sparsity parameters on the performance of fake news detection.

The rest of the paper is organized as follows. In Section II, we present basic background knowledge related to fake news detection and privacy-enhanced FL. In Section III, we introduce the system design in detail and present the corresponding algorithm. In Section IV, we show the experimental results and the corresponding analysis. We then compare with related work and finally give the conclusion in Section V and VI, respectively.

5.2 Background

5.2.1 Machine Learning for Fake News Detection

Machine learning classification algorithms have been extensively applied in diverse domains including healthcare, agriculture, engineering, sports, entertainment, economics, management, and social sciences [118]. Some researchers have applied machine learning techniques to a database of disinformation articles and factual information mined from media news databases and found that the classifier is suitable for fake news detection [119]. The detection can be viewed as a binary classification problem, where the goal is to process and analyze text files of data to determine whether it can be considered fake news.

Naive Bayes classifier has been used to detect data on Facebook news posts. The authors stated that the problem of false information detection could be solved by artificial intelligence [120]. There are also studies based on other classifiers, such as Support Vector Machine (SVM) [121] and Long Short-Term Memory (LSTM) [122], Convolutional Neural Network (CNN) [123], etc. They all show the feasibility of various machine-learning methods for fake news detection.

We preliminary experimented on a fake news dataset with commonly used classifiers. The results in Fig. 5.1 show that the FNN with a simple structure can achieve good results. The following research will also be based on this network structure.

5.2.2 Intelligent Fake News Detection

In [124], swarm learning is used for fake news detection, and the authors also apply human-in-the-loop to improve model accuracy by manually intervening in the training process. The idea of swarm learning is similar to federated learning, but the focus is on decentralization, that is, without a central server. However, the experimental setup in this paper is limited to a maximum of 8 participating nodes, which has certain limitations in practical applications. At the same time, it requires users to correct the model and expand the training set during the training process, which puts additional requirements on the participants and increases the cost of local computing, thus reducing the efficiency of the system.

In addition, the authors of [125] propose a federated learning-based COVID-19 fake news detection model with deep self-attention network named FL_FNDM. In this article, the main work lies in the design of machine learning models, and the discussion on federated learning itself is insufficient, and the number of participants and the impact of local data size on federated learning are not discussed. Only three participants are set up in this paper, and the experimental results are preliminary and not very convincing. In addition, there is no discussion about the security threats faced by federated learning itself and no communication-related optimizations.

Compared with the above works, we focus on the performance of federated learning

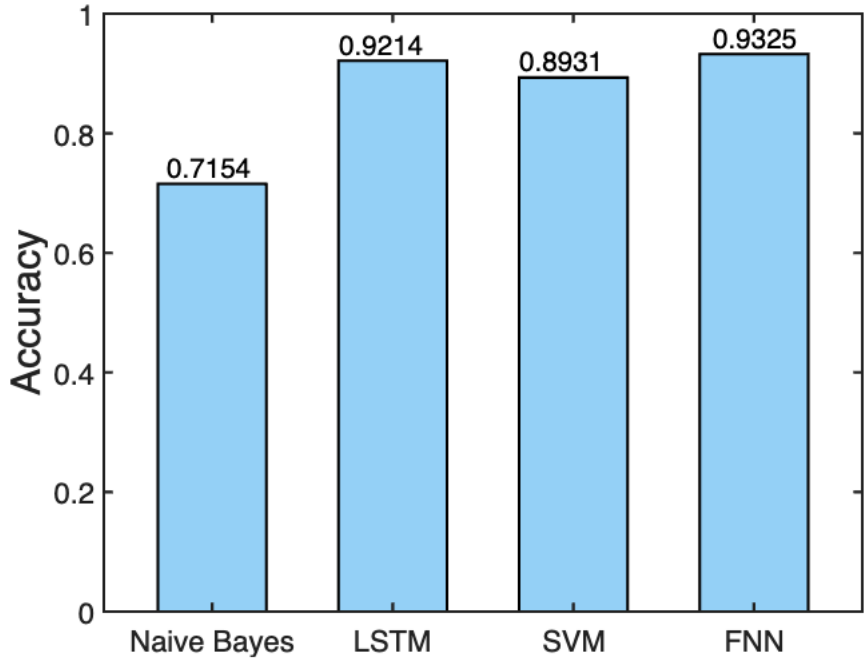


Figure 5.1: Primary test on different machine learning models.

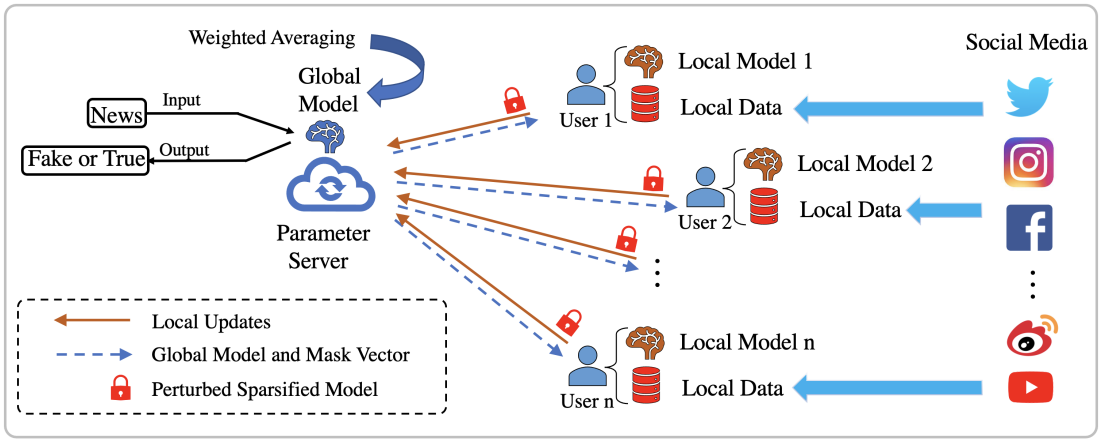


Figure 5.2: FIND: federated learning for intelligent fake news detection

itself in fake news detection, assume 30 users, and conduct experiments on multiple metrics in federated learning. In addition, we also focus on privacy preservation in federated learning, inspired by [126], propose a sparsified update perturbation method, and discuss the impact of privacy-related settings on accuracy.

5.2.3 Federated Learning (FL)

With the development of big data, edge computing, large cloud computing platforms and various open source frameworks, machine learning, and other artificial intelligence technologies are applied to multiple industries at an unprecedented speed, which also brings a new challenge - data privacy protection [127]. Internet data of various industries are scattered in different enterprises and organizations, forming the phenomenon of ”data silos” [128], and data cannot be directly shared or exchanged under data privacy protection.

Therefore, federated learning, a distributed machine learning framework that delegates data processing tasks to the client without revealing privacy, was born [129]. [130] proposed a federated learning framework where they have used local datasets from banks to build fraud detection models and shared FDS to protect sensitive customer information. In the field of healthcare, federated learning has gained significant applications. It is a collaborative algorithm known as an aggregated algorithm, which trains local healthcare datasets across different fog nodes and shares the learned knowledge with central server [131]. In vehicular networks, federated learning (FL) is also being applied to address various challenges and enhance the performance of intelligent transport systems [132].

5.2.4 Differential Privacy in FL

Recent research shows that the privacy of client data can be leaked by the gradient parameters of the global model in FL, i.e. it is not enough to protect privacy by keeping the data local, and that privacy-preserving techniques can protect privacy at the expense of model accuracy. Thus, differential privacy techniques are often needed in FL to

protect client privacy and to select reasonable parameters for model accuracy improvement. Differential Privacy [116], also known as statistical disclosure control, inference control, privacy-preserving data mining, and private data analysis is a framework for privacy guarantee evaluation provided by a mechanism designed to protect privacy.

Centralized differential privacy (DP) is difficult to implement in federated learning [117]. Thus, local differential privacy (LDP) is often applied to federated learning [133].

The differential privacy-based encryption model is more advantageous for FL gradient information encryption because it mainly adds noise to the gradient information with weak communication costs. Although it may affect the accuracy of the model convergence, it can be achieved under many iterations [116]. At the same time, the security-based multi-party computation is mainly implemented through complex communication protocols or encryption mechanisms on both sides of the C/S.

FL allows end-customer data to be aggregated and separated from the cloud's mechanistic learning model to protect sensitive client data [134] and is becoming more widely used because of its clear advantages in the sensitive issue of privacy and security. However, clients still leak sensitive information [135] when performing model updates with the cloud.

