DESIGN AND DEVELOPMENT OF SECURE FEDERATED LEARNING FRAMEWORK FOR DATA-SENSITIVE APPLICATIONS

Thesis submitted to

Delhi Technological University

for the award of the degree of

DOCTOR OF PHILOSOPHY

in

Information Technology

by

Manu Narula

(2K21/PHDIT/02)

Under the Supervision of

Prof. Dinesh Kumar Vishwakarma

&

Dr. Jasraj Meena

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly known as Delhi College of Engineering)

NEW DELHI-110042, INDIA

(FEB, 2026)

Department of Information Technology
Delhi Technological University, Delhi
Delhi, INDIA - 110042

Candidate's Declaration

I hereby declare that the work being presented in this thesis entitled "**Design and Development of Secure Federated Learning Framework for Data-Sensitive Applications**" in partial fulfillment of the requirements for the award of the degree of **Doctor of Philosophy** and submitted in the Department of Information technology of the Delhi Technological University, Delhi is an authentic record of my own work carried out during a period from August 2021 to January 2026 under the supervision of **Prof.Dinesh Kumar Vishwakarma**, Head, Department of Information Technology, Delhi Technological University, Delhi and **Dr.Jasraj Meena**, Associate Professor, School of Computer and System Sciences, Jawaharlal Nehru University, New Delhi.

I have not submitted the matter presented in this thesis for the award of any other degree at this or other organization.

(Manu Narula)
(Regn. No. 2K21/PHDIT/02)

This is to certify that the above statement made by the candidate is true to the best of our knowledge and belief.

Dr. Jasraj Meena
(Joint-Supervisor)
Associate Professor
School of Computer and System Sciences
Jawaharlal Nehru University, Delhi, India

Prof. Dinesh Kumar Vishwakarma
(Supervisor)
Head
Department of Information Technology,
Delhi Technological University, Delhi, India

Place: Delhi, INDIA
Date: February 2026

Department of Information Technology
Delhi Technological University, Delhi
Delhi, INDIA - 110042

Certificate

It is to certify that the work contained in this thesis entitled "**Design and Development of Secure Federated Learning Framework for Data-Sensitive Applications**" submitted by Manu Narula (2K21/PHDIT/02) for the award of the degree of **Doctor of Philosophy** at Delhi Technological University, Delhi, India contains original research work conducted under our guidance and supervision.

We attest that Manu Narula has successfully met all the stipulated requirements and standards for the submission of the thesis. The research work presented in this thesis is original, as far as we know, and has not been utilized as a foundation for the conferral of any other degree or similar recognition.

We, as supervisor and joint supervisor, affirm the authenticity and originality of the research work conducted by Mr. Manu Narula and presented in this thesis.

Dr. Jasraj Meena
(Joint-Supervisor)
Associate Professor
School of Computer and System Sciences
Jawaharlal Nehru University, Delhi, India

Prof. Dinesh Kumar Vishwakarma
(Supervisor)
Head
Department of Information Technology,
Delhi Technological University, Delhi, India

Place: Delhi, INDIA
Date: February 2026

Dedicated to our deities and my family

ACKNOWLEDGMENT

The completion of this doctoral thesis has been a challenging yet immensely rewarding journey. With profound gratitude and appreciation, I take this opportunity to acknowledge all those who have contributed, in various capacities, to the successful completion of this work.

I would first like to express my deepest gratitude to my thesis supervisor, **Prof. Dinesh Kumar Vishwakarma**, and Joint-Supervisor **Dr. Jasraj Meena**, whose continuous guidance, insightful feedback, and unwavering support have been invaluable to the development of this thesis. Their expertise and dedication to my academic progress have been a constant source of inspiration. I am sincerely grateful for the countless hours they devoted to reviewing my work, providing constructive criticism, and motivating me to strive for higher standards. Their encouragement and belief in my abilities, even during the most challenging times, were instrumental in bringing this thesis to completion.

I would also like to extend my heartfelt thanks to the former Head of the Department, **Prof. Kapil Sharma**, for his time, thoughtful suggestions, and continued support. His constructive feedback significantly enhanced the quality of this research, and I have benefited greatly from his expertise.

On a personal note, I express my deepest appreciation to my family for their unwavering love, patience, and encouragement throughout this journey. I am especially grateful to my parents, **Mr. Manish Narula** and **Mrs. Aikta Narula**, my sister **Manvi Narula**, and my spouse **Anushka Narula** for their constant emotional and intellectual support. Their encouragement and belief in my potential have been a continuous source of motivation.

Finally, I would like to thank my friends **Anita, Meenakshi, Shivani, and Lakshita**, along with all those who may not be mentioned individually but have contributed in various ways. Whether through intellectual discussions, moral support, or acts of kindness, your contributions are deeply appreciated.

February, 2026 Manu Narula

ABSTRACT

In an era where security breaches and cyber fraud have become increasingly common, concerns about data privacy have also risen exponentially. This presents a substantial hurdle for training Artificial Intelligence (AI) applications that handle sensitive data, such as healthcare and finance. Federated Learning (FL) provides a solution for Data-Sensitive Applications (DSA), enabling collaboration between competing parties while guaranteeing complete control over data privacy. Over time, the attacks on FL have evolved, and the native technology alone is insufficient to maintain the expected privacy levels when handling such data. Hence, efforts are made to integrate other technologies with FL to enhance certain aspects, such as resource management, security, privacy, and efficiency, with varying results and pros and cons.

This thesis examines the integration of security mechanisms with FL, focusing on networks with resource scarcity, which include AdHoc networks, Internet of Things, and Internet of Medical Things (IoMT), commonly found in the field of healthcare and other DSA, addressing critical issues such as data privacy, security, access control, and scalability. It proposes solutions to enhance data security and privacy in healthcare systems by examining various existing studies and frameworks.

FL is adopted for its ability to facilitate the integration of AI in domains where conventional centralized training approaches are infeasible due to data distribution constraints, privacy concerns, or competitive restrictions. Furthermore, FL enables remote or decentralized systems to achieve performance comparable to state-of-the-art centralized methods while incurring minimal resource overhead. By eliminating the need for centralized data aggregation, FL significantly mitigates the risks associated with large-scale data breaches and unauthorized access.

This thesis makes a substantial **contribution** in summarizing the existing state of FL, its various types, supporting technologies, and challenges, designing hybrid security frameworks for FL aimed at DSA networks, and performing extensive evaluation of the state-of-the-art FL techniques. Our contributions are as follows:

- An extensive systematic literature review has been conducted to investigate the cur-

rent state of FL implementations in DSA fields, such as healthcare and finance. This analysis highlights the complex challenges of implementing FL in mainstream healthcare applications, including concerns about transmission costs, data security, privacy, and data/system heterogeneity. Additionally, we provide a detailed taxonomy for the existing literature, focusing on DSA. Alongside identifying the gaps, the review also highlights the fundamental challenges that FL may pose in real-world scenarios.

- As organizations and developers explore FL solutions for various applications, the multitude of FL tools and frameworks can feel overwhelming for a beginner. We studied various tools and frameworks available in both open and proprietary forms to provide a concise view of their advantages, limitations, and utility, thereby filling a gap in the proposed literature.

- It contributes to improving the security in resource-constrained DSA networks such as IoMT by employing quantization. Unlike prior techniques, we incorporate a workload-aware client selection scheme to overcome the quantization loss, optimize bandwidth, and provide stability to the training network by minimizing the straggler nodes with negligible performance overhead. Thus, providing a reliable security solution to resource-constrained DSA. The experimental results show a 77% to 95% decrease in straggler nodes and an 8% to 0% decrease in accuracy compared to standard FL, depending on the dataset's complexity.

- To ensure security in small edge networks, IoMT networks, and AdHoc networks, a dynamic, lightweight cryptographic FL framework is proposed. Conventional cryptographic techniques are computationally expensive for small networks. Thus, we propose a block cipher with an ever-changing key that takes multiple client and network attributes into account and is unique to all participants. The overhead induced by the cipher and key generation is significantly lower than that of state-of-the-art solutions available. The experimental evaluation indicates that the proposed scheme's relative performance is comparable to that of native FL. The results reveal an increase of approximately 4.5% to 8% in per-round computation time compared to standard FL. However, this marginal computational overhead is justified by the significantly enhanced security guarantees provided by the proposed approach, particularly in resource-constrained network environments, where conventional security mechanisms with similar computational budgets offer comparatively lower protection

- To ensure privacy in networks with sufficient resources, we propose a hybrid FL

framework that utilizes differential privacy in conjunction with homomorphic encryption. To counter the loss incurred by the differential privacy scheme, we employ the workload-aware client selection. The simulation results demonstrate minimal computational overhead and comparable performance to other state-of-the-art techniques. The proposed Fed-HDVE imposes a minimal time overhead of 2.8 to 3.5 seconds, depending on the model size.

The performance evaluation, analysis, and experimental results indicate that the proposed solutions offer a viable and effective security solution. Moreover, the comparative study demonstrates that the suggested approaches perform on par with existing solutions, with lower computational overhead and resource consumption.

Contents

Declaration iii

Certificate v

Dedication vii

Acknowledgments ix

Abstract x

List of Acronyms/Abbreviations xvii

List of Tables xix

List of Figures xxi

List of Algorithms xxiv

1 Introduction 1

1.1 Background . . . 1

1.1.1 Terminology and Security Objectives . . . 2

1.1.2 Data-Sensitive Application . . . 2

1.1.3 Federated Learning . . . 3

1.2 Federated Learning: Concepts and Preliminaries . . . 4

1.2.1 Machine Learning and Deep Learning . . . 5

1.2.2 Distributed Learning . . . 6

1.2.3 Active Learning (AL) . . . 7

1.2.4 Advent of Federated Learning (FL) . . . 8

1.3 Impact of FL on DSA . . . 14

1.4 Issues and Challenges of Federated Learning . . . 18

1.4.1 Basic Attacks and Privacy Concerns in FL . . . 19

1.4.2 Transmission Costs . . . 21

1.4.3 System Heterogeneity . . . 22
1.4.4 Data Heterogeneity . . . 22
1.5 Motivation . . . 23
1.6 Problem Statement . . . 24
1.6.1 Research Objectives . . . 24
1.6.2 Threat Model and Adversarial Assumptions . . . 24
1.6.3 Contribution in the Thesis . . . 24
1.7 Thesis Outline . . . 25

2 Literature review **29**
2.1 Federated Learning Implementations with Supporting Technologies . . . 31
2.2 Analysis of FL Implementations in Data-Sensitive Applications . . . 33
2.2.1 FL in Healthcare . . . 35
2.2.2 FL in Finance . . . 46
2.2.3 Potential Upgrades for FL in DSA . . . 49
2.2.4 Tools and Frameworks . . . 56
2.2.5 Analysis of Dataset Availability for DSAs . . . 59
2.2.6 Real-world Case Studies of FL . . . 61
2.2.7 Research Gaps and Problem Analysis . . . 62

3 Workload-aware secure federated learning framework with quantization **65**
3.1 Introduction . . . 65
3.1.1 Incentive-Aware FL (IFL) . . . 66
3.1.2 Major Issues and Challenges identified with quantization based FL . . . 67
3.2 Related Work in Quantization and Incentive-Aware FL . . . 68
3.2.1 Quantization . . . 68
3.2.2 Incentive-Aware FL and Workload-Aware FL . . . 69
3.3 Proposed Quantization-based Hybrid Frameworks . . . 70
3.3.1 Dynamic Resource-Aware Federated Framework for Secure and Sustainable Learning in Data Sensitive Applications (DRAFFSS) . . . 70
3.3.2 Federated Workload-Aware Quantized Framework for Secure Learning in Data-Sensitive Applications (Fed-WALQ) . . . 80
3.4 Summary of the Chapter . . . 97

4 Secure federated learning framework with dynamic lightweight ciphers for resource-constrained data-sensitive applications **99**

4.1 Introduction . . . 99
4.1.1 Scalability and utility of Fed-DLC . . . 100
4.2 Related Work . . . 101
4.2.1 FL in the Internet of Medical Things . . . 101
4.2.2 Lightweight Ciphers for FL-Based Techniques . . . 102
4.3 Proposed Framework (Fed-DLC) . . . 103
4.3.1 Dynamic Lightweight Cipher . . . 103
4.3.2 Fed-DLC . . . 110
4.3.3 Deep dive in Key Generation and Encryption . . . 112
4.3.4 Resistance to Known-Plaintext Attack (KPA) . . . 115
4.3.5 Resistance to Chosen-Plaintext Attack (CPA) . . . 115
4.4 Experimental Results and Analysis . . . 116
4.4.1 Baseline Models . . . 117
4.4.2 Experimental Setup for Fed-DLC Simulation . . . 118
4.4.3 Dataset used for Fed-DLC Simulation . . . 118
4.4.4 Results of Simulations of Fed-DLC . . . 119
4.5 Summary of this Chapter . . . 127

5 Secure federated learning framework with differential privacy and homomorphic vector encryption 129

5.1 Introduction . . . 129
5.1.1 Differential Privacy . . . 130
5.1.2 Homomorphic Encryption . . . 131
5.1.3 Assumptions for Scalable Real-World Implementation of Fed-DHVE 131
5.2 Related Work in differential privacy and homomorphic encryption with federated learning . . . 132
5.2.1 Differential Privacy with Federated Learning . . . 132
5.2.2 Homomorphic Encryption with Federated Learning . . . 132
5.2.3 Hybrid Approaches (Differential Privacy and Homomorphic Encryption) . . . 133
5.3 Proposed Framework (Fed-DHVE) . . . 133
5.3.1 Standard FL algorithm . . . 134
5.3.2 Differential Privacy-based security with Federated Learning . . . 134
5.3.3 Conjugate Search Problem - Homomorphic Encryption . . . 135
5.3.4 Fed-DHVE . . . 137

5.3.5 Privacy and Leakage Discussion 138
5.4 Experimental Results and Analysis for Fed-DHVE Simulation 139
5.4.1 Baseline Model for simulation of Fed-DHVE 139
5.4.2 Experimental setup and datasets used for Fed-DHVE Simulation . . 140
5.4.3 Results of Fed-DHVE Simulation 140
5.5 Summary of the Chapter 142

6 Conclusion and Future Scope **143**
6.1 Conclusion 143
6.2 Future Directions 144

List of Publications **146**

Bibliography **147**

Proof of Publication **181**

Plagiarism Verification **187**

Curriculum Vitae **188**

List of Acronyms/Abbreviations

AL	Active Learning
AI	Artificial Intelligence
DSA	Data-Sensitive Applications
DFL	Decentralized Federated Learning
DL	Deep Learning
DP	Differential Privacy
DDP	Distributed Differential Privacy
Dist. L.	Distributed Learning
DRAFFSS	Dynamic Resource-Aware Federated Framework for Secure and Sustainable Learning in DSA
EFDLS	Efficient Federated Distillation Learning System
FL	Federated Learning
Fed-DHVE	Federated Learning Framework with Differential Privacy and Homomorphic Vector Encryption
Fed-DLC	Federated Learning Framework with Dynamic Lightweight Cipher
FTL	Federated Transfer Learning
Fed-WALQ	Federated Workload-Aware Quantized Framework for Secure Learning in DSA
FDMA	Frequency Division Multiple Access
HE	Homomorphic Encryption
HFL	Horizontal Federated Learning
IFL	Incentive-Aware Federated Learning
IID	Independent and Identically Distributed
IIoT	Industrial Internet of Things
IoMT	Internet of Medical Things
IoT	Internet of Things
IoV	Internet of Vehicles
IPFS	InterPlanetary File System
LDP	Local Differential Privacy
ML	Machine Learning

MPC	Multi-Party Computing
PFL	Personalized Federated Learning
PSI	Private Set Intersection
RL	Reinforcement Learning
RFL	Resource-Aware Federated Learning
TL	Transfer Learning
UAV	Unmanned Aerial Vehicle
VFL	Vertical Federated Learning

List of Tables

1.1 Comparison of FL with Traditional ML and DL 14
1.2 Security techniques frequently used with FL 18

2.1 Summary of existing surveys for federated learning 30
2.2 FL Implementations with Supporting Technologies 34
2.3 Recent Advances of FL in Adaptive Healthcare and Medical Imaging 37
2.4 Recent Advances of FL in IoT and IoMT-Enabled Healthcare 40
2.5 Recent Advances of FL in IoT and IoMT-Enabled Health Monitoring 41
2.6 Recent Advances of FL in MRI-based diagnosis and Covid-19 diagnosis. . . . 43
2.7 Recent Advances of FL in Critical Illness Diagnosis. 45
2.8 Recent Advances of FL in Critical Illness Diagnosis and Misc. 47
2.9 Recent Advances of FL in the Finance Sector 50
2.10 Adoptable Techniques from FL Implementations suitable for adoption in DSA for Resource and Model Optimization. 53
2.11 Adoptable Techniques from FL Implementations suitable for adoption in DSA for Incentive Aware Applications 54
2.12 Adoptable Techniques from FL Implementations suitable for adoption in DSA for Privacy Enhancement and Utilities. 55
2.13 Adoptable Techniques from FL Implementations suitable for adoption in DSA for Gradient and Model Security. 57
2.14 Popular Frameworks for FL integration 59
2.15 Characteristics of Public Datasets in Healthcare Corresponding to DSA . . . 60

3.1 Datasets used for Experimentation of Fed-WALQ. 77
3.2 Results achieved by DRAFFSS in various Datasets 77
3.3 Performance comparison of DRAFFSS against Fed-MSQE and Standard FL over the two medical datasets 79
3.4 Experimental Setup 88
3.5 Datasets used for Experimentation of Fed-WALQ. 90

3.6 Accuracy and Straggler Node Percentage achieved by Fed-WALQ over Different Datasets 91
3.7 Accuracy Achieved by Standard FL, FED-MSQE, and FED-WALQ Over Different Datasets 94
3.8 Average Straggler Node Percentage Observed (Lower is Better) 96

4.1 Performance of Fed-DLC over Various Datasets 119
4.2 Performance Comparison of Fed-DLC against state-of-the-art techniques in ResNet-20 Model 120
4.3 Performance Comparison of Fed-DLC against Standard FL, Fed-MSQE, VerifyNet, and Turbo-Aggregate in ResNet-50 Model 125
4.4 Performance Comparison of Fed-DLC against Standard FL, Fed-MSQE, VerifyNet, and Turbo-Aggregate in InceptionV3 Model 126

5.1 Performance of Fed-DHVE over Various Datasets 140
5.2 Performance Comparison of Fed-DHVE against state-of-the-art techniques . 141

List of Figures

1.1 Cloud/Edge-based Federated Learning . . . 3
1.2 Types of Learning in Artificial Intelligence . . . 5
1.3 Data distribution in Horizontal Federated Learning . . . 9
1.4 Data distribution in Vertical Federated Learning . . . 9
1.5 Data distribution in Federated Transfer Learning . . . 10
1.6 Data Distribution in Confederated Learning . . . 10
1.7 General Architecture of Collaborative Horizontal Federated Learning for Data-Sensitive Applications . . . 17
1.8 General Architecture of Collaborative Vertical Federated Learning for Data-Sensitive Applications . . . 17
1.9 General Architecture of Collaborative Transfer Federated Learning for Data-Sensitive Applications . . . 18

2.1 Proposed Taxonomy . . . 63

3.1 Homogeneous FL setup with Identical Data Distribution across Clients . . . 67
3.2 Heterogeneous FL setup with Different Data Distribution across Clients . . . 67
3.3 Architecture of Proposed Framework (DRAFFSS) . . . 71
3.4 Architecture of the baseline ResNet-18 model . . . 76
3.5 Sample images from Diabetic Retinopathy Dataset . . . 76
3.6 Sample images from RSNA Bone Age Dataset . . . 77
3.7 Test Accuracy Comparison for Diabetic Retinopathy Dataset . . . 78
3.8 Comparison of straggler nodes present during training over Diabetic Retinopathy Dataset . . . 78
3.9 Test Accuracy Comparison for RSNA Bone Age Dataset . . . 79
3.10 Comparison of straggler nodes present during training over RSNA Bone Age Dataset . . . 79
3.11 Proposed architecture of Fed-WALQ . . . 80
3.12 Example of workload allotment process of Algorithm 3.6 . . . 84
3.13 Architecture of the baseline Resnet 18 model used in Fed-WALQ . . . 88

3.14 Sample images from Datasets used in experiments 90
3.15 Legend for figures 3.16 and 3.17 93
3.16 Comparison of Test Accuracy of Fed-WALQ, Fed-MSQE, and Standard FL over various datasets during training in 3, 5, and 7-bit quantization. 93
3.17 Comparison of Straggler Node % observed in Fed-WALQ, Fed-MSQE, and Standard FL over various datasets during training in 3, 5, and 7-bit quantization. 94
3.18 Legend for Figures 3.19 and 3.20 95
3.19 Comparison of Test Accuracy of Fed-WALQ, Fed-MSQE, and Standard FL over various datasets during training in 4 and 6-bit quantization. 95
3.20 Comparison of Straggler Node % observed in Fed-WALQ, Fed-MSQE, and Standard FL over various datasets during training in 4 and 6-bit quantization. 96

4.1 Proposed Architecture of Fed-DLC. 103
4.2 Key matrix at different stages in Key_Gen method. 113
4.3 64X4 Key matrix 113
4.4 Key matrix at different stages in Key_Gen method. 113
4.5 Demo run of encryption process. 114
4.6 The architectures of ResNet-20 and ResNet-50 models. 117
4.7 The architectures of the InceptionV3 model. 118
4.8 Confusion Matrices for Fed-DLC applied over ResNet-20 over Various Datasets. 121
4.9 Accuracy Over Epoch Graphs for Fed-DLC in ResNet-20 Model over Different Datasets. 121
4.10 Loss Over Epoch Graphs for Fed-DLC in ResNet-20 Model over Different Datasets. 121
4.11 Confusion Matrices for Fed-DLC applied over ResNet-50 over Various Datasets. 122
4.12 Accuracy Over Epoch Graphs for Fed-DLC in ResNet-50 Model over Different Datasets. 122
4.13 Loss Over Epoch Graphs for Fed-DLC in ResNet-50 Model over Different Datasets. 122
4.14 Confusion Matrices for Fed-DLC applied over InceptionV3 over Various Datasets. 123
4.15 Accuracy Over Epoch Graphs for Fed-DLC in InceptionV3 Model over Different Datasets. 123
4.16 Loss Over Epoch Graphs for Fed-DLC in InceptionV3 Model over Different Datasets. 123

5.1 Proposed Architecture of Fed-DHVE. 134

List of Algorithms

3.1 Quantization of Parameters in FL Setup (Fed-q) 72
3.2 Workload Prediction (Round_Load) 74
3.3 DRAFFSS 75
3.4 Quantizing Function (Qbound) 82
3.5 Federated Quantization of Parameters (Fquantize) 83
3.6 Client Workload Prediction (Pred_Epoch) 84
3.7 Server_Side_Client_Selection 86
3.8 Fed-WALQ 87
4.1 Key_Gen 105
4.2 Dynamic_Cipher_Encrypt 107
4.3 Dynamic_Cipher_Decrypt 109
4.4 Fed-DLC 111
5.1 Differentially Private Federated Learning 135
5.2 CSP-based Homomorphic Encryption in Federated Learning 136
5.3 Fed-DHVE 137

Chapter 1

Introduction

The purpose of this chapter * *is to provide background information about federated learning, data-sensitive applications, and an overall introduction to the thesis. Also, the chapter explains the motivation, objectives, and structure of the thesis.*

1.1 Background

With the emergence of Artificial Intelligence (AI), there has been a growth of inconceivable standards in technology affecting nearly every field, however minuscule or vast it may be [1]. AI applications require significant data to train Machine Learning (ML) and Deep Learning (DL) models for accurate results. In both, the data is stored centrally in one place or shared/copied to support parallel training. This meant that all the data, directly or indirectly, was controlled by a third party, which may or may not be trusted by the data owners. This was not a major concern for applications dealing with general or casual data like e-commerce, recommender systems, etc. However, it became a respectable concern for the applications that dealt with sensitive data like finance, healthcare, surveillance systems, etc. Moreover, the data involved with these applications are either in much more significant amounts that cannot be stored centrally or in the form of isolated data islands. It became a pivotal issue and limited the acceptance of ML and DL in these applications. This is because when ML or DL is applied over scattered data, the amount of data available for the training process at one location is insufficient for a good accuracy output. Since Data-Sensitive Applications (DSA) frequently deal with scattered data islands, ML and DL are of limited utility.

*The part of this chapter has been published/accepted/communicated in: Manu Narula, Jasraj Meena, and Dinesh Kumar Vishwakarma, "A comprehensive review on Federated Learning for Data-Sensitive Application: Open issues & challenges," *Engineering Applications of Artificial Intelligence*, Elsevier, Volume 133, Part B, pp. 108128, 2024, DOI: `https://doi.org/10.1016/j.engappai.2024.108128`. **(SCIE, IF=8.0)**.

1.1.1 Terminology and Security Objectives

In this thesis, the terms *privacy*, *security*, and *protection* are used with distinct meanings to ensure consistency across the proposed frameworks. *Privacy* refers to preventing the reconstruction or inference of sensitive training data from gradients, model parameters, or aggregated updates, primarily achieved through Differential Privacy mechanisms. *Security* refers to safeguarding the federated learning pipeline against adversarial actions, such as eavesdropping, model interception, and malicious modification of updates, by leveraging lightweight cryptography and homomorphic encryption. *Protection* is an operational umbrella term that describes the practical integration of privacy and security mechanisms in heterogeneous, resource-constrained environments. Accordingly, the key objectives of this work are to ensure confidentiality of model updates, resistance to inference attacks, robustness against semi-honest or malicious participants, and scalable deployment with minimal computational and communication overhead.

1.1.2 Data-Sensitive Application

A data-sensitive application or data-sensitive field is conceptualized as a set of implementations that depend on data of a highly confidential nature, such as data related to the identity or social footprint of individuals [2]. In addition to this, in most of these deployments, accuracy takes center stage, and the slightest variation in data processing can result in invalid outcomes. Some of the works in DSA are Cerebral ataxia diagnosis [3], Fraudulent transaction detection [4, 5], Tumor characterization [6, 7], etc. A major challenge when dealing with DSA is that it is difficult to apply direct encryptions within the training and analysis process without affecting performance, which renders such measures inappropriate. Another challenge in these applications is that one entity cannot have sufficient data for good-quality training in practical environments. The concerned institutions are largely resistant to collaborating, given the nature of the data involved and the competitive nature amongst them.

To cater to the demands of DSA, Federated Learning (FL) came out as the top contender for private training and analysis. The existence of FL is credited to the event when Google put forward a proposal to use local model parameters instead of user data to train a globally acceptable model back in 2016 [8]. Since then, it became a staple for Google to integrate FL-based content suggestions and predictions in its ventures, such as Android [9, 10]. Apple, in 2017, implemented FL to predict power-hungry or storage-hungry websites and emoji

suggestions while typing on its devices. Since then, the technique has inspired many improvements, modifications, and a varied set of applications, especially in fields that depend on sensitive data.

1.1.3 Federated Learning

FL can be regarded as a modified form of Distributed Learning due to similar architectures but different approaches for transmission and training. Distributed Learning creates a web of nodes to enforce high-performance computing by implementing parallel computation [11] working over the same or cloned datasets. FL acknowledges that the distributed nature of data sources enables training over multiple isolated entities. Another significant difference between Distributed Learning and FL is that the former does not implement any mechanism to handle data heterogeneity.

Fig. 1.1 shows the basic organization of an FL setup. Here, multiple client devices (mobile, laptop, tablet) communicate with a cloud/edge server to synchronize model parameters. These parameters are used to carry out localized training on client devices. The updated set of parameters is uploaded back to the server for aggregation into the central model. The process is repeated in multiple rounds, each time with a randomly selected subset of devices.

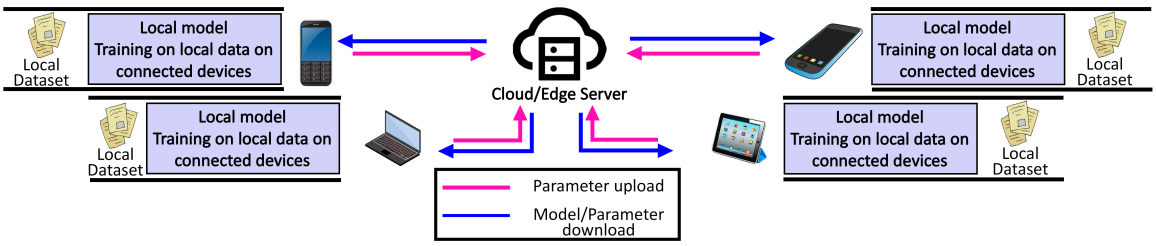


Figure 1.1: Cloud/Edge-based Federated Learning

Moving to the privacy aspect, FL eliminates the need for sharing user data by incorporating user devices as computation nodes. It also enables the simultaneous training of multiple models through the periodic transmission of model parameters across the network. But, FL too, has its share of challenges [12–17] that stem from the integration of a network-based architecture. These often include:

- Inference and inversion attacks on the network level and the model itself.
- Practical problems like transmission overheads, synchronization issues.
- Bias in model weights due to heterogeneity in configurations and datatypes involved [18–23].

- and minute performance concerns due to data poisoning and model poisoning attacks.

To tackle these, FL is sometimes layered with another technology (block-chain [24–33], Edge computing [33–43], etc.) for additional security at potential breach points. The extent of this layering is decided by the degree of vitality of the data in question.

Security can be implemented via FL in almost every ML and DL-based application [4, 44, 45]. However, DSA, like health care and finance, provides ample ground for it to flourish due to the sheer amount of data generated and the highly confidential nature associated with it. Hence, even with the associated transmission overheads and some unique threats, FL provides a viable solution for training. The integration of FL in DSA has seen an exponential rise in recent years and is rapidly expanding with new, hybrid, and modified implementation ideas. We provide a summarized view of this expanse along with the various supporting technologies and the future possibilities that open with each addition. Some authors discussed advanced FL. The works of Xing et al. [46] introduced advanced forms of FL and proposed an efficient federated distillation learning system (EFDLS) for multi-task scenarios in time-series classification and displayed improvements in communication efficiency while maintaining on-par accuracy with state-of-the-art FL implementations.

Renda et al. [47] and Luis et al. [48] studied the utility of FL toward explainable AI. The former highlighted its utility in vehicular networks using advanced 5G and 6G systems and the latter investigated FL's role in providing data security in general explainable ML and DL models.

1.2 Federated Learning: Concepts and Preliminaries

Over the years, AI has seen many technological advances which helped in its widespread application. The extent of these applications is seen in computer vision implementations like self-driving cars, facial recognition systems, intrusion detection, and real-time image processing. FL is the latest technology in the gradual evolution of learning algorithms under AI. It is essential to understand the necessity and tradeoff of the said technique and the core differences from distributed learning. Hence, in this section, we discuss, in brief, the various innovations and their drawbacks that led to the development of FL. We also discuss the basic blocks and various types of FL in detail before moving on to their utility towards DSA in the next section. The various types and sub-types of learning in AI in

existence are depicted in Fig. 1.2.

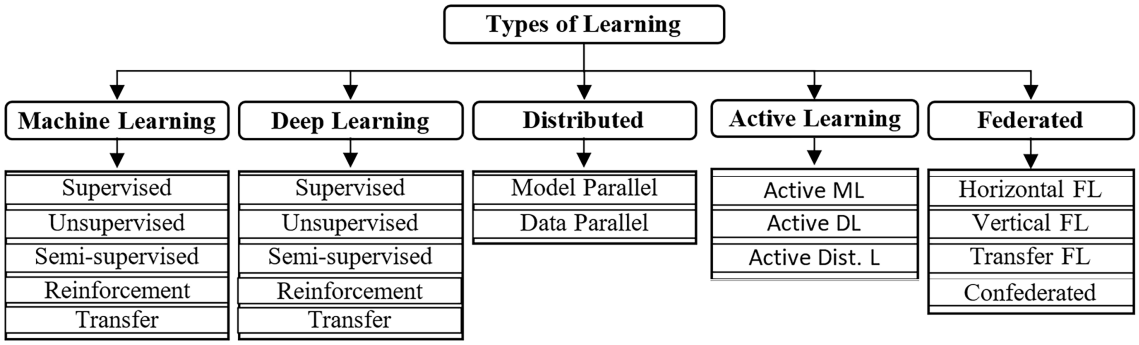


Figure 1.2: Types of Learning in Artificial Intelligence

1.2.1 Machine Learning and Deep Learning

The emergence of AI was marked by algorithms that enabled computers to make active decisions. The collective set of such algorithms and models was termed ML [49–51]. The paradigm quickly caught a lot of attention. ML was soon found in almost all conceivable applications, ranging from automated factories to disease identification [52, 53]. The most popular applications of ML came out in the form of computer vision, which enabled both passive and real-time image processing for various tasks. These include automated vehicles, recognition systems, and feature extractors, which implement a variety of ML training techniques with known outputs (Supervised), Unknown outputs for regressive workloads (Unsupervised), or a mix of both (Semi-supervised). Despite having a large application base, ML is restricted to a linear approach of if-then-else. Additionally, ML requires direct human interaction for proper functioning throughout the process, especially in feedback-based implementations of reinforcement learning. Thus, the major issue faced by ML is the absence of evolution or self-learning. Additionally, frequent human interaction becomes challenging when the associated data is vast, which limits its utility in DSA.

To tackle this issue, neural networks have been developed that mimic the functioning of the human brain. The ML algorithms focused on using these networks are termed DL algorithms [54]. These provided a way to introduce flexibility and adaptability in the training and analysis process and required less or no human interaction in several applications (Cyber Security, Feature Matching, Object Detection, etc.) [55–62]. Although a significant improvement over ML, DL introduced the need for massive amounts of data to achieve desirable results. The computational power required for the process also increased with the complexity of both the algorithm and the neural networks. This, in turn, shot up the time

spent training a model of appreciable accuracy, along with some other minor issues [63]. Though the complexities can be reduced using transfer learning models, given the nature of the data involved and the resource-constrained scenarios encountered in many DSAs, the utility of DL is strictly limited.

1.2.2 Distributed Learning

The cost presented by DL algorithms in terms of data, computation, and time led the researchers to actively focus on finding alternative approaches. The concept of Distributed Learning presented a positive opportunity to achieve both high computation power and low training periods. It does so by allocating multiple devices tied by the parallel processing paradigm, updating a standard central model [64]. This is implemented in either of the two possible ways, depending on the model used (model parallel and data parallel).

In the data parallel approach, the dataset is split into smaller sets of equal size. The data members are randomly chosen from the original set to maintain the quality of input for all subsets. These subsets are then assigned to nodes (devices), which in turn train the central model using the provided data. The results obtained by all nodes are aggregated, and the model is updated at regular intervals. The drawback of using this method is that the dataset involved must be sufficiently large to be split into subsets of appreciable dimensions. Failing this, each node will have a low amount of data, and the overall quality of training may suffer [64].

On the other hand, in the model-parallel approach, the model is split into distinctly trainable entities or sub-models. Here, the dataset is cloned to every node for training. Every node is allotted a sub-section of the central model for training. Hence, different sections of the model are trained simultaneously. This approach eliminates the need for the dataset to be enormously large. However, according to our study, not many models can be split in the required ways [64].

Apart from approach-based issues and requirements, Distributed Learning fails to consider the practical challenges of system and data heterogeneity. System heterogeneity occurs when the devices in the network have varying capabilities in terms of storage, processing power, bandwidth, and response delays. This mismatch in hardware is not efficiently handled in most Distributed Learning implementations and hence is still a field of research [50]. Data heterogeneity [18, 65, 66], on the other hand, is a result of how the data is processed on individual nodes. Data heterogeneity is not encountered by distributed learning, as the

same dataset is used on each node. Hence, the issue remains unaddressed at any level.

The technology offers a significant advantage for clients without high-power setups, enabling them to implement DL directly. However, large-scale deployment faces significant difficulties due to the above-mentioned hurdles. This limits the scale and scope of applications toward DSA. The situation led to the development of a new technology to evaluate the dataset for optimal data-point selection for the training process. This is done through mid-process training of a second model. The technology is popularized as Active Learning (AL).

1.2.3 Active Learning (AL)

The scarcity of pre-identified outputs poses a challenge in many AI algorithms. Hence, AL was developed. AL is a hybrid case of supervised and unsupervised learning. Here, the ratio of labelled data is significantly less than the ratio of unlabeled data. This can be encountered either when the unlabeled data is available inexpensively or when labelling the data is not feasible. AL interacts with the data to predict if the data contributes to improving the model [56, 67, 68]. It employs a two-step training process for this interaction. The first step is to train a model on the limited labelled data to gain an understanding of the model's requirements. In the second step, the learned parameters are used to evaluate entries from the unlabeled pool and assign labels to them. This process of self-labelling and selecting optimal portions of data results in minimizing the size of the required dataset compared to standard ML and DL approaches, attaining the same accuracy. Some examples of AL implementation can be found in the field of smart education [69], handling imbalanced data [70], natural disaster classification [71], biomedical imaging [72], and managing unlabelled data [73, 74].

Although AL provides a more efficient way to implement AI with fewer data points, it remains a temporary or limited solution. As people become more security-aware, along with the rise in cyber-criminal activities, the reluctance to share data is also increasing. This impedes the development process of most AI solutions. Enterprises cannot support the data-hungry nature of these services solely on their own data. Different companies, or in many cases, different departments within the same company, do not typically share data. This, in turn, results in the formation of what can be called isolated data islands. These isolated chunks of data are insufficient for any AI application, even if AL is involved. Since data islands are widespread in DSA, the usability of AL is limited.

1.2.4 Advent of Federated Learning (FL)

The rapidly growing concern for data privacy gave birth to the paradigm of FL. FL is a hybrid form of learning that combines concepts of distributed learning and deep learning, with a greater emphasis on user privacy. Unlike Distributed Learning, no training data is collected or replicated in any form from the clients. Instead, model parameters are shared with all the clients, and localized training is done on client devices. The updated weights from these models are aggregated and integrated into the global model over multiple rounds. Repeating the same process in each round with randomly selected nodes. Since the training data never leaves the end devices, the security quotient rises manifold.

The system heterogeneity issues are also tackled efficiently as the resource requirement associated with Distributed Learning diminishes in the case of FL. Each node device has its private data upon which it trains the model according to its processing power. The result required, that is, the model parameters, are independent of the underlying process in the device. This makes problems with system heterogeneity mostly irrelevant. Data heterogeneity, on the other hand, is taken care of within the architecture adopted for implementing FL, for the concerned scenario, as discussed ahead.

1.2.4.1 Types of FL

Based on the nature of data sourcing and compatibility, FL can be broadly divided into four major categories: Horizontal FL, Vertical FL, Federated Transfer Learning, and Confederated Learning.

1. Horizontal Federated Learning (HFL)

   Horizontal or homogenous FL [75, 76] works upon the assumption that the training data available to different devices are identical in terms of feature set and differ only in sampling method and sample size. For example, several persons purchase medicines from different medical shops, and the shopkeepers maintain a record of sales as Date, Bill no, Person Name, Item Name, Amount. Here, the number of customers for each shop can vary, but the features are the same, qualifying it as horizontal data distribution. This is the most widely used partitioning method for current implementations of FL, as it is much less challenging in practical implementations [19,77,78]. A depiction of the Attribute space between two contributors in horizontal partitioning is shown in Fig. 1.3.

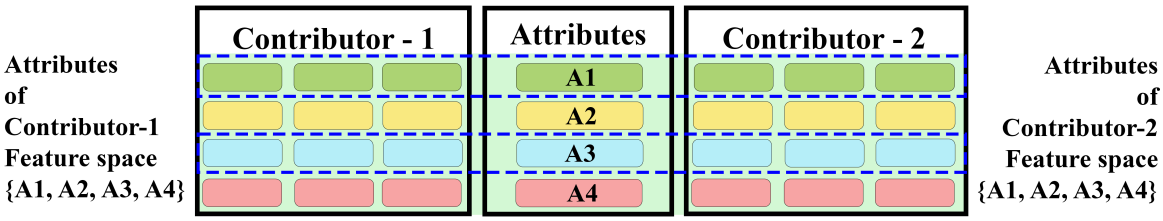


Figure 1.3: Data distribution in Horizontal Federated Learning

2. Vertical Federated Learning (VFL)

   Vertical or heterogeneous FL [16, 79, 80] focuses on data partitioning, which does not sport the same attributes throughout the subsets. In practical applications, this can be pictured as a collaboration between a bank and an insurance agency [81]. The primary benefit of such a collaboration is that more attributes are available in the combined dataset, which results in a greater quality of training. The dissimilarities in the data features are taken care of by a single controller (Host) who establishes a common set with all the concerning attributes, which are then filled up by individual clients involved. Thus, the clients do not need to disclose the data, only the features required for training. The common identification keys for entries that contain data from different owners in separate columns in the same row are calculated via Private Set Intersection (PSI) schemes [82–88]. Most of these schemes are based on inexpensive crypto ciphers. Due to the limitations associated with implementations of VFL, along with the scarcity of highly secure PSI mechanisms, the field has much scope for research. Fig. 1.4 depicts the attribute space in a vertically partitioned setup between two contributors.

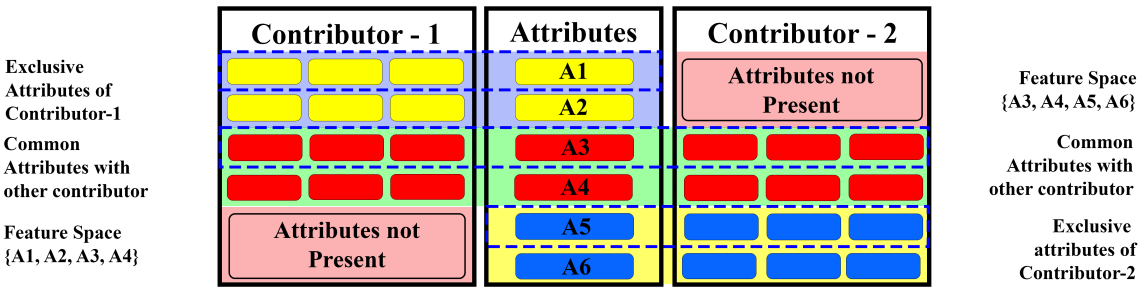


Figure 1.4: Data distribution in Vertical Federated Learning

3. Federated Transfer Learning (FTL)

   In the field of AI, most of the training methods are typically restricted to a single domain at a time while solving a problem. This implies that the training and testing sets share identical distributions of defining features. When these distributions

change or become dissimilar, retraining becomes necessary in most cases over new data. This translates to a high data acquisition cost and is sometimes difficult to train. In such cases, the concept of Transfer Learning (TL) is adopted, that is, propagating helpful knowledge from a source to a target destination model [89–94] having a similar application. Popular from way back in 1995, researchers have experimented with applying TL over AI algorithms, including Reinforcement Learning (RL), DL, Distributed Learning, and Manifolds-based learning. As of present, TL has made a firm stand in the technological landscape and, with no surprise, is swiftly making its way into FL [33, 95–99]. Given the similar nature of applications in DSA, TL has a massive application base to expand upon. A depiction of attribute space between two entities using transfer learning is provided in Fig. 1.5.

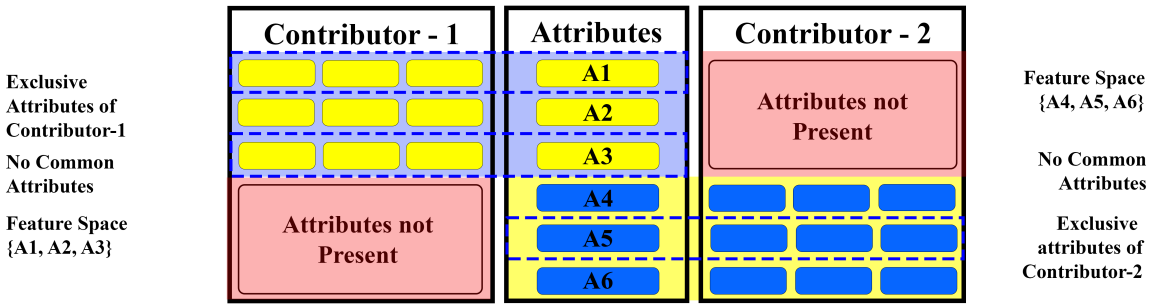


Figure 1.5: Data distribution in Federated Transfer Learning

4. Confederated Learning

The works of Liu et al. [100] discuss an FL implementation in a unique scenario encountered in large-scale intelligence systems. The data in question presents a horizontal division at the individual's level, a vertical division at the datatype level, and is separated by identity without any primary matching key. The authors employed a mix of representation learning, generative AI, imputations, and data-augmentation methods. An example of such a scenario in healthcare applications is shown in Fig. 1.6.

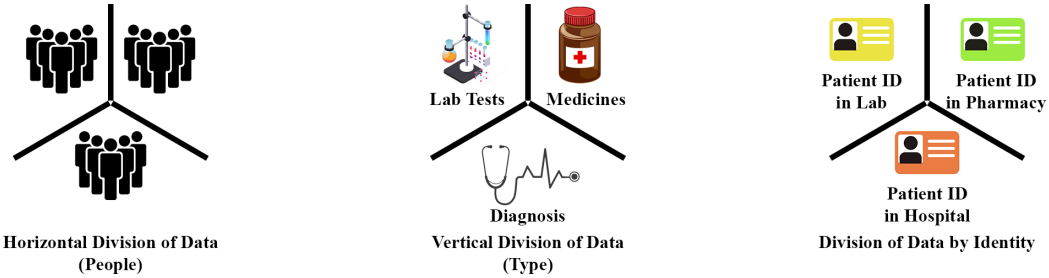


Figure 1.6: Data Distribution in Confederated Learning

1.2.4.2 Special Types of FL

Apart from the standard FL types based on attribute space, some special types of FL exist based on the utility and computing architecture. The implementations of such types are mostly case-specific and are adjusted according to the immediate goals.

1. Personalized FL (PFL)

   The most popular among these types comes in the form of PFL [101]. PFL can be explained as a case of FTL applied over either HFL or VFL. Here the globally trained model of traditional implementation is retrained over client-specific data to provide more intimate and accurate results. Some of the recent works in PFL can be seen in Health Monitoring [42], Seizure Detection [102], and Privacy-Preserving Graph Neural Networks [103]. In addition, PFL also adds a layer of privacy for data at the client level.

2. Decentralized FL (DFL)

   When FL is implemented over either block-chain or Edge/IoT networks [104] in such a way that in addition to the dataset storage, the computation also gains a decentralized structure [61]. In other words, the global model is also stored and aggregated in a decentralized format, then the implementation is regarded as DFL. DFL adds multiple benefits such as energy efficiency, additional encryptions, tamper-proof structure (block-chain), and model/data security [105]. On the other hand, the additional features add significant overhead in computation, which limits the usage of DFL to applications that can afford the trade-off [26].

3. Resource Aware FL (RFL) and Incentive Aware FL (IFL)

   RFL is a special case of FL that focuses on setups with hard constraints on storage and computing power. This is employed mainly in low-powered IoT [33], Internet of Medical Things (IoMT) [34, 106], and Edge [37] networks but is often expanded onto bigger networks to better manage resource utilization [31]. RFL is a relatively young field when compared to FL, and hence, a limited number of implementations and studies are done at the time of writing. But, as the paradigm of IoMT is gaining traction, RFL for such low-powered setups has become a field of growing focus.

   IFL, along with RFL is a relatively young field of implementations for FL. IFL is based on the process of tracking the contribution of every participant node in a training

round. This tracking is then used to allocate rewards to the nodes. The reward can be of any form such as Monetary, Reputation point, etc. These rewards motivate the nodes to actively participate in the subsequent rounds as well as attract new nodes to the training network. These implementations can be based on block-chain technology [107–109], 6G enabled Edge [110], or the architecture of RL [111].

4. Cross-Silo and Cross-Device FL

A dataset exclusive to a device or a private network, overall, when used for distributed or federated learning, is regarded as a data island. The corresponding device or network is often termed a data silo. When all the silos involved in an FL network house identical or comparable attributes, the FL can be classified as HFL. But when the attributes and datatypes differ across various silos, the FL is classified as cross-silo FL. It can be regarded as saturated VFL and requires extreme care when implemented. Du et al. [133] proposed an attack-resistant framework for privacy preservation in cross-silo FL. The framework makes use of blinding techniques and batching to minimize collusion attacks and enhance computation. Liu et al. [100] acknowledged Confederated Learning, which involves cross-silo structure across various levels in a complex way, as discussed in section 3.4.1.4. Most recently, Carvalho Bertoli et al. [112] proposed a stacked unsupervised FL method dealing with non-IID cross-silos for Intrusion Detection Systems, showing a remarkable performance over the state-of-the-art.

Another challenge encountered by FL is the compatibility of data at the cross-device level. Two devices connected in the same network or data silo may store data in different formats. This results in the degradation of generalization in the training process. Many researchers have studied and designed approaches to overcome this issue. Singh et al. [113] put forth a layer-based framework for cross-device FL., which regularizes the local training step via the knowledge distillation scheme between global and local models utilizing the small proxy dataset. The layer-wise knowledge distillation mechanism updates the local model to exploit the common representation from different layers. Another researcher, Le et al. [114], used cross-device FL to provide a security boost in Android Web services. They employed a model that learned from the browsing patterns of the user locally and generated a unique web security model locally that is further sent to the server for a globally robust model. Both approaches show comparable results to a centralized ML system.

5. Trustworthy FL

   Sanchez et al. [115] proposed that privacy preservation and security in FL alone are irrelevant to developing trust in the system and separate measures should be taken to incorporate and promote the element of trust in FL. The authors put forth privacy, robustness, fairness, explainability, federation, and accountability as the six prominent pillars of trustworthiness along with many minor ones. The purpose of the proposed algorithm is to calculate a trust score of various FL implementations based on these metrics and provide a better evaluation from this point of view.

6. Federated Meta-Learning

   Considering the statistical and systematic challenges in collaborative ML models across distributed networks that present substantial bottlenecks in the real-world applications of FL, Chen et al. [116] proposed a meta-learning FL approach FedMeta, which replaced the shared models with a shared parameterized algorithm called meta-learner. The approach showed a higher convergence rate than traditional approaches while working with less data, resulting in lower communication costs over the training period. The approach also maintains the inherent privacy of data and the performance of the trained global model.

Independent of which type of FL is in place, FL shares some similarities with Dist. L. This includes the need for a base algorithm upon which the architecture is installed. Depending on the complexity and data availability for the task to be performed, the underlying mechanism may use any type of learning algorithm. There are several research works that show a variety of ML and DL algorithms under FL. The authors in [29] put forward an ML-based federated architecture secured through a block-chain, while in [117], an IoT-based skin disease detection is highlighted using federated ML. FL found the most ground over neural networks with applications such as water-marking [118], steganalysis [97] and image classification [119], [120, 121], wireless networks [11, 45, 122], banking [4, 5, 123], and various healthcare advances [100, 124–127].

Any FL algorithm comprises aggregations over multiple rounds of training. Each of these rounds follows the following sequence of tasks.

1. Firstly, a subset is created from the members of the network (federation). This subset comprises randomly chosen members(clients) and is typically a tiny portion of the population.

2. Each client is provided with the current weights from the server.

3. The models are trained locally on clients with client data for one epoch and the updated weights are shared with the server.

4. The server collects the updates and aggregates them (conventionally by averaging) and improves the global model.

This process is repeated numerous times, each time with a different subset of data. The small size of subsets was initially desired because Google implemented the idea on Android devices, which were in the millions at the time of FL's introduction. This was further improved almost immediately by replacing model weights with gradient updates to ensure that none of the original data was transmitted over the network. A summarized view of the differences between FL and standard learning (ML, DL) is presented in Table 1.1.

Table 1.1: Comparison of FL with Traditional ML and DL

Federated Learning (FL)	**Machine Learning / Deep Learning (ML/DL)**
Data is unevenly distributed across the network	Data is centralized or cloned to all devices
Data owners have complete control over which data is shared	Data owners lose control once data is shared
Security is required at the network level	Security is required at the data storage location
High bandwidth utilization	Minimal bandwidth utilization
Complex security mechanisms needed	Simpler security mechanisms suffice
Enables training on diverse datasets for robust models	Limited dataset leads to less generalized models
Limited model support; specialized FL frameworks needed	All standard models supported with existing frameworks

FL shows a prominent scope of utility towards DSA regarding privacy, security, and computational requirements even in its basic form. This prominence has attracted researchers over the years, resulting in many implementations in various DSAs. Some of these implementations use linear approaches, while others propose a modified structure to mold FL better to the task at hand.

1.3 Impact of FL on DSA

In this section, we discuss the impact of FL on the DSA. In such applications, privacy is at an equal priority level as that of accuracy. Even a tiny leakage in data or a miscalculation in predictions can cause serious repercussions. These applications can originate from many sources in everyday life, such as banking (mobile and net), medical checkup ledger,

online shopping, etc. Some specific examples of these applications are critical illness diagnosis [3,6,7,32,120,128–132], financial fraud detection [4,5,123,133], etc. These all generate large amounts of data relating to the personal routines of individuals. If this data becomes accessible to unauthorized or malicious entities, it can result in targeted fraud, leverage, or other forms of cybercrime against the clients while also hampering the institution's reputation.

Coming to the other side of these issues, this data is used to train models for predicting crucial strategic and statistical decisions. Any modification/encryption to the data in such scenarios, either via cyber-attack or via security measures, can result in a low-quality predictor/recommender, which can prove costly not only in terms of capital but in some cases, life too (Healthcare). This results in limitations placed upon the level of encryption that can directly affect the data structure or features. Additionally, as discussed, the data is of the utmost level of sensitivity to both clients and institutions, which makes them reluctant to collaborate with one another. This presents a hurdle in getting a diverse and overall complete training set. Based on this, we can define the essential characteristics of a DSA as follows.

1. The data required for training the models are of a highly confidential nature.
2. The performance of the trained model is crucial and cannot afford modifications in data induced by encryptions or other schemes.
3. The data required for training is either present as isolated data islands (mutually exclusive and private datasets of multiple individuals or organizations) or is too large in amount to be stored centrally.

The mutual competitiveness and data security collectively put a hard bound on the size and quality of datasets from individual institutions. Hence, for a substantial amount of time, getting a highly accurate model for the tasks at hand becomes a very difficult chore.

With the advent of FL, researchers found a way to negate most of these associated issues. The most pivotal feature of FL surfaces is the change in transmitted data. That is, sharing only model parameters and gradients in place of the whole dataset. It reduces the hesitation of institutions to collaborate, which in turn, results in the formation of a bigger and more varied dataset for the concerned application. Various diagnostic applications across multiple hospitals, as well as fraud identifiers across multiple banks, are examples of such applications. FL not only accommodates the distributed data islands but also provides a sufficient level

of guarantees for privacy and security. With ongoing experimentations, FL has become more resource-optimized and has found an even greater footing in these fields. It enables resource-constrained setups to leverage distributed training. In addition, FL eliminates the possibility of fraud, theft, and leakage of data to a great extent. Healthcare and Finance emerge as the most prominent fields reaping the benefits of FL. This is evidenced by the rapid research and implementation of layered technologies tailored to various usage levels.

A generalized architecture for collaborative HFL for DSA is shown in Fig. 1.7. Here, FL is implemented at two levels, the institutional level and the global level, interacting with each other. The institutions involved are provided with the global model and its initial parameters (step 1). Then, every institution trains a federated model within their organization using local private data over the received model (step 2) and aggregates the local parameters to update an institution-level model at their central edge server (level 1) (step 3). The model parameters from these edge servers of different institutions are then shared with a central cloud server (Level 2) (step 4).

While interacting at any of the two levels, additional encryption techniques can be employed to further boost the security of the data. The cloud server aggregates all the inputs to update the global model, whose modified parameters are shared back to all the participants for the next round of training (steps 5-7). This architecture can be modified according to the problem requirements and may integrate different technologies (Blockchain, Edge, IoT) to provide intermediate support in security, processing, and storage. These integrations, in many scenarios, may also result in additional functionality in the form of resource optimization, incentive mechanisms, PFL, etc.

Fig. 1.8 shows the generalized architecture for collaborative VFL for DSA. Here, the main difference from collaborative HFL is the presence of a "Host" or moderator between participants. The host examines the attribute distribution and ensures uniformity among training datasets by applying a suitable Private Set Intersection scheme (step 0). Apart from this, the architecture operates similarly to the HFL architecture, with forward and backward propagation of updated parameters between the institute level and the cloud/edge level. The possibility of integrating supporting technologies and making modifications based on the application at hand is also prominent in VFL.

Fig. 1.9. shows the generalized architecture for collaborative FTL for DSA. Here, a model pre-trained on similar data is used to reduce the training costs. Parameters in all layers except the final few are frozen leaving only a fraction of trainable parameters minimizing the

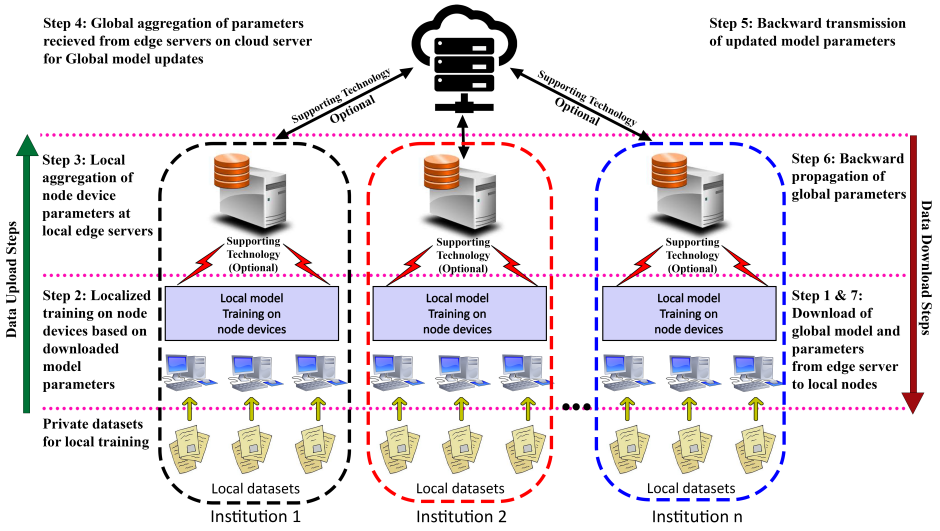


Figure 1.7: General Architecture of Collaborative Horizontal Federated Learning for Data-Sensitive Applications

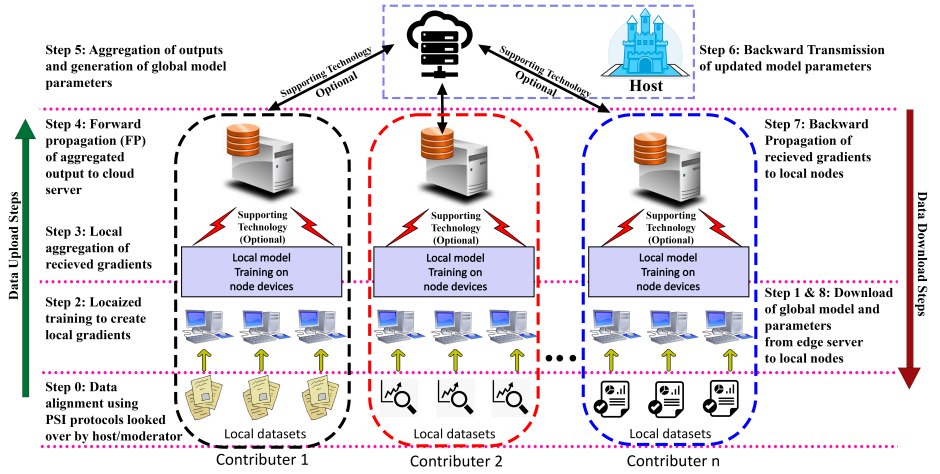


Figure 1.8: General Architecture of Collaborative Vertical Federated Learning for Data-Sensitive Applications

training time and resource consumption. After the selection of layers, the architecture works identically to the HFL architecture with forward and backward propagation of updated parameters between the institute level and cloud/edge level.

FL has been implemented with a host of advanced encryption techniques to provide varying levels of security, each with its advantages and drawbacks. The security and encryption techniques used frequently with FL are detailed in Table 1.2.

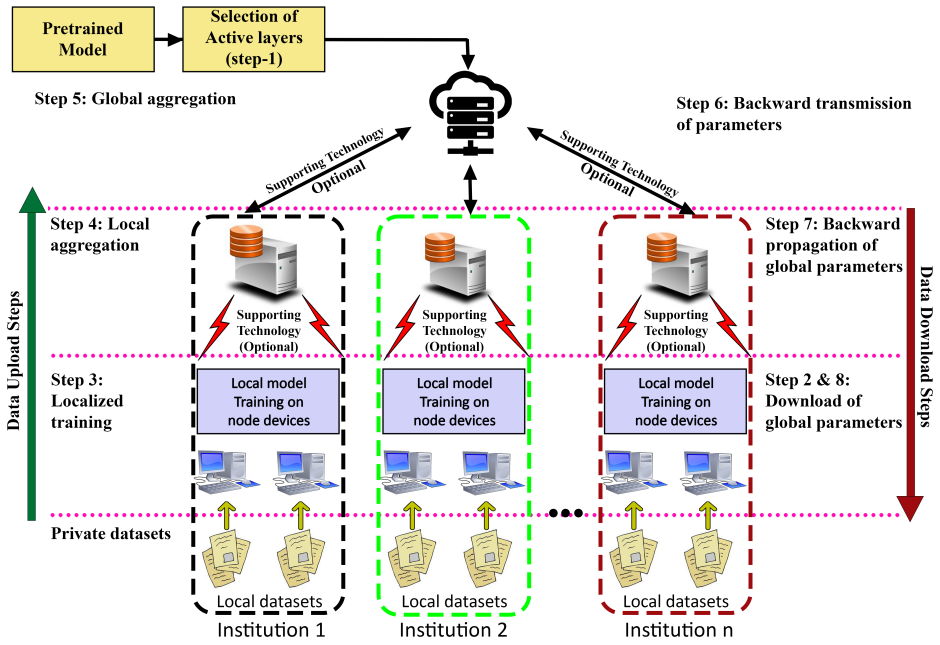


Figure 1.9: General Architecture of Collaborative Transfer Federated Learning for Data-Sensitive Applications

1.4 Issues and Challenges of Federated Learning

FL emerges as a promising technology in many fields that depend upon AI and trend analysis. However, as discussed, DSAs provide a solid foundation for the future. A wide range of configurations of FL, on its own and with supporting technologies for DSAs, was compiled in this work. Each has its unique set of advantages and drawbacks. However, the issues that plagued other types of learning have extremely low efficiency against FL. It has an altogether new form of threats that may result in poor or corrupted training and privacy leakage, or may present themselves as technical limitations of involved devices.

Table 1.2: Security techniques frequently used with FL

Technique	Advantages	Drawbacks
Differential Privacy [134]	• Protection from linkage attacks • Targeted access to partially correct data • Control over privacy levels	• Individual-level analysis not possible after applying DP • Ineffective with small datasets
Homomorphic Encryption [28, 86]	• Data remains encrypted at all times • Minimal or no trade-off between usability and privacy	• Poor performance • Cannot be applied to highly complex problems
Parameter Compression [135, 136]	• Bandwidth conservation • Reduced risk of model inversion	• Performance loss in the final model

Broadly classifying, in this section, we highlight the major challenges faced by FL along with potential solutions for each.

1.4.1 Basic Attacks and Privacy Concerns in FL

The journey of FL from concept to deployment revealed the practical challenges and vulnerabilities of the newly conceived technology. Most of these issues included efficiency, accuracy, and security concerns. The former two issues are well researched, making FL on par, if not above, traditional mechanisms in the concerned area [137]. Security is still not much explored; the major security breaches faced by federated systems are Inference attacks, Data poisoning attacks, and Model & gradient inversion attacks, combined, taking 80% of the research focus. In addition to these, the minor attacks and case-specific attacks are inherited from the distributed architecture, and the model involved has negligible efficiency in compromising the security of the implementation. There have been many studies showing that even if the data is localized to user nodes, the model itself can be exploited to deduce the personal information involved. This is done via inversion, poisoning, and inference attacks. Prevention of such attacks is challenging as among the massive number of nodes connected, any node can be malicious, which is sufficient to initiate an attack. The same is possible with server-side attacks for model inversion and inferences. Although with very little possible success, the mere possibility of such attacks happening can compromise the confidentiality of the system. Zhang et al. [38] present a GAN-based implementation of a model poisoning attack specifically designed for FL in their work. While Sun et al. [138] provide a comprehensive list of potential data poisoning schemes against FL. Some of the potential safeguards are put forth by Wen et al. [139], Ma et al. [140], and Fang et al. [141].

1.4.1.1 Inference Attack

As conveyed by its name, it is based on inference or comparisons of outputs generated by the model during different rounds to infer which data elements contributed to the training process [142, 143]. It is important to note that the attacker doesn't have direct access to either the dataset or the victim model, and the inferences are solely based on confidence scores calculated by a shadow model or attack model. This does not diminish the probability of this attack deducing the participating samples and thus compromising the security. Authors in [143] experimented and concluded that the inferences can be effectively made even if only the output labels are accessible by the attacker, highlighting an improvement in the shadow model's effectiveness. Many attempts have been made over the years to tackle

inference attacks such as DP, k-anonymity, and l-diversity [134,144,145]. Though beneficial to a certain extent [146], none of the techniques is designed to deal directly with the problem. Rather, these focus on providing better inherent data privacy. In [147], the authors try to resolve the issue via data sanitization on a collective level by manipulating certain attributes of the data set, creating added noise. This results in a decrease in inference accuracy.

1.4.1.2 Data Poisoning Attacks

These attacks are based on the contamination of data or model outcomes during either the training phase or the testing phase [38,49,138,139,141,148]. Based on this, these are divided into two categories: Causative attacks and Exploratory attacks [63]. Causative attacks manipulate the training data directly and thus heavily influence the training phase, while exploratory attacks are based on miscalculations from the model itself during the testing and predicting phases. The latter does not affect the training process of the algorithm. For a successful poisoning attack, the attacker needs direct or indirect access to the training data. Hence, most works and developments in data poisoning are centered around the system where training data is stored or transmitted to a common location. Since FL employs decentralized and private data sets, these attacks have a limited success rate. However, these attacks are an active field of research and are being rapidly upgraded to adapt to the shifting paradigm. This is evidenced by the works in [138] as well as the new category of Byzantine poisoning attacks [141, 149]. These modified Byzantine attacks use encryption over manipulated data to camouflage the poisoning from the existing defence mechanisms that focus on irregularities in data. The authors in [140] propose a double-door homomorphic encryption scheme to break the efficiency of these attacks in FL and have shown 60-80 percent improvements over traditional methods.