Differential privacy (DP) is a technique employed to safeguard the privacy of individual data points when performing statistical analyses or machine learning tasks. It provides a rigorous mathematical framework for quantifying and controlling the privacy risks associated with the release of sensitive information. The main objective of DP is to ensure that an adversary cannot confidently determine whether a particular individual's data was included in a dataset. In the realm of federated learning, where multiple clients collaborate to construct a shared model without sharing their raw data, DP assumes a critical role in preserving privacy [136]. In the context of federated learning, centralized differential privacy (CDP) is primarily utilized to protect privacy during the data aggregation process at the central server [137]. The central server meticulously incorporates calibrated noise into the aggregated updates to prevent the exposure

of sensitive information about any individual participant's data. The privacy guarantees offered by CDP guarantee that the global model does not disclose specific details regarding any individual participant's data. To uphold privacy standards, the server adheres to a predefined privacy budget, ensuring that the total privacy cost incurred during the aggregation process remains within the designated budget. This control mechanism prevents the accumulation of privacy breaches and ensures a consistent level of privacy protection throughout the communication rounds in the federated learning process. In contrast, the local differential privacy (LDP) approach focuses on privacy protection at the client's level, primarily concerning their local data and local model updates [138]. Clients introduce noise or employ privacy-preserving algorithms during the training process to protect the privacy of individual data. Local model updates are modified using LDP mechanisms, such as noise addition or random perturbation, to safeguard against potential inference attacks and information leakage. The central server receives the perturbed model updates from participants and aggregates them to construct a global model. The perturbation introduced by LDP prevents the direct identification of individual contributions, thereby preserving privacy at the client's level.

5.3 The Proposed FIND System

In this section, we present the design details of the proposed FIND system. We first give the basic introduction and assumptions of the system and specify the training process and objectives. Then we give a description of the algorithm and elaborate on the key steps, including local training, sparsified model perturbation and parameter aggregation, respectively. Finally, we also look forward to the feasibility of the system in a peer-to-peer architecture.

5.3.1 Training Goals

We adopt a conventional centralized FL architecture, where each user i, represented as a smart device, has access to a vast amount of social media data containing both true

Table 5.1: Symbols and Variables

Symbol	Variable
n	The number of total users
m	Number of randomly selected users
T	The number of total epochs
D_i	The local dataset of user i
p_i	The weight of user i
ω_0	The initial model.
$\omega(t)$	The global model at epoch t
$\omega_i(t)$	User i's local model at epoch t
$\widetilde{\omega}_i(t)$	Perturbed model of user i at epoch t
$F_i(\omega)$	The loss function for user i with model ω
$\nabla F_i(\cdot)$	The gradient of function F_i
$\|D_i\|$	The local data size of user i
η	The learning rate.

and fake information as depicted in Figure 5.2.

They can all communicate with the parameter server instead of direct mutual communication. In this system, these devices collect authentic and fake news from social networks and store them locally as strings. These data are discernible as true or false. In other words, these data are labeled. We use D_i to denote the local dataset of user i and $|D_i|$ to refer to the amount of data. To defend against fake news, these users collaboratively train a detection model. Note that $\boldsymbol{w}_i$ is denoted as the model parameter of user i.

Since $F(\omega)$ denotes the empirical loss, the training objective of FL to minimize $F(\boldsymbol{w})$ can be written as follows:

$$\min F(\omega) = \frac{1}{|D|} \sum_{i=1}^{N} |D_i| F_i(\omega), \tag{5.1}$$

where $|D|$ denotes the total amount of data for all users that is equal to:

$$|D| = \sum_{i=1}^{N} |D_i|. \tag{5.2}$$

$|D_i|/|D|$ in Equation 5.1 is the weight of user i, which is a common assumption in FL. Moreover, we can notice that in Equation 5.1, $F(\omega)$ is obtained by weighted

averaging the loss of each user. In FL, the global loss cannot be directly calculated due to its distributed nature. The specific per-user loss is then shown in the following equation:

$$F_i(\omega) = \frac{1}{|D_i|} \sum_{j=1}^{|D_i|} f(\omega; (x_j, y_j)), \tag{5.3}$$

where $f(\omega; (x_j, y_j)$ denotes the value of loss function (i.e., prediction error) given the model ω and the training data sample (x_i, y_i).

5.3.2 Threat Model

Although user data is kept local throughout the FL training, recent research has shown that attackers can launch various attacks to obtain the original information. For instance, gradient leakage attacks [139], and reconstruction attacks [113]. Analyzing the client's local gradient updates history makes these attacks possible. Since the parameter server typically collects updates from the clients' local models in plaintext, this type of leakage is also feasible for the aggregator. This work considers the standard threat model: the honest but curious parameter server or the curious and colluding user. We assume that the attackers are honest, i.e., they follow the training protocol we specify. However, at the same time, they are curious about other participants' private information and try to infer private data by collecting and analyzing uploaded data. Participants may also collaborate to obtain private information by inspecting the information transmitted in multiple rounds with the server. We also clarify that attackers do not maliciously inject information that interferes with training, such as malicious model uploading.

5.3.3 FIND Detailed Operations

Next, we describe the details of the system operation in conjunction with the algorithm description from both user and server-side perspectives. We assume that a total of n users participate in the training of T epochs, with m users randomly selected before the start of each training round (i.e., this is a common assumption for FL with the

Algorithm 7 FIND: Federated Learning for Intelligent Fake News Detection

1: **Input:** Initial model $\boldsymbol{w}(0)$, initial mask vector $\boldsymbol{v}(0)$, learning rate η;
2: **Output:** $\boldsymbol{w}$;
3: **for** t = 1 T **do**
4: **for** each user $i = 1, 2, 3, ..., m$ in parallel **do**
5: $\Delta_i(t) = OnDeviceTraining(\boldsymbol{w}(t), \eta)$
6: $\widetilde{\Delta}'_i(t) = SparsifiedPerturbation(\Delta_i(t), \boldsymbol{v}(t))$
7: Send perturbed sparsified update $\widetilde{\Delta}'_i(t)$ to parameter server
8: $\boldsymbol{v}(t), \boldsymbol{w}(t) = GlobalAggregation(\widetilde{\Delta}'_i(t))$
9: Send mask vector $\boldsymbol{v}(t)$ and new global model $\boldsymbol{w}(t)$ to all users
10: **end for**
11: **end for**
12: **Return:** $\boldsymbol{w}_i(t)$

Algorithm 8 On-Device Training

1: **Input:** Initial model $\boldsymbol{w}(0)$, learning rate η
2: **Output:** Local update $\Delta_i(t)$
3: $\boldsymbol{v}_i(t) = \boldsymbol{v}(t-1)$
4: $\boldsymbol{w}_i(t) = \boldsymbol{w}(t-1)$
5: $\boldsymbol{w}_i(t) = \boldsymbol{w}_i(t) - \eta \nabla F_i(\boldsymbol{w}_i(t))$
6: $\Delta_i(t) = \boldsymbol{w}(t-1) - \boldsymbol{w}_i(t)$
7: **Return:** $\Delta_i(t)$

primary purpose of improving communication efficiency).

User-Side

The users first receives the global model and the mask vector from the server. Before the first training round starts, the global model $\boldsymbol{w}(0)$ is generated by the server and the mask vector is a vector $\boldsymbol{v}(t) \in \{0, 1\}^d$ of length d consisting of 0 and 1, in particular $v(0) = \{1\}^d$.

As in lines 3 to 6 of Algorithm 7, the selected users first initialize the local models with the received global model and then update the model parameters by gradient descent on their local data to get the local update Δ_i^t of epoch t.

The updates obtained at this point have the same structure as the model. The users generate sparsified updates by multiplying the local updates with the corresponding mask vector received from the server. To avoid privacy leakage by honest but curious servers and to ensure local sparsified updates security, we adopt the local differential privacy scheme to provide privacy guarantees. Here, we give the comprehensive defini-

Algorithm 9 Sparsified Perturbation

1: **Input:** Local update $\Delta_i(t)$, mask vector $v(t)$
2: **Output:** Perturbed sparsified update $\widetilde{\Delta}'_i(t)$
3: $\Delta'_i(t) = \Delta_i(t) \odot \boldsymbol{v}(t)$
4: $\widetilde{\Delta}'_i(t) = \Delta'_i(t) + Lap(\Delta s/\epsilon) \odot \boldsymbol{v}(t)$
5: **Return:** $\widetilde{\Delta}'_i(t)$

tion and Laplace mechanism:

Definition 2 *(ϵ-LDP). For $\epsilon > 0$, a random algorithm M satisfies the ϵ-local differential privacy, if and only if any two inputs x, $x^{'}$ in the domain of possible values for user data D, for any output x^*, respectively. Hence, we have:*

$$\frac{Pr[M(x) = x^*]}{Pr[M(x') = x^*]} \leq e^{\epsilon}. \tag{5.4}$$

Definition 3 *Laplace mechanism. Given data input value x and a function f, the Laplace mechanism is defined as:*

$$f(x) = x + Lap(\Delta s/\epsilon), \tag{5.5}$$

where $Lap(\cdot)$ is the Laplace distribution, $(\Delta s/\epsilon)$ is the scale parameter. And the Δs denotes the local sensitivity of two data inputs on the objective function. Then the Laplace noise is added to raw sparsified updates. The users then send the perturbed sparsified updates to the parameter server for global aggregation as Algorithm 9.

Server-Side

The parameter server randomly selects m users to participate in the next round of training before the start of each round and sends the global model and the new mask vector to these users.

Once the updates are received from all m users, the server performs a weighted average of these updates to derive the global update, which is then used to obtain the new global model. The weighted average takes into account the contribution of each user's update based on predefined weights or other factors. After obtaining the global update,

Algorithm 10 Global Aggregation

1: **Input:** Perturbed update $\widetilde{\Delta}_i'(t)$
2: **Output:** Mask vector $\boldsymbol{v}(t)$ and new global model $\boldsymbol{w}(t)$
3: $\boldsymbol{w}(t) = \boldsymbol{w}(t-1) - \frac{1}{|D(t)|}\sum_{i\in[m]} |D_i|\widetilde{\Delta}_i'(t)$
4: $\Delta(t) = \boldsymbol{w}(t-1) - \boldsymbol{w}(t)$
5: Generate mask vector according to the selected top γ coordinates of $|\Delta(t)|$
6: **Return:** $\boldsymbol{v}(t)$ and $\boldsymbol{w}(t)$

the server proceeds with the sparsification process. This process involves selecting the coordinates based on the Top-γ strategy as Equation 5.6 illustrates.

$$[Top_\gamma(\boldsymbol{v})]_j = \begin{cases} [\boldsymbol{v}]_j, & \text{if } j \in Sort(\boldsymbol{v}, \gamma) \\ 0, & \text{otherwise} \end{cases}, \tag{5.6}$$

where the elements in $\boldsymbol{v}$ are sorted by value and the top γ ones are selected and their position subscript j is recorded into the set $Sort(\boldsymbol{v}, \gamma)$. Once the top γ coordinates are identified, the server generates a new mask vector.

Compared with [126], the main difference in the sparsified model perturbation is the generation of mask vectors. The authors mention that a common method is to collect part of the data on the server side, and use the global model to train for several rounds, compare the parameter changes, and generate the corresponding mask vector. This assumption, although feasible, is essentially contrary to the idea of federated learning. In our paper, we strictly require that the server does not store any data but determines the mask vector used in the next round of local training by comparing the changes of the global model in two adjacent epochs. This indirect generation method also shows a great training effect and achieves high model accuracy in our experimental setting.

This mask vector acts as a filter or selector, assigning a value of 1 to the selected coordinates and 0 to the remaining coordinates. By applying this mask vector to the local update, the clients effectively sparsifies the model by zeroing out or excluding certain coordinates, reducing its overall size and complexity.

According to Lemma 9 in [126], we likewise have the bounded sparsification as:

$$\mathbb{E}||Top_\gamma(\boldsymbol{v}) - \boldsymbol{v}||^2 \leq \mathbb{E}||rand_\gamma(\boldsymbol{v}) - \boldsymbol{v}||^2 \tag{5.7}$$

$$= \sum_{j=1}^{d} (\frac{\gamma}{d}([\boldsymbol{v}]_j - [\boldsymbol{v}]_j)^2 + (1 - \frac{\gamma}{d})[\boldsymbol{v}]_j^2) \tag{5.8}$$

$$= (1 - \frac{\gamma}{d})||\boldsymbol{v}||^2, \tag{5.9}$$

where $rand_\gamma(\boldsymbol{v})$ is a random sparsifier, i.e., γ elements are randomly selected from the vector $\boldsymbol{v}$ and the rest elements are set to 0.

5.4 Simulation Experiments

5.4.1 Datasets and Model

We selected the Kaggle fake news dataset [140] as our benchmark for mainly two reasons. Firstly, this dataset is easily accessible, allowing us to obtain the necessary data conveniently. Secondly, it has already been utilized in previous relevant research [141], indicating its suitability for evaluating and comparing our detection methods with existing approaches. By leveraging this widely-used dataset, we can ensure the comparability of our results with prior studies.

The dataset consists of three files, train.csv, test.csv and submit.csv. Since the dataset is used in the kaggle competition, test.csv is missing the label compared to the training set. Submit.csv is the file template used to submit the results. Therefore, we only select the file train.csv for the following experiments.

Among them, train.csv includes 20,800 records. It includes five attributes: id, title, author, text, and label. We choose two attributes, text and label, to train the model. Since some of the data lack text attribute values and the strings cannot be fed directly to the model, we need to process the data further. Fig. 5.3 provides a direct and brief overview of the key steps in data preprocessing by giving an example of the first piece of data in the dataset. We choose the feedforward neural network model which includes

id	title	author	text (str)	label
0	…	…	…	1

Remove unwanted attribute columns.

text (str)	label
House Dem Aide: We Didn' t Even See Comey' s Letter Until Jason Chaffetz ...	1

Clean data, remove stopwords, special symbols, etc.

text (list)	label
['house', 'dem', 'aide', 'even', 'see', 'comey', 'letter', 'jason', 'chaffetz', 'tweeted' ...	1

Generate vector representations.

text (list)	label
[-0.15896325 0.0524237 0.24444875 0.28692251 -0.31936178 -0.05843308 ...	1

Figure 5.3: Preprocessing of training data.

three hidden layers with a total of 163570 trainable parameters. The activation functions of the first three dense layers are chosen as the commonly used relu, and the last one is chosen as softmax.

5.4.2 Results and Analysis

We assume that users participate and jointly train a neural network model for fake news detection. Each user has the same number of local data, and they component the local dataset by random sampling from the training set. Here we present the main experimental parameters as shown in Table 5.2.

First, we explore the effectiveness of FL in this usage scenario, then we further explore the effect of local data volume on convergence speed and model accuracy.

The Feasibility of Federated Learning in Fake News Detection

We conducted experiments on 30 users. Each user randomly selected 200 non-repeated data from the training set to compose the local dataset. Fig. 5.4 shows the local data composition of ten randomly selected users from the 30 participants, and we stipulate that there is no overlap between the local data of each user, and the percentage of classification as true news and false news is obtained based on sampling, so as to simulate the data heterogeneity in real scenarios. After 50 rounds of global training, we obtained the experimental results and visualized them as Fig. 5.5 and Fig. 5.6.

We compare the convergence speed and accuracy of federated learning, centralized

Table 5.2: Experimental Parameters

Parameters	Values
OS	Ubuntu 18.04
GPU	Nvidia GTX 3070
Dataset	Kaggle fake news
Number of users	30
Data volume of each user	50, 100, 150, 200
Privacy budget ϵ	9
Communication rounds	50, 100
Learning rate	0.001
Batch size	64

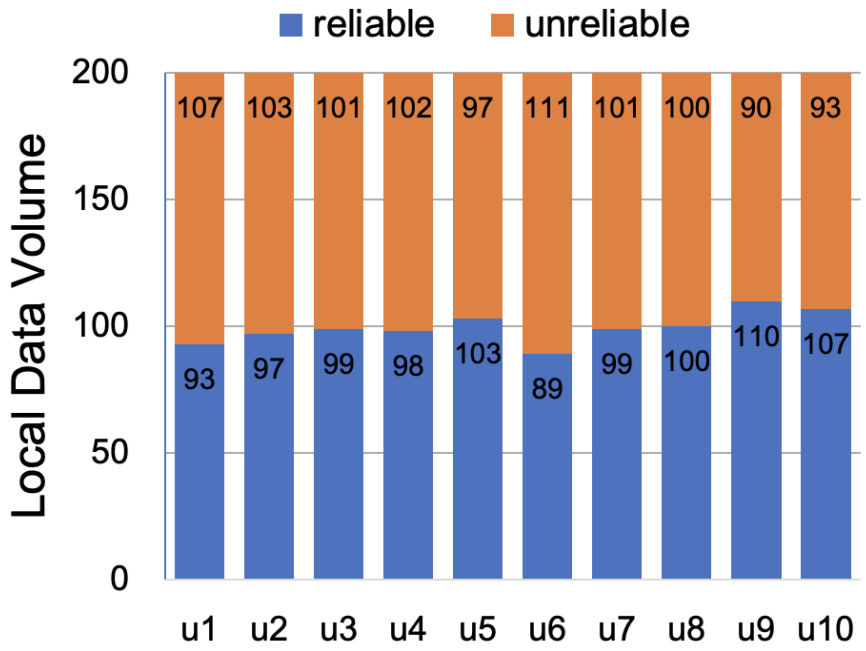


Figure 5.4: The data composition of the local dataset (i.e., Unlikable refers to data labeled as fake news.).