1.4.1.3 Model Inversion and Gradient Inversion Attacks

Though the FL system provides high-level security, it is still at risk from a state-of-the-art inversion attack from the server side. The attacker can infer the training samples by evaluating the model parameters of individual local models that are transmitted to the server. The associated process is called model inversion [150–152], hence the name. Gradient inversion [153], on the other hand, focuses on leaked gradient values shared over the FL network rather than the local model interactions. Here, the gradients are evaluated to reconstruct the used dataset, and the results can be pixel-wise accurate for images.

This implies that the key defining components, or the sensitive part of the data, can be restored via such an attack. To minimize the possibility of such a scenario, typically, noise is introduced in the gradients and model parameters before transmitting. The introduction of the said noise prevents the inversion directly, but also levies a cost in terms of the quality of the aggregated model. Authors in [152] propose a scheme to provide the same effect as the general approach without compromising quality. This is achieved by splitting the local model into sub-models, which are then aggregated, and the required values are shared.

Experimentally, the global model trained by these aggregated values is identical to the case of naïve FL. While also concluding that the inversion attack on these transmitted values gets heavily diluted or vague, sensitive details of the used dataset. Another technique for preserving the quality of training is discussed in [153], focusing on the field of IoMT, dealing with highly sensitive patient data. The authors propose a hybrid implementation of proximal gradients and split learning to provide an encryption layer to transmitted attributes. Another successful attempt at minimizing the quality compromise is based on quantizing parameters with strict constraints [135] experimented on over the Internet of Medical Things.

We have studied these attacks in depth in Chapter 2 and proposed four frameworks, DRAFFSS, Fed-WALQ, Fed-DLC, and Fed-HDVE in following chapters to counter their effectiveness.

1.4.2 Transmission Costs

The focus of FL is privacy; the data is stored locally on the training nodes. Considering the scale of implementations, the nodes involved can range from a few hundred to millions. In such large-scale deployments, the number of gradients or weights shared may cause a network jam if the network's bandwidth bottlenecks the process. The major issue caused by the jam is a decrease in transmission efficiency. As the model updates are periodic and the nodes have varying computational power, a delay in uploads can result in a scenario where devices spend more time transmitting rather than in actual training. The problem expands with the inclusion of more and more devices in the training network and occurs in both centralized and decentralized FL. Given that one can broadly ignore the cost associated with dealing with traffic for sensitive applications, the possibility of getting a complete deadlock in training is a major concern. Another aspect of the transmission problem is the difference between uplink and downlink speeds provided by the network. Usually, a significant drop is recorded when comparing down-speed to up-speed, which increases the

costs of uploading model data [154].

Research is aimed at two potential areas to tackle this issue: the first is to limit or balance the number of communication rounds required for good-quality training, along with an optimized selection process. This is done to limit the devices contributing to a particular round. The second area is to optimize the size of data shared from the device to the server, minimizing the payload and overhead.

We propose two quantisation-based frameworks in Chapter 2, which optimize the bandwidth usage for both powerful data centres as well as resource-limited networks such as IoMT.

1.4.3 System Heterogeneity

FL relies on connected nodes or devices to facilitate model training. The connections can be made in any number of ways (wired or wireless) with any type of device that exists in the network, irrespective of the computational power level or resources of the device. This induces heterogeneity at the system level, which may impart delays in mitigation and fault tolerance of the learning process. Apart from the variation in configurations, the nodes may not be idle for processing, or at a required battery power level, and in many cases may get offline without prior intimation. These uncertainties make one consider the issues of latency and fault tolerance. Both of which are of pivotal importance in DSAs.

However, research has been done on how to prevent the latency hit. Many researchers in recent years have come up with implementations to minimize the impact of heterogeneity in data, but few have tried to tackle this on the hardware level. A sub-grouping approach towards connected nodes based on their capabilities may be able to provide a potential direction for future works [13].

1.4.4 Data Heterogeneity

While system heterogeneity represents physical variations, heterogeneity in data corresponds to the variation among the datatypes, and the distribution space from different contributors in the training process. Such irregularity containing data is termed non-IID. Non-IID data makes the modelling and evaluation process much more complex. Variation in data also decreases the guarantee of an unbiased model when using stochastic gradient descent, a widely implemented scheme in FL as of now. Biased training causes a model to predict incorrectly with a high confidence value, making it difficult to rectify or trace this issue. A number of works regarding statistical heterogeneity have surfaced in recent years, ranging

from aggregators [18] to edge-based implementations [21, 22] and generalized FL solutions [23, 155]. The performance gains attained are marginal to satisfactory in nature and hence, provide a base for improvements.

We counter the effect of hetrogeneity in our proposed frameworks by introducing incentive and workload-aware mechanisms to optimize the load distribution across client systems according to their sustainable limits.

These challenges that plague FL implementations in real-world deployments form the basis of our motivation and research objectives in order to provide a robust and secure way of training sensitive applications.

1.5 Motivation

After extensively studying the state-of-the-art in the field of federated learning for data-sensitive applications, we identified several research gaps that remain unaddressed. These gaps are addressed in our research objectives, which are highlighted as follows.

- Privacy and security of data have been a rising concern in recent times at the individual and organizational levels.

- The accompanying insecurity significantly restricts collaboration and data sharing, resulting in the formation of isolated data islands. These islands individually do not house sufficient data for a good-quality training of any AI applications.

- Both the privacy issue and the scattered distribution of data can be effectively handled using Federated Learning, which minimizes the data exchange over a distributed node setup.

- Data-sensitive applications emerge as the most welcoming field for FL implementations and have seen rapid growth in research and experimentation over the last few years.

1.6 Problem Statement

1.6.1 Research Objectives

1. **Objective 1:** To study, analyze, and compare various implementations of Federated Learning and their utility toward Data Sensitive Applications.
2. **Objective 2:** To implement secure Incentive-Aware Federated Learning mechanism for Data Sensitive Applications.
3. **Objective 3:** To propose secure Federated Learning framework for Data Sensitive Applications.

1.6.2 Threat Model and Adversarial Assumptions

To clearly position the security guarantees of the proposed frameworks, we define the assumed threat model considered throughout this thesis. The central aggregation server is primarily assumed to be *honest-but-curious*; that is, it correctly follows the prescribed federated learning protocol but may attempt to infer sensitive information from the received model updates. Communication channels are assumed to be vulnerable to eavesdropping and man-in-the-middle interception attacks.

Client devices are considered semi-trusted participants. A subset of clients may behave in a *malicious or colluding* manner by attempting to infer information about other participants, reconstruct training data from shared gradients, or disrupt the training process through manipulated updates. However, large-scale coordinated Byzantine attacks are considered outside the primary scope of this work.

Under this threat model, the objective of the proposed frameworks is to protect model updates during transmission, prevent inference attacks on gradients, and reduce the risk of information leakage from both communication and aggregation stages while maintaining practical computational efficiency.

1.6.3 Contribution in the Thesis

In this thesis, we focus on federated learning and its utility for applications that deal with highly sensitive data in resource-constrained environments and briefly address a few research gaps associated with the field of federated learning and data-sensitive applications. In this

thesis, three research objectives were defined as mentioned in Section 1.6.1, and we address all these objectives one by one discussed below, as a contribution of the thesis:

1. Objective 1: *To study, analyze, and compare various implementations of Federated Learning and their utility toward Data Sensitive Applications.*

   In this objective, we studied the various forms and applications of federated learning, their implementation tools, potential supporting technologies, and active applications in healthcare and finance, along with an in-depth examination of dataset availability, potential attacks, and the drawbacks of each implementation covered in Chapter 2.

2. Objective 2: *To implement secure Incentive-Aware Federated Learning mechanism for Data Sensitive Applications.*

   In this objective, we propose two secure frameworks for federated learning, incorporating incentive-awareness mechanisms and quantization in Chapter 3. The first framework, DRAFFSS, focuses on enhancing the resource utilization in the training network while also making it stable in the long run. The second framework, Fed-WALQ, builds on DRAFFSS and targets security while maintaining optimal resource management in the network.

3. Objective 3: *To propose secure Federated Learning framework for Data Sensitive Applications.*

   To achieve this objective, we propose two secure frameworks that implement encryption and differential privacy to provide robust defence against various attacks. The first framework, Fed-DLC, discussed in Chapter 4, introduces a dynamic, lightweight cipher that changes depending on the training device attributes, such as the time of joining the network, among others, over a simple block structure to provide security at minimal computational cost. The second framework, Fed-DHVE, discussed in Chapter 5, incorporates differential privacy and homomorphic encryption to secure big training networks with heavy computational resources.

1.7 Thesis Outline

The thesis comprises six chapters, each addressing key aspects of federated learning that are pivotal in data-sensitive applications.

Chapter 1: Introduction

This chapter provides the foundational insights into the fundamental working of federated learning, its various types, architectures, tools and frameworks. This chapter outlines the motivation for the thesis, presents the problem statement, describes the research objectives, and highlights the thesis's contributions. This chapter also includes a concise outline of the thesis.

Chapter 2: Literature Review

This section provides an extensive analysis of studies addressing the utility and extent of adoption of federated learning in different data-sensitive fields and their potential limitations at the time and also provides the comparisons in approaches, their advantages, and drawbacks with different tools and structures in practice. This chapter concludes by identifying research gaps in the existing literature.

Chapter 3: Workload-aware secure federated learning framework with quantization

This chapter presents two hybrid frameworks that integrate the popular quantization approach with workload-aware epoch distribution to reduce bandwidth consumption, optimize resource utilization, minimize stragglers, and provide security against model inversion attacks. The discussion encompasses both resource-constrained and resource-rich networks, which are tested using a variety of datasets of varying complexity for detailed analysis.

Chapter 4: Secure federated learning framework with dynamic lightweight ciphers for resource-constrained data-sensitive applications

This chapter introduces a lightweight cipher scheme modified to work with federated learning. The keys used by the scheme are generated dynamically and change over rounds to provide security to transmitted parameter vectors. The discussion covers the impact on performance, overall training time, and level of security provided by the proposed framework, along with future adjustments that can be made to suit different types of training networks.

Chapter 5: Secure federated learning framework with differential privacy and homomorphic vector encryption

This chapter introduces a hybrid framework aimed at improving privacy aspect of federated learning by implementing differential privacy over a homomorphic encryption layer for transmitted parameter vectors. The discussion covers the enhancements in security gained by

the hybrid in targetted applications as well as practical challenges and adaptability needed to be solved for widespread adoption of same.

Chapter 6: Conclusion and Future Scope

This final chapter outlines the findings and potential directions for future inquiry in the research work. This chapter examines the significance of proposed solutions in enhancing security and safeguarding privacy.

Chapter 2

Literature review

The objective of this chapter [*]*. is to provide an exhaustive literature review on the current state of federated learning, its implementations in data-sensitive applications with and without supporting technologies, data availability, and associated challenges. This chapter analyses the utilities of existing state-of-the-art and pinpoints existing research gaps.*

Ever since put forth, the idea and implementation of FL have been an area of focus for active research. Over the course of our analysis, we found many surveys and studies that explored the paradigm in various fields. Though none of the works focus on DSA directly, some partial coverage is seen in a few. Qu et al. [26] discuss at length the mutual benefits and shortcomings of block-chain-integrated FL, along with the threats this paradigm faces in its current state. Issa et al. [24] expand upon these fundamentals and provide a study of blockchain-assisted FL for the Internet of Things (IoT). Wahab et al. [156] provide extensive classifications of FL based on applications, supporting technologies, and scope of implementation, while also providing a detailed analysis of common and uncommon threats. A wider view of FL, its surrounding structure, hype, and challenges are provided by Li et al. [157], Zhang et al. [14], Aledhari et al. [158], Bonawitz et al. [159], Rodriguez Barroso et al. [160], Huang et al. [75], Liu et al. [76], and Abdulrahman et al. [161] following a general point of view. As well as providing a cleaner picture of the transition that took place in research and applications from centralized learning to distributed decentralized learning, and accompanying issues. Authors in [12, 13, 15, 17, 138, 162, 163] take a more scrutinizing

[*]The part of this chapter has been published/accepted/communicated in: Manu Narula, Jasraj Meena, and Dinesh Kumar Vishwakarma, "A comprehensive review on Federated Learning for Data-Sensitive Application: Open issues & challenges," *Engineering Applications of Artificial Intelligence*, Elsevier, Volume 133, Part B, pp. 108128, 2024, DOI: `https://doi.org/10.1016/j.engappai.2024.108128`. **(SCIE, IF=8.0)**.
&
Manu Narula, Jasraj Meena and Dinesh Kumar Vishwakarma, "A Comprehensive Study of Federated Learning Tools and Frameworks for Data-Sensitive Applications",*Puneet Kumar Gupta (eds), Computational Models for Intelligence and Automation*, SCRS, India, 2025, pp. 127-135. DOI: `https://doi.org/10.56155/978-81-975670-1-8-12`

approach and focus on various present and potential concerns, flaws, and security threats of FL. Lim et al. [36] and Wang et al. [35] highlight the advantages and milestones achieved by FL when integrated into mobile edge networks. Gupta et al. [164] and Bellavista et al. [165] presented a study of FL in a distributed and decentralized environment, respectively. Lavaur et al. [137] and Khokhar et al. [119] provide a more application-specific coverage of FL in intrusion detection and image analysis respectively. Moving on to partial coverage of DSAs, the works by Nguyen et al. [124], Sadilek et al. [166], Kaissis et al. [167], and Xu et al. [125] highlight the utility of FL in the field of healthcare. The application of FL in the field of banking is put forth in the works of Long et al. [168]. An overview of the findings and approaches of these surveys is given in Table 2.1, highlighting the covered areas of applications and the challenges discussed in the respective works.

Table 2.1: Summary of existing surveys for federated learning

Ref.	**Year**	**Areas Covered**						**Challenge Area Discussed**				
		FL	BC	IoT	Sec	NW/E	DSA	TC/C	S/A	RP	HET	
											D	S
Li et al. [13]	2019	✓	✗	✗	✓	✓	✗	✓	✓	✗	✓	✓
Kairouz et al. [162]	2019	✓	✗	✗	✓	✗	✗	✗	✓	✗	✓	✓
Aledhari et al. [158]	2020	✓	✓	✓	✓	✓	✗	✗	✓	✗	✓	✓
Zhang et al. [14]	2021	✓	✗	✗	✓	✗	✗	✓	✗	✗	✓	✓
Li et al. [157]	2021	✓	✗	✗	✓	✓	✗	✓	✓	✓	✓	✓
Bonawitz et al. [159]	2022	✓	✗	✓	✗	✓	✗	✗	●	✗	✗	✓
Rodriguez-Barroso et al. [160]	2023	✓	✗	✗	✓	✗	✗	✗	✓	✗	✓	✗
Liu et al. [76]	2022	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗
Huang et al. [75]	2022	✓	✗	✗	✓	✗	✗	✓	✓	✗	✓	✗
Bellavista et al. [165]	2021	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✓
Gupta et al. [164]	2022	✓	●	✓	✗	✓	✗	●	✗	●	✗	●
Issa et al. [24]	2022	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
Qu et al. [26]	2022	✓	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓
Wang et al. [35]	2019	✓	✗	✗	✗	✓	✗	✓	✓	✓	✓	●
Lim et al. [36]	2020	✓	✗	✗	✓	✓	✗	✓	✓	✓	✓	✓
Wahab et al. [156]	2021	✓	✗	✗	✗	✓	✗	✓	✗	✗	✓	✗
Abdulrahman et al. [161]	2021	✓	✗	✗	✗	✓	✗	✓	✓	✓	✓	✓
Khan et al. [15]	2021	✓	✗	✓	✓	✗	✗	✓	✓	✓	✓	✓
Lyu et al. [17]	2020	✓	✗	✗	✓	✗	✗	✗	✓	✗	✓	✓
Bouacida et al. [12]	2021	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗	✗
Truong et al. [163]	2021	✓	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗
Sun et al. [138]	2022	✓	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
Lavaur et al. [137]	2022	✓	✗	✗	✓	✗	✗	●	✗	●	✗	✗
Kaissis et al. [167]	2020	✓	✗	✗	✓	✗	●	✓	✓	✓	✓	✓
Long et al. [168]	2021	✓	✗	✗	●	✗	●	✗	✓	✗	✓	✓
Sadilek et al. [166]	2021	✓	✗	✗	✓	✗	●	✗	✓	✗	✓	✗
Xu et al. [125]	2021	✓	✗	✗	✗	✗	●	✓	✓	✗	●	✗
Khokhar et al. [119]	2022	✓	✗	✗	✓	✓	✗	✗	✓	✗	✗	✗
Nguyen et al. [124]	2023	✓	✗	✓	✓	✗	●	✗	✓	✓	✓	✓
Narula et al. [169]	2024	✓	●	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓ represents complete coverage, ✗ represents no coverage, and ● represents partial coverage of the field. FL: Feder-! ated Learning, BC: Block-Chain, IoT: Internet of Things, Sec: Security, NW/E: Network/Edge, DSA: Data-Sensitive Application, TC/C: Transmission Costs/Communication, S/A: Security/Attacks, RO: Resource Optimization, Het: Heterogeneity, D: Data, S: System.

2.1 Federated Learning Implementations with Supporting Technologies

Apart from the inherent privacy of FL, another technology is often layered over FL to provide a substantial improvement in the level of security. The supporting technology can range from block-chain and multilevel transforms to light homomorphic ciphers and edge networks, as highlighted in Table 2.2.

In [26], Qu et al. discuss the implementation of FL supported by block-chain technology. In their study, the authors conclude that block-chain serves FL systems with not only robustness against various poisoning attacks. But also reduce the possibility of a single-point failure by decentralizing the process. In addition, the incentive mechanism associated with proof of work motivates the nodes involved to participate more actively in the training process. In exchange, FL provides block-chain with an energy-saving consensus algorithm and a secure way for cross-chain communications relating to the same model. Looking at the other side of this integration, the computation costs rise significantly with the involved overheads.

Authors Korkmaz et al. [29], Sun et al. [25, 30], and Wang et. al. [107], take the approach further with practical experimentation. Their experiments also highlight the overheads incorporated with using block-chain in implementation. The work of Korkmaz et al. [29] is focused on the problem of single-point failures in centralized FL and puts forward a decentralized FL framework to address the issues. Sun et al. [25, 30] take an additional step and introduce additive homomorphic encryption on the transmitting data, focusing on privacy and leakage issues in general FL schemes. Wang et al. [107], on the other hand, highlight the benefits of incentive mechanisms powered by block-chain in FL. All discussed implementations show performance at par with centralized FL.

Another exciting example of block-chain-assisted FL is seen in the Industrial IoT (IIoT) field, discussed by Jia et al. [28]. The author proposes a multi-level, security-enabled aggregator for FL. The implementation is achieved using Differential Privacy (DP) and homomorphic encryption over a decentralized structure (block-chain). The work is focused on crucial data generated by industries utilizing IIoT. From the experimental results, significant improvements are seen in accuracy and f1-score while uplifting the overall security of the system. The implementation, however, suffers from increased computational complexity due to block-chain integration as shown in Table 2.2.

In addition to these technologies, the Internet of Vehicles (IoV) is no stranger to their experimentation. He. et al. [27], extended the decentralized approach to connected and/or autonomous vehicles and proposed a new consensus algorithm based on FL to resist potential adversaries. They also integrate the InterPlanetary File System (IPFS), the potential successor of HTML to provide a more reliable networked storage system. The overall results highlight an effective resistance towards malicious node inputs but compromise the accuracy.

Pham et al. [170] highlight the low-powered nature of wireless networks and propose a dynamic Unmanned Aerial Vehicle (UAV) based communication system for sustainable FL in this environment. The communication is achieved via Frequency Division Multiple Access (FDMA) in asynchronous mode. The simulation results provided by the authors show significantly less power consumption than other schemes (Fixed CPU Frequency, Fixed Transmission Time, and Fixed UAV placement). However, the testing and simulation are done with only one multiple-access technique, Hence, further experimentation is needed with synchronous techniques.

Switching from IoV to IoMT, Dara et al. [171], proposed a combined implementation of IoMT, DL, Transfer Learning, and FL to detect COVID-19 from Computer Tomography (CT) scan images. Such a hybrid is prepared to increase the scalability of FL applications at different levels (local and global). The results show a promising increase in accuracy when compared to other similar models (Xception, DenseNet, VGG-16, etc.). However, the amalgamation also results in an increased computational complexity, which may narrow down the accepting applications. In addition, the work, though discussing potential usage for encrypted traffic, currently focuses only on non-encrypted images.

Another hybrid approach is given by Singh et al. [172]. The authors employ block-chain and FL to preserve privacy in the IIoT. With more specific usage in Industry 5.0, the proposed mechanism incorporates a validation and verification layer to boost privacy and authentication. This is done through miners at the block-chain level. On the FL level, the different departments of the industry allow for localized training updates over an automated network. In addition, a Distributed Hash Table (DHT) is incorporated at the cloud level to provide decentralized secure storage. The approach shows slightly better accuracy when compared to existing schemes (FedAVG, FedSGV).

Xue et al. [34], Li et al. [39], and Mo et al. [40] extended FL to devices with low power or low computational resources by integrating edge computing. Implementing in the commu-

nication, healthcare, and Personalized FL (PFL) fields, respectively, using a Reinforcement Learning (RL) algorithm. The authors also present the use of edge servers with connected client nodes for group-based computation with a decentralized setup. A two-level update is adopted for aggregations, encompassing both the group and global models. The results shown in the work highlight that the performance results are just slightly behind those of a traditional high-powered training network. But enables real-time predictions for such constrained networks. The research, however, is still in its early stages and has considerable scope for experimentation remaining.

Zhang et al. [33] propose a block-chain, Edge, and FL hybrid system for IIoT. Their algorithm uses TL over edge nodes for rapid and efficient training in resource-constrained environments. They also propose a blockchain-based device authentication scheme to limit unauthorized access. Though the work is promising, the accuracy of TL suffers, and overheads of block-chain make the process slower, which may not be feasible in some cases.

Considering the ever-changing nature of the network, authors Pinyoanuntapong et al. [122] and Morell et al. [41] proposed edge-based implementations dealing with wireless networks and low client-density networks, respectively. While the former focuses on improving transmission efficiency in dynamic wireless systems, the latter proposes a fault-tolerant mechanism for networks with very low clients that might also drop out to zero. Both works show improvements over traditional FL with slight margins. The drawback for the former is a limited improvement in computational efficiency. For the latter, if the network becomes stable with very low clients and with zero dropout rate and addition rate for a prolonged period, the accuracy of the model suffers greatly due to statistical heterogeneity. The mutual benefits gained by these collaborations between technologies, along with the associated disadvantages, are summarized in Table 2.2.

2.2 Analysis of FL Implementations in Data-Sensitive Applications

DSA represents a variety of applications from many fields that both, deal with sensitive information, and have a resource-constrained environment. Each of these fields presents us with a similar level of requirements and importance. FL satisfies most of these requirements of DSA. While healthcare put forth the most rapid progress in adopting FL technology, Banking, Education, E-commerce, and other fields have barely touched the surface. Many

Table 2.2: FL Implementations with Supporting Technologies

Reference	Technologies involved along with FL	Benefits gained by FL	Benefits of FL for supporting technology	Disadvantages
Qu et al. [26], Korkmaz et al. [29], and Sun et al. [30]	• Block-chain • Homomorphic encryption	• Prevention of single-point failure. • Incentive mechanism to reward participation. • Robustness against Byzantine and poisoning attacks.	Low-power consensus algorithm for block-chain and a secure way for cross-chain communication.	High overheads lead to increased computational complexity.
Sun et al. [25]	• Block-chain	• Prevention of single-point failure. • Improved security in case of data leak via noise-based encryption that defaults to zero when aggregated. • Audit mechanism for malicious user tracking based on behaviour chain.	N.A.	N.A.
Wang et al. [107]	• Block-chain	• Prevention of single-point failure. • Incentive mechanism to reward participation to attract active participation.	N.A.	N.A.
Jia et al. [28]	• Blockchain • Industrial IoT • Homomorphic Encryption	• Prevention of single-point failure. • Gains a tamper-proof medium for cross-communication, increasing data security. • Robustness against inference and inversion attacks.	N.A.	N.A.
He et al. [27]	• Blockchain • Edge computing • IoV	• Prevention of single-point failure. • Decentralized storage, and robustness against poisoning. • Faster and secure transmissions across the networks.	Secure and reliable training solution (IoV) and decreased poisoned data in training.	Measures addressing data poisoning result in a significant decrease in model accuracy.
Pham et al. [170]	• IoV	• Sustainability for FL-based low-powered wireless networks. • Energy-efficient network.	N.A.	N.A.
Dara et al. [171]	• IoT	N.A.	Privacy guarantees for patient data in real-life applications.	...
Singh et al. [172]	• Blockchain • 5G • Industry 5.0	• Data authentication and verification mechanism, improving reliability and security. • Decentralized architecture preventing single-point failure. • Reduced latency.	Guarantee of privacy in Industry 5.0.	Overheads from verification and validation as well as block-chain result in greater computational complexity.
Mo et al. [40]	• Edge computing	• Minimized energy consumption in transmissions.	N.A.	The accuracy of models lies slightly behind when compared to models trained in general environments.
Xue et al. [34]	• Edge computing • IoMT • SDN • Homomorphic encryption	• Energy optimization. • Prevention of single-point failure. • Boost privacy and security while transmitting. • Decentralized storage and computing.	Smooth execution of ML and DL training over resource-constrained environments.	N.A.
Li et al. [39]	• Edge computing	• Energy efficiency and resource optimization.	N.A.	N.A.
Zhang et al. [33]	• Industrial IoT • Blockchain • Edge computing	• Device authentication, improving reliability of clients. • Supporting computation power for low-powered devices.	N.A.	Overheads make the process slower, which may not be feasible in some cases.
Pinyoanuntapong et al. [122]	• Edge networks	• Transmission efficiency.	N.A.	N.A.
Morell et al. [41]	• Edge networks	• Dynamic adjustment mechanism for handling heterogeneity and user dropouts.	N.A.	If client devices are too low with zero change in numbers, accuracy suffers greatly.

of these advances are generalized in nature as they can be expanded to other applications with relative ease than creating a similar mechanism from scratch.

2.2.1 FL in Healthcare

Out of all DSA fields, Healthcare has attracted the most focus towards implementations based on FL for improvement in various aspects (Healthcare systems, Monitoring, and Diagnosis). The works range from linear FL applications to multi-layered complex architectures incorporating one or many different technologies with FL. The various initiatives that came forward in healthcare, along with their used datasets, performance metrics, and key contributions are summarized in Tables 2.3, 2.4, 2.5, 2.6, 2.7, and 2.8.

Starting with the work of Vepakomma et al. [173] on splitNN, the author proposes different configurations for different data setups like vertical, horizontal, centralized, etc. The work shifts the focus toward resource optimization as its approach makes the Deep Neural Network (DNN) implementations accessible to low-powered and resource-constrained setups. They summarize their results in terms of storage and processing. The results highlight a drop in requirements by 95% in computing and about 50% to 90% in storage consumption.

Poirot et al. [126], present another approach involving splitNN and highlight the utility of collaborative FL based on DL models in healthcare. They experimented and compared deep collaborative learning using splitNN with central and non-collaborating setups over healthcare datasets as shown in Table 2.3. The results presented in the work show a performance gain of 3% to 60%, depending on the number of participants, as compared to centralized and traditional training models. Their technique, however, is currently limited to a single protocol, presenting a substantial scope for improvement.

Boughorbel et al. [174] sighted the acceleration provided by ML to precision medicine diagnosis. They proposed an FL method with the capability of reducing the impact of uncertainty in Electronic Health Records (EHR) data. The feat was achieved by actively rejecting contributions that seemed more fluctuating compared to the previous rounds. The proposed scheme introduces ensembling to the models by taking only one final aggregated result for each contributing entity for the final aggregated global update. The mechanism, however, is only tested with small datasets with simple rejection criteria which may not be suitable for complex applications.

Moving to image analysis, Yan et al. [121] propose a variation-aware FL over decentralized data. Their approach uses a modified Generative Adversarial Network (GAN) and

CycleGAN to synthesize a dataset based on raw image data of clients. The synthesized set comprises images similar in characteristics but significantly different from raw data which makes them suitable to share with other clients. This is done to normalize the image space of the data from the cross-client federated setup. The approach performs well for multi-client networks with varied datasets. However, since the image transformation may reduce the image space present in the raw data of a client to match it with the data of less image space of another client, distortion may occur in the augmented results. This presents a minor accuracy bump when implemented in DSAs dealing with critical disease diagnoses like prostate cancer discussed by the authors. In addition, the approach currently cannot adjust to the manual annotations done by physicians or other experts to the raw data.

Zhang et al. [18] propose a split-learning-based aggregator applicable to any FL implementation that deals with biased or heterogeneous datasets in image classification tasks. The authors propose an architecture partially like that discussed in fig.1.7. and enable forward and backward propagation to manage variations in inputs and outputs. The approach uses gradient sharing to slightly boost the security of the setup in case of a leak. The aggregator is pitted against known algorithms and achieves promising results with improvements ranging from 50% to 75% over medical image datasets. The most prominent limitation herein is that the whole work is theorized and experimented with statistic heterogeneity only. There is no estimate of how well this will work with other forms of heterogeneity like resources/system heterogeneity.

Another attempt to boost security in image data is seen in the work of Adnan et al. [175]. The authors propose using DP over the parameter updates used for FL training in a decentralized architecture. The experiments conducted by them highlight that DP significantly increases the guarantee of privacy while maintaining the overheads in feasible constraints. The approach is tested against centralized FL and traditional non-FL setups for histopathology image analysis and shows at-par performance with both, Independent and Identically Distributed (IID) and Non-IID distributions.

Linardos et al. [176], in their simulations, highlighted the prowess of FL with decentralized datasets. They employed FL in medical imaging to identify cardiovascular diseases over a distributed dataset. The dataset was first augmented to boost the number of data points for good training. The main aim of the study is to exhibit robustness against data heterogeneity for a model trained via FL and improve the privacy of patient data. The results are provided for all configurations (No augmentation, basic augmentation, shape augmentation, intensity

augmentation, collaborative learning, and FL) and show slightly increased accuracies. The tradeoff here involves the high sensitivity of CMR diagnosis to shape. The augmentation induced only proves to be advantageous when applied at a low scale. Table 2.3 summarizes these advances in adaptive healthcare systems and medical imaging.

Table 2.3: Recent Advances of FL in Adaptive Healthcare and Medical Imaging

Reference	**Main Area**	**Dataset**	**Performance Metrics**					**Key Contributions**	**Limitations**
			Acc	Pre	Rec	F1	ReU		
Vepakomma et al. [173]	Healthcare	Cifar-10 and cifar-100	✗	✗	✗	✗	✓	An FL-based method with split-learning for EHR training	Convergence stability under heterogeneous data distributions is not analyzed
Poirot et al. [126]	Collaborative DL and Healthcare	Diabetic Retinopathy [177] and Chexpert [178]	✓	✗	✗	✗	✗	Analysis of performance gains of Dist. L systems and their future in the medical field	Limited comparison with other distributed learning algorithms.
Boughorbel et al. [174]	Uncertainty-aware FL	Self-created dataset from data by Cerner Health Facts EHR	✓	✗	✗	✗	✗	FL approach for dealing with uncertainty in model updates	While data heterogeneity is addressed, the challenge of non-IID data is not fully resolved.
Yan et al. [121]	Medical Imaging	Prostatex Dataset [179]	✓	✗	✗	✗	✗	A novel FL approach for prostate cancer detection	The paper does not analyze the communication overheads and asynchronous participation which are cruicial for the target network.
Zhang et al. [18]	Medical Imaging	Diabetic Retinopathy [177], RSNA Bone Age [180], and BraTs 2017 [181]	✓	✗	✗	✗	✗	Method to deal with data heterogeneity at various levels.	The analysis or potential directions for system heterogeneity are not explored.
Adnan et al. [175]	Medical Imaging	Lung cancer dataset (LUAD and LUSC) [182, 183]	✓	✗	✗	✗	✗	DP in medical imaging applicable for both IID and Non-IID distributions	Although the framework achieved strong differential privacy, the paper does not deeply analyze how different privacy budgets affect clinical utility or model robustness.
Linardos et al. [176]	Image Diagnosis	M&M [184] and ACDC [185]	✓	✗	✗	✗	✗	Novel FL approach for sensitive image diagnosis.	Broader validation on other cancer types or imaging modalities (MRI, CT, etc.) was not conducted, limiting generalizability.

Switching from diagnosis to monitoring, Krall et al. [150] discuss the sharp increase in data availability in healthcare systems powered by IoT. Considering the advantages of this

availability for analytics and diagnosis and the risk of theft of sensitive data. The authors propose a gradient-based mechanism to deflect model inversion attacks. The approach is implemented over a DP-enabled setup and shows a significant drop in attack accuracy at the cost of slight decreases in model accuracy. This drop may be considered significant when considered with respect to DSA.