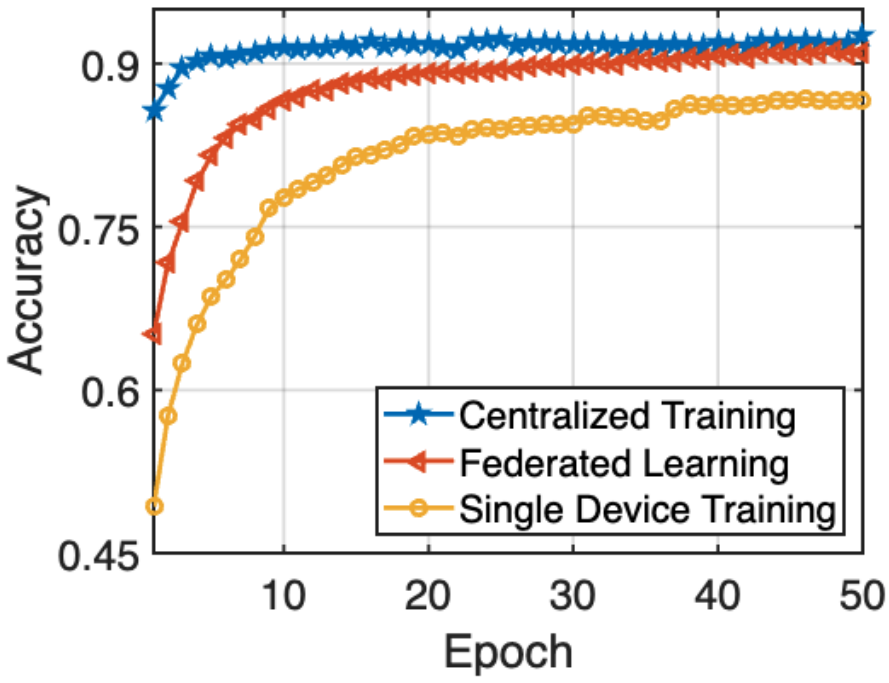
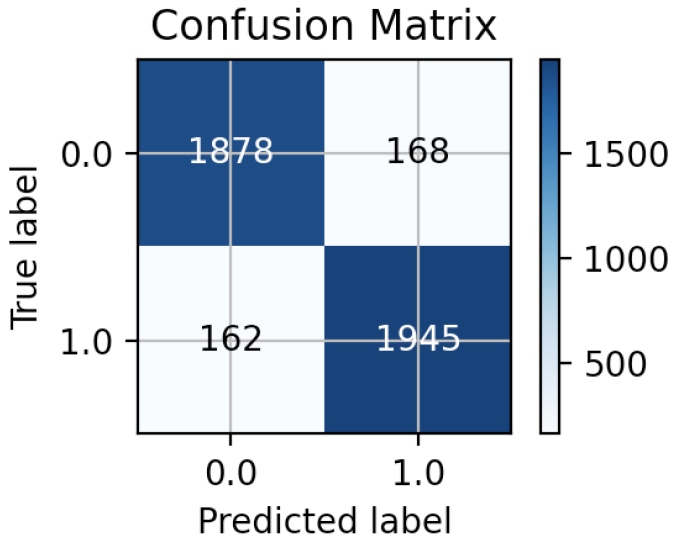
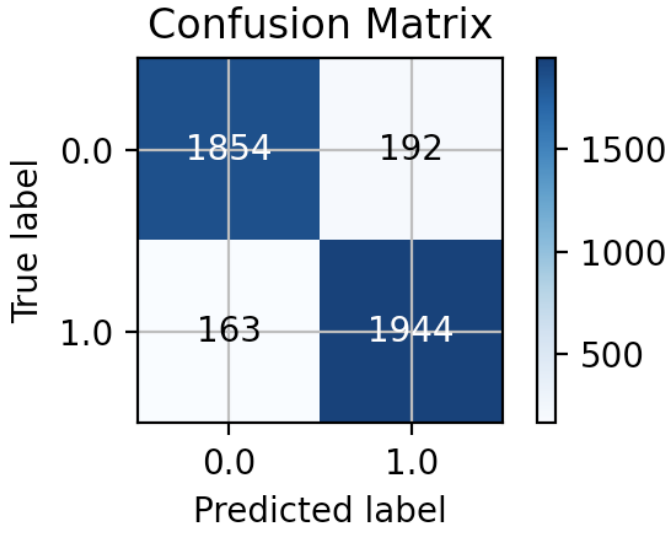


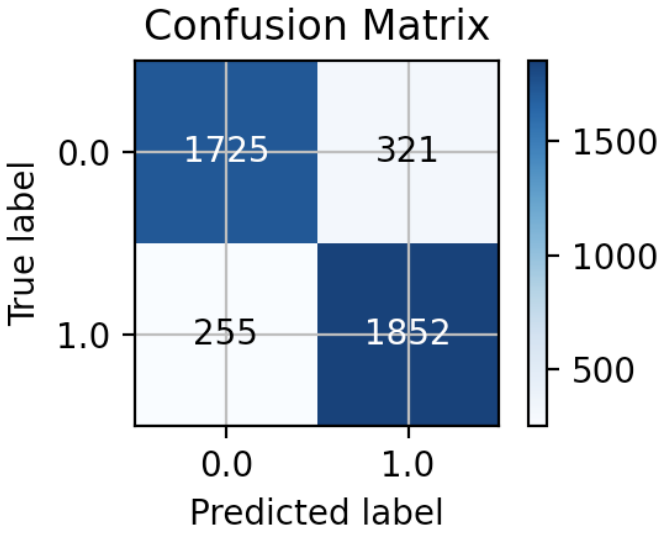
Figure 5.5: Comparison of accuracy versus epoch in centralized training, federated learning, and single device training.



(a) Centralized Training



(b) Federated Learning



(c) Single Device Training

Figure 5.6: Confusion matrix of different machine learning scenarios.

training, and single device learning. From Fig. 5.5, we can intuitively see that centralized method is the fastest in terms of convergence speed, because the data are pooled together, which is the most efficient, but lacks feasibility in terms of the need to protect user privacy without gathering their data. Training on a single device, on the other hand, has the slowest convergence rate and the lowest accuracy. After ten epochs of training, the accuracy was only 0.77. In comparison, federated and centralized training reached accuracy values of 0.86 and 0.90, respectively, which are about 12% and 17% higher than single device training. This is because the limited amount of local data for a single user cannot support the training of a high-precision model, which is where big data comes in, and in general, much data can lead to higher model accuracy.

Fig. 5.6 visualizes the model performance of the above three methods after 50 epochs of training by constructing confusion matrix. The darker color of the main diagonal line represents better performance. The testing accuracy of the FL model is about 0.91, which is only 1.1% lower than centralized learning, and 5.8% higher than single device training. We can conclude that FL is feasible and superior in this scenario by aggregating only the user's local models, which both protects the security of the user's local data and benefits from aggregating the training results of multiple users to improve model accuracy and convergence speed.

The Impact of Model Sparsification

In order to investigate the impact of sparsification on system performance, we explored different sparsity levels, represented by the value of γ in Algorithm 10. We defined γ values such as 0.5, 0.4, and 0.3, which indicate selecting and retaining a percentage of 50%, 40%, and 30% of the model parameters, respectively, while setting the remaining elements to 0. This further allows for additional compression. We choose a relatively large privacy budget, and here we choose $\epsilon = 9$ for the experiment. We conducted experiments with different γ values and the results are presented in Table 5.3.

According to the results in Table 5.3, we can analyze the effect of different values

Table 5.3: Impact of Sparsification on System Performance

γ	Epoch=20	Epoch=40	Epoch=60	Epoch=80	Epoch=100
1	0.8026	0.8471	0.8612	0.8878	0.8892
0.8	0.8041	0.8435	0.8589	0.8863	0.8991
0.6	0.7928	0.8324	0.8434	0.8584	0.8725
0.4	0.6393	0.6960	0.7414	0.7917	0.8221
0.2	0.5020	0.6248	0.6997	0.7541	0.7629

of γ on the performance of the system. The table provides accuracy values for each γ value across different epochs (20, 40, 60, and 80).

We observe that as the value of γ decreases, indicating higher sparsity levels, the accuracy of the system tends to decrease as well. For example, at epoch 20, the accuracy decreases from 0.8026 (for $\gamma = 1$) to 0.7928 (for $\gamma = 0.6$) and further to 0.5020 (for $\gamma = 0.2$). This trend suggests that higher sparsity levels lead to a loss of information or important features, resulting in a decline in system performance.