The improvements offered by FL also extend to existing implementations of general healthcare systems that are based on IoT. Lin et al. [2] demonstrate an efficient access control scheme based on the social footprint of clients to enable better security for in-home healthcare monitoring systems. This is achieved by evaluating their relationships with other clients and estimating a trust level to provide applicable authorization. Their proposed system, Secure Access Control Mechanism (SACM), as shown by their studies, also provides a promising direction in data integrity protection.

Another implementation in IoMT-based healthcare is put forth by Rehman et al. [127]. The authors propose a block-chain-integrated FL approach to boost privacy in large-scale healthcare systems in IoMT networks. The approach is coupled with an Intrusion Detection System (IDS) to analyze malicious user inputs in case of compromised nodes. The results are compared with traditional setups using pure FL (Random Forest, KNN, SVM linear) and show marginal to significant improvements in accuracy for intrusion detection and healthcare analysis. However, the implementation also suffers the overheads associated with the complexities of block-chain and overlapping IDS.

Talking about wearable tech, it is important to note that most of these devices are linked to our smartphones. When using touch-based devices for client input, the resulting vibrations of taps recorded by sensors of the said device may be used for eavesdropping on the information entered. A detailed study of such side channel attacks and their possible prevention method is presented by Rehman et al. [186]. They demonstrated their experiments over attacks based on smartphones. Though the results obtained are positive, further experimentation is required to be certain as the method is tested with only a single and comparatively simple configuration.

Apart from common challenges and implementations, some rare instances show how complicated the FL methods can get in real-life applications. One such case is studied by Liu et al. [100], where the data partitioning is horizontal in general but vertical in internal structure. The authors encountered this hybrid distribution in large-scale systems and proposed a modified FL technique: confederated learning. The technique effectively adjusts to the

biased data by virtue of a network of models trained on separate disconnected data streams from multiple entities. The entities collaborate in real-time to create a global model. There are many limitations to this approach as discussed by the authors. The most significant is the requirement of a connecting attribute across all the silos, which may be derived, but can result in negligence of important interactions among data attributes. The advances integrating IoT and IoMT with FL in healthcare monitoring are discussed in Table 2.4.

Wu et al. [42] and Chen et al. [92] further expanded PFL for health monitoring systems. While the former focused on general applications, the latter proposed a similar approach for wireless wearable devices. Both works follow the same structure, only differing in protocols due to topological changes. An appreciable performance improvement in terms of accuracy and alert times is depicted in their experiments and implementations. Both authors have theoretically proven the expandability of the systems to a wide range of monitoring systems.

An advanced version of Human Activity Recognition from wearable devices is proposed by Xiao et al. [190] which uses a Perception Extraction Network (PEN); a combination of a relation network (Long Short-Term Memory and attention mechanism) and a feature network (Convolution Block). Focusing on both global relations and local features, the PEN outperforms all existing Human Activity Recognition algorithms on four widely used datasets.

Xu et al. [135] highlighted that IoMT networks employed for monitoring and diagnosis systems with FL provide improved security to IoMT devices and the data itself but cannot guarantee model security. This implies that the model may be prone to inversion or inference attacks at the server level. The authors propose a novel quantizer to clip the transmitted parameters while minimizing quantization error. Though minimized, the quantization error still decreases the accuracy of the model by a noticeable margin. In addition, if used for a network with few devices, the overheads become infeasible. The advances integrating IoT and IoMT with FL in healthcare monitoring are discussed in Table 2.5.

Moving to core DSA, a basic implementation of FL for Brain data analysis is proposed by Silva et al. [193] for distributed medical databases. The experimentation is done with both synthetic and real data of neuroimaging used to analyze Parkinson's Disease and Alzheimer's Disease. The results show the consistency of the proposed framework for small-scale imaging but have not yet been tested with large-scale deployments.

Another naïve approach is put forth by Srivastava et al. [130] in their work on detecting in-

Table 2.4: Recent Advances of FL in IoT and IoMT-Enabled Healthcare

Reference	Main Area	Dataset	Performance Metrics						Key Contributions	Limitations
			Acc	Pre	Rec	F1	Sen	Int		
Krall et al. [150]	Healthcare analytics	Self Created Dataset	✓	✗	✗	✗	✗	✗	Gradient-based defense mechanism against inversion attacks	The tuning of the control variable is difficult with small variations leading to steep change in performance of the model.
Lin et al. [2]	IoT-enabled Healthcare	Facebook-like social network [187]	✗	✗	✗	✗	✗	✓	FL system for trust evaluation of users by weighing social footprint	A limited number of attacks are tested, with many prominent attacks are not addressed.
Rehman et al. [127]	Healthcare	Parkinson's disease dataset [188] and NSL-KDD dataset [189]	✓	✗	✗	✗	✓	✗	Hybrid healthcare system based on block-chain and FL	Deep learning–based predictions (RTS-DELM) are not explainable, which limits clinical trust and adoption in sensitive healthcare decision-making
Rehman et al. [186]	Privacy in healthcare data	Self Created Dataset	✓	✓	✓	✓	✗	✗	Approach for side-channel attack detection	The experiments were performed on a minimalistic setup incapable of detecting problems associated with large training networks.
Liu et al. [100]	Confederated learning in healthcare	Self Created Dataset	✓	✗	✗	✗	✗	✗	Method to deal with hybrid partitioning in FL	The model performs weakly against rare diseases while also lacking generalizability.

Table 2.5: Recent Advances of FL in IoT and IoMT-Enabled Health Monitoring

Reference	Main Area	Dataset	Performance Metrics						Key Contributions	Limitations
			Acc	Pre	Rec	F1	Sen	Int		
Wu et al. [42]	In-home health monitoring	MobiAct dataset [191]	✓	✓	✓	✓	✗	✗	PFL approach for remote monitoring	Very limited analysis of personalizability of the model.
Chen et al. [96]	Wearable healthcare monitoring	Heterogeneity Human Activity Recognition [192]	✓	✗	✗	✓	✗	✗	TFL for low-powered wearable device networks	Experiments have very limited users, limiting evaluation of scalability, network heterogeneity, and communication efficiency in large federated deployments.
Xiao et al. [190]	Wearable healthcare monitoring	Heterogeneity Human Activity Recognition [192]	✓	✗	✗	✓	✗	✗	Enhanced Feature Extraction for better performance in Human Activity Recognition	The inclusion of homomorphic encryption provided security but causes notable computational and communication overhead even for just 10 clients.
Xu et al. [135]	Wearable healthcare monitoring	MNIST and Cifar-10	✓	✗	✗	✗	✗	✗	Quantization-based security enhancement with minimum quantization loss	The quantization boundaries remain static during training; dynamic model updates are not explored, leaving potential security and performance improvements.

tracranial hemorrhage using a distributed convolutional network. While the implementation itself shows marginal improvements in accuracy compared to traditional networks (Resnet, DenseNet), FL is just proposed as a future expansion and is not implemented. Nevertheless, the distributed architecture adopted in this work promises further improvements when implemented with FL.

Dayan et al. [131] and Dou et al. [132] took up COVID-19 diagnosis and proposed techniques for predicting clinical outcomes and detecting lung abnormalities respectively. The former's experiments highlighted an increase of 38% in the generalizability of the model when compared with the same model trained without FL. The latter's work evaluates the feasibility of FL integration in COVID-related lung abnormalities and shows significant improvements in accuracy over the traditional central model.

Samuel et al. [32], combined FL with block-chain technology to observe symptoms of covid-19 in patients. The proposed approach attracted researchers to the field as it is of a hybrid nature that also helps identify fake information circulating regarding COVID-19 measures. The work, however, is limited to simulation on empirical calculations as no public dataset is available regarding public awareness. A summary of these works is presented in Table 2.6.

Diving a little further into critical implementations, Ahmed et al. [194], proposed a novel approach for mental health evaluation by combining FL with natural language processing. They used the responses of individuals for the PHQ-9 questionnaire and manual annotations to create a hypergraph-based dataset for training an RNN model. Their work, when compared to existing techniques, showed slightly better results in terms of the F1 score while maintaining privacy. The testing, however, is limited due to the small feature set taken and can be improved in the future by adding more training attributes.

Hossein et al. [117] propose a collaborative FL approach for skin disease identification over decentralized data. This solves significant challenges of dataset size and collaboration hurdles. They demonstrate their experiments over custom-made datasets showing performance at par or better with other existing schemes (VGG-16, AlexNet). Their approach also promises a better security level compared to general FL. Their work provides potential paths for FL in IoMT. The only drawback is the requirement of a certain amount of training data, failing which, the results fall far behind the competing algorithms.

Baghersalimi et al. [102], developed a PFL system for epileptic seizure detection. The global model trained by general FL is retrained over the data of a specific person to provide

Table 2.6: Recent Advances of FL in MRI-based diagnosis and Covid-19 diagnosis.

Reference	**Main Area**	**Dataset**	**Performance Metrics**		**Key Contributions**	**Limitations**
			Accuracy	RMSE		
Silva et al. [193]	Meta-analysis of brain data	Self Created Dataset of MRI Images	✗	✓	FL for distributed databases in brain analysis	The paper does not provide theoretical guarantees or empirical evaluation under non-IID or unbalanced data distributions across sites.
Srivastava et al. [130]	Intracranial Haemorrhage	Self Created Dataset of CT Scans	✓	✗	Federated DNN model for Haemorrhage detection	The study was unable to fully validate their approach to full-scale implementations, which restricts experimental validation and scalability to larger medical datasets.
Dayan et al. [131]	Covid-19 clinical outcomes	Self Created Dataset of EMR Data	✓	✗	Approach for predicting oxygen requirements for Covid-19 patients	The paper utilizes black-box architecture models for experiments, in addition the explainable AI is not fully explored, which limits the interpretability of the results for validation by actual physicians.
Dou et al. [132]	Covid-19 lung abnormalities	Self Created Dataset	✓	✗	FL approach based on multinational collaborative data	Despite being based on multinational data, the diversity and quantity is severely limited, making the approach less robust.
Samuel et al. [32]	Covid-19 healthcare system	Self Created Dataset in collaboration with over 100 Centers for Disease Control	✓	✗	FL-Blockchain hybrid to ensure trust and data immutability.	The paper does not address the scalability challenges, such as increased blockchain transaction volume, communication latency, and network congestion in large distributed healthcare networks.

a more intimate and accurate result. The experiments highlight an increase of around 12% in sensitivity and accuracy compared to general FL. Their work paves the way for highly accurate personalized monitoring systems for rare diseases. However, all the results provided are obtained via simulation only and may present a different set of challenges in real-life implementation. In addition, no prominent step is induced to boost security other than that inherent in FL.

Covering the problem of cerebral ataxia, Ngo et al. [3] put forth an FL-based DNN approach for diagnosis and auto-extraction of the feature set. Based on image transformations, the dataset for the experiments is created using motion sensors over volunteers from geographically separate clinics while maintaining horizontal partitioning. Their experiments show 2% to 3% improvements over traditional less secure methods. The prospects of the implementation are boasted through the integration of block-chain and model stacking for better privacy control.

Nair et al. [129] proposed a collaborative framework based on FL for detecting facial paralysis. The approach uses the inherent privacy preservation of FL in a multi-client system among hospitals sharing image data while preserving patients' identities. The tests conducted show an increased convergence rate and slightly improved accuracy over MNIST and covid-pneumonia datasets. However, no additional measure is taken to improve the privacy and security of data other than those inherent in FL. A summary of these frameworks is presented in Table 2.7

A similar method for larynx cancer survival prediction is put forth by Ronn Hansen et al. [128]. The method is trained and tested across three centers with local datasets varying in size and sampling. The results demonstrate a well-calibrated model with high power of discrimination. The approach, similar to the work in [114], does not undertake any additional measure to boost privacy beyond that inherent in FL. Liu et al. [198] proposed an FL-based Vertebral Body Segment Framework (FLVBSF) to create highly accurate vertebral segmentation models for spinal surgery. The framework proposed enabled collaboration between hospitals and clinics, which helped acquire a substantial dataset of MRI images for effective learning. This was previously limited due to a smaller number of required images at an individual level. The approach again is based on naïve FL, and no additional security measure is adopted.

Switching to miscellaneous works in healthcare, Gudur et al. [22], in their work, propose a collaborative FL approach for human activity recognition, also dealing with statistical and

Table 2.7: Recent Advances of FL in Critical Illness Diagnosis.

Reference	Main Area	Dataset	Performance Metrics						Key Contributions	Limitations
			Acc	Pre	Rec	F1	Sen	RMSE		
Ahmed et al. [194]	Mental health analysis	Self Created Dataset using PHQ9-questionnaire	✗	✗	✗	✓	✗	✗	Hybrid approach with Natural language processing and FL for mental health diagnosis.	The performance can be improved only through "fine-tuning of hyperparameters" and "longer training", indicating sensitivity to configuration, which can limit scalability.
Hossen et al. [117]	Dermatology	Self Created Dataset [195]	✓	✓	✓	✗	✗	✗	The proposed model outperforms existing models with a high level of privacy.	The model requires 500 epochs and extensive hyperparameter tuning, implying long training times and resource-intensive computation, which may not be feasible for all IoMT devices.
Baghersalimi et al. [102]	Epileptic seizure detection	Epileptic seizures dataset [196]	✓	✗	✗	✓	✓	✗	FL framework for low-power embedded devices.	The training data is not cross-validated, and the real-time feasibility is theoretical, not experimental at the time.
Ngo et al. [3]	Diagnosis of Cerebellar Ataxia	Self Created Dataset	✓	✗	✗	✗	✗	✗	Auto-crafted feature extractor for image classification.	The patients for rare disorders like cerebellar ataxia may be small, which may result in over-fitting, bias, and lack of robustness in real-world deployments.
Nair et al. [129]	Facial paralysis	MNIST and Covid-Pneumonia [197]	✓	✓	✓	✓	✗	✗	FL-based collaboration framework for facial paralysis diagnosis	The datasets used for experiments are relatively small scale, and do not represent the massive, heterogenous, and noisy data of real-world hospitals.

model heterogeneity amongst end devices. They use label-based aggregation to solve non-IID challenges. In addition, they propose the transfer of model scores in place of weights to update the global model. Their experimental technique shows performance gains of up to 11%. Since the approach is based on label aggregation, if multi-dimensional labels are somehow obtained by an attacker, it can cause privacy concerns for an otherwise efficient mechanism.

Rahman et al. [199] propose another implementation of naïve FL to predict the length of stay of patients in a hospital to better manage resources. FL helps reduce the reluctance for collaboration amongst hospitals and enables collaborative learning over distributed datasets. The approach does not involve anything additional to strengthen security or privacy besides that inherent in FL.

Zhu et al. [23] proposed an FL technique integrating Graph Neural Networks to tackle data heterogeneity problems in molecular studies. The data required for such a study is quite expensive as much as $50 million USD, hence public contribution from institutes for data is not possible due to the intrinsic value itself. FL removes this barrier by allowing collaborative learning without sharing data. The approach is tested over a number of different datasets, as shown in Table 6, to simulate heterogeneity in data. The results show statistically significant margins of improvement in the model's sensitivity over the traditional approach. However, this implementation does not integrate anything other than native FL to boost security and privacy. A summarized view of these critical illness diagnosis studies in FL-based healthcare is provided in Table 2.8.

Healthcare, without any doubt, takes the lead spot for integrating federated learning in its practices toward secure and private applications. However, the data-sensitive group has many member fields that do not have the same level of seriousness given to them regarding such research. The field of finance ranks second to healthcare, and while one may think that being at such a spot will see similar, if not a bit less effort from researchers, it's not true.

2.2.2 FL in Finance

Compared to medical implementations of simple, complex, and hybrid learning models, finance has been struggling to accept FL into its work, as evidenced by the bare minimum work done. The most prominent examples of these works are summarized in Table 2.9.

Starting with the work of Byrd et al. [205] which provides a generalized approach for secure computations applicable to most financial applications. The proposed protocol amalgamates

Table 2.8: Recent Advances of FL in Critical Illness Diagnosis and Misc.

Reference	Main Area	Dataset	Performance Metrics						Key Contributions	Limitations
			Acc	Pre	Rec	F1	Sen	RMSE		
Ronn Hansen et al. [128]	Larynx Cancer Survival	Self Created Dataset	✓	✗	✗	✗	✗	✗	Survival prediction model for larynx cancer based on open-source FL	The model needed a recalibration (scaling update) when applied across different centers, indicating that the model's baseline hazard or risk-calibration did not generalize fully without adjustment.
Liu et al. [200]	Vertebral body segmentation	Spine segmentation in 3d MRI [201]	✓	✗	✗	✗	✗	✗	Robust FL model for vertebral segmentation analysis/diagnosis.	The model depends heavily on the balance between iteration count and participating nodes in each round, harming robustness of the proposed approach.
Gudur et al. [22]	Statistic heterogeneity	Heterogeneity Human Activity Recognition [192]	✓	✗	✗	✗	✗	✗	Method to deal with heterogeneity in labels in on-device FL.	Many real mobile activity recognition deployments have extreme heterogeneity at the hardware level: different sensors, sampling rates, user behaviours, missing data, and device drop-outs that are not taken into account in performance.
Rahman et al. [202]	Patient-stay prediction	Open-source dataset of hospital administrative information [203]	✓	✗	✗	✗	✗	✗	FL approach to increase resource and quality management based on patient stay predictions.	Evaluation is focused on the predictive accuracy of LOS. Other important metrics, like fairness across different patient groups or interpretability of predictions are not deeply analyzed.
Zhu et al. [23]	Study of molecular properties	FreeSolv, Lipophilicity, ESOL, QM9, Tox21, SIDER, ClinTox, BBBP, and BACE [204]	✓	✗	✗	✗	✗	✓	FL A novel implementation of graph NN in a heterogeneous setting.	N.A.

the concepts of DP and Multi-Party Computation (MPC) to provide a substantial increase in privacy level. DP provides encryption to model parameters while MPC provides an additional layer of anonymity to the data. The simulations are done over a real-world dataset for credit card transactions, and the results put forth in the work highlight the efficiency against inference and inversion attacks.

Moving to the subfield of accounting, Schreyer et al. [206], proposed an FL-based statement audit approach with DP and split learning. The approach is tested with three accounting datasets from different cities, injected with a small fraction of global and local anomaly data points. The results show the anomaly detection performance is directly proportional to the number of model parameters shared when the scope shifts from local to global. This restricts the highest precision attainable by the model to the consent of the user as to how many model parameters can be shared in collaboration.

Lv et al. [5] proposed a detection system based on cross-industry fraud. The authors pointed out the norm that financial fraud goes through a number of exchanges between social and financial circles. Without collaboration between industries for a combined dataset, no proper training can be achieved. Hence an FL technique is proposed to reduce the reluctance and enable training over socio-financial data. The results show an improvement of 1% over centralized approaches. However, no substantial work is done except to implement FL directly, hence, further research is required to guarantee the security levels desired.

Du et al. [123], applied FL as a monitoring and detection system over the decentralized transactions in digital currency for malicious interchanges. The authors propose graph-based learning over a federated architecture to leverage the relationships between nodes and transactions, which is generally ignored. The approach also prevents the risk of leakage of feature data that can occur in traditional systems. When compared to logistic regression and multilayer perceptron, the proposed model not only outperforms both but also trains in fewer epochs. Currently, the experiments are only done over Bitcoin, and further experimentation with other types of digital currencies is required to make the system more robust and reliable.

Shingi et al. [180] extended FL to predict possible loan defaults. Given the highly unbalanced data availability for the field, the authors applied a synthetic minority oversampling to balance out the data points. The proposed implementation motivates collaboration between banks to better train the predictor by ensuring the privacy of their data by localization. The results show slight improvements over centralized and FedAVG approaches. Being a simple FL approach with a slightly modified aggregator, the system is susceptible to heterogeneity,

which limits the scalability of the approach.

Myalil et al. [4], proposed a robust transaction evaluating FL scheme. Adding to the data localization, the authors implement an aggregator that evaluates participants for malicious intent. While considering the non-IID nature of collaboration. The results based on the data simulations show marginal improvements, increasing with the increase in the number of malicious participants. The model is shown to be adaptive in nature to preserve the quality of the process. Since the data publicly available for such an endeavor is limited, a synthetic dataset made via paysim is used, which prevents us from evaluating practical challenges and impacts effectively. Another scope of upgrade is identified by the authors, as the model requires continuous input from poisoning nodes, which results in ineffectiveness against stealth or short-term poisoning.

Another attempt at promoting collaboration between industries is put forth by Gupta et al. [78], where they propose a risk prediction model for life insurance based on FL. Life insurance coverage depends on a number of elements from different industries, such as banking, healthcare, etc., which makes the use of federated learning more than feasible. The experiments conducted show performance close to traditional implementations and hence validate the possibility of advanced techniques over distributed learning. Since the experimental data is linearly distributed, the averaging algorithm is straightforward, and no results with uneven or biased data are present. A summarized view of applications in the Finance Sector is presented in Table 2.9.

2.2.3 Potential Upgrades for FL in DSA

Apart from these works, there is little to no work done toward DSA in other fields. This, however, does not imply that other implementations cannot benefit the focused applications. Tables 2.10, 2.11, 2.12 and 2.13 lists the FL implementations, which, while not directly affecting DSA, provide us with a base to gather inspiration and build upon.

Starting with the works focusing on resource allocation and optimization. Yang et al. [215], took into notice, the increase in computational power of wireless devices and the possibility of applying FL in this environment. The authors propose a protocol for the wireless implementation of the same while highlighting the issues of transmission noise and the dynamic nature of the setup. Three scheduling algorithms are evaluated for providing resource optimization in this setup, which forms the basis of the study (Random Scheduling, Round Robin, and Proportional Fair). However, since the number of algorithms is very small, no

Table 2.9: Recent Advances of FL in the Finance Sector

Reference	Main Area	Dataset	Performance Metrics					Key Contributions	Limitations
			Acc	Pre	Rec	F1	Kappa		
Byrd et al. [205]	Financial applications	Credit Card Fraud Detection [207]	✓	✗	✗	✗	✗	Multi-Party Computation in Finance with Differential Privacy.	The computation cost of MPC can be high resulting in limited scalability of the implementations.
Schreyer et al. [206]	Financial statement audit	Philadelphia payments [208], Chicago payments [209], and York Payments [210]	✗	✓	✗	✗	✗	FL system to process multi-client audits autonomously and privately.	Different clients may have different accounting systems, formats, transaction types, which may make federated modeling challenging.
Lv et al. [5]	E-Banking fraud detection	Self Created Dataset	✓	✓	✓	✗	✗	Novel method to apply VFL for fraud detection.	The convergence and communication analysis is very limited in addition to very few contemporary algorithms being tested.
Du et al. [123]	Digital Currency	Elliptic Dataset [211]	✓	✓	✓	✓	✗	Method to detect malicious transactions in a decentralized setup.	The approach is not tested with any type of privacy attack and the baseline comparisons are fairly limited.
Shingi et al. [212]	Loan defaults prediction	The dataset released as part of LTFS Data Science FinHack-2019 [213]	✓	✓	✓	✓	✗	Method to deal with privacy issues with collaborating banks and data imbalance.	The tested model has a linear lightweight architecture that is not at par with state-of-the-art models as well as the simulation works on assumption that all clients have 100% response and participation that is not possible in the real-world.
Myalil et al. [4]	Fraudulent Transactions	Self Created Dataset via Paysim [118]	✗	✗	✗	✓	✗	FL approach to deal with malicious nodes in the presence of non-IID data.	The work acts as a proof of concept and not a practically tested implementation at any level with minimal analytical information presented.
Gupta et al. [81]	Risk prediction for life insurance	Prudential Life Insurance Assessment [214]	✗	✗	✗	✗	✓	The paper showcases utility of FL in the insurance industry to train models without compromising data privacy.	The field of insurance presents high data heterogeneity and varied audits that complicate and limit the integration of FL in current the system.

proper inference can be made from the data. Another attempt at resource optimization is presented by Nguyen et al. [37] by utilizing a multi-access edge server. The authors propose a resource-sharing model for CPU cycles and bandwidth among simultaneous FL services. Their approach shows significantly less energy consumption and optimized resource utilization in simulations. However, the experimentation is not tested in real-life scenarios and may result in a different or less efficient outcome.

Zeng et al. [216] take the problem of communication efficiency & resource optimization and propose an Over-The-Air (OTA) mechanism for computation over Edge. The mechanism is implemented to improve global aggregation times in VFL. Their approach aims at theoretically minimizing the Mean Squared Error (MSE) in OTA computation to reduce the optimality gap. This is achieved by superiorizing the bounded perturbation with a convergence guarantee. The simulations show faster computation times with smaller MSEs than SemiDefinite Relaxation (SDR) and Successive Convex Approximation (SCA) algorithms. The approach is defined with respect to basic VFL implementations and may perform differently in complex scenarios.

Taking resource optimization further, Zheng et al. [217] proposed compressing model parameters from a high-dimensional tensor space to a low-dimensional subspace. This approach results in reduced bandwidth consumption and storage requirements for FL. In addition, a novel update method based on hierarchical tensor backpropagation is developed to reduce the memory requirement, improving training efficiency in Edge nodes. The improvements, however, come at the cost of precision reductions in the overall results.

Li et al. [155] fused the resource optimization scheme with workload estimations and proposed an adaptive FL framework over AL for resource-constrained setups. The workload assigned to a node in the network is actively calculated based on the average loss in results from the last training round. This in turn accelerates the convergence of the model. The authors propose a q-FFL mechanism to provide adaptive robustness against heterogeneity in data. However, the convergence obtained in the proposed algorithm incorporating q-FFL is slightly slower than FedAVG systems without straggler nodes that bottleneck training, hence showing a small degradation.

Zhu et al. [112] recognized the shift in focus from 5G to 6G due to the commercialization of the former. They proposed a green energy-efficient mechanism over an incentive and reputation-based reward system for short-term and long-term incentive calculations. The approach is tested with IID and non-IID setups and highlights robustness in both.

Yang et al. [191] shifted their approach from resource optimization to model optimization to deal with the invalid parameters created in VFL. They integrate homomorphic encryption and gradient compression schemes to optimize the transmission load. The loss of information is avoided by grouping the unsent information and relaying it when it gains a substantial size. The approach shows on-par accuracy with traditional methods but currently has a very limited base of applications.

The various research that can be implemented in DSA for resource and model optimization are summarized in Table 2.10

The authors in [77, 78, 107, 108, 110] highlight various incentive-based approaches for IFL implementations, including reputation evaluation, interactivity, blockchain, per-epoch contribution, and RL. Each of these works is either in a very early stage or has substantial overheads associated as shown in Table 2.11.

A number of implementations based on DP and homomorphic encryptions are proposed by Choudhury et al. [192], Malekzadeh et al. [193], Zhu et al. [194], Brunetta et al. [99], Kerkouche et al. [195], and Wu et al. [97] individually. The approaches successfully boost privacy at different levels, but all the simulations highlight a heavy gain in overhead, which may or may not be feasible depending on the target application. The approach in [195] also shows slight vulnerability from server-side attacks.

Moving to general utility, Gamboa-Montero et al. [219] proposed a new framework for incremental learning in human-robot interaction. The simulations show promise for the future, but since the approach is new in an already new paradigm, accepting it in general FL will take substantial time. A summary of all the non-DSA implementations for future improvements in privacy and utility is provided in Table 2.12.

FL, in most cases, uses a centralized structure for resource allocation and global model aggregations. Li et al. [31], in their works, highlight the issue of single-point failure that can occur in a centralized environment. They propose a decentralized approach by integrating blockchain technology while also analyzing the impact of malicious lazy nodes in the network. Their approach shows a minor decrease in performance with increasing concentration of such nodes and presents a promising future research direction in the security of FL. However, the testing is done on simple datasets only, and no real-life or complex dataset results are present as of now for the implementation's performance.

So et al. [196] propose a modified aggregation scheme to simplify the time complexity of

Table 2.10: Adoptable Techniques from FL Implementations suitable for adoption in DSA for Resource and Model Optimization.

Reference	**Benefiting Area**	**Techniques Applied**	**Key Results**	**Limitations/ Potential Improvements**
Yang et al. [215]	Resource Allocation and Optimization	FL, round robin, random scheduling, and proportional fair	Impacts scheduling schemes have on the convergence rate of FL.	Only three scheduling schemes are considered for the experiment.
Nguyen et al. [37]	Resource Allocation and Optimization	FL, Virtualization, and MS-FEDL (proposed)	Approach for resource sharing across simultaneous FL services.	No testing corresponding to DSAs.
Zeng et al. [216]	Resource Optimization and Utility	FL and 5G//6G networking.	OTA computation for faster aggregation in FL.	The work is currently simulation-based only.
Zheng et al. [217]	Resource Optimization	FL and edge computing.	Compression approach to reduce bandwidth requirements for FL-based Edge networks.	The approach shows promise for IID data, but suffers in the case of non-IID data.
Li et al. [155]	Resource Optimization	FL	Workload-based client selector for heterogeneous systems.	N.A.
Zhu et al. [110]	Resource optimization and Incentive awareness	FL and 5G/6G networking.	Reputation-based contribution evaluation for energy-efficient FL systems with 6G technology.	The work is currently simulation-based only.
Yang et al. [218]	Model Optimization	FL and homomorphic encryption.	Improvements in modeling speed and generalization in FL.	The approach is tested over textual data only at the time of writing.

Table 2.11: Adoptable Techniques from FL Implementations suitable for adoption in DSA for Incentive Aware Applications

Reference	**Benefiting Area**	**Techniques Applied**	**Key Results**	**Limitations/Potential Improvements**
Kang et al. [108]	Incentive Awareness	FL and Block-chain	Reputation-based selection scheme for clients using Block-chain.	The set of parameters for reputation calculations can be expanded further for improved accuracy.
Zhao et al. [77]	Incentive Awareness	FL, convolutional neural network (for HFL), and Linear regression (for VFL).	Approach for weighing individual user contributions in terms of interactive-ness and resource-saving.	The approach can be modified to reward participation to attract more nodes and motivate existing ones.
Wang et al. [107]	Incentive Awareness	FL and Blockchain.	Decentralized incentive-aware FL to prevent single-point failures.	N.A.
Wang et al. [78]	Incentive Awareness	FL	Dynamic contribution evaluation of clients on a per-epoch basis and robustness against non-IID data.	N.A.
Lee et al. [110]	Incentive Awareness	Federated Reinforced Learning (FRL).	Approach exploiting device scores calculated in RL to provide an incentive system in FRL.	Experiments are done in a controlled setup. Further tests in different IoT setups are needed for a complete evaluation.

Table 2.12: Adoptable Techniques from FL Implementations suitable for adoption in DSA for Privacy Enhancement and Utilities.

Reference	**Benefiting Area**	**Techniques Applied**	**Key Results**	**Limitations/Potential Improvements**
Choudhury et al. [220]	Privacy-Enhancement	FL and DP	Trade-off present between privacy enhancement and learning accuracy.	The rate of convergence is not verified in experiments.
Malekzadeh et al. [221]	Privacy-Enhancement	FL, SqueezeNet, and parallel differentially private stochastic gradient descent.	Method to balance between the guarantee of differential privacy and model accuracy.	High overhead in computation.
Zhu et al. [222]	Privacy-Enhancement	FL	Approach for preventing leakage of data labels in multi-owner systems of VFL.	The involved encryptions introduce a substantial overhead in training and learning times.
Brunetta et al. [105]	Privacy-Enhancement	FL, DP, and homomorphic encryption.	Non-interactive, verifiable, and secure aggregation scheme.	N.A.
Kerkouche et al. [154]	Privacy-Enhancement and optimization	FL and DP	Mechanism with privacy-enhanced resource allocation.	The mechanism may be vulnerable to some Server-side attacks based on Deep RL.
Wu et al. [103]	Personalization and privacy enhancement	FL and graph neural network.	A method to protect the user interaction history with the aggregation server.	N.A.
Gamboa-Montero et al. [219]	Utility	FL	An interactive mechanism for incremental learning.	Based on human-robot interactions, much work is needed to translate for generalized FL.

aggregation from traditional $O(n^2)$ to $O(nlog(n))$ with a tolerance for 50% client dropout. The results of the experiments showed a forty times faster performance while verifying the dropout tolerance of the mechanism using LaGrange coding. In addition to this, the guarantees of privacy, robustness, and lower overheads are also verified successfully. Apart from the experimentation, the approach lacks a real-world implementation to fully test the theoretical claims.

Rey et al. [197] study in detail the impact of malicious attacks on FL in IoT setups and propose a malware detection scheme to secure the network. Though the scheme does not hamper the accuracy, the study of adversarial attacks is limited to supervised learning scenarios only.

Yang et al. [93] proposed an FTL scheme (FedSteg) to extend the utility of FL to the steganalysis of encrypted images. The mechanism boosted the security of the data manifolds as shown by the results compared to state-of-the-art techniques (WoW, S-Uniward, hILL).

Rodriguez-Barrosso et al. [149] proposed a dynamic aggregator to tackle byzantine attacks on FL implementations. The approach evaluates and discards nodes with either suspecting behavior or poor contribution to the overall training. This results in two prominent accomplishments of the experiments: the defense against Byzantine attacks without harming the model accuracy, and improving accuracy by filtering out weak nodes from the aggregation. The limitation, however, is that the approach is ineffective if the number of malicious or weak nodes is very low.

Tekgul et al. [103] shifted focus to model theft and proposed a watermarking scheme to highlight ownership over the model. The experiments show negligible impact on the FL process in the case of a single owner. However, if multiple owners exist in the case of a partnership or shared structure, the scheme may induce substantial overheads.

A summary of all implementations that may enhance transmitted gradient security or model security is provided in Table 2.13

2.2.4 Tools and Frameworks

- **Pysyft** was proposed by Ryffel et al. [225] as the first standard that enabled FL to implement with support for both Multiparty Computation (MPC) and Differential Privacy (DP) in a resource effective way. The protocol is implemented through the PySyft framework, which is compatible with TensorFlow [226] and PyTorch [227]

Table 2.13: Adoptable Techniques from FL Implementations suitable for adoption in DSA for Gradient and Model Security.

Reference	**Benefiting Area**	**Techniques Applied**	**Key Results**	**Limitations/Potential Improvements**
Li et al. [31]	Security	Block-chain and FL.	Improvements in modeling speed and generalization in FL.	The approach is tested over textual data only at the time of writing.
So et al. [223]	Security	FL and Turbo-Aggregate (proposed)	Method for Secure aggregation over wireless topologies.	The results reported are not very detailed.
Rey et al. [224]	Security	FL	Malware detection in IoT network devices in both supervised and unsupervised learning.	The impact of adversarial attacks is studied for supervised scenarios only.
Yang et al. [97]	Security	FTL and steganography.	FTL approach for secure steganalysis.	N.A.
Rodriguez-Barrosso et al. [149]	Security	FL, DL, and adversarial attacks	Dynamic selection scheme for adversarial and weak client devices in FL.	The approach is less effective when adversarial clients are in fewer numbers.
Tekgul et al. [118]	Model security	FFL and watermarking.	Watermarking approach for demonstration of ownership to prevent model theft.	The approach may induce respectable overhead in case of shared ownership of the model.

libraries, making it popular amongst developers.

- **TensorFlow Federated** is an open-source framework for machine learning over decentralized data [228]. It is developed to promote open research and experimentation with FL. It enables users to simulate various predefined federated learning algorithms on their models and data, and can incorporate custom algorithms. A predefined set of models and algorithms also helps implement expanded FL applications, such as federated analytics.

- **Nvidia Clara** is a suite of computing solutions developed by Nvidia to empower AI in the medical domain [198]. The edge AI computing platform Nvidia EGX in Clara is designed to integrate APIs and SDK to support FL with relative ease in healthcare applications such as those discussed in [6, 7].

- **OpenFL** is a community-driven open-source initiative developed to provide a plugin-type functionality for AI libraries in Python3 [199]. Like Pysyft, the OpenFL framework seamlessly integrates with both TensorFlow and PyTorch. The tool has extended Flash and HTML5 content support and is entirely portable.

- **FATE** is another open-source project developed by Webank aimed at supporting a federated ecosystem of AI applications [229]. It provides a modular approach with ready-to-use libraries that simplifies the development process for new and learning developers. It supports MPC and big data protocols with FL-enabled libraries for graphics pipelines, interfaces, and scheduling.

- **Substra** [230] is an open-source framework developed by Owkin and hosted by the Linux Foundation for AI. It doubles as a simulator for FL-based testing on a single-node setup and has a native web application to facilitate FL at a large scale.

- **IBM Federated Learning** is new and innovative sollution developed by IBM [231]. Its prominent achievements include optimizing and compressing models before uploading to the central FL server, which minimizes bandwidth utilization. Another pivotal example is the "DeTrust" framework, which employs consensus-based cryptography to securely transmit model parameter vectors.

- **PaddleFL** is based on Paddle, a deep learning platform developed by Baidu [232]. It supports both DP and MPC. It supports most state-of-the-art algorithms, such as FedAvg. But at the time, there is no support for Vertical FL algorithms. The

development of PaddleFL is still in its early stages, and additional support for new technologies may be introduced in the future.

- **FedML** is an FL framework with support for native benchmarking. Based on PyTorch, a team developed it at the University of Southern California [233]. Algorithms supported by FedML do not consider adversaries, but the support for DP is integrated as a safety measure. It supports simulation, distributed computing, and single-node implementation.

A summary of all popular Frameworks for FL is provided in Table 2.14.

Table 2.14: Popular Frameworks for FL integration

Name	Open-Source	Pytorch support	Tensorflow support	Optimizable	Secure
Pysyft [225]	✓	✓	✓	✗	✓
TensorFlow Federated [228]	✓	✗	✓	✓	✓
Nvidia Clara [198]	✗	✓	✓	✓	✓
OpenFL [199]	✓	✓	✓	✓	✓
IBM Federated Learning [231]	✗	✗	✗	✗	✓
FATE [229]	✓	✓	✓	✓	✓
Substra [230]	✓	✓	✓	✓	✓
PaddleFL [232]	✓	✓	✓	✓	✗
FedML [233]	✓	✓	✓	✓	✗

✓ represents coverage, ✗ and represents non-coverage

For our research and frameworks, we have utilized TensorFlow Federated and PySyft frameworks for their secure and open-source nature.

2.2.5 Analysis of Dataset Availability for DSAs

Given the nature of data associated with DSAs, most of the work done in the field is implemented over either a private self-accumulated dataset or a non-DSA dataset available publicly (CIFAR-10, MNIST, FMNIST). A fraction of work is done on datasets relevant to the field with sensitive user data in synthetic or anonymized formats available in the public domain. Resonating with different forms of input (Image, Text, Signals. etc.), a summary of the characteristics of these datasets is provided in Table 2.15.

The publicly available datasets focus on Healthcare and Finance data. A large chunk of Healthcare related data is available in the form of images of varied formats [177–180, 184, 185, 197, 201], used for diagnosis in critical illness and basic healthcare systems. The rest of the datasets available are either text-based [182, 183, 196], or voice-based [188]. Moving to Finance, given the transactional and accounting nature of the field, all the datasets available (synthetic or otherwise) are of textual formats [207–211, 213, 214] (CSV, txt, etc.). Apart

from Health and Finance, some social network-based datasets [220] in the form of network graphs (Tnet, UCInet) also exist.

It is important to note that none of these datasets is of a distributed nature. Since a benchmark distributed dataset is not available, these are used for simulating real-world scenarios by dividing them into smaller datasets representing the private local data of client nodes. These subsets are induced with imbalance and bias using augmentation at different levels.

Table 2.15: Characteristics of Public Datasets in Healthcare Corresponding to DSA

Reference	**Year**	**Field**	**Datapoint Type**	**Size (Elements/ File-size)**	**Key Attributes**
Parkinson's disease dataset [188]	2008	Healthcare	Text and Voice	197 / -	MDVP:Fo(Hz), MDVP:Fhi(Hz), MDVP:Flo(Hz), MDVP:jitter(%), MDVP:Jitter(Abs), NHR, HNR, Label
Lung Adenocarcinoma (LUAD) [182]	2012	Healthcare	Text & Binary (txt, maf, vcf, etc.)	28,562 / 134.96TB	TCaseID, Gender, Mutations, Genes, Slides
Diabetic Retinopathy [177]	2015	Healthcare	Image	- / 88.29GB	Image, Level
Lung Squamous Cell Carcinoma (LUSC [31])	2016	Healthcare	Text	Binary (XML, maf, vcf, etc.)	25,600 / 3.32PB & CaseID, Gender, Mutations, Genes, Slides
Prostatex Dataset [179]	2017	Healthcare	Image	- / 15.1GB	ProxID, studydate, ImageUID, DIM, VoxelSpacing, SpacingBetweenSlices, etc.
RSNA Bone Age [180]	2017	Healthcare	Image	4800 / 9.97GB	CaseID, Sex
Automated Cardiac Diagnosis Challenge (ACDC [185])	2018	Healthcare	Image (Nifti)	150 / -	Height, Weight, ExamID
Epileptic seizures dataset [196]	2018	Healthcare	Text	11500 / 7.64MB	Response Variable, Explanatory variable
Spine segmentation in 3d MRI [193]	2018	Healthcare	Image	200 / 1.21GB	Ground Truth Image, Spinal MRI Image
Chest Xray: Covid-Pneumonia [197]	2020	Healthcare	Image	6,432 / 2.3GB	Image Labels
M&M [184]	2020	Healthcare	Image (MRI Scans)	375 / -	Country, ImageID
Chexpert [178]	2020	Healthcare	Image (Radiographs)	2,24,316 / 11.47GB	Image Name
Facebook-like social network [187]	2011	Social	Tnet, UCINET	1899 Users 59835 Messages	Gender, Age, Course Attended
Chicago payments [209]	2011	Finance	Text	Data Updated Daily	Vendor_Name, Amount, Department_Name, Voucher_Number
York payments [210]	2012 -2023	Finance	Text	23388 / 5MB (2022 Version)	Supplier_Id, Expense Area, Expense Type, Amount
Credit Card Fraud Detection [207]	2013	Finance	Text	2,84,807 / 150MB	Time, Amount, V1-V28
Prudential Life Insurance Assessment [214]	2015	Finance	Text	- / 3.4MB	Id, Product_Info_1-7, Employment_Info_1-6, InsuredInfo_1-6, Insurance_History_1-9, Family_History_1-5, Medical_History_1-41
Philadelphia payments [208]	2017	Finance	Text	Data Updated Daily	Vendor_Name, Transaction_Amount, Sub_Obj, Sub_Obj_Title
Elliptic Dataset [211]	2019	Finance	Text	2,03,769 Nodes- 2,34,355 Edges / 697.43MB	Time step, No. of Input/output, Transaction_fee, Incoming transactions, BTC received/sent, Std Deviation, Correlation coefficients
Dataset released as part of LTFS Data Science FinHack-2019 [213]	2019	Finance	Text	143400 / 113.52MB	PRI NO OF ACCTS, PRI ACTIVE ACCTS, PRI OVERDUE ACCTS, PRI CURRENT BALANCE

We have experimented and simulated our proposed frameworks with image datasets from the field of healthcare including RSNA BoneAge, Chexpert, and Diabetic Retinopathy along with some standard datasets like MNIST, FMNIST, EMNIST, and Cifar-10.

2.2.6 Real-world Case Studies of FL

In this section, we will go over some instances from the real world where FL is employed to deal with sensitive data.

1. Collaborative FL Project for Brain Tumor Detection (Intel Labs and University of Pennsylvania)

   The Intel Labs collaborated with the University of Pennsylvania and many other medical institutions in 2018 [234] to enhance security in medical imaging-related learning. To eradicate the need to share raw data, the project team applied federated learning over a highly expanded/collaborative version of the BraTS [181] dataset, having more than 25000 MRI scans, spanning over 5TB across the network of 70 geographical centers in six continents. Intel SGX and OpenFL were used in an open-source library OS environment. The result comparison shows 33% improvements in accuracy over the standard ML/DL models trained on a centralized level.

2. FL for Contextual Sensitive Data in Healthcare (Ministry of Trade, Industry and Energy and Korea Institute for Advancement of Technology under the International Cooperative R&D program)

   Jeon et al. [235] highlighted the need for automation in contextual data quality assessment in local sites of federated learning environments to identify patterns of incorrect or malicious data. The joint venture proposed a contextual data quality assessment mechanism to identify the feasibility of a local model fitted for purposes of healthcare applications by using DL models and semantic ontologies. The mechanism considered five parameters Value added, Relevancy, Timeliness, Completeness, and Appropriate amount of data to judge and segregate the available models best suited for sensitive applications. The approach was tested on a self-generated Ontology and demonstrated appreciable performance.

The case studies and experiments studied in this manuscript provide an understanding of the infrastructure required for minimal implementation of secure FL. The major requirements are summarized as follows:

- At least one moderately powered server for aggregation in network patches or the whole network centrally. The number may change with the size of the network.

- The central server present in low-powered networks like IoMT should be powerful enough to compensate for the limited computing power of the nodes.
- The data available in the various connected networks/ data islands should provide data substantially different than what is available at other islands in the network to assure a good quality training value and reduce redundancy.
- Most importantly, a stable network with a high bandwidth is required to minimize packet drops and straggling during training rounds.

The Taxonomy of the existing federated learning implementations based on their types, supporting technologies, area of applications, and security measures is shown in figure 2.1

2.2.7 Research Gaps and Problem Analysis

Although the existing literature provides extensive coverage of federated learning and its supporting technologies, most prior work addresses security, privacy, resource efficiency, and scalability in isolation rather than as an integrated solution. Studies focusing on Differential Privacy primarily address inference attacks but often overlook communication overhead and system heterogeneity. Similarly, works on Homomorphic Encryption emphasize secure aggregation but introduce significant computational complexity, limiting practical deployment in resource-constrained environments. Blockchain-based and cryptographic approaches enhance trust and integrity but typically impose substantial latency and bandwidth costs. Furthermore, limited research simultaneously considers straggler mitigation, workload awareness, lightweight cryptography, and hybrid privacy-preserving mechanisms within a unified framework. Consequently, a clear research gap remains in designing scalable, resource-aware, and security-integrated federated learning frameworks that balance privacy guarantees, computational efficiency, and real-world deployability.

Figure 2.1: Proposed Taxonomy

Chapter 3

Workload-aware secure federated learning framework with quantization

*The purpose of this chapter * is to introduce federated learning based secure hybrid frameworks that combine the paradigm of Workload-aware federated learning with quantization. Such a system optimizes the resource utilization while also lowering the upper bound on the minimal required resources in a federated learning implementation. It is vital to evaluate these hybrid frameworks with the state-of-the-art on multiple metrics, such as time overheads, resource utilization, and performance of the trained model, to highlight the utility of the proposed approaches for resource-constrained training networks.*

3.1 Introduction

The basics of FL and DSA have been covered in detail in Chapter 2. In this chapter, we propose two hybrid frameworks corresponding to the second objective of this research, which implement a special type of FL, Incentive-Aware FL (IFL), which motivates client nodes to actively participate in and contribute to global training. We also incorporate quantization-based parameter encryption to increase the security quotient of the proposed frameworks. The basics of IFL and quantization are discussed as follows.

*The part research work covered in this chapter has been published/accepted/communicated in: Manu Narula, Jasraj Meena, and Dinesh Kumar Vishwakarma, "Dynamic Resource-Aware Federated Framework for Secure and Sustainable Learning in Data Sensitive Applications," 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), Pune, India, 2024, pp. 1-8, 2025,https://doi.org/10.1109/I2CT61223.2024.10543390.
&
Manu Narula, Jasraj Meena, and Dinesh Kumar Vishwakarma, "Federated workload-aware quantized framework for secure learning in data-sensitive applications," *Future Generation Computer Systems*, Elsevier, vol. 168, pp.107772, 2025, DOI: https://doi.org/10.1016/j.future.2025.107772. **(SCIE, IF=6.1)**

Terminology and Key Concepts for Chapter 3 For clarity and readability, key terms used throughout this chapter are briefly defined. *Quantization* refers to the process of reducing the precision of model parameters or gradients by representing them with fewer bits, thereby decreasing communication cost and computational burden in federated learning environments. In this work, quantization is primarily used as a resource-efficiency mechanism to reduce bandwidth consumption and mitigate straggler effects in heterogeneous client networks. The *security context* in this chapter refers to protecting model updates and training stability in resource-constrained environments, where traditional heavy cryptographic techniques may not be practical. Accordingly, the proposed framework aims to balance communication efficiency, training stability, and basic confidentiality while maintaining acceptable model accuracy.

3.1.1 Incentive-Aware FL (IFL)

IFL [107, 108, 110, 236, 237] is a unique form of FL that tracks contributions from various devices. This record is used to allot rewards and workloads to the devices in subsequent rounds, providing motivation for computation nodes and optimizing the overall training performance of the network. The associated reward system is also employed to attract new nodes to the network and refresh dead or idle nodes to participate.

Fig. 3.1 shows the essential organization of an FL setup in homogenous data distribution, and Fig. 3.2 shows the same for heterogeneous data distribution. Multiple client devices (mobile, laptop, tablet) communicate with a cloud/edge server to synchronize model parameters. These parameters are used to carry out localized training on client devices. A host system may supervise the parameters in case of a heterogeneous setup. The updated set of parameters is uploaded to the server for aggregation into the central model. The process is repeated in multiple rounds with a randomly selected subset of devices. The randomness in subset selection ensures fairness and diversity in the training data samples included over numerous rounds.

Though FL eliminates the possibility of a direct data leak, studies have shown the dataset used for training can still be effectively replicated if one can access the shared gradients [238]. The efficiency of this inference highly depends on a case-by-case basis but still poses a considerable threat. In their work, Xu et al. [135] proposed a simple FL-based scheme for security enhancement focusing on the Internet of Medical Things (IoMT). The quantization employed by the proposed scheme over the transferred gradients/parameters dissipates

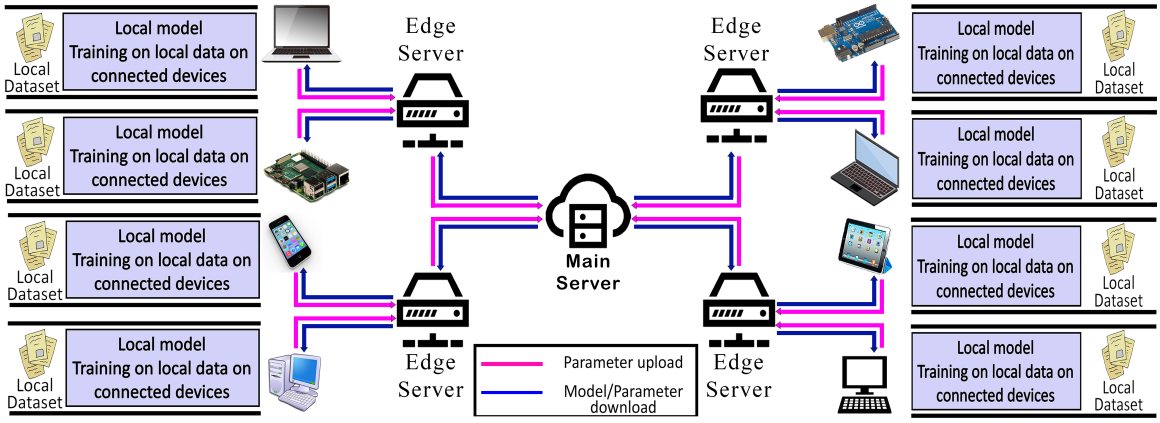


Figure 3.1: Homogeneous FL setup with Identical Data Distribution across Clients

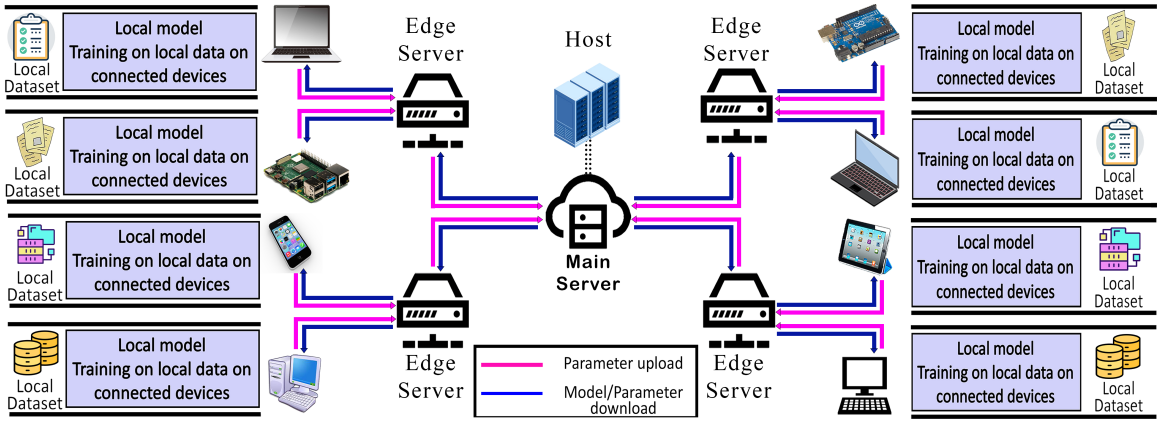


Figure 3.2: Heterogeneous FL setup with Different Data Distribution across Clients

the data retrieval efficiency of the attacker if the gradients are eavesdropped. The security, however, comes at the cost of a slight decrease in the global model's accuracy due to quantization error. This restricts the scheme's usage in applications that depend heavily on accurate results. Kerkouche et al. [154] provide another example of quantization for bandwidth optimization with a similar cost imbued on accuracy. Such a loss in performance may be feasible in general-purpose applications but presents a significant challenge in data-sensitive applications as explained in section 1.1.1.

3.1.2 Major Issues and Challenges identified with quantization based FL

Throughout our study of the related work and state-of-the-art methods for quantization-based FL, we found the following significant challenges:

- The quantization at any level in a training method causes an accuracy drop proportional to the quantization loss, limiting the extent of quantization.

- The efficiency of training suffers greatly depending on the frequency of dropouts from the network in addition to the native heterogeneity in client devices.

3.2 Related Work in Quantization and Incentive-Aware FL

Security is a non-exhaustive field of work. New threats, solutions, and improvements emerge daily, rapidly making state-of-the-art systems obsolete. FL is no stranger to this trend, as the leakage of the bare minimum transmitted parameters or gradients can enable attackers to infer the used dataset with near-perfect accuracy in some cases [143, 238]. Many researchers have found solutions to minimize or prevent this leak/inference. One of the most successful techniques is found to be quantization.

3.2.1 Quantization

There has been considerable research done in quantization at various levels. Initially, quantization methods were applied to the weights of a Neural Network (NN) model to optimize storage and bandwidth. Works of Li et al. [239], Chenghao et al. [240], and Bao et al. [241] highlight attempts made toward weight-based model compressions. The focus soon shifted to quantizing activation functions due to their larger size. Choi et al. [242] proposed Parameterized Clipping Activation for Quantized NN (PACT), which compresses the activation functions during the training phase. In contrast, Li et al. [243] proposed a Parameterized Max Scale (PAMS) to overcome performance loss in quantized Super Resolution NN. However, both these approaches rely on training, making the process computationally expensive. Fang et al. [244] proposed Piece-Wise Linear Quantization (PWLQ) to estimate the quantized parameters and deduce optimal breakpoints, minimizing the quantization error. However, using quantization identically for parameters with dissimilar probability distributions results in a limited reduction of the quantization error. Dong et al. [245] proposed Hessian Aware Quantization (HAWQ) of NN, adopting a layer-wise quantization approach, using the Hessian spectrum of layers as the influencing factor for selecting quantization precision optimized for each layer. As a method for optimizing bandwidth by compressing NN models, quantization showcases the potential for boosting security in FL applications.

FL natively improves the security and privacy of the involved data by limiting the data transfer across the network to model parameters. Some recent studies effectively demon-

strated the reconstruction of private data inferred from these gradients/parameters in case of a leak [238]. Xu et al. [135] proposed a quantization-based enhancement for FL to hinder the effectiveness of these inferences. Using 3-bit and 5-bit quantization with clip-ping over various image datasets, the proposed Federated Learning with Minimum Square Quantization Error (Fed-MSQE) model shows considerably less quantization error when compared to state-of-the-art Apot and Uniform quantizers. The approach, however, suffers from a loss in accuracy by 0.02% to 15%, depending on the complexity of the dataset used. Rendering the approach of limited utility toward DSAs.

Kerkouche et al. [154] presented another security approach partially focusing on resource optimization in constrained networks like the Internet of Medical Things (IoMT). The results highlight massive drops in bandwidth consumption to 3.14% of the original usage. This drop reduces the strain on resources in the network and minimizes the time taken per round of the FL algorithm in resource-constrained environments. However, similar to the works of Xu et al., the experimentations show a loss in accuracy ranging from 2% to 5% on simple datasets, which may increase with the complexity of features. IFL techniques over the base quantizer system can compensate for some of these performance gaps.

3.2.2 Incentive-Aware FL and Workload-Aware FL

In addition to the straightforward division of horizontal and vertical FL, some special techniques are implemented in certain cases that improve the feasibility and performance of the technology. IFL adds a step at the end of the training round, along with the aggregation step, to determine the individual contributions of nodes to the global model. Wang et al. [107] proposed an IFL method by incorporating blockchain technology in order to introduce trust and credibility between nodes. While proving to be a significantly stable, secure, and reliable technique, the accompanying overhead of blockchain [246, 247] is difficult to ignore in most use cases for constrained setups like IoMT. Nishio et al. [248] propose FedCS, which shortlists clients for subsequent rounds by evaluating their resource budget to prevent dropouts. However, judging whether a device has sufficient resources for the task is a fairly grueling chore. Abdulrahman et al. [249] propose another client selection framework, FedMCCS, which incorporates multiple grounds to evaluate and shortlist clients, including location, resource, etc. However, multiple hard constraints may result in fewer participants, thus slowing down the convergence. Ferdinand et al. [250] and Reisizadeh et al. [251] propose Anytime Mini-batch and QuanTimed-DSGD, respectively. The two frameworks attempt to create a novel training process using constant computation time for

devices instead of defining the number of training epochs.

A special application of IFL is proposed by Li et al. [155], which implements the contribution evaluation for individual clients based on their capacity for handling calculations. The approach allows optimal workload to be allocated in gradual steps to the nodes, decreasing the chance of dropouts during training. This is achieved through one of the two algorithms implementing inverse ratio increase (FedSAE-IRA) and steep increase followed by slow rise in workload (FedSAE-FASSA), respectively [155]. Both approaches show improvements in accuracy compared to the traditional FL approach. We, therefore, couple a modified version of this technique with a base quantizing algorithm to generate a hybrid algorithm to improve security while minimizing the accuracy loss experienced in raw quantization.

3.3 Proposed Quantization-based Hybrid Frameworks

We propose two hybrid frameworks that combine workload-aware FL with Quantization to minimize the performance impact of security provided by native quantization. The frameworks target sustained training with optimal resource utilization while also minimizing bandwidth requirements in a secure transmission scheme.

3.3.1 Dynamic Resource-Aware Federated Framework for Secure and Sustainable Learning in Data Sensitive Applications (DRAFFSS)

This section describes in detail our first hybrid approach combining the security aspect of quantization with optimized throughput gained by virtue of workload-aware load selection. A detailed view of the complete framework with all the sub-functions is shown in Fig. 3.3.

At the start of each round, a subset of client nodes is randomly selected and the shortlisted nodes are queried about their capabilities. The client nodes respond with the lower & upper bound epoch, and the maximum sustainable epoch by the device. These nodes are provided with the global model parameters (step 1). The devices take into account the individually calculated workload for the round, and train the model locally. After training, the updated models are quantized based on the quantization boundaries of the parameters. These boundaries are employed to serve two purposes: To provide security in case the

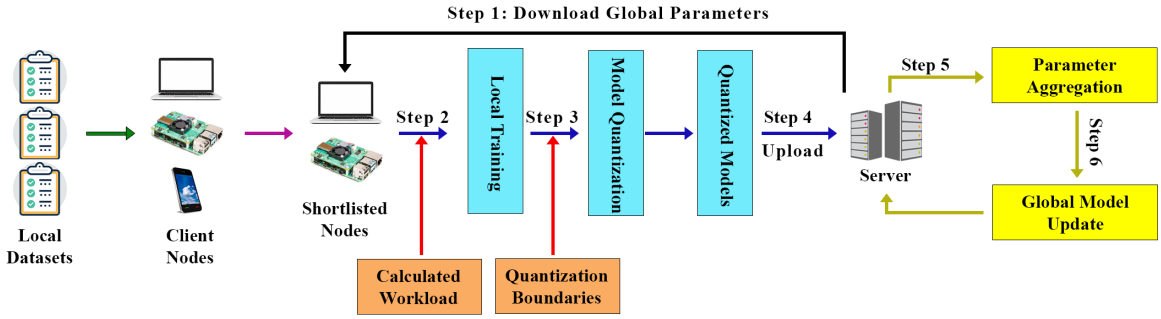


Figure 3.3: Architecture of Proposed Framework (DRAFFSS)

transmitted parameters are compromised and to minimize the network load. A copy of the original model is stored on the client node, and the quantized model is uploaded to the server for aggregation and updating the global model (step 6). The process may run for multiple rounds, depending on the required level of training and the variety of data in the client nodes.

3.3.1.1 Quantization based Security Method

The security aspect of the approach relies on the quantization of model parameters while maintaining minimal quantization loss. An element in consideration can be quantized toward the lower or upper bound as feasible. The probabilities of an element x being quantized towards either boundary value are calculated as follows:

$$\Pr(x \to a_{i-1}) = \frac{a_i - x}{a_i - a_{i-1}} \tag{3.3.1}$$

$$\Pr(x \to a_i) = \frac{x - a_i}{a_i - a_{i-1}} \tag{3.3.2}$$

Given that all parameters of the model are arranged in the form of a vector $\vec{p}_v$, having a probability distribution $f(x)$, for an element x falling between boundaries a_{i-1} and a_i, the total squared quantization error σ^2 is calculated as:

$$\sigma^2 = \sum_{i=0}^{2^b-1} \int_{a_{i-1}}^{a_i} \Big[-x^2 + (a_{i-1} + a_i)x - a_i a_{i-1} \Big] f(x)\, dx \tag{3.3.3}$$

Using partial derivatives, the authors in [135] derived the quantization probability of the i^{th} element that minimizes the quantization loss as:

$$\Pr_i = \frac{\int_{a_{i-1}}^{a_{i+1}} f(x)\,dx - \int_{a_{i-1}}^{a_{i+1}} x f(x)\,dx}{a_{i+1} - a_{i-1}} \tag{3.3.4}$$

Algorithm 3.1 elaborates on the quantization employed in a federated learning (FL) setup. At the initiation of each training round t, a subset of client nodes R is selected by the server. All nodes in R receive the model from the global server. This model is denoted as m_0 or m_{t-1} (line 2).

The clients then train the model k times over their local data and update the model to m_t (lines 3–6). The model is then quantized by applying equation 3.3.4 and rounding off the parameters accordingly using stochastic rounding (lines 7–8) to create $\tilde{m}_t$.

The final model m_t is retained with the client, while the quantized model $\tilde{m}_t$ is uploaded to the server for aggregation (line 9).

Algorithm 3.1 Quantization of Parameters in FL Setup (Fed-q)

Require: Initial model m_0
Ensure: Final model m_t
1: **function** FED-Q(m_0)
2: $m_0 \leftarrow m_{t-1}$
3: **for** $\tau = 0$ to $K - 1$ **do**
4: $\hat{m}_{\tau+1} \leftarrow x_\tau - \eta_t \nabla f(x_\tau)$
5: **end for**
6: $m_t \leftarrow \hat{m}_K$
7: Apply Equation 3.3.4 on each parameter: $\{a_\tau^t\}_{\tau=0}^{2^5-1}$
8: Quantize model: $\tilde{m}_t \leftarrow Q(m_t, \{a_\tau^t\}_{j=0}^{2^b-1})$ ▷ stochastic rounding
9: Upload $\{a_\tau^t\}_{j=0}^{2^b-1}$ and $f(\tilde{m}_t)$ to server
10: **return** m_t
11: **end function**

3.3.1.2 Optimization through Client Selection

Quantization strengthens the security of the data from inference attacks, but the induced compression results in a loss of accuracy in the performance of the global model. This restricts the usage in critical applications like the DSAs. To compensate for this loss, a workload optimization technique is integrated based on [155]. The authors propose workload-aware client selection and show improvements in the accuracy of the trained model compared to traditional FL. The approach also highlights a stable network with minimal stragglers. We layer this approach over quantization to regain the accuracy lost in algorithm 3.1. The authors define the predicted workload for a client in a particular round by increasing the

workload of the same client in the previous round by a fraction of itself. If the client drops out of the network, then the workload for that client in the next round is halved. Both cases are depicted by 3.3.5.

$$\hat{E}_k^{(t+1)} = \begin{cases} \frac{1}{2}\hat{E}_k^t, & \text{if client drops out,} \\ \hat{E}_k^t + \frac{\mu}{\hat{E}_k^t}, & \text{otherwise.} \end{cases} \tag{3.3.5}$$

Here $\hat{E}_k^t$ is the workload predicted for client k in round t, and μ is a hyper-parameter defined on an application-to-application basis. The selection of the value of μ is crucial because if the value is too high, the stragglers will rise, and if the value is too low, the convergence time will be infeasible. Algorithm 3.2 shows the process for the calculation of the optimal workload for a client in the next round. The algorithm takes input for the sustainable capacity of the client to process workload: $\tilde{E}_k^t$, the lower and upper limits of epoch for workload assignment: $\hat{L}_k^t$ and $\hat{H}_k^t$. Lines 2–5 initiate the second portion of (3.3.5). The first portion of (3.3.5) is handled by lines 6–10. Lines 11–14 are triggered when the lower limit exceeds the sustainable workload value for the client. It sets the workload of the next round as `NULL` and halves both limits as per (3.3.5), highlighting a client drop from the algorithm.

Algorithm 3.2 Workload Prediction (Round_Load)

Require: Locally sustainable workload $\tilde{E}_k^t$ of client k at round t, the lower limit of the epoch for $\tilde{E}_k^t$, $\hat{L}_k^t$, and the upper limit of the epoch for $\tilde{E}_k^t$, $\hat{H}_k^t$.