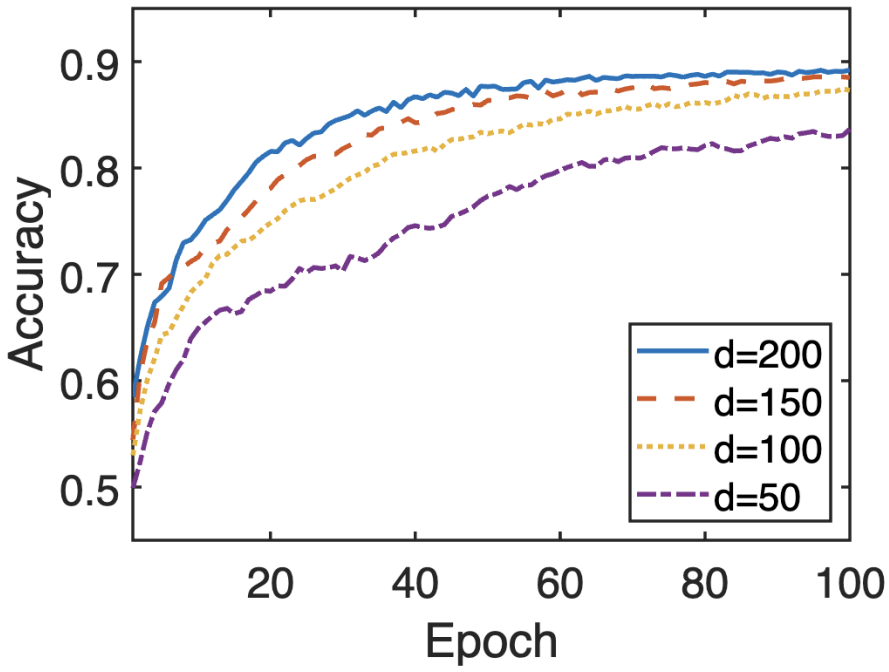
However, it is noteworthy that the system achieves better performance when γ is set to 0.8 and 0.6. For instance, at epoch 20, the accuracy is 0.8041 for $\gamma = 0.8$ and 0.7928 for $\gamma = 0.6$. This indicates that an appropriate choice of γ can reduce communication overhead without significantly impacting the model's performance.

Considering the trade-off between model complexity and performance, we have decided to select $\gamma = 0.6$ for subsequent experiments. This value strikes a balance between achieving a certain level of sparsity and maintaining satisfactory system performance.

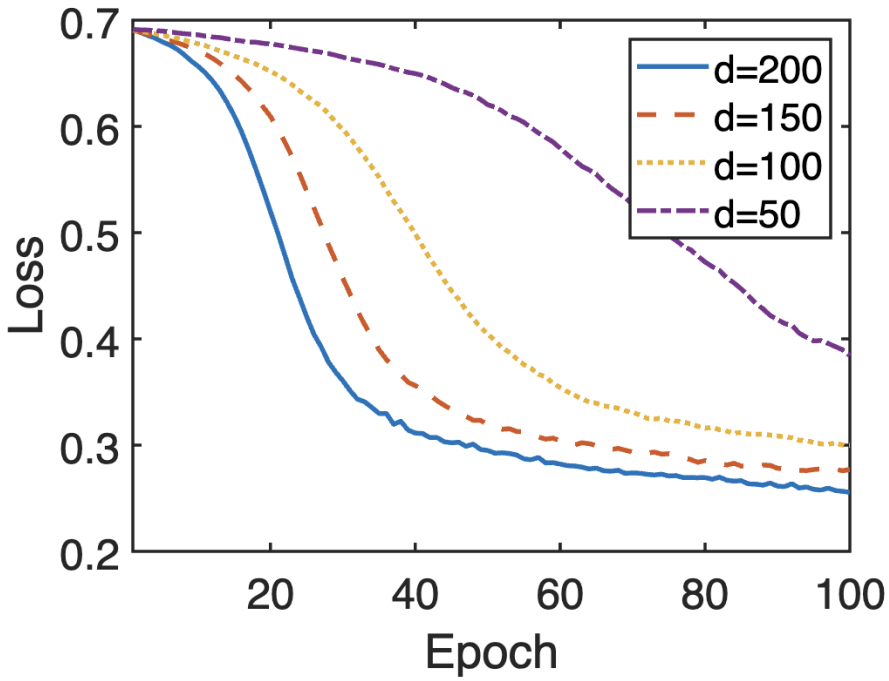
The Impact of Local Data Volume on Training

In this part, we explore the impact of users' local data volume d on the training performance. The local data is sampled in the same way as in the previous experiments. We tested at d taking values 50, 100, 150 and 200. After training for 100 epochs, we plot the changes in accuracy and loss as shown in Fig. 5.7.

We can clearly see that the larger the amount of local data, the higher the accuracy of the model and the lower the loss value after the same number of training rounds. This is in line with our expectations, and more data should yield better results in each iteration



(a) Accuracy versus epochs.



(b) Loss versus epochs.

Figure 5.7: Trends of accuracy and loss under different local data volumes.

of the user's local model. After 50 rounds of training, in terms of accuracy, when the amount of local training data is 50, the accuracy is only 0.77, while when the amount of data is 200, the accuracy reaches 0.877, an improvement of about 14%. The gap is shrinking, but in the end, the testing accuracy with the largest amount of local data is still the highest. The results are similar for loss. The smaller the amount of local data, the slower the loss value decreases and the model converges. Through this experiment, we verified that the more data, the higher the accuracy of the model and the faster the convergence speed.

5.5 Conclusion

To address the issue of fake news, we propose FIND, a federated learning-based detection system. FIND leverages the distributed training capability of federated learning to train a highly accurate detection model while ensuring user privacy protection. Additionally, we introduce the sparsified update perturbation method, which further enhances the system's resilience to inference attacks by sparsifying the model and introducing artificial noise. In the experimental section, we evaluate FIND using the Kaggle Fake News dataset. We consider a scenario with 30 users and vary the local data volume per user, ranging from 50 to 200 instances. We also explore different sparsity levels by setting the sparsity parameter, γ, to values from 0.2 to 1. Our experiments demonstrate the superiority of FIND in terms of accuracy, communication efficiency, and privacy preservation, reinforcing its effectiveness in combating fake news.

5.6 Future Research

In our future research, we will primarily concentrate on two aspects. Firstly, we aim to delve deeper into the application of unsupervised federated learning in the context of fake news detection. Given the substantial amount of news data often lacking labeled annotations, unsupervised learning methods are better suited for addressing this common scenario. We will explore novel approaches and algorithms within the unsu-

pervised federated learning framework to enhance the accuracy and efficiency of fake news detection models.

Secondly, we intend to explore the role of large-scale models in supporting fake news detection. Large models, such as those based on transformer architectures, have demonstrated impressive capabilities in various natural language processing tasks. We will investigate how these models can be effectively leveraged for fake news detection. Specifically, we will utilize prompt learning techniques, including pre-training and fine-tuning strategies, to harness the power of large models and adapt them to the specific challenges of identifying and combating fake news.

By focusing on these two aspects, we aim to advance the state-of-the-art in fake news detection and contribute to developing robust and efficient solutions. Through our research, we strive to provide valuable insights and practical tools that can help mitigate the spread of misinformation and promote trustworthiness in online information ecosystems.

Chapter 6

Conclusion and Future Directions

6.1 Conclusion

This dissertation has been structured to address the pressing challenges in Federated Learning (FL), with a particular focus on data privacy and communication efficiency. The first chapter of this dissertation sets the stage by providing an overarching background on Federated Learning (FL), its significance, and the challenges it faces. It introduces the key concepts of data privacy and communication efficiency within the FL paradigm, laying the groundwork for the subsequent chapters. Following the introduction, the second chapter outlines the structure of the dissertation, detailing how each subsequent chapter contributes to the overarching themes of data privacy and communication efficiency in FL.

Chapter 3 dives deep into the application of FL in the Internet of Medical Things (IoMT). This chapter presents a novel framework that integrates blockchain technology to ensure secure and transparent model updating. Beyond mere security measures, the framework is designed with an emphasis on adaptability, incorporating personalized learning algorithms that are tailored to meet individual healthcare needs. By doing so, this work transcends the traditional boundaries of data privacy and steps into the realm of improving healthcare quality. It paves the way for a new generation of medical services that are not only secure but also highly personalized, thereby changing the way

healthcare providers engage with machine learning technologies.

Following this, Chapter 4 introduces DEEP-FEL, a decentralized system that aims to optimize FL specifically for healthcare applications. The system employs a unique hierarchical ring topology and a heuristic algorithm, which facilitate efficient data aggregation. One of the standout features of DEEP-FEL is its innovative parameter aggregation algorithm. This algorithm is engineered to minimize data transmission, thereby significantly reducing communication overhead—a critical factor in real-world healthcare settings where data transmission costs can be prohibitive. DEEP-FEL stands as a testament to what can be achieved when machine learning is thoughtfully integrated into healthcare services, offering a robust solution that does not compromise data privacy or system efficiency.

The final research chapter, Chapter 5, shifts the focus towards the societal implications of FL, examining its role in the critical area of fake news detection. This chapter introduces the FIND system, a pioneering approach that leverages advanced natural language processing techniques in conjunction with FL. The system is designed to train a global model capable of detecting fake news while ensuring that all user data remains localized. This dual focus on societal impact and data privacy makes FIND an exemplary model for how FL can be applied for social good, especially in today's age of information overload and misinformation.