Ensure: Predicted minimum workload $\hat{L}_k^{(t+1)}$, predicted maximum workload $\hat{H}_k^{(t+1)}$, and training workload $\hat{E}_k^t$.

1: **function** ROUND_LOAD($\hat{L}_k^t, \hat{H}_k^t, \tilde{E}_k^t$)
2: **if** $\tilde{E}_k^t$ is greater than $\hat{H}_k^t$ **then**
3: $\hat{L}_k^{(t+1)}, \hat{H}_k^{(t+1)} \leftarrow \hat{L}_k^t + \frac{u}{\hat{L}_k^t},\ \hat{H}_k^t + \frac{u}{\hat{H}_k^t}$
4: $\hat{E}_k^t \leftarrow \hat{H}_k^t$
5: **else if** the actually affordable workload falls into $[\hat{L}_k^t, \hat{H}_k^t]$ **then**
6: $\hat{L}_k^{(t+1)} \leftarrow \min\left(\hat{L}_k^t + \frac{u}{\hat{L}_k^t} + \frac{1}{2}\hat{H}_k^t\right)$
7: $\hat{H}_k^{(t+1)} \leftarrow \max\left(\hat{L}_k^t + \frac{u}{\hat{L}_k^t} + \frac{1}{2}\hat{H}_k^t\right)$
8: $\hat{E}_k^t \leftarrow \hat{L}_k^t$
9: **else**
10: $\hat{L}_k^{(t+1)}, \hat{H}_k^{(t+1)} \leftarrow \frac{1}{2}\hat{L}_k^t,\ \frac{1}{2}\hat{H}_k^t$
11: $\hat{E}_k^t \leftarrow 0$
12: **end if**
13: **return** $\hat{L}_k^{(t+1)}, \hat{H}_k^{(t+1)}, \hat{E}_k^t$
14: **end function**

3.3.1.3 Hybrid Algorithm (DRAFFSS)

Combining the two aspects of security and optimization from the discussed works, we propose Dynamic Resource-Aware Federated Framework for Secure and Sustainable Learning to allow secure training with minimal straggler nodes and accuracy loss. The framework employs a 5-bit quantizer for parameter compression to optimize bandwidth consumption and prevent inference in case of eavesdropping on the network. The quantization loss induced is compensated by adding a layer for dynamic workload-allotment which reduces the chance of client drop out by allocating gradually increasing workload to the devices. The complete process of DRAFFSS is explained in Algorithm 3.3. At the start of each communication round, a subset of clients is selected based on a probability list and global weights are shared with these clients (lines 1-3). The clients respond with the respective sustainable and minimum workloads (line 4). These values are passed on to a function employing algorithm 3.2 to determine workloads for the ongoing round, which are passed to the Fed-q function (algorithm 3.1) to carry out training (lines 5-8). Finally, the weights returned by Fed-q are aggregated into the global model (line 9).

Algorithm 3.3 DRAFFSS

Require: Number of training rounds T, list of clients in the network $\mathcal{R}$
Ensure: Global trained model weights w^T

for each training round $t = 1$ to T **do**
  $C_t \leftarrow$ shortlist clients based on probability list generated for $\mathcal{R}$
  Server passes the global weights w_t to shortlisted clients
  $\{(\hat{L}_k^t, \hat{H}_k^t, \tilde{E}_k^t)\} \leftarrow$ receive replies from C_t ▷ Request initial workloads and sustainable limits
  for each client $k \in C_t$ **in parallel do**
    $(\hat{L}_k^t, \hat{H}_k^t, \hat{E}_k^t) \leftarrow$ ROUND_LOAD$(\hat{L}_k^t, \hat{H}_k^t, \tilde{E}_k^t)$ ▷ Algorithm 3.2
    $\Delta_k^t, h_k^t \leftarrow$ FED-Q(m_k) ▷ trained with workload values $(\hat{L}_k^t, \hat{H}_k^t, \hat{E}_k^t)$, Algorithm 3.1
  end for
  $w^{(t+1)} \leftarrow w^t - \dfrac{\sum_{k \in C_t} \Delta_k^t}{\sum_{k \in C_t} h_k^t}$
end for

3.3.1.4 Result and Analysis of DRAFFSS

This section elaborates on our experimental environment and results over the various datasets used. We compared our model with standard FL as well as FL with 5-bit quantization to present a detailed view of the gains achieved by the hybrid approach.

3.3.1.4.1 Baseline Model for simulation of DRAFFSS

We have used Resnet-18 [252], an 18-layer model with ReLU activation. We used the SGD optimizer with an adaptive learning rate of 0.1, divided by 10 when the error graph plateaus, similar to that used in [238]. The architecture of the model used is depicted in Fig. 3.4.

3.3.1.4.2 Experimental Setup for DRAFFSS simulation

Our simulations are carried out on a system running Windows 11 Pro version 22H2 equipped with an Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz, 128GB DDR4 RAM, and two NVIDIA Titan RTX GPUs having 24GB VRAM each. We implemented our models in Python 3.9.16 with Cuda 12.1 [253], Tensorflow 2.12 [226] , and Pysyft 0.9.0 [254] over Jupyter Notebook in Conda virtual environment.

To replicate a network with heavy traffic and constraints implied by the bandwidth, we simulate 1000 nodes with slightly varying resources such as more or less RAM, CPU-cycles, and data-location (GPU/RAM).

3.3.1.4.3 Datasets used for simulation of DRAFFSS

We tested our algorithm against the implementation discussed in [135] and standard FL

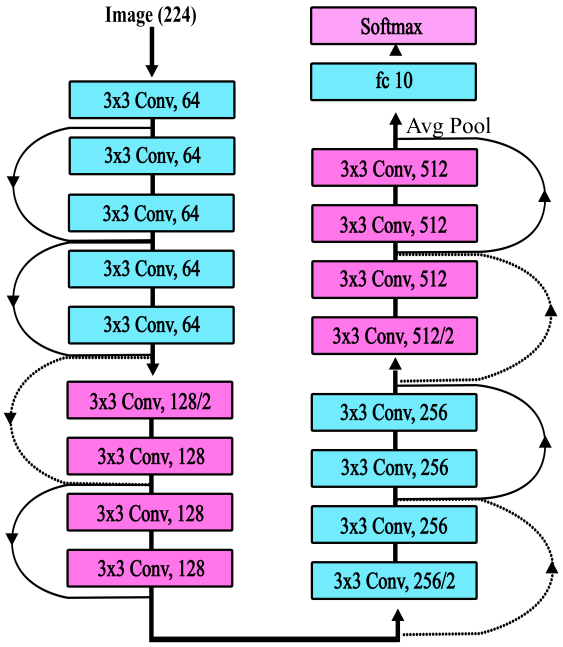


Figure 3.4: Architecture of the baseline ResNet-18 model

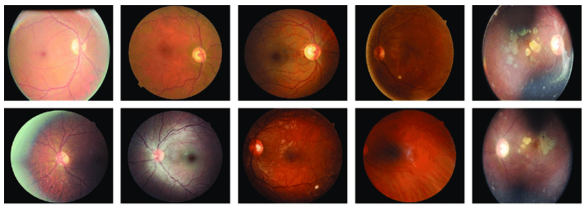


Figure 3.5: Sample images from Diabetic Retinopathy Dataset

over two publicly available medical image datasets having dissimilar complexity and sample size. A brief description of the datasets and their composition is as follows:

1. Diabetic Retinopathy

   Diabetic Retinopathy Dataset [177] is a collection of high-resolution retinal images labelled 0 through 4 depicting absence or presence of diabetic retinopathy in the patient. There are two separate images for the left and right eyes for each patient. Fig. 3.5 shows some sample images from the Diabetic Retinopathy Dataset.

2. RSNA Bone Age

   RSNA Bone Age Dataset [180] is an image dataset housing X-ray scans of the hands of multiple children. The aim of the dataset is to predict both age and gender of the individuals by analyzing bone structures. We utilize the dataset for image classification on gender basis. Fig. 3.6 shows some sample images from the dataset.

Table 3.1 provides a summary of the datasets used in the experimentation.

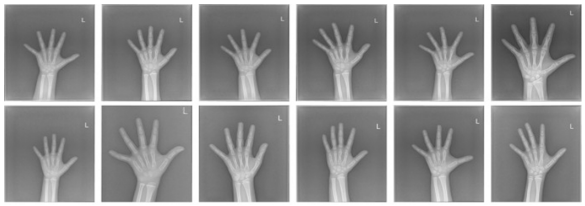


Figure 3.6: Sample images from RSNA Bone Age Dataset

Table 3.1: Datasets used for Experimentation of Fed-WALQ.

Dataset	**Description**	**Sample Size**	**Train Split**	**Test Split**
Diabetic Retinopathy	Image classification dataset with high-resolution retinal images labelled 0-4 depicting the presence of diabetic retinopathy.	High Resolution Retinal Images (3000px x 2000px) (RGB)	35,126	53,576
RSNA Bone Age	Image classification dataset with X-ray scans of hands of children of varying ages.	X-ray scans	12,611	1,625

3.3.1.4.4 Results of Simulations of DRAFFSS

We benchmarked DRAFFSS over the two datasets discussed in the previous section and evaluated it against standard FL and Fed-MSQE [135] with the 5-bit quantization mode. Primarily, four metrics are used for evaluation: Accuracy, Straggler node percentage, Precision, and Recall. The straggler node percentage is defined as follows:

$$\text{Straggler}\% = \frac{1}{T}\sum_{t=1}^{T}\frac{\text{Nodes active in } C_t \times 100}{1000} \tag{3.3.6}$$

Our DRAFFSS demonstrated improved accuracy compared to the Fed-MSQE while also showing significantly less straggler nodes in all the comparisons. A summary of results attained by DRAFFSS is shown in table 3.2.

Table 3.2: Results achieved by DRAFFSS in various Datasets

Dataset	**Accuracy**	**Straggler Node (%)**
Diabetic Retinopathy	79.20	14
RSNA Bone age	82.60	12

1. Comparison in Diabetic Retinopathy Dataset.

   Being the simpler of the two datasets, the results of all techniques in comparison appear close, well within the margin of error due to randomness. The standard FL tops the accuracy curve, Fed-MSQE shows a loss of roughly 5.70%, and DRAFFSS shows a loss of just 2.50%, highlighting significant improvement in the narrow range

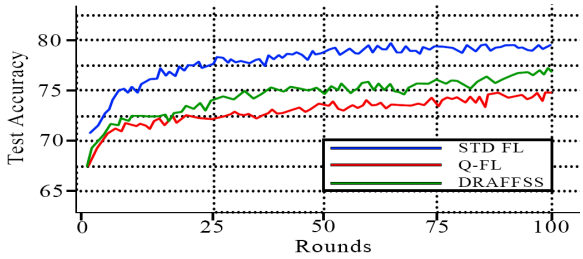


Figure 3.7: Test Accuracy Comparison for Diabetic Retinopathy Dataset

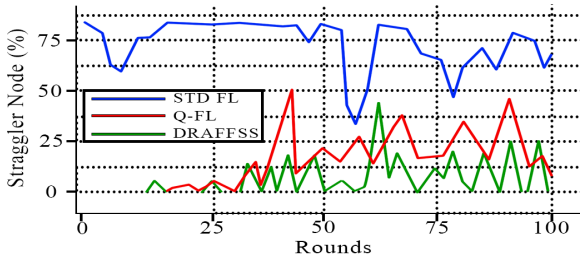


Figure 3.8: Comparison of straggler nodes present during training over Diabetic Retinopathy Dataset

available. A comparison of test accuracies of the three techniques over the Diabetic Retinopathy Dataset is shown in Fig. 3.7.

Due to quantization and workload allotment, the number of stragglers is also seen to be reduced significantly when compared to traditional FL and Fed-MSQE as highlighted by the graph in Fig. 3.8.

2. Comparison in RSNA Bone Age Dataset.

   The dataset is designed to predict the age of the patient based on X-rays. Hence, the accuracy tested corresponds to the predictions that lie within 5.4 months of the actual ages. Here the standard FL attained a testing accuracy of 90%. While Fed-MSQE attained a testing accuracy of 77.2%, DRAFFSS attained an accuracy of 82.6% highlighting a drop of only 8.2% against 14.2% of Fed-MSQE. The comparison of testing accuracy for all three techniques is shown in Fig. 3.9.

   Comparing the straggler percentages, standard FL suffered from 73.5% straggler node presence. While Fed-MSQE and DRAFFSS dealt with 14% and 12% presence respectively. Fig. 3.10. Shows the straggler node variations throughout training for the three techniques.

   A summary of all the test accuracy results is tabulated in Table 3.3.

   Based on the results observed, DRAFFSS demonstrates its capability to compensate for performance loss induced by quantization. The compensation may be marginal or

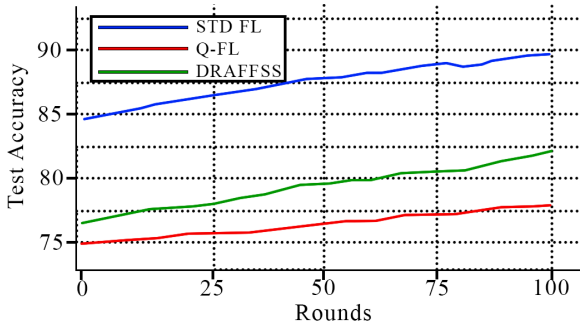


Figure 3.9: Test Accuracy Comparison for RSNA Bone Age Dataset

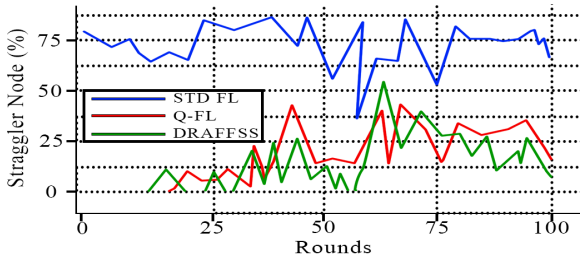


Figure 3.10: Comparison of straggler nodes present during training over RSNA Bone Age Dataset

Table 3.3: Performance comparison of DRAFFSS against Fed-MSQE and Standard FL over the two medical datasets

Method/ Dataset	**Diabetic Retinopathy**				**RSNA Bone Age**			
	Test Accuracy	**Straggler %**	**Precision**	**Recall**	**Test Accuracy**	**Straggler %**	**Precision**	**Recall**
Standard FL	79.80	72.00	79.76	79.79	90.00	73.50	90.01	90.00
Fed-MSQE (5-bit)	75.00	17.00	75.12	75.12	77.20	14.00	77.20	77.21
DRAFFSS (Proposed)	77.50	14.00	77.50	77.49	82.60	12.00	82.60	82.60

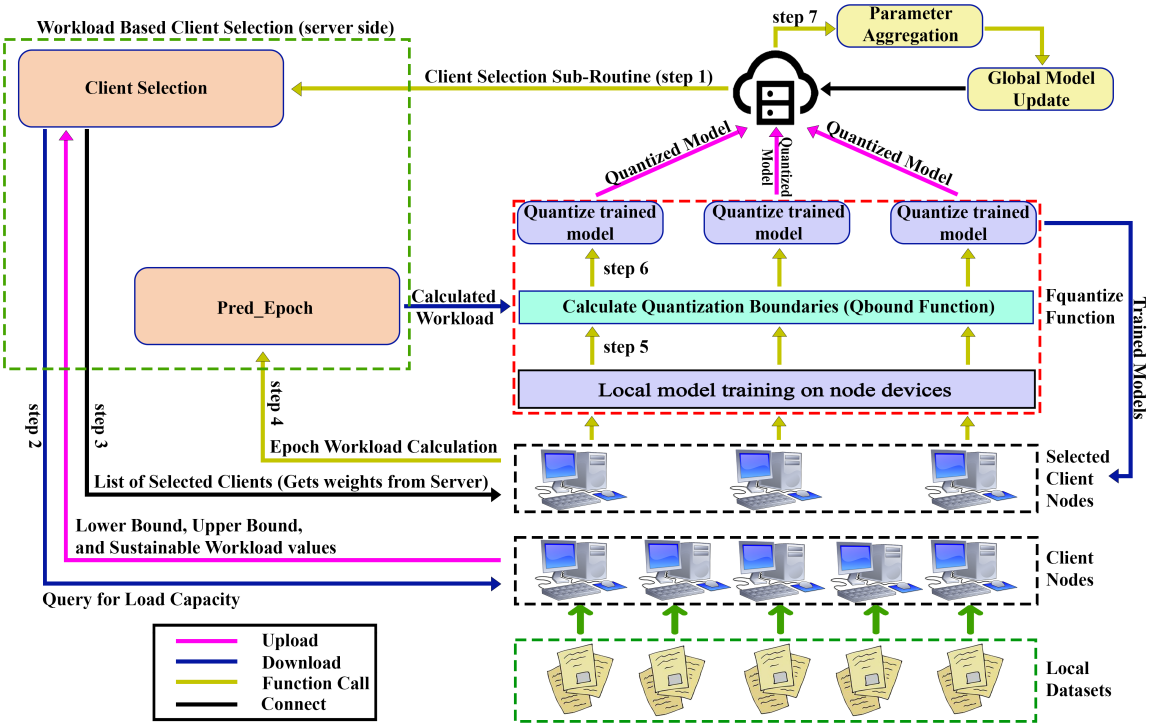


Figure 3.11: Proposed architecture of Fed-WALQ

significant depending on the complexity of the data used for training. The reduction in straggler nodes, however, is significant in all the test cases, highlighting a stable system compared to the existing techniques. The improved stability allows the network to train more effectively, as there are fewer interruptions due to node dropouts.

Accessing from the point of view of the node devices, the workload allotment helps keep resources from being overutilized. This, in turn, enables the devices to continue their local routines without issues.

3.3.2 Federated Workload-Aware Quantized Framework for Secure Learning in Data-Sensitive Applications (Fed-WALQ)

This section describes in detail our hybrid approach, which combines the security aspect of quantization with optimizing the throughput gained through workload-aware client selection. The Fed-WALQ framework extends the DRAFFSS framework discussed in previous section and subjects it to experimentation over wider variety of datasets and quantization configurations. A detailed view of the complete framework with all the sub-functions is shown in Fig. 3.11.

At the start of each round, the server initiates the Server_Side_Client_Selection (step 1), which queries the connected clients to know their capabilities (step 2). The client nodes respond with three values: The initial lower and upper bound epoch and the maximum workload sustainable by the device. Based on these values, a list of high-performing client nodes is created, and a fixed number of clients are randomly selected from this list (step 3). The chosen clients are provided with the global model parameters. The Pred_Epoch function calculates the workload allocated to them in the current round (based on the load allotted in the previous round) (step 4). The clients then train the model locally for the allotted epoch and call the quantizer subroutine individually. The quantizer sub-routine first calculates the quantization boundaries of each model parameter through the Qbound function (step 5). These boundaries are then used to quantize the model through the Fquantize function (step 6) to serve two purposes: To provide security in case the transmitted parameters are compromised and to minimize the network load. A copy of the original model is stored on the client device, and the quantized model is sent to the server for aggregation and updating of the global model (step 7). The process may run for multiple rounds depending on the required level of training and the variety of data in the client nodes.

3.3.2.1 Quantization based Security for Parameter Gradients

The core of the security boosting portion of the method relies on quantizing parameters with minimal quantization loss as discussed in section 3.3.1.1. An element can be rounded/quantized toward the lower or upper bound as feasible. The probabilities of element x being quantised towards either boundary value can be calculated using the equations 3.3.1 and 3.3.2. The total squared quantization error and the minimized quantization error can be calculated using the equations 3.3.3 and 3.3.4, respectively.

Algorithm 3.4 represents the **Qbound** function implementing equation 3.3.4 and calculates quantization boundaries for a given parameter vector $\vec{p_v}$ at a given bit limit ranging from 3 to 7 bits denoted by 'b'. All the boundary values are initialized in lines 1-5. The numerator portion of equation 3.3.4 is performed at line 10 for each element by multiplying the numeric part of p_c belonging to a sorted parameter vector p_s by a_{i+1} and then subtracting the sum of p_c. The rest of equation 3.3.4 is calculated in line 11, where the result of line 10 is divided by $a_{i+1} - a_{i-1}$ and floored, providing idx, corresponding to $\Pr_i$.

Algorithm 3.4 Quantizing Function (Qbound)

Require: Parameter vector $\vec{p}_v$
Ensure: Quantization boundaries $\{a_i\}_{i=0}^{2^b-1}$

1: $a_0 \leftarrow \min(\vec{p}_v)$
2: $a_{2^b-1} \leftarrow \max(\vec{p}_v)$
3: **for** $i = 1$ **to** $2^b - 2$ **do**
4: $\quad a_i \leftarrow a_0 + i \times \dfrac{a_{2^b-1} - a_0}{2^b - 1}$
5: **end for**
6: $\vec{p}_s \leftarrow \text{Sort}(\vec{p}_v)$ ▷ ascending order
7: **while** True **do**
8: $\quad \{\hat{a}_i\}_{i=0}^{2^b-1} \leftarrow \{a_i\}_{i=0}^{2^b-1}$ ▷ store previous boundaries
9: $\quad$ **for** $i = 1$ **to** $2^b - 2$ **do**
10: $\quad\quad p_c \leftarrow \text{find}(a_{i-1} \leq \vec{p}_s \leq a_{i+1})$
11: $\quad\quad t \leftarrow \text{num}(p_c) \times a_{i+1} - \text{sum}(p_c)$
12: $\quad\quad \text{idx} \leftarrow \left\lfloor \dfrac{t}{a_{i+1} - a_{i-1}} \right\rfloor$
13: $\quad\quad a_i \leftarrow \text{idx}^{\text{th}}$ element of p_c
14: $\quad$ **end for**
15: $\quad$ **if** $\{\hat{a}_i\}_{i=0}^{2^b-1} = \{a_i\}_{i=0}^{2^b-1}$ **then**
16: $\quad\quad$ **break** ▷ convergence reached
17: $\quad$ **end if**
18: **end while**
19: **return** $\{a_i\}_{i=0}^{2^b-1}$

The approach discussed in Algorithm 3.4 is expanded to the FL architecture in Algorithm 3.5. At the beginning of each training t, the server selects a random subset of client devices R. All clients belonging to t receive the initial parameters from the global server. Let this model be m_0 or m_{t-1} represented in line 1. The clients then update the model by running the Stochastic Gradient Descent-based training loop k times locally and obtain a new model m_t (lines 2-5). We then quantize the model by first calling the **Qbound** function in Algorithm 3.4 (line 6) and then applying the stochastic rounding $Q(\cdot)$ (line 7), which generates a quantized model $m_{\sim t}$. The final model m_t is kept with the client, while quantized $m_{\sim t}$ is uploaded to the server for aggregation (line 8).

Algorithm 3.5 Federated Quantization of Parameters (Fquantize)

Require: Initial model m_0
Ensure: Final model m_t
1: $m_0 \leftarrow m_{t-1}$
2: **for** $\tau = 0$ **to** $K - 1$ **do**
3: $\quad m_{\tau+1} \leftarrow x_\tau^- - \eta_t \nabla f(x_\tau^-)$ ▷ local update step
4: **end for**
5: $m_t \leftarrow m_K^-$
6: $\{a_\tau^t\}_{\tau=0}^{2^b-1} \leftarrow \text{Qbound}(m_t)$ ▷ call Algorithm 3.4
7: $m_{\sim t} \leftarrow Q(m_t, \{a_\tau^t\}_{\tau=0}^{2^b-1})$ ▷ quantize parameters locally
8: Upload $\{a_\tau^t\}_{\tau=0}^{2^b-1}$ and $f(m_{\sim t})$ to server ▷ send quantized parameters and local loss to server
9: **return** m_t

3.3.2.2 Client Selection based Optimization Method

Quantization ensures the security of the datasets from inference attacks over transmitted parameters, but the compression induced also causes a loss of accuracy in the performance of the global model. This restricts usage in critical applications, such as the DSAs. To compensate for this loss, a client selection technique is integrated based on the work of Li et al. [155]. The authors propose workload-aware client selection directed towards heterogeneous setups and showed improvements in the accuracy of the trained model when compared to traditional FL. We layer this mechanism over quantization to recover the accuracy lost in algorithms 3.4 and 3.5. The approach is divided into two parts: workload prediction being the first, and client selection being the second. The authors define the predicted workload for a client in a particular epoch by increasing the workload of the same client in the previous epoch by a fraction of itself. If the client drops out of the network, then the workload for that client in the next epoch is halved. Both cases are summarized in Equation 3.3.5. Algorithm 3.6 shows the process for calculating the optimal workload for a client for the next epoch.

The process takes input for the sustainable capacity of the client to process workload: $\tilde{E}_k^t$, the lower and upper limits of the epoch for workload assignment: $\hat{L}_k^t$ and $\hat{H}_k^t$. Lines 1-4 trigger the first branch of the algorithm if the sustainable workload exceeds the current upper limit. This branch sets the workload of the next epoch as the upper limit and then increases the limits by the factor defined in equation 3.3.5. The second branch (lines 5-9) deals with a straggling client having a sustainable value between the limits. This branch sets the workload of the next epoch as the lower limit (line 9) and lowers the limits as defined

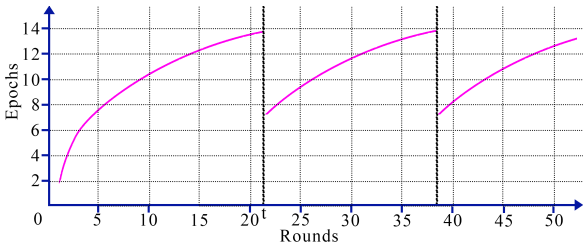


Figure 3.12: Example of workload allotment process of Algorithm 3.6

in lines 7 and 8. The final branch (lines 10-13) is triggered when the lower limit exceeds the sustainable workload value for the client. It sets the workload of the next epoch as NULL and halves both limits as per equation 3.3.5, highlighting a client drop from the process.

Algorithm 3.6 Client Workload Prediction (Pred_Epoch)

Require: Sustainable workload $\tilde{E}_k^t$ of client k at round t, lower limit $\hat{L}_k^t$, upper limit $\hat{H}_k^t$

Ensure: Predicted minimum workload $\hat{L}_k^{(t+1)}$, predicted maximum workload $\hat{H}_k^{(t+1)}$, training workload $\hat{E}_k^t$

1: **if** $\tilde{E}_k^t > \hat{H}_k^t$ **then**

2: $\quad \hat{L}_k^{(t+1)} \leftarrow \hat{L}_k^t + \dfrac{u}{\hat{L}_k^t}$

3: $\quad \hat{H}_k^{(t+1)} \leftarrow \hat{H}_k^t + \dfrac{u}{\hat{H}_k^t}$

4: $\quad \hat{E}_k^t \leftarrow \hat{H}_k^t$ ▷ workload capped at upper limit

5: **else**

6: $\quad$ **if** the actually affordable workload falls into $[\hat{L}_k^t, \hat{H}_k^t]$ **then**

7: $\quad\quad \hat{L}_k^{(t+1)} \leftarrow \min\left(\hat{L}_k^t + \dfrac{u}{\hat{L}_k^t} + \dfrac{1}{2}\hat{H}_k^t\right)$

8: $\quad\quad \hat{H}_k^{(t+1)} \leftarrow \max\left(\hat{L}_k^t + \dfrac{u}{\hat{L}_k^t} + \dfrac{1}{2}\hat{H}_k^t\right)$

9: $\quad\quad \hat{E}_k^t \leftarrow \hat{L}_k^t$

10: $\quad$ **else**

11: $\quad\quad \hat{L}_k^{(t+1)} \leftarrow \dfrac{1}{2}\hat{L}_k^t$

12: $\quad\quad \hat{H}_k^{(t+1)} \leftarrow \dfrac{1}{2}\hat{H}_k^t$

13: $\quad\quad \hat{E}_k^t \leftarrow 0$ ▷ no workload affordable

14: $\quad$ **end if**

15: **end if**

16: **return** $\hat{L}_k^{(t+1)}, \hat{H}_k^{(t+1)}, \hat{E}_k^t$

Fig. 3.12 shows an example of the workload allotting algorithm defined in Algorithm 3.6. If a client k drops out at round t, the workload calculated for round $t+1$ is reduced to 50% of that in round t. If the client does not drop out, the workload is increased by $\mu/\hat{E}_k^t$.

At the start of a round, the server randomly selects a subset of clients and evaluates the training value or importance associated with each client. The virtue of cross-entropy loss determines this value. The calculated importance is converted into a probability list governing the final selection of clients. For any client k in round t, the associated importance is calculated using the equation 3.3.5. The SoftMax function is used on the server side to determine the probability by employing the following equation:

$$p_k^t = \frac{e^{\beta v_k^t}}{\sum_{k=1}^{K} e^{\beta v_k^t}} \tag{3.3.7}$$

Here, the hyperparameter β is introduced to prevent the training value from being exceptionally large or insufficiently small to depict the relations among raw data. Incorporating active learning in the process tips the balance by preferring clients with high losses in the final selection process. Therefore, to ensure fairness in the selection, q-FFL is combined into the algorithm, and the global objective function is adjusted to the following equation:

$$\min f(w) = \sum_{k=1}^{N} \frac{1}{(q+1)} F_k^{(q+1)}(w) \tag{3.3.8}$$

Here N represents the number of clients in the network, and $F_k(w)$ is the objective function for client k. The fairness level is controlled through the exponent q. Even though q is directly proportional to the fairness level, only some values of q benefit the process by increasing the aggregation weight of clients having high losses. This increase limits the variance among locally trained models and achieves a stable global convergence rate. The final process, integrating equations 3.3.5, 3.3.7, and 3.3.8, ensures privacy and fairness, as given in Algorithm 3.7.

Algorithm 3.7 Server_Side_Client_Selection

Require: Initial global weights w^0, $\hat{L}_k^0$, $\hat{H}_k^0$, $\tilde{E}_k^0$, learning rate η
Ensure: Updated global weights $w^{(t+1)}$

for each training round $t = 1$ **to** T **do**
  $C_t \leftarrow$ shortlist clients based on probability list ▷ select clients for this round
  Server passes the global weights w_t to clients in C_t
  $\{(\hat{L}_k^t, \hat{H}_k^t, \tilde{E}_k^t)\} \leftarrow$ receive replies from C_t ▷ request initial workloads and sustainable limits
  for each client $k \in C_t$ **in parallel do**
    $(\hat{L}_k^t, \hat{H}_k^t, \hat{E}_k^t) \leftarrow \text{Pred_Epoch}(\hat{L}_k^t, \hat{H}_k^t, \tilde{E}_k^t)$
    $\Delta_k^t, H_k^t \leftarrow$ client-side training results ▷ compute updates and local workload
  end for
  $w^{(t+1)} \leftarrow w^t - \dfrac{\sum_{k \in C_t} \Delta_k^t}{\sum_{k \in C_t} H_k^t}$ ▷ aggregate updates to global model
end for

The proposed workload assignment strategy is heuristic and intentionally designed to balance practicality and computational efficiency in heterogeneous federated environments. Exact optimization of client selection and workload allocation would require solving a global scheduling problem that depends on dynamic client availability, bandwidth, and processing capabilities, which is computationally expensive and impractical for real-time federated learning rounds. Therefore, a lightweight heuristic approach is adopted to provide near-optimal allocation with minimal overhead.

The computational complexity of the workload assignment process per round is $\mathcal{O}(N \log N)$, dominated by sorting clients by their predicted workloads and resource capabilities, followed by linear selection of the top K eligible participants. This complexity is negligible compared to the cost of local model training and communication, ensuring that the proposed strategy does not introduce significant additional overhead to the federated learning pipeline.

3.3.2.3 Hybrid Algorithm (Fed-WALQ)

Merging the two aspects of security and optimization from the discussed works, we propose Workload-Aware Federated Learning with Linear Quantization (Fed-WALQ) to allow secure training over sensitive data in bandwidth-constrained networks with minimal performance loss. The framework uses a 5-bit quantizer for parameter compression in trained models to optimize bandwidth consumption and prevent inference in case the transmitted data is leaked. The accuracy loss induced by the quantization loss is compensated by adding a sub-routine for workload-based client selection, which caters to two purposes: It selects

high-value or high-performing clients from the random subset of each round to maximize training throughput, and it also reduces the chance of client drop out by allocating gradually increasing workload to the devices. The complete process of Fed-WALQ is given in Algorithm 3.8 based on a modified version of Algorithm 3.7 to integrate quantization. At the start of each communication round, a subset of clients is selected based on a probability list, and global weights are shared with these clients (lines 1-3). The clients respond with sustainable and minimal workloads (line 4). These values are passed on to a function employing algorithm 3.6 to determine workloads for the ongoing round, which are passed to the Fquantize function (algorithm 3.5) to train (lines 5-7). Finally, the weights returned by Fquantize are aggregated into the global model (line 9).

Algorithm 3.8 Fed-WALQ

Require: Number of training rounds T, list of clients in the network R
Ensure: Global trained model weights w^T
1: **for** each training round $t = 1$ to T **do**
2: $C_t \leftarrow$ shortlist clients based on the probability list generated for R
3: Server passes the global weights w_t to shortlisted clients
4: $\{(\hat{L}_k^t, \hat{H}_k^t, \tilde{E}_k^t)\} \leftarrow$ receive reply from C_t ▷ Request initial workloads and sustainable limits
5: **for** each client $k \in C_t$ in parallel **do**
6: $(\hat{L}_k^t, \hat{H}_k^t, \hat{E}_k^t) \leftarrow$ Pred_Epoch$(\hat{L}_k^t, \hat{H}_k^t, \tilde{E}_k^t)$ ▷ Algorithm 3.6
7: $\Delta_k^t, H_k^t \leftarrow$ Fquantize(m_k) ▷ Trained with workload values $(\hat{L}_k^t, \hat{H}_k^t, \hat{E}_k^t)$; see Algorithm 3.5
8: **end for**
9: $w^{(t+1)} \leftarrow w^t - \dfrac{\sum_{k \in C_t} \Delta_k^t}{\sum_{k \in C_t} H_k^t}$
10: **end for**

3.3.2.4 Result and Analysis of Fed-WALQ

3.3.2.4.1 Baseline Model for simulation of Fed-WALQ

We have used Resnet-18 [252], an 18-layer model with ReLU activation. We used the SGD optimiser with an adaptive learning rate of 0.1, divided by 10 when the error graph plateaus, similar to that used in [238]. The exact model used is depicted in Fig. 3.13.

3.3.2.4.2 Experimental Setup for simulation of Fed-WALQ

Our simulations are carried out on a system running Windows 11 Pro version 22H2 equipped with an Intel (R) Xeon(R) Silver 4114 CPU @ 2.20GHz, 128GB DDR4 RAM, and two NVIDIA Titan RTX GPUs in SLI having 24GB VRAM each. We implemented our models in Python 3.9.16 with Cuda 11.8 [253], Pytorch 2.0.0 [227], Numpy 1.23, and Pysyft 0.8.0 [254]

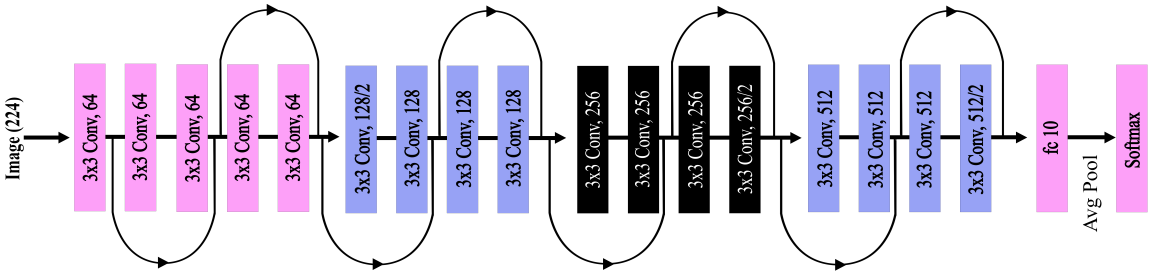


Figure 3.13: Architecture of the baseline Resnet 18 model used in Fed-WALQ

over Jupyter Notebook in Conda virtual environment.

The details for all hardware, software, and simulation parameters are provided in Table 3.4.

Table 3.4: Experimental Setup

Element	Value
Hardware Specifications	
Processor	Intel (R) Xeon(R) Silver 4114 CPU @ 2.20GHz
System Memory	128 GB - DDR4
Graphics Processor	2x NVIDIA Titan RTX
Graphics Memory	2x 24 GB
Software Specifications	
Operating System	Windows 11 Pro - version 22H2
Coding Language	Python 3.9.16
Cuda	Cuda 11.8
Pytorch	Pytorch 2.0.0
Numpy	Numpy 1.23
Pysyft	Pysyft 0.8.0
Model Specifications	
Model	ResNet-18
Layers	18
Parameters	11,000,000
Activation	Relu
Optimizer	Stochastic Gradient Descent
Learning Rate	0.1 (divided by 10 when error graph plateaus)
Simulated Devices	
Number of Nodes Simulated	1000
Number of Servers Simulated	1
Number of Nodes participating in a Round	100

3.3.2.4.3 Datasets used for simulation of Fed-WALQ

We tested our algorithm against the implementation discussed in [135] and standard FL over four openly available image datasets having dissimilar complexity, color space, and sample size. A brief description of the datasets and their composition is as follows:

1. MNIST

MNIST [255] is an image classification dataset created in 1994, having 10 classes representing handwritten digits 0–9. The dataset comprises 28x28-bit grayscale images, divided into 60,000 training samples and 10,000 testing samples. Some images from the dataset are shown in Fig. 3.14a.

2. Fashion-MNIST

   Fashion MNIST [256] is an image dataset initially created to replace the MNIST digit dataset. Like MNIST, Fashion MNIST houses 28x28-bit grayscale images of 10 classes divided into 60000 training and 10000 testing samples. The classes represent various fashion accessories and clothing (shirts, bags, sandals, etc.). Fig. 3.14b. shows some samples from the dataset.

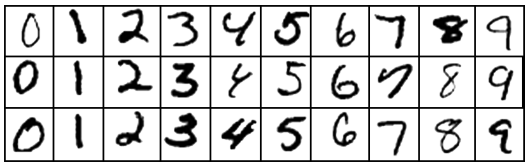
3. Federated Extended MNIST (FEMNIST)

   Federated Extended MNIST (FEMNIST) [257] is a modular extended version of the MNIST dataset created from the NIST database following a similar method as the MNIST. The digit version of FEMNIST houses 28x28-bit grayscale images from 10 classes (0-9) divided into 2,40,000 training and 40,000 testing samples. Sample images of the extended dataset are shown in Fig. 3.14c.

4. Cifar-10

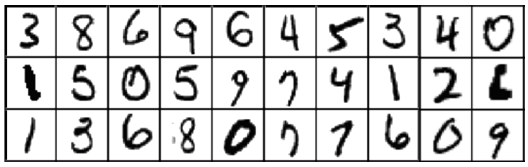
   CIFAR-10 [258] is a labeled subset of the "80 million tiny images" dataset [259]. It houses 32x32-bit RGB color images of 10 classes divided into 50000 training and 10000 testing samples. Each class individually has 5000 training and 1000 testing samples and represents objects like cars, planes, dogs, etc. A sample subset of the Cifar-10 is shown in Fig. 3.14d.

Table 3.5 provides a summary of the datasets used in the experimentation.

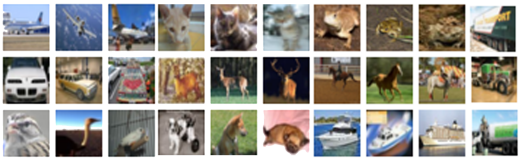


(a) Sample Images from the MNIST Dataset

(b) Sample Images from the Fashion MNIST Dataset



(c) Sample Images from the FEMNIST Dataset



(d) Sample Images from the CIFAR10 Dataset

Figure 3.14: Sample images from Datasets used in experiments

Table 3.5: Datasets used for Experimentation of Fed-WALQ.

Dataset	**Description**	**Sample Size**	**Train Split**	**Test Split**
MNIST	Image classification dataset with 10 classes of handwritten digits 0–9	28×28 (Gray)	60,000	10,000
Fashion MNIST	Image classification dataset with 10 classes of fashion accessories	28×28 (Gray)	60,000	10,000
FEMNIST	Extended version of the MNIST dataset created from the NIST database	28×28 (Gray)	240,000	60,000
CIFAR-10	Labelled subset of the "80 million tiny images " with RGB images of 10 classes	32×32 (RGB)	50,000	10,000

Table 3.6: Accuracy and Straggler Node Percentage achieved by Fed-WALQ over Different Datasets

Metric	MNIST	F-MNIST	FEMNIST	CIFAR-10
Acc (%)-7bit	99.04	83.20	77.25	91.10
Straggler (%)-7bit	15	18	35	38
Acc (%)-6bit	98.52	82.93	76.88	90.42
Straggler (%)-6bit	10.3	10.5	24.6	30.1
Acc (%)-5bit	98.05	82.60	76.50	89.60
Straggler (%)-5bit	8	4	20	24
Acc (%)-4bit	96.61	81.81	75.04	87.94
Straggler (%)-4bit	4.6	2.7	8.2	12.3
Acc (%)-3bit	95.05	81.36	74.60	87.15
Straggler (%)-3bit	3	1.3	4	6

3.3.2.4.4 Results of Simulations of Fed-WALQ

We benchmarked Fed-WALQ over the four datasets discussed in section 3.3 and compared it against standard FL and Fed-MSQE at 3-bit, 4-bit, 5-bit, 6-bit, and 7-bit quantization of parameters. The small number of techniques in comparison is attributed to the limited work done in quantisation-based FL, and even fewer implementations show an acceptable quantisation loss in the case of DSAs. Our Fed-WALQ demonstrated improved accuracy over Fed-MSQE in all the comparisons, ranging from significant to marginal gains depending on the complexity of the dataset. A summary of the accuracy observed in our proposed framework is shown in Table 3.6. The legend for the graphs in Fig. 3.16 and Fig. 3.17 are given in Fig. 3.15, while that for Fig. 3.19 and Fig. 3.20 is given in Fig. 3.18.

It is observed in table 3.6 that applying quantization from 7-bit to 5-bit leads to a steep reduction in the percentage of straggler nodes. However, the improvement becomes less significant when further reducing the precision to 4-bit or 3-bit quantization. This behaviour can be attributed to the fact that, prior to quantization, the model parameter size was large and difficult for the simulated clients to process within the required time constraints. Once 7-bit quantization is applied, the parameter size falls within the processing capabilities of most clients, resulting in a stabilized system and a sharp drop in straggler occurrences. Further reductions in bit-width continue to decrease the parameter size and slightly ease computation, but since the workload is already within client capabilities at higher bit-widths, the additional improvement in straggler reduction is comparatively marginal.

1. Performance Comparison in MNIST

Being the least complex dataset, the results of all techniques in comparison appear very close, well within the margin of error due to randomness. While standard FL tops the accuracy curve, Fed-MSQE shows a loss of roughly 0.46%, and Fed-WALQ shows a loss of just 0.2%, highlighting significant improvement in the narrow range available. A comparison of test accuracies of the three techniques over MNIST is shown in Fig. 3.16a for 3, 5, and 7-bit configurations and in Fig. 3.19a) for 4 and 6-bit configurations. Due to quantization and workload allotment, the number of stragglers is significantly reduced compared to traditional FL and Fed-MSQE, as highlighted by the graph in Fig. 3.17a and Fig. 3.20a.

2. Performance Comparison in Fashion-MNIST

   Moving to a slightly complex dataset, Fed-WALQ achieves an accuracy of 82.6% against the 83.1% of standard FL, showing a loss of 0.6%, marginally more significant than MNIST due to an increase in complexity. While Fed-MSQE achieves only 79.8% accuracy. The comparisons of test accuracies of the techniques over Fashion-MNIST are shown in Fig. 3.16b and Fig. 3.19b. When the ratio of stragglers is compared, Fed-WALQ outperforms standard FL and Fed-MSQE again. The straggler node ratio comparisons for Fashion MNIST are shown in Fig. 3.17b and Fig. 3.20b.

3. Performance Comparison in FEMNIST

   FEMNIST provides four times the number of samples than MNIST, introducing quantity and variation in the handwritten content for analysis, adding to the complexity. The standard FL achieves a test accuracy of 77.3% against the 76.5 achieved by Fed-WALQ in 5-bit mode, showing a 1% loss. The comparisons of test accuracies over FEMNIST for the techniques are shown in Fig. 3.16c and Fig. 3.19c. The reduction in straggler nodes is significant but not to the level seen in the case of MNIST and Fashion-MNIST. The graphs highlighting the same is shown in Fig. 3.17c and Fig. 3.20c.

4. Performance Comparison in CIfar-10

   CIFAR-10, the most complex dataset in our testing benchmark, showed the highest accuracy differences in comparisons. While standard FL shows test accuracy of 91.2, Fed-MSQE reaches 88.9 (2.52% loss) in 5-bit mode, and Fed-WALQ reaches 89.6 (1.75% loss) in 5-bit mode, highlighting a marginal gain over Fed-MSQE. Looking at the comparison graphs in Fig. 3.16d and Fig. 3.19d, we can infer that the gain will increase if the number of rounds increases. The straggler ratio sees the gap between standard FL and Fed-WALQ widen while that between Fed-WALQ and Fed-MSQE becomes negligible in the case of CIFAR-10,

as shown in Fig. 3.17d and Fig. 3.20d.

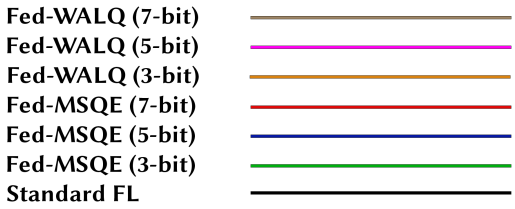
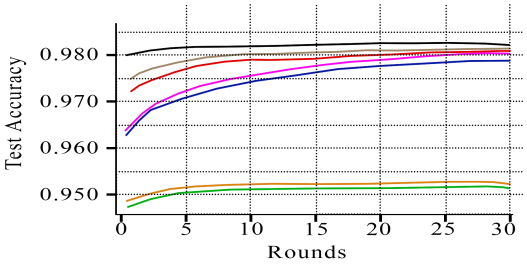
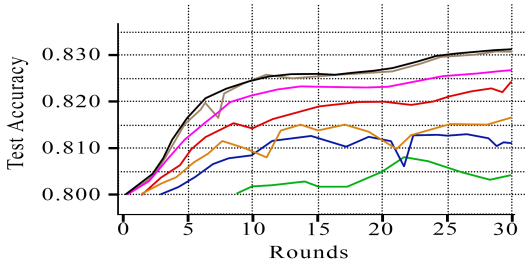


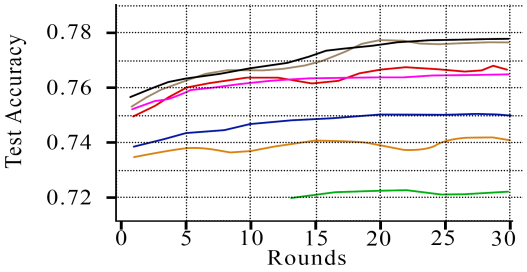
Figure 3.15: Legend for figures 3.16 and 3.17



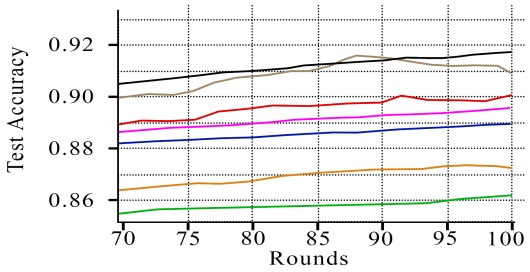
(a) MNIST



(b) Fashion MNIST



(c) FEMNIST



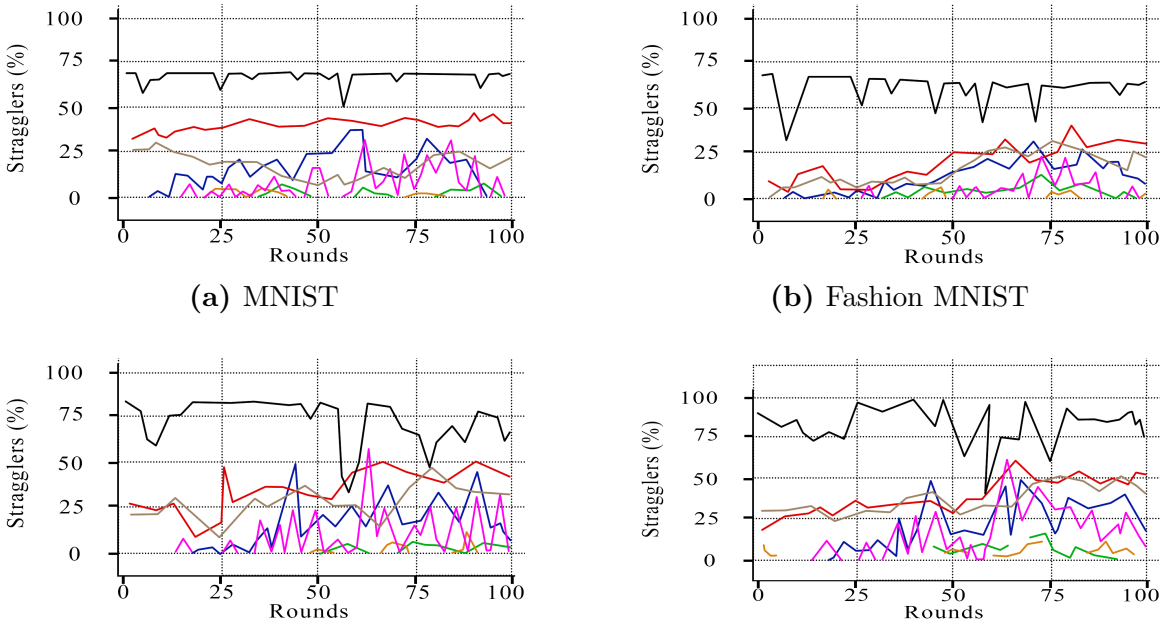
(d) CIFAR-10

Figure 3.16: Comparison of Test Accuracy of Fed-WALQ, Fed-MSQE, and Standard FL over various datasets during training in 3, 5, and 7-bit quantization.

In Figure 3.16a and Figure 3.19a, it is observed that the curves corresponding to 3-bit and 4-bit quantization form a distinct group near the lower region of the comparison plots. This occurs because 3-bit and 4-bit representations introduce significantly higher information loss compared to 5-bit or higher precision quantization, causing the achievable accuracy to saturate at a lower level. Furthermore, the scale of variation in these graphs is relatively small compared to that in other plots in the figures, making the separation between the lower-precision and higher-precision curves more visually prominent. A summary of all the test accuracy results is tabulated in Table 3.7.

The average straggler node percentage observed in various simulations is summarized in Table 3.8.

Based on the results observed, Fed-WALQ demonstrates its capability to compensate for



(a) MNIST **(b)** Fashion MNIST

(c) FEMNIST **(d)** CIFAR-10

Figure 3.17: Comparison of Straggler Node % observed in Fed-WALQ, Fed-MSQE, and Standard FL over various datasets during training in 3, 5, and 7-bit quantization.

Table 3.7: Accuracy Achieved by Standard FL, FED-MSQE, and FED-WALQ Over Different Datasets

Technique/Dataset	**MNIST**	**FMNIST**	**FEMNIST**	**CIFAR-10**
Standard FL	98.25	83.10	77.30	91.20
Fed-WALQ (7-bit)	99.04	83.20	77.25	91.10
FED-MSQE (7-bit)	98.80	81.20	75.57	90.79
Fed-WALQ (6-bit)	98.52	82.93	76.88	90.42
FED-MSQE (6-bit)	98.21	80.67	74.92	89.97
Fed-WALQ (5-bit)	98.05	82.60	76.50	89.60
FED-MSQE (5-bit)	97.80	79.80	74.38	88.90
Fed-WALQ (4-bit)	96.61	81.81	75.04	87.94
FED-MSQE (4-bit)	95.15	78.80	74.58	88.89
Fed-WALQ (3-bit)	95.50	81.36	74.60	87.15
FED-MSQE (3-bit)	94.70	80.13	72.06	85.04

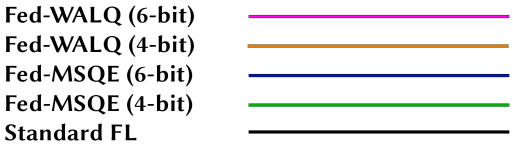
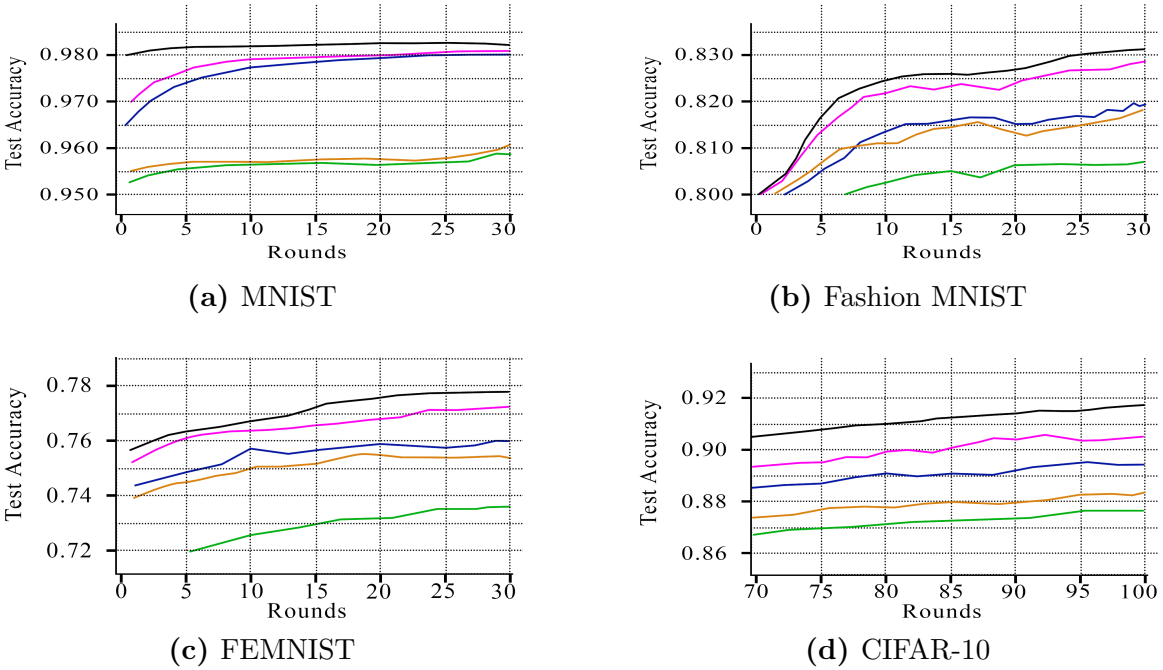


Figure 3.18: Legend for Figures 3.19 and 3.20



(a) MNIST

(b) Fashion MNIST

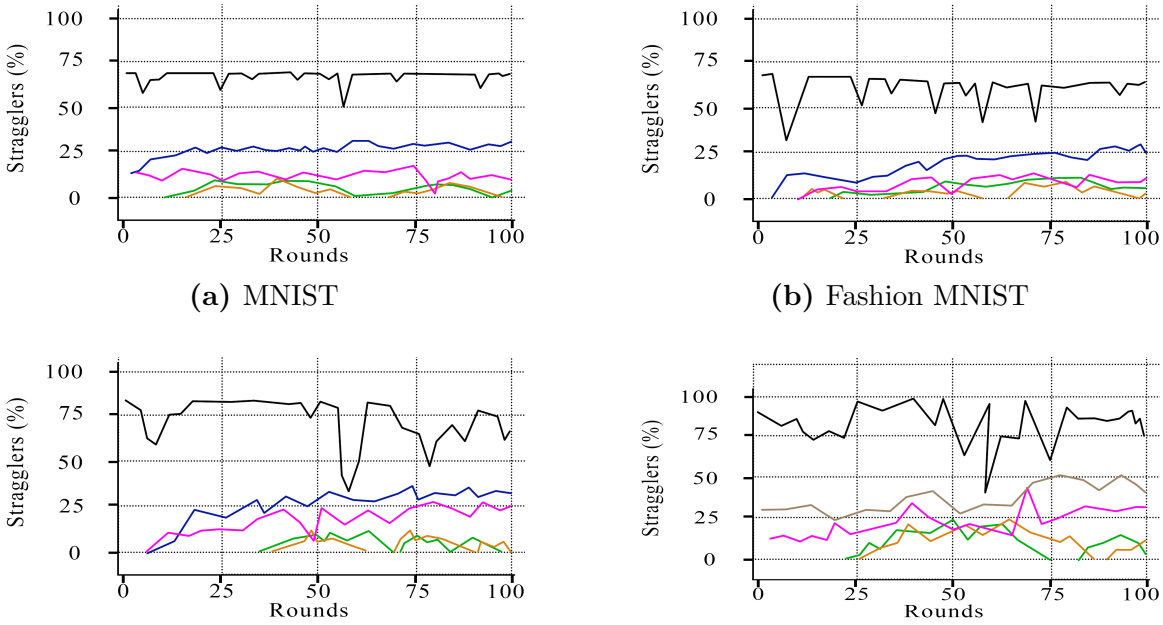
(c) FEMNIST

(d) CIFAR-10

Figure 3.19: Comparison of Test Accuracy of Fed-WALQ, Fed-MSQE, and Standard FL over various datasets during training in 4 and 6-bit quantization.

accuracy loss induced by quantization at various quantization levels. The compensation may be marginal or significant depending on the complexity of the training data and the quantization level, as evidenced by the varying gains over tests in dissimilar datasets. The most optimal gain is seen at the 5-bit quantization level, which also provides substantial security through scrambling vectors as compared to a 7-bit mode, which is almost identical to native vectors, and the 3-bit mode introduces a loss too significant to recover at the present stage.

However, the reduction in straggler nodes is substantial in all the test cases, highlighting a stable system compared to the existing techniques. The reduction observed is attributed to two prominent factors. 1) The reduction in parameter vector size through quantization minimizes the load on the network and, in turn, packet drops. 2) The active workload management for individual nodes is done at the server, which doesn't push a node beyond its optimal capabilities, minimizing the chance of withdrawal from the network.



(a) MNIST **(b)** Fashion MNIST

(c) FEMNIST **(d)** CIFAR-10

Figure 3.20: Comparison of Straggler Node % observed in Fed-WALQ, Fed-MSQE, and Standard FL over various datasets during training in 4 and 6-bit quantization.

Table 3.8: Average Straggler Node Percentage Observed (Lower is Better)

Technique/Dataset	**MNIST**	**FMNIST**	**FEMNIST**	**CIFAR-10**
Standard FL	68.0	66.0	71.0	79.0
Fed-WALQ (7-bit)	15.0	18.0	35.0	38.0
FED-MSQE (7-bit)	45.0	27.1	41.4	43.9
Fed-WALQ (6-bit)	10.3	10.5	24.6	30.1
FED-MSQE (6-bit)	21.0	21.2	30.6	32.1
Fed-WALQ (5-bit)	8.0	4.0	20.0	24.0
FED-MSQE (5-bit)	12.0	9.0	22.0	25.0
Fed-WALQ (4-bit)	4.6	2.7	8.2	11.1
FED-MSQE (4-bit)	5.1	6.6	10.4	12.3
Fed-WALQ (3-bit)	3.0	1.3	4.0	6.0
FED-MSQE (3-bit)	3.1	5.7	5.9	6.8

3.4 Summary of the Chapter

In this chapter, we proposed two federated learning frameworks (DRAFFSS and Fed-WALQ) aimed at solving the problems of resource optimization, client node stability, and security in resource-constrained training networks such as IoMT attaining the goal of our second research objective. Both frameworks utilize quantization to compress and encrypt the parameter vectors transmitted between the client and the server, and employ a workload-awareness mechanism to allot optimal load to individual participating devices. After collectively testing over 6 image classification datasets of varying complexity with a ResNet-18 model, the proposed frameworks showcase improvements in stability and staggler node percentages ranging from 18% to 65% decline depending on the extent of quantization applied. This is achieved while managing on-par accuracy ($\tilde{1}$% to 3.55% decline) with the standard FL and negligible time or computational overheads.

Chapter 4

Secure federated learning framework with dynamic lightweight ciphers for resource-constrained data-sensitive applications

---

*The frameworks proposed in chapter 3 rely on quntization for security. Sometimes the nodes in the network are too scarse that the client selection scheme cannot recover the quantization loss incurred. The purpose of this chapter * is to introduce federated learning based secure framework that utilize lightweight cipher with dynamic key for encrypted transmissions of training vectors. The approach presents a lossless encryption-based scheme with minimal resource demand and, therefore, no requirement for performance recovery. It is vital to evaluate the proposed framework with the state-of-the-art on multiple metrics, such as time overheads, resource utilization, and performance of the trained model, to highlight the utility of the proposed approaches for resource-constrained training networks.*

4.1 Introduction

Healthcare has been the most active field for FL to flourish [124, 169]. However, it is mainly dependent on the client nodes that belong to IoMT networks [260] for training the local models and, in turn, the global model, and the lack of security measures that do not impose a high computational burden poses a significant challenge. Many techniques have emerged, working towards better resource optimization in FL as discussed by authors in [34, 37, 261].

*The research work covered in this chapter has been Communicated in: Manu Narula, Jasraj Meena, and Dinesh Kumar Vishwakarma, "Secure Federated Learning Framework with Dynamic Lightweight Cipher (Fed-DLC) for Resource-Constrained Networks in Data Sensitive Applications,"

However, security implementations in networks with resource scarcity achieve very little success, as most of these security layers incur significant overhead [262]. We propose a light-computation secure Federated Learning Framework with Dynamic Lightweight Cypher (Fed-DLC) corresponding to our third objective for resource-constrained setups encountered in the Internet of Medical Things (IoMT) and DSA. We summarize the main contributions as follows:

- We propose a lightweight FL framework that provides state-of-the-art security with negligible computational overhead for low-powered DSA setups.
- To achieve real-time processing, we integrate a node-level dynamic cipher with linear time complexity.
- Through extensive experimentation across various classification datasets, we show that our framework provides real-time security with negligible computational overhead compared to existing techniques.

4.1.1 Scalability and utility of Fed-DLC

4.1.1.1 Scalability and Performance over non-IID data

The proposed Fed-DLC functions as a security plugin layer between a federated learning network's server and client nodes. This makes the performance over a dataset independent of the encryption process and solely dependent on the underlying implemented model and training algorithms. Hence, the Fed-DLC can be applied over non-IID data scenarios and is expected to show a similar level of security at negligible overheads. Furthermore, the encryption proposed used a short key aiming towards resource-constrained networks, which can be expanded with slight or no alteration in the algorithm, making it suitable for resource-rich networks.

4.1.1.2 Utility for non-DSA applications

The work focuses on DSA and resource-constrained environments like IoMT. Hence, the parameters proposed in the Fed-DLC are small. These parameters, such as key length, block size, and number of encryption rounds, can be tweaked to increase the security level for networks that allow for more computation power, effectively expanding the scope of the proposed framework outside the bounds of DSA and IoMT networks.

4.1.1.3 Threat model for Fed-DLC

In this chapter, the federated learning environment is assumed to operate under a semi-trusted setting where both communication channels and participating entities may pose security risks. The central server is considered honest-but-curious, meaning it follows the prescribed protocol but may attempt to infer sensitive information from transmitted model updates. Communication between clients and the server is assumed to be vulnerable to eavesdropping, replay, and man-in-the-middle attacks. Client devices are treated as semi-trusted nodes; a subset of clients may attempt to infer information about other participants or intercept transmitted updates. However, large-scale Byzantine attacks and fully compromised infrastructures are considered outside the scope of this work. Under this threat model, the primary objective of the proposed dynamic lightweight cipher is to ensure confidentiality and integrity of model updates during transmission while maintaining low computational overhead suitable for resource-constrained environments such as IoT and IoMT networks.

4.2 Related Work

Security is a non-exhaustive field of work. New threats, solutions, and improvements emerge daily, rapidly making state-of-the-art systems obsolete. FL is no stranger to this trend as the leakage of the bare minimal transmitted parameters, or gradients, can enable attackers to infer the used dataset with near-perfect accuracy in some cases [238]. Many researchers have found solutions to minimize or prevent this leak/inference. However, very few attempts have been made to secure FL in IoMT or similar fields due to a scarcity of resources.

4.2.1 FL in the Internet of Medical Things

In 2021, Farhad et al. [263] proposed FL integration in wearable IoMT sensors to analyze and supervise patients with persistent health conditions using DL. The emphasis emphasized FL from an efficiency point-of-view in the IoMT paradigm. However, the discussion is limited to the generic integration of FL without optimization. Samuel et al. [32] presented FL-based blockchain technology to diagnose COVID-19. The work, however, is limited to simulation on empirical calculations as no relevant public dataset is available.

Arya et al. [264], in 2022, emphasized the decentral nature of the IoMT data streams and the concerns about patient confidentiality in a centralized ML or DL approach. The authors proposed the integration of FL to enable ensemble learning in a decentralized manner, en-

suring data privacy and diverse collaborations between health centers. The work discussed the native security assurances of FL and did not consider the modern attacks developed to affect FL setups. Ahmed et al. [106] proposed a secure framework for FL in IoMT by integrating 6G and blockchain technologies for dynamic resource allocation and monitoring over the network. Their framework proved to be an improvement over the then state-of-the-art dynamic user allocation approaches regarding security and efficiency. Still, blockchain added a heavy overhead to the overall process. Khowaja et al. [153] proposed another technique for preserving the quality of training focusing on the field of IoMT, dealing with susceptible patient data. The authors propose a hybrid implementation of proximal gradients and split learning to provide an encryption layer to transmitted attributes. Lakhan et al. [265] proposed a privacy-preserving task scheduling system for IoMT based on blockchain integration. Their proposed system outperforms most of the contemporary fraud analysis and data validation techniques but also suffers from the heavy overheads of blockchain.

The limited success and research toward security enhancements in FL-based IoMT networks paved the way for experimentation with lightweight encryption schemes.