In conclusion, this dissertation represents a rigorous foray into enhancing Federated Learning (FL) for the Internet of Medical Things (IoMT) and social computing spheres. It stands as a testament to the potential of FL to operate with heightened efficiency and improved privacy safeguards, as encapsulated in the title "Optimizing Federated Learning for IoMT and Social Computing Based on Efficiency and Privacy Enhancements". Our work bridges current gaps and lays the groundwork for future explorations, positing a resilient architecture for the forthcoming wave of FL systems that are both privacy-conscious and efficiency-oriented.

6.2 Current Work's Limitations and Future Directions

6.2.1 Current Limitations:

- Scalability Issues in Large-Scale Networks: The current research primarily focuses on small-scale user networks. There is a lack of in-depth study on the scalability and stability of large-scale node networks. In big data environments, balancing algorithm efficiency and privacy protection while managing and optimizing numerous nodes remains an unresolved challenge.

- Insufficient Comprehensive Security Consideration: Although the research emphasizes data privacy protection, it falls short in addressing broader security challenges such as data tampering, model leakage, and network attacks. The defensive mechanisms against these sophisticated threats require further strengthening.

- Lack of Practicality and Feasibility Verification: The current studies predominantly remain theoretical and experimental, lacking sufficient validation and testing of federated learning systems in real-world application environments.

6.2.2 Future Research Directions:

- Optimization for Large-Scale Networks: Future research should focus on developing federated learning algorithms suitable for large-scale networks. This includes efficient node management, dynamic optimization strategies, and load-balancing techniques to achieve scalability and stability in large-scale applications.

- Enhancing Security Protection Mechanisms: It's essential to enhance security protection mechanisms against complex network attacks, including improved encryption technologies, secure multi-party computation, and defensive strategies to enhance the overall security of federated learning systems.

- Testing and Validation in Real Applications: Collaborative empirical studies and

system tests should be conducted with industry partners, applying federated learning in real-world scenarios. Through collaborations with industry partners, the performance and stability of federated learning systems in actual environments can be validated, and feedback can be collected for iterative improvements.

- Standardization and Interoperability Research: Promoting the standardization of federated learning is crucial. Developing universal protocols and standards for data sharing, model training, and model aggregation is necessary.

By exploring these directions, federated learning can better adapt to various application requirements while protecting data privacy, enhancing its usability and impact in real-world settings.

Acknowledgment

I am profoundly grateful to my supervisor, Prof. Chunhua Su, whose invaluable guidance, insightful feedback, and unwavering support have been crucial throughout my research journey. His expertise and mentorship have been instrumental in the completion of this dissertation.

I would also like to extend my sincere appreciation to the members of my dissertation committee: Prof. Akihito Nakamura, Prof. Xin Zhu, and Prof. Yasuyuki Kachi, for their constructive critiques and insightful suggestions, which have significantly enhanced the quality of my work.

My heartfelt gratitude goes to Eyes, Japan Co., Ltd., and President Mr. Jun Yamadera for their invaluable support and assistance, which have been essential to my academic and personal development.

I am deeply thankful to the NEC C&C Foundation for their generous grant, which has been a cornerstone in supporting my research endeavors.

I express my gratitude to the department staff for their administrative and technical support, which has greatly facilitated my research process.

Lastly, I would like to offer my heartfelt thanks to my family and friends for their constant encouragement and understanding throughout this challenging yet rewarding journey. Their support has been a source of strength and motivation.

References

[1] Google Brain Team, "Tensorflow federated," https://www.tensorflow.org/federated.

[2] IBM Research, "Ibm federated learning," https://www.ibm.com/docs/en/watsonx-as-a-service?topic=models-federated-learning.

[3] NVIDIA, "Nvidia clara federated learning," https://docs.nvidia.com/clara/clara-train-sdk/federated-learning/federated_learning.html.

[4] WeBank's AI Group, "Federated ai technology enabler," https://fate.fedai.org.

[5] OpenMined, "Pysyft: A python library for secure and private deep learning," https://github.com/OpenMined/PySyft.

[6] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Transactions on Mobile Computing*, vol. 21, no. 8, pp. 2818–2832, 2020.

[7] M. F. Khan, T. M. Ghazal, R. A. Said, A. Fatima, S. Abbas, M. Khan, G. F. Issa, M. Ahmad, and M. A. Khan, "An iomt-enabled smart healthcare model to monitor elderly people using machine learning technique," *Computational Intelligence and Neuroscience*, vol. 2021, 2021.

[8] R. P. Singh, M. Javaid, A. Haleem, R. Vaishya, and S. Ali, "Internet of medical things (iomt) for orthopaedic in covid-19 pandemic: Roles, challenges, and applications," *Journal of Clinical Orthopaedics and Trauma*, vol. 11, no. 4, pp. 713–717, 2020.

[9] G. Srivastava, J. Crichigno, and S. Dhar, "A light and secure healthcare blockchain for iot medical devices," in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, 2019, pp. 1–5.

[10] B. Ghimire and D. B. Rawat, "Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8229–8249, 2022.

[11] D. Chowdhury, S. Banerjee, M. Sannigrahi, A. Chakraborty, A. Das, A. Dey, and A. D. Dwivedi, "Federated learning based covid-19 detection," *Expert Systems*, vol. n/a, no. n/a, p. e13173. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.13173

[12] S. Sanyal, D. Wu, and B. Nour, "A federated filtering framework for internet of medical things," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[13] F. Alsubaei, A. Abuhussein, and S. Shiva, "Security and privacy in the internet of medical things: taxonomy and risk assessment," in *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*. IEEE, 2017, pp. 112–120.

[14] X. Xiao, Z. Tang, C. Li, B. Xiao, and K. Li, "Sca: Sybil-based collusion attacks of iiot data poisoning in federated learning," *IEEE Transactions on Industrial Informatics*, 2022.

[15] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[16] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 1698–1707.

[17] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–9.

[18] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3400–3413, 2019.

[19] V. Kulkarni, M. Kulkarni, and A. Pant, "Survey of personalization techniques for federated learning," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. IEEE, 2020, pp. 794–797.

[20] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of things intrusion detection: Centralized, on-device, or federated learning?" *IEEE Network*, vol. 34, no. 6, pp. 310–317, 2020.

[21] M. Seliem and K. Elgazzar, "Biomt: Blockchain for the internet of medical things," in *2019 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. IEEE, 2019, pp. 1–4.

[22] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4298–4311, 2020.

[23] 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.

[24] A. Gatouillat, Y. Badr, B. Massot, and E. Sejdić, "Internet of medical things: A review of recent contributions dealing with cyber-physical systems in medicine," *IEEE internet of things journal*, vol. 5, no. 5, pp. 3810–3822, 2018.

[25] F. Al-Turjman, M. H. Nawaz, and U. D. Ulusar, "Intelligence in the internet of medical things era: A systematic review of current and future trends," *Computer Communications*, vol. 150, pp. 644–660, 2020.

[26] M. V. Ramesh, S. Anand, and P. Rekha, "A mobile software for health professionals to monitor remote patients," in *2012 Ninth international conference on wireless and optical communications networks (WOCN)*. IEEE, 2012, pp. 1–4.

[27] W. Wang, Q. Chen, Z. Yin, G. Srivastava, T. R. Gadekallu, F. Alsolami, and C. Su, "Blockchain and puf-based lightweight authentication protocol for wireless medical sensor networks," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8883–8891, 2021.

[28] H. Xiong, C. Jin, M. Alazab, K.-H. Yeh, H. Wang, T. R. Gadekallu, W. Wang, and C. Su, "On the design of blockchain-based ecdsa with fault-tolerant batch verification protocol for blockchain-enabled iomt," *IEEE journal of biomedical and health informatics*, vol. 26, no. 5, pp. 1977–1986, 2021.

[29] I. A. Khan, N. Moustafa, I. Razzak, M. Tanveer, D. Pi, Y. Pan, and B. S. Ali, "Xsru-iomt: Explainable simple recurrent units for threat detection in internet of medical things networks," *Future Generation Computer Systems*, vol. 127, pp. 181–193, 2022.

[30] A. Almogren, I. Mohiuddin, I. U. Din, H. Almajed, and N. Guizani, "Ftm-iomt: Fuzzy-based trust management for preventing sybil attacks in internet of medical things," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4485–4497, 2020.

[31] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Bedgehealth: A decentralized architecture for edge-based iomt networks using blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 743–11 757, 2021.

[32] Z. Zhao, J. Xia, L. Fan, X. Lei, G. K. Karagiannidis, and A. Nallanathan, "System optimization of federated learning networks with a constrained latency," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 1095–1100, 2021.

[33] D. Byrd and A. Polychroniadou, "Differentially private secure multi-party computation for federated learning in financial applications," pp. 1–9, 2020.

[34] G. Long, Y. Tan, J. Jiang, and C. Zhang, "Federated learning for open banking," in *Federated learning*. Springer, 2020, pp. 240–254.

[35] S. Warnat-Herresthal, H. Schultze, K. L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N. A. Aziz *et al.*, "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, 2021.

[36] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen *et al.*, "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data," *Scientific reports*, vol. 10, no. 1, pp. 1–12, 2020.

[37] G. Kaissis, A. Ziller, J. Passerat-Palmbach, T. Ryffel, D. Usynin, A. Trask, I. Lima, J. Mancuso, F. Jungmann, M.-M. Steinborn *et al.*, "End-to-end privacy preserving deep learning on multi-institutional medical imaging," *Nature Machine Intelligence*, vol. 3, no. 6, pp. 473–484, 2021.

[38] D. C. Nguyen, Q.-V. Pham, P. N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. Dobre, and W.-J. Hwang, "Federated learning for smart healthcare: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–37, 2022.

[39] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[40] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," 2019.

[41] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[42] M. Arambakam and J. Beel, "Federated meta-learning: Democratizing algorithm selection across disciplines and software libraries," in *7th ICML Workshop on Automated Machine Learning (AutoML)*, 2020.

[43] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.

[44] J. Chen, K. Li, and S. Y. Philip, "Privacy-preserving deep learning model for decentralized vanets using fully homomorphic encryption and blockchain," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 633–11 642, 2021.

[45] J. Chen, K. Li, and P. S. Yu, "Privacy-preserving deep learning model for decentralized vanets using fully homomorphic encryption and blockchain," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 633–11 642, 2022.

[46] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2019.

[47] Y. Li, C. Chen, N. Liu, H. Huang, Z. Zheng, and Q. Yan, "A blockchain-based decentralized federated learning framework with committee consensus," *IEEE Network*, vol. 35, no. 1, pp. 234–241, 2020.

[48] 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.

[49] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[50] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[52] X. Luo, Y. Wu, X. Xiao, and B. C. Ooi, "Feature inference attack on model predictions in vertical federated learning," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 181–192.

[53] A. Hatamizadeh, H. Yin, P. Molchanov, A. Myronenko, W. Li, P. Dogra, A. Feng, M. G. Flores, J. Kautz, D. Xu *et al.*, "Do gradient inversion attacks make federated learning unsafe?" *IEEE Transactions on Medical Imaging*, 2023.

[54] R. Verma, "Smart city healthcare cyber physical system: Characteristics, technologies and challenges," *Wireless personal communications*, pp. 1–21, 2021.

[55] B. Yin, H. Yin, Y. Wu, and Z. Jiang, "Fdc: A secure federated deep learning mechanism for data collaborations in the internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6348–6359, 2020.

[56] Z. Tang, R. Zhu, P. Lin, J. He, H. Wang, Q. Huang, S. Chang, and Q. Ma, "A hardware friendly unsupervised memristive neural network with weight sharing mechanism," *Neurocomputing*, vol. 332, pp. 193–202, 2019.

[57] Y. Guo, F. Liu, Z. Cai, L. Chen, and N. Xiao, "Feel: A federated edge learning system for efficient and privacy-preserving mobile healthcare," in *49th International Conference on Parallel Processing-ICPP*, 2020, pp. 1–11.

[58] D. Lee, "Strategies for technology-driven service encounters for patient experience satisfaction in hospitals," *Technological forecasting and social change*, vol. 137, pp. 118–127, 2018.

[59] Z. Obermeyer and S. Mullainathan, "Dissecting racial bias in an algorithm that guides health decisions for 70 million people," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019, pp. 89–89.

[60] S. A. Latif, F. B. X. Wen, C. Iwendi, F. W. Li-li, S. M. Mohsin, Z. Han, and S. S. Band, "Ai-empowered, blockchain and sdn integrated security architecture for iot network of cyber physical systems," *Computer Communications*, vol. 181, pp. 274–283, 2022.

[61] M. Shabbir, A. Shabbir, C. Iwendi, A. R. Javed, M. Rizwan, N. Herencsar, and J. C.-W. Lin, "Enhancing security of health information using modular encryption standard in mobile cloud computing," *IEEE Access*, vol. 9, pp. 8820–8834, 2021.

[62] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[63] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *Journal of Healthcare Informatics Research*, vol. 5, no. 1, pp. 1–19, 2021.

[64] Y. Liu, Y. Kang, X. Zhang, L. Li, Y. Cheng, T. Chen, M. Hong, and Q. Yang, "A communication efficient collaborative learning framework for distributed features," *arXiv preprint arXiv:1912.11187*, 2019.

[65] 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)*. IEEE, 2019, pp. 739–753.

[66] Y. Chen, Y. Ning, M. Slawski, and H. Rangwala, "Asynchronous online federated learning for edge devices with non-iid data," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 15–24.

[67] Z. Wang, Y. Hu, J. Xiao, and C. Wu, "Efficient ring-topology decentralized federated learning with deep generative models for industrial artificial intelligent," *arXiv preprint arXiv:2104.08100*, 2021.

[68] 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)*, 2019, pp. 739–753.

[69] 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*. PMLR, 2017, pp. 1273–1282.

[70] A. Lalitha, O. C. Kilinc, T. Javidi, and F. Koushanfar, "Peer-to-peer federated learning on graphs," *arXiv preprint arXiv:1901.11173*, 2019.

[71] 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.

[72] H. Wang, L. Muñoz-González, D. Eklund, and S. Raza, "Non-iid data rebalancing at iot edge with peer-to-peer federated learning for anomaly detection," in *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2021, pp. 153–163.

[73] Y. Liu, Z. Ai, S. Sun, S. Zhang, Z. Liu, and H. Yu, "Fedcoin: A peer-to-peer payment system for federated learning," in *Federated Learning*. Springer, 2020, pp. 125–138.

[74] I. Hegedűs, G. Danner, and M. Jelasity, "Gossip learning as a decentralized alternative to federated learning," in *IFIP International Conference on Distributed Applications and Interoperable Systems*. Springer, 2019, pp. 74–90.

[75] Y. Wu, "Cloud-edge orchestration for the internet of things: Architecture and ai-powered data processing," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 792–12 805, 2020.

[76] M. Ibrar, L. Wang, G.-M. Muntean, J. Chen, N. Shah, and A. Akbar, "Ihsf: An intelligent solution for improved performance of reliable and time-sensitive flows in hybrid sdn-based fc iot systems," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3130–3142, 2020.

[77] 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.

[78] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated learning-based computation offloading optimization in edge computing-supported internet of things," *IEEE Access*, vol. 7, pp. 69 194–69 201, 2019.

[79] 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.

[80] 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)*, 2020, pp. 1–6.

[81] A. Ziller, D. Usynin, R. Braren, M. Makowski, D. Rueckert, and G. Kaissis, "Medical imaging deep learning with differential privacy," *Scientific Reports*, vol. 11, no. 1, pp. 1–8, 2021.

[82] Y. Guo, Y. Wu, Y. Zhu, B. Yang, and C. Han, "Anomaly detection using distributed log data: A lightweight federated learning approach," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[83] O. Choudhury, A. Gkoulalas-Divanis, T. Salonidis, I. Sylla, Y. Park, G. Hsu, and A. Das, "Differential privacy-enabled federated learning for sensitive health data," *arXiv preprint arXiv:1910.02578*, 2019.

[84] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[85] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, "Ldp-fed: Federated learning with local differential privacy," in *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, 2020, pp. 61–66.

[86] B. B. Gupta, K.-C. Li, V. C. Leung, K. E. Psannis, S. Yamaguchi *et al.*, "Blockchain-assisted secure fine-grained searchable encryption for a cloud-based healthcare cyber-physical system," *IEEE/CAA Journal of Automatica Sinica*, 2021.

[87] G. N. Nguyen, N. H. Le Viet, M. Elhoseny, K. Shankar, B. Gupta, and A. A. Abd El-Latif, "Secure blockchain enabled cyber–physical systems in healthcare using deep belief network with resnet model," *Journal of Parallel and Distributed Computing*, vol. 153, pp. 150–160, 2021.

[88] M. M. Baig, H. GholamHosseini, and M. J. Connolly, "Mobile healthcare applications: system design review, critical issues and challenges," *Australasian physical & engineering sciences in medicine*, vol. 38, no. 1, pp. 23–38, 2015.

[89] M. Alhussein and G. Muhammad, "Voice pathology detection using deep learning on mobile healthcare framework," *IEEE Access*, vol. 6, pp. 41 034–41 041, 2018.

[90] A. Subasi, M. Radhwan, R. Kurdi, and K. Khateeb, "Iot based mobile healthcare system for human activity recognition," in *2018 15th Learning and Technology Conference (L T)*, 2018, pp. 29–34.

[91] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2134–2143, 2019.

[92] A. Qayyum, K. Ahmad, M. A. Ahsan, A. Al-Fuqaha, and J. Qadir, "Collaborative federated learning for healthcare: Multi-modal covid-19 diagnosis at the edge," *arXiv preprint arXiv:2101.07511*, 2021.

[93] J. Li, J. Cai, F. Khan, A. U. Rehman, V. Balasubramaniam, J. Sun, and P. Venu, "A secured framework for sdn-based edge computing in iot-enabled healthcare system," *IEEE Access*, vol. 8, pp. 135 479–135 490, 2020.

[94] Z. H. Ahmed, "A hybrid genetic algorithm for the bottleneck traveling salesman problem," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 12, no. 1, pp. 1–10, 2013.

[95] S. N. Kabadi and A. P. Punnen, "The bottleneck tsp," in *The Traveling Salesman Problem and Its Variations*. Springer, 2007, pp. 697–735.

[96] K. Helsgaun, "General k-opt submoves for the lin–kernighan tsp heuristic," *Mathematical Programming Computation*, vol. 1, no. 2, pp. 119–163, 2009.

[97] Helsgaun, Keld, "Solving the clustered traveling salesman problem using the lin-kernighan-helsgaun algorithm," 2014.

[98] J. LaRusic and A. P. Punnen, "The asymmetric bottleneck traveling salesman problem: algorithms, complexity and empirical analysis," *Computers & Operations Research*, vol. 43, pp. 20–35, 2014.

[99] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[100] A. Gibiansky, "Bringing hpc techniques to deep learning," *Baidu Research, Tech. Rep.*, 2017.

[101] F. A. D. Santos, "Predicting human eye diseases," https://www.kaggle.com/fabianogalaxy/dataset-with-catarats-images, accessed: 2021-11-22.

[102] M. Jennings, "Skin-cancer-identification," https://github.com/Matt-Jennings-GitHub/Skin-Cancer-Identification, accessed: 2021-11-22.

[103] J. Zhao, Y. Zhang, X. He, and P. Xie, "Covid-ct-dataset: a ct scan dataset about covid-19," *arXiv preprint arXiv:2003.13865*, 2020.

[104] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2012.

[105] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD explorations newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[106] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–36, May 2017. [Online]. Available: https://www.aeaweb.org/articles?id=10.1257/jep.31.2.211

[107] S. M. YarAdua *et al.*, "Influence of digital images on the propagation of fake news on twitter in russia and ukraine crisis."

[108] V. Balakrishnan, W. Z. Ng, M. C. Soo, G. J. Han, and C. J. Lee, "Infodemic and fake news – a comprehensive overview of its global magnitude during the covid-19 pandemic in 2021: A scoping review," *International Journal of Disaster Risk Reduction*, vol. 78, p. 103144, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212420922003636

[109] W. Wang, Q. Chen, Z. Yin, G. Srivastava, T. R. Gadekallu, F. Alsolami, and C. Su, "Blockchain and puf-based lightweight authentication protocol for wireless medical sensor networks," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8883–8891, 2022.

[110] S. Pandya, G. Srivastava, R. Jhaveri, M. R. Babu, S. Bhattacharya, P. K. R. Maddikunta, S. Mastorakis, M. J. Piran, and T. R. Gadekallu, "Federated learning for smart cities: A comprehensive survey," *Sustainable Energy Technologies and Assessments*, vol. 55, p. 102987, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2213138822010359

[111] M. Hina, M. Ali, A. R. Javed, G. Srivastava, T. R. Gadekallu, and Z. Jalil, "Email classification and forensics analysis using machine learning," in *2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI)*, 2021, pp. 630–635.

[112] J. A. Alzubi, O. A. Alzubi, A. Singh, and M. Ramachandran, "Cloud-iiot based electronic health record privacy-preserving by cnn and blockchain-enabled federated learning," *IEEE Transactions on Industrial Informatics*, 2022.

[113] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 937–16 947, 2020.

[114] H. Ren, J. Deng, and X. Xie, "Grnn: generative regression neural network—a data leakage attack for federated learning," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–24, 2022.

[115] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[116] C. Dwork, "Differential Privacy: A Survey of Results," in *Theory and Applications of Models of Computation*. Berlin, Germany: Springer, 2008, pp. 1–19.

[117] A. E. Ouadrhiri and A. Abdelhadi, "Differential privacy for deep and federated learning: A survey," *IEEE Access*, vol. 10, pp. 22 359–22 380, 2022.

[118] Y. Muhammad, M. A. Hassan, S. Almotairi, K. Farooq, F. Granelli, and L. Strážovská, "The role of socioeconomic factors in improving the performance of students based on intelligent computational approaches," *Electronics*, vol. 12, no. 9, p. 1982, 2023.

[119] C. K. Hiramath and G. C. Deshpande, "Fake news detection using deep learning techniques," in *2019 1st International Conference on Advances in Information Technology (ICAIT)*, 2019, pp. 411–415.

[120] M. Granik and V. Mesyura, "Fake news detection using naive bayes classifier," in *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, 2017, pp. 900–903.

[121] J. Shaikh and R. Patil, "Fake news detection using machine learning," in *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*, 2020, pp. 1–5.

[122] R. Sharma, V. Agarwal, S. Sharma, and M. S. Arya, "An lstm-based fake news detection system using word embeddings-based feature extraction," in *ICT Analysis and Applications*. Springer, 2021, pp. 247–255.

[123] R. K. Kaliyar, A. Goswami, P. Narang, and S. Sinha, "Fndnet–a deep convolutional neural network for fake news detection," *Cognitive Systems Research*, vol. 61, pp. 32–44, 2020.

[124] X. Dong, S. Sarker, and L. Qian, "Integrating human-in-the-loop into swarm learning for decentralized fake news detection," in *2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*. IEEE, 2022, pp. 46–53.

[125] S. Ouyang, J. Du, B. Wang, B. Yu, Y. Wang, and M. Liang, "Federal learning based covid-19 fake news detection with deep self-attention network," in *2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS)*, 2021, pp. 296–299.

[126] R. Hu, Y. Gong, and Y. Guo, "Federated learning with sparsified model perturbation: Improving accuracy under client-level differential privacy," *arXiv preprint arXiv:2202.07178*, 2022.

[127] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, "When Machine Learning Meets Privacy: A Survey and Outlook," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–36, Mar. 2021.

[128] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Comput. Ind. Eng.*, vol. 149, p. 106854, Nov. 2020.

[129] W. Wang, Y. Yang, Z. Yin, K. Dev, X. Zhou, X. Li, N. M. F. Qureshi, and C. Su, "Bsif: Blockchain-based secure, interactive, and fair mobile crowdsensing," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 12, pp. 3452–3469, 2022.

[130] W. Yang, Y. Zhang, K. Ye, L. Li, and C.-Z. Xu, "FFD: A Federated Learning Based Method for Credit Card Fraud Detection," in *Big Data – BigData 2019*. Cham, Switzerland: Springer, Jun. 2019, pp. 18–32.

[131] P. Tiwari, A. Lakhan, R. H. Jhaveri, and T.-M. Gronli, "Consumer-centric internet of medical things for cyborg applications based on federated reinforcement learning," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2023.

[132] A. R. Javed, M. A. Hassan, F. Shahzad, W. Ahmed, S. Singh, T. Baker, and T. R. Gadekallu, "Integration of blockchain technology and federated learning in vehicular (iot) networks: A comprehensive survey," *Sensors*, vol. 22, no. 12, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/12/4394

[133] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, "LDP-Fed: federated learning with local differential privacy," in *EdgeSys '20: Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 61–66.

[134] C. Ma, J. Li, M. Ding, H. H. Yang, F. Shu, T. Q. S. Quek, and H. V. Poor, "On Safeguarding Privacy and Security in the Framework of Federated Learning," *IEEE Network*, vol. 34, no. 4, pp. 242–248, Mar. 2020.

[135] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, "Protection Against Reconstruction and Its Applications in Private Federated Learning," *arXiv*, Dec. 2018.

[136] 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.

[137] S. Chen, D. Yu, Y. Zou, J. Yu, and X. Cheng, "Decentralized wireless federated learning with differential privacy," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6273–6282, 2022.

[138] X. Zhang, X. Chen, M. Hong, Z. S. Wu, and J. Yi, "Understanding clipping for federated learning: Convergence and client-level differential privacy," in *International Conference on Machine Learning, ICML 2022*, 2022.

[139] J. Wang, S. Guo, X. Xie, and H. Qi, "Protect privacy from gradient leakage attack in federated learning," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 580–589.

[140] "Kaggle fake news dataset," https://www.kaggle.com/competitions/fake-news/data.

[141] V. M. Krešňáková, M. Sarnovský, and P. Butka, "Deep learning methods for fake news detection," in *2019 IEEE 19th International Symposium on Computational Intelligence and Informatics and 7th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Sciences and Robotics (CINTI-MACRo)*, 2019, pp. 000 143–000 148.