4.2.2 Lightweight Ciphers for FL-Based Techniques

Chandramouli et al. [266] put forth a mechanism to optimize encryption algorithms according to the battery consumption for low-powered devices. The approach shows significant performance gain in some algorithms but has limited viability in complex ciphers. Dini et al. [267] proposed a Lightweight Authenticated Re-Keying scheme (LARK) targeted toward Wireless Sensor Networks (WSN). The re-keying scheme provides a dynamic security layer for both forward and backward propagations and is immune to most long-term network attacks. Harn et al. [268] proposed an aggregation-based encryption technique for WSNs that enables piggybacking over network sensor nodes. The technique uses small key and data packets as opposed to the conventional larger key format hence exchanging security level for bandwidth optimization. Singh et al. [269] discussed an image encryption scheme targeting health applications based on optimization and the Lorenz system. The approach proved resilient to many attacks over benchmark Kaggle datasets. The approach can be enhanced further by replacing the base model with an efficient deep neural network. Wang et al. [270] and Adil et al. [271] took a step further and fused lightweight encryption with an authentication mechanism for WSNs and smart healthcare, respectively, tackling cyber-physical attacks.

4.3 Proposed Framework (Fed-DLC)

This section describes in detail, our hybrid approach combining the security aspect of quantization with optimizing throughput gained through workload-aware client selection. A detailed view of the complete framework with all the sub-functions is shown in figure 4.1. At the start of each round, the server shortlists a subset of random nodes from the nodes

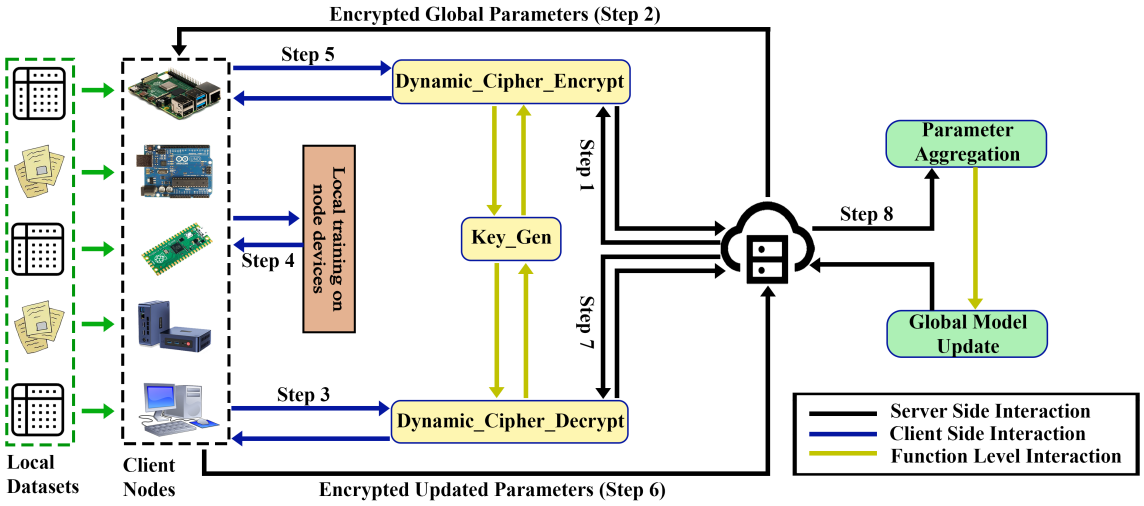


Figure 4.1: Proposed Architecture of Fed-DLC.

available in the network. The global model parameters are then encrypted via the Dynamic_Cipher_Encrypt function (Algorithm 4.1) called individually for all selected nodes (step 1) and are shared over the subset (step 2). The client devices receive the update and subsequently call the Dynamic_Cipher_Decrypt function (Algorithm 4.2) to access actual parameters (step 3). Local models are trained over the node's private data (step 4) and trained parameters are encrypted again by calling the Dynamic_Cipher_Encrypt function (step 5) and uploaded back to the server (step 6). Uploaded parameters are decrypted via Dynamic_Cypher_Decrypt (step 7) and aggregated into the global model (step 8) at the end of the round. Both algorithm 4.2 and algorithm 4.3 have access to the Key_Gen function (Algorithm 4.1) which generates encryption keys based on hardware and network configurations of the calling nodes.

4.3.1 Dynamic Lightweight Cipher

Given the resource scarcity associated with IoMT and most DSA in healthcare, security approaches with heavy encryptions or high computations do not yield success and may harm the network efficiency beyond acceptable levels. Hence lightweight ciphers are integrated

for security assurances in IoMT, however, lightweight means less computation which often translates to low levels of security against brute force or complex attacks.

To compensate for this issue, we integrate a lightweight cipher of a highly dynamic nature based on the works of Roy et al. [272]. We have modified the basic architecture proposed by the authors to fulfill the requirements of a federated learning network and split the overall process into three blocks interacting with each other and the network entities.

The Key Generation presents the dynamic portion of the process and is at the heart of the framework. Algorithm 4.1 shows the Key_Gen function that implements the same. The function accepts a 256-bit character input, which serves as the base key and is created by merging the MAC address, IP address and Name of the concerned node at the time of calling the function. The key is reshaped into a grid of 16x16 bits (line 1). The odd-numbered rows (1, 3, 5... 15) are subjected to cyclic left shift operations "row" number of times, i.e., row 1 is shifted 1 time, row 3 is shifted 3 times, and so on (lines 3-7). The even-numbered rows (2,4,6... 16) are subjected to cyclic right shift operations "row" number of times (lines 8-12). Similarly, odd and even-numbered columns are subjected to cyclic down shift and cyclic up shift operations, respectively, "column" number of times (lines 14-25). After all the shifting operations culminate, the square matrix is reshaped to a 64x4 grid representing 64 keys k[0], k[1]... k[63] having 4 bits each (lines 26-27). The sub-keys are then rearranged into 4 sets of 16 keys in the form of a 3D matrix of size [4][16][4] according to a predetermined order (line 28), and this 3D matrix is returned to the calling function (line 29).

Algorithm 4.1 Key_Gen

Require: Key K of size [256]
Ensure: Subkey set SK of size [4][16][4]
Reshape K from [256] to [16][16]
for $r = 1$ to 16 **do**
    if r is odd **then**
        for $i = 1$ to r **do**
            Cyclic left shift the r^{th} row
        end for
    else
        for $i = 1$ to r **do**
            Cyclic right shift the r^{th} row
        end for
    end if
end for
for each column $c = 1$ to 16 **do**
    if c is odd **then**
        for $i = 1$ to c **do**
            Cyclic down shift the c^{th} column
        end for
    else
        for $i = 1$ to c **do**
            Cyclic up shift the c^{th} column
        end for
    end if
end for
Reshape K from [16][16] back to [256]
Reshape K from [256] to [64][4]
Initialize $SK[4][16][4]$
$SK[0] \leftarrow \{K[0], K[8], K[15], K[1], K[2], K[9], K[14], K[3],$
    $K[4], K[10], K[13], K[5], K[6], K[11], K[12], K[7]\}$
$SK[1] \leftarrow \{K[16], K[24], K[31], K[17], K[18], K[25], K[30],$
    $K[19], K[20], K[26], K[29], K[21], K[22], K[27], K[28], K[23]\}$
$SK[2] \leftarrow \{K[32], K[40], K[47], K[33], K[34], K[41], K[46],$
    $K[35], K[36], K[42], K[45], K[37], K[38], K[43], K[44], K[39]\}$
$SK[3] \leftarrow \{K[48], K[56], K[63], K[49], K[50], K[57], K[62],$
    $K[51], K[52], K[58], K[61], K[53], K[54], K[59], K[60], K[55]\}$
return SK

The cumulative time complexity of the key generation process represents constant time and hence does not present a significant computation load. The other important aspect of the security implementation is the encryption and decryption of parameters using the key acquired from algorithm 4.1.

4.3.1.1 Key Randomness, Entropy, and Attack Surface Analysis

The proposed Fed-DLC framework derives encryption keys using client- and network-specific attributes such as device identifiers, IPv6 segments, name fragments, and time-of-join information. While this design ensures client-level uniqueness and lightweight implementation, it raises important considerations regarding entropy, predictability, and potential attack surface.

4.3.1.1.1 Entropy Analysis

Let the generated master key be denoted as:

$$K = \mathcal{F}(A_{name}, A_{IPv6}, A_{MAC}, T_{join}) \tag{4.3.1}$$

where $\mathcal{F}(\cdot)$ represents the key derivation process and the inputs correspond to client-specific attributes. The effective entropy of K depends on the unpredictability of each attribute, the independence between attributes, and the diffusion strength of the key generation algorithm. Certain attributes (e.g., MAC addresses or IPv6 prefixes) may exhibit structured or partially predictable patterns. If an adversary can approximate these values, the effective key space may be reduced. Therefore, the true entropy is bounded by the least unpredictable component used in the derivation.

4.3.1.1.2 Attack Surface Evaluation

The attack surface includes exposure of client network identifiers, reuse of partially predictable attributes across sessions, and linear key mixing operations that may not provide sufficient diffusion. If an adversary can model or approximate the key derivation function $\mathcal{F}$, the search space may be significantly reduced compared to a uniformly random key of equivalent length.

Algorithm 4.2 depicts the encryption mechanism for parameters. The function takes three inputs; the parameters to encrypt, the 256-bit key, and a 2-bit selector value. The key is passed to the Key_Gen function and the 3D key matrix is obtained (line 1). The parameter array is split into blocks of 256 bits and stored in a message array (lines 2-3). A 2D sub-key matrix is calculated from the 3D key matrix based on the selection bits (line 4). Each message block is then subjected to 16 rounds of encryption. In every odd-numbered round t, the t^{th} set of 4-bits in the message is passed through an Xor gate with the bits in the

t^{th} row of the sub-key, and the resulting bits are swapped as {1-1, 2-65, 3-129, 4-193}in the 256-bit block (lines 7-10). For every even-numbered round u, the u^{th} set of 4-bits in the message is passed through an Xnor gate with the bits the u^{th} row of the sub-key and the resulting bits are swapped as {1-193, 2-129, 3-65, 4-1}in the 256-bit block (lines 11-14). Once every block is processed, the message is reshaped into 1D array and returned to the calling function (lines 17-18).

Algorithm 4.2 Dynamic_Cipher_Encrypt

Require: Message array m_0, Key K, and Selector value S
Ensure: Encrypted message m_t

1: $Key[4][16][4] \leftarrow Key_Gen(K)$ ▷ Algorithm 4.1
2: $n \leftarrow \text{length}(m_0)/256$
3: Split m_0 into 256-bit blocks and store in $Message_blocks[\,]$
4: Determine $Sub_Key[16][4]$ based on selector bits $(S1, S2)$:

$$Sub_Key \leftarrow \begin{cases} Key[1], & S1 = 0 \text{ and } S2 = 0, \\ Key[2], & S1 = 0 \text{ and } S2 = 1, \\ Key[3], & S1 = 1 \text{ and } S2 = 0, \\ Key[4], & S1 = 1 \text{ and } S2 = 1 \end{cases}$$

5: **for** $b = 1$ to n **do**
6: **for** $t = 1$ to 16 **do**
7: **if** t is odd **then**
8: XOR the t^{th} 4 bits of Message_blocks[b] with the t^{th} row of Sub_Key
9: Swap these 4 bits according to:

$$\begin{cases} 1 \rightarrow 1, \\ 2 \rightarrow 65, \\ 3 \rightarrow 129, \\ 4 \rightarrow 193 \end{cases}$$

10: **else**
11: XNOR the t^{th} 4 bits of Message_blocks[b] with the t^{th} row of Sub_Key
12: Swap these 4 bits according to:

$$\begin{cases} 1 \rightarrow 1, \\ 2 \rightarrow 65, \\ 3 \rightarrow 129, \\ 4 \rightarrow 193 \end{cases}$$

13: **end if**
14: **end for**
15: **end for**
16: Reshape Message_blocks into 1-dimensional array m_t
17: **return** m_t

The cumulative time complexity of the encryption process represents linear time dependent on the message's length or the model parameters' size. Since all nodes on the training network will work on the same model configuration, the parameter size becomes constant, and the time complexity in a particular application decreases from linear to constant.

Algorithm 4.3 defines the decryption process, which follows the exact reverse steps discussed in algorithm 4.2 since both Xor and Xnor gates are self-inverting. The function takes three inputs: the parameter array, a 256-bit key, and a 2-bit selector value. The key is passed to the Key_Gen function, and a 3D key matrix is acquired (line 1). The parameter array is split into 256-bit blocks for processing (lines 2-3). A 2D sub-key matrix is calculated from the 3D key matrix based on the selection bits (line 4). Each message block is then subjected to 16 rounds of decryption. In every odd-numbered round t, the t^{th} set of 4-bits in the message are swapped as {1-1, 2-65, 3-129, 4-193} in the 256-bit block and then passed through an Xor gate with the bits the t^{th} row of the sub-key (lines 6-10). For every even-numbered round u, the u^{th}set of 4-bits in the message are swapped as {1-193, 2-129, 3-65, 4-1} in the 256-bit block and then passed through an Xnor gate with the bits the u^{th} row of the sub-key (lines 11-14). Once every block is processed, the message is reshaped into a 1D array and returned to the calling function (lines 17-18).

Algorithm 4.3 Dynamic_Cipher_Decrypt

Require: Message array m_t, Key K, and Selector value S
Ensure: Decrypted message m_0

$Key[4][16][4] \leftarrow Key_Gen(K)$ ▷ Algorithm 1
$n \leftarrow \text{length}(m_t)/256$
Split m_t into 256-bit blocks and store in $Message_blocks[\,]$
Determine $Sub_Key[16][4]$ based on selector bits $(S1, S2)$:

$$Sub_Key \leftarrow \begin{cases} Key[1], & S1 = 0 \text{ and } S2 = 0, \\ Key[2], & S1 = 0 \text{ and } S2 = 1, \\ Key[3], & S1 = 1 \text{ and } S2 = 0, \\ Key[4], & S1 = 1 \text{ and } S2 = 1 \end{cases}$$

for $b = 1$ to n **do**
    for $t = 1$ to 16 **do**
        if t is odd **then**
            Swap the t^{th} 4 bits of Message_blocks[b] using:

$$\begin{cases} 1 \rightarrow 193, \\ 2 \rightarrow 129, \\ 3 \rightarrow 65, \\ 4 \rightarrow 1 \end{cases}$$

            XOR the t^{th} 4 bits of Message_blocks[b] with the t^{th} row of Sub_Key
        else
            Swap the t^{th} 4 bits of Message_blocks[b] using:

$$\begin{cases} 1 \rightarrow 193, \\ 2 \rightarrow 129, \\ 3 \rightarrow 65, \\ 4 \rightarrow 1 \end{cases}$$

            XNOR the t^{th} 4 bits of Message_blocks[b] with the t^{th} row of Sub_Key
        end if
    end for
end for
Reshape Message_blocks into 1-dimensional array m_0
return m_0

The cumulative time complexity of the decryption process represents linear time dependent on the message's length or the size of the model parameters. Since all nodes on the training network will work on the same model configuration, the parameter size becomes constant, and the time complexity in a particular application reduces from linear to constant time.

4.3.2 Fed-DLC

Combining the dynamic cipher with FL, Fed-DLC considers the hardware and network configurations of the client nodes to add more randomness to the encryption-decryption process. Since every device poses different configurations in the network, no two devices share the same keys, adding to the security quotient in case one device's keys are compromised. Algorithm 4.4 outlines the exact process adopted in Fed-DLC. It takes two inputs: the number of training rounds T and a list of devices connected in the network. The algorithm is divided into two parts: the server side and the client side.

The process initiates on the server side, where a subset of client devices is selected randomly and queried for their IP address, MAC address, and device name (lines 2-4). For every client in the subset, the key is generated by taking the first ten characters of its name (replaced by blank spaces if the name is not long enough), sixteen characters of its IP address, and the time of joining the network in HH:MM:SS format, totaling thirty-two characters or 256 bits (line 6). The last two bits of the device's MAC address are taken as selection bits (line 7). Encrypted parameters are acquired by passing global parameters, key, and selection bits to the Dynamic_Cipher_Encrypt function (Algorithm 4.2) and are passed to the respective client (lines 8-10). When the responses from clients are received, the received parameters are passed to Dynamic_Cipher_Decrypt (Algorithm 4.3) along with the key and selection bits of the respective client to acquire decrypted parameters (lines 11-16). At the end of the round, all parameters are aggregated into the global model (line 17).

On the client side, when the client receives encrypted global parameters, it creates a key by combining the first ten characters of its device name, sixteen characters of its IP address, and the time of joining the network in HH:MM:SS format. It passes the parameters, key, and last two bits of the MAC address to the Dynamic_Cipher_Decrypt function to acquire decrypted parameters (lines 19-22). The client then trains the local model for one epoch on its private dataset (line 23) and passes the updated parameters along with the key and selection bits to the Dynamic_Cipher_Encrypt function before uploading back to the server (lines 24-25).

Algorithm 4.4 Fed-DLC

Require: Number of training rounds T, list of clients in the network R
Ensure: Trained global model weights w_t

for $t = 1$ to T **do**
  // On Server
  $C_t \leftarrow N$ randomly selected clients from R ▷ N remains constant each round
  Query each client in C_t for IP address, client name, and MAC address
  for each client c in C_t **do**
    $Key \leftarrow$ first 10 chars of client name
    $Key \leftarrow Key \,\|$ 16 chars of IPv6 address
    $Key \leftarrow Key \,\|$ time-of-join in HHMMSS format
    $S \leftarrow$ last 2 bits of MAC address
    $M \leftarrow$ Dynamic_Cipher_Encrypt(w_t, Key, S) ▷ Algorithm 4.2
    Transmit M to client c
  end for
  for each client c in C_t **do**
    Receive encrypted updated weights w_u from client
    $Key \leftarrow$ first 10 chars of client name
    $Key \leftarrow Key \,\|$ 16 chars of IPv6 address
    $Key \leftarrow Key \,\|$ time-of-join in HHMMSS format
    $S \leftarrow$ last 2 bits of MAC address
    $w_u \leftarrow$ Dynamic_Cipher_Decrypt(w_u, Key, S) ▷ Algorithm 4.3
  end for
  Aggregate all decrypted w_u to update global model w_t
  // On Clients
  Receive encrypted global weights w from server
  $Decrypted_w \leftarrow$ Dynamic_Cipher_Decrypt(w, Key, S) ▷ Algorithm 4.3
  Train local model for 1 epoch using $Decrypted_w$ to obtain w_u
  $w_u \leftarrow$ Dynamic_Cipher_Encrypt(w_u, Key, S) ▷ Algorithm 4.2
  return w_u
end for

The cumulative time complexity of the Fed-DLC is calculated to be in linear time dependent on the size of the parameters. Since in an FL training network, models with identical architecture are used across all devices, the size becomes a constant which reduces the time complexity to the constant time domain. In addition to this, the devices may disconnect and reconnect in between rounds which will result in a change in the IP addresses of the devices. This adds to the randomness of the key generation mechanism at the individual level. The security provided by Fed-DLC does not compromise the quality of training results as well as does not add any significant amount of time in the training rounds even within low-powered networks. This is demonstrated at length in the subsequent sections.

4.3.3 Deep dive in Key Generation and Encryption

We take a sample message and pass it through the functions to better explain the key generation, encryption, and decryption process.

4.3.3.1 Key Generation

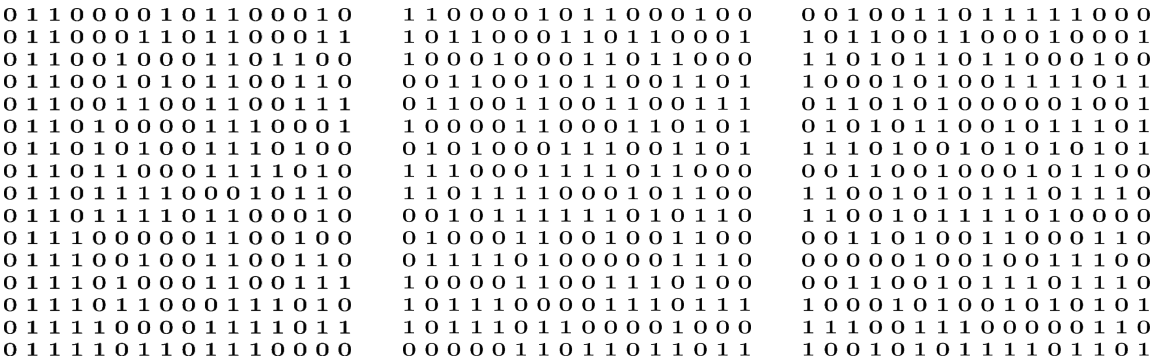
- Let the 256-bit key be "abcdefghijklmnopqrstuvwxyzabcdef"
- The key in binary form:

  01100001 01100010 01100011 01100100 01100101 01100110 01100111 01101000 01101001 01101010 01101011 01101100 01101101 01101110 01101111 01110000 01110001 01110010 01110011 01110100 01110101 01110110 01110111 01111000 01111001 01111010 01100001 01100010 01100011 01100100 01100101 01100110

- The Key_gen method converts the key to a 16x16 matrix for processing as shown in Figure 4.2a
- The key matrix is subjected to cyclic row shifts based on row numbers, and we obtain the modified matrix as shown in Figure 4.2b.
- The row-shifted matrix is then subjected to cyclic column shifts based on column numbers, and we obtain the matrix shown in Figure 4.2c.
- This matrix is then reshaped to dimensions 64x4 as shown in Figure 4.3. having rows 0 to 63.
- The rows are then rearranged into 4 planes or sub-keys having sixteen 4-bit rows according to the pattern described in the Key_Gen algorithm. The planes are illustrated in Figure 6.
- The array containing the 4 planes is returned to the calling method.

4.3.3.2 Message Encryption

- Let the message to be encrypted be "This is a sample message text 32" which has exactly 256-bits.
- The message in binary form:



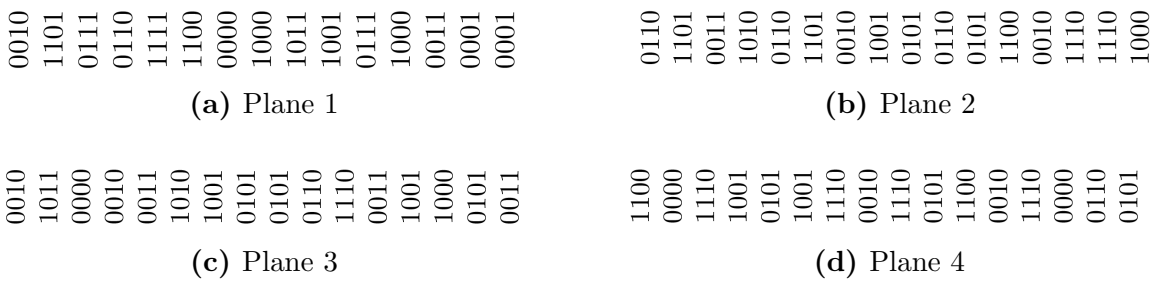
(a) Original key matrix before operations

(b) Key matrix after row operations

(c) Key matrix after column operations

Figure 4.2: Key matrix at different stages in Key_Gen method.

Figure 4.3: 64X4 Key matrix



(a) Plane 1

(b) Plane 2

(c) Plane 3

(d) Plane 4

Figure 4.4: Key matrix at different stages in Key_Gen method.

01010100 01101000 01101001 01110011 00100000 01101001 01110011 00100000 01100001 00100000 01110011 01100001 01101101 01110000 01101100 01100101 00100000 01101101 01100101 01110011 01110011 01100001 01100111 01100101 00100000 01110100 01100101 01111000 01110100 00100000 00110011 00110010

- Let the selection bits required for sub-key selection be '0' and '1', hence sub-key-2 will be used in the encryption process.

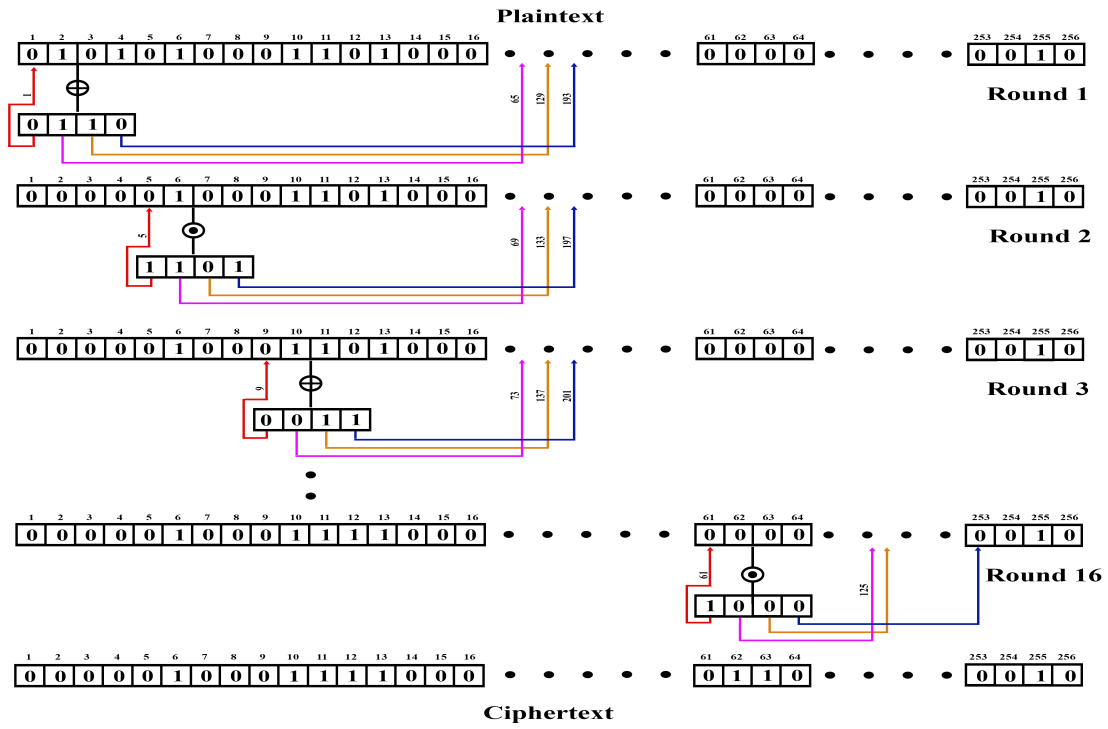


Figure 4.5: Demo run of encryption process.

- The plain text message is now passed through 16 rounds of encryption.

- In i^{th} round, the i^{th} 4-bit block of plain text is subjected to XOR (odd round) or XNOR (even round) operation with i^{th} row of sub-key-2 and the resulting bits are swapped as {1 - 1, 2 - 65, 3- 129, 4 - 193}. as shown in Figure 7.

- The ciphertext obtained at the end can be decrypted by reversing the encryption steps on destination machine.

‘

4.3.4 Resistance to Known-Plaintext Attack (KPA)

In a Known-Plaintext Attack (KPA), the adversary is assumed to observe one or more plaintext–ciphertext pairs:

$$(m_0, C_0) \tag{4.3.2}$$

Since the proposed encryption mechanism relies on XOR/XNOR operations, the ciphertext can be expressed as:

$$C = M \oplus K_{sub} \quad \text{or} \quad C = \neg(M \oplus K_{sub}) \tag{4.3.3}$$

where M denotes the plaintext block and K_{sub} represents the selected subkey.

If multiple plaintext–ciphertext pairs are available under the same key, linear relationships between M and C may potentially be exploited due to the linearity of XOR-based operations.

However, within the Fed-DLC framework, the following properties mitigate such risks:

- The encryption key is dynamically generated for each client.
- The time-of-join is incorporated into the key derivation process.
- Selector bits derived from device-specific MAC information vary across clients.

As a result, cross-client key reuse is minimized, thereby reducing the feasibility of large-scale linear key recovery across the network.

4.3.5 Resistance to Chosen-Plaintext Attack (CPA)

In a Chosen-Plaintext Attack (CPA), the adversary is allowed to select arbitrary plaintexts and obtain their corresponding ciphertexts under the same encryption key. Formally, the attacker queries the encryption oracle with chosen inputs:

$$M_1, M_2, \ldots, M_q$$

and observes the resulting ciphertexts:

$$C_i = \text{Enc}(M_i), \quad \forall i \in \{1, \ldots, q\}.$$

Given that the proposed Dynamic_Cipher_Encrypt algorithm relies on alternating XOR/XNOR operations, the encryption of a block can be represented as:

$$C = M \oplus K_{sub} \quad \text{or} \quad C = \neg(M \oplus K_{sub}),$$

where K_{sub} is the dynamically selected subkey determined by the selector bits.

Due to the linear nature of XOR-based transformations, if the same key were reused across multiple encryptions, carefully chosen plaintexts could potentially allow an adversary to derive linear relationships and recover portions of the subkey.

However, within the Fed-DLC framework, CPA resistance is strengthened by the following design characteristics:

- Client-specific dynamic key generation incorporating name fragments, IPv6 segments, and time-of-join.
- Selector-based subkey switching derived from device-specific MAC address bits.
- Round-dependent variability in key construction, limiting repeated-key exposure.

These properties significantly reduce the practicality of adaptive chosen-plaintext attacks across clients. Nevertheless, since the cipher does not currently provide a formal indistinguishability proof under CPA (IND-CPA), it cannot be claimed to achieve provable CPA security under standard cryptographic definitions. Establishing such guarantees remains an important direction for future enhancement.

4.4 Experimental Results and Analysis

In this section, we elaborate on our baseline models, experimental setup, the various datasets used, and the results obtained from the models discussed in the previous section on those datasets.

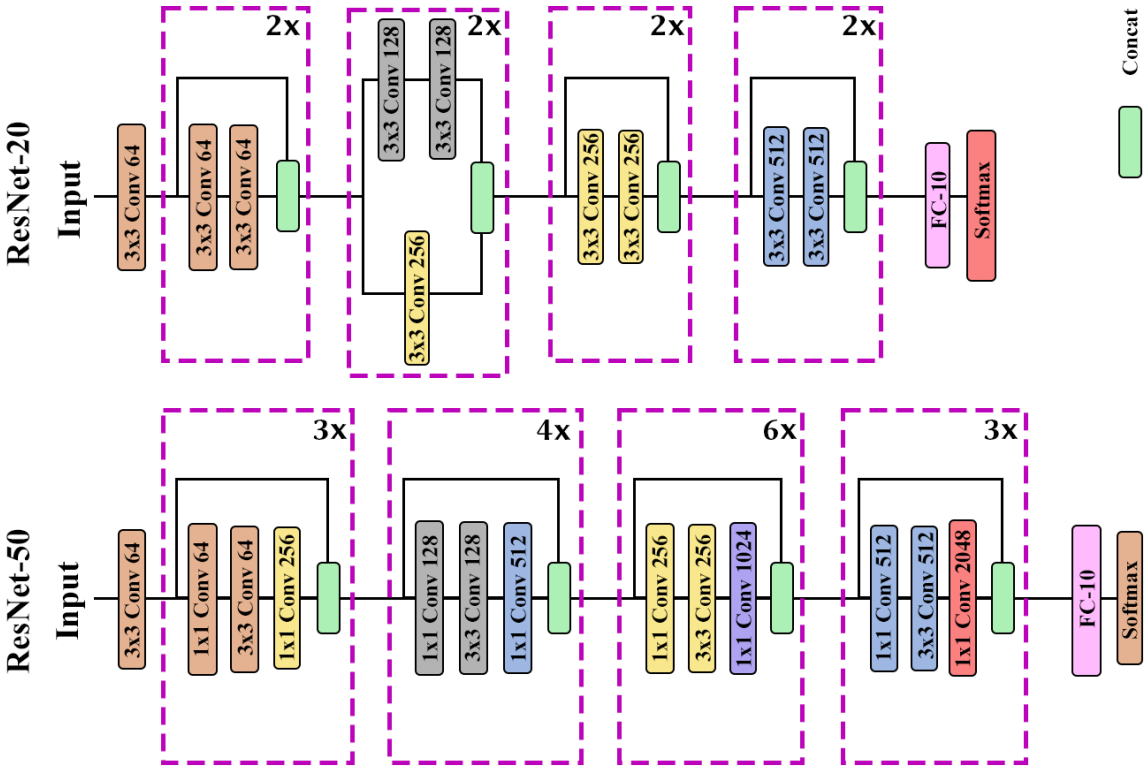


Figure 4.6: The architectures of ResNet-20 and ResNet-50 models.

4.4.1 Baseline Models

For a detailed demonstration of Fed-DLC's capabilities, we tested our framework on three models (ResNet-20, ResNet-50, and InceptionV3) over four datasets (MNIST, Fashion-MNIST, Extended MNIST, and CIFAR-10) with learning rate set to 0.01 with SGD optimizer. The three models used are as follows.

4.4.1.1 ResNet-20 and ResNet-50

The idea of Residual Blocks was designed to address the vanishing gradient issue [273]. A method known as skip connections is applied over the neural network which bypasses some levels in between by linking layer activations to subsequent layers. This method results in the creation of a leftover block. These leftover blocks, when stacked together are termed Residual Networks or ResNet. We employ two variations of ResNet having 20 and 50 layers respectively for our tests. The architectures of ResNet-20 and ResNet-50 are shown in Figure 4.6.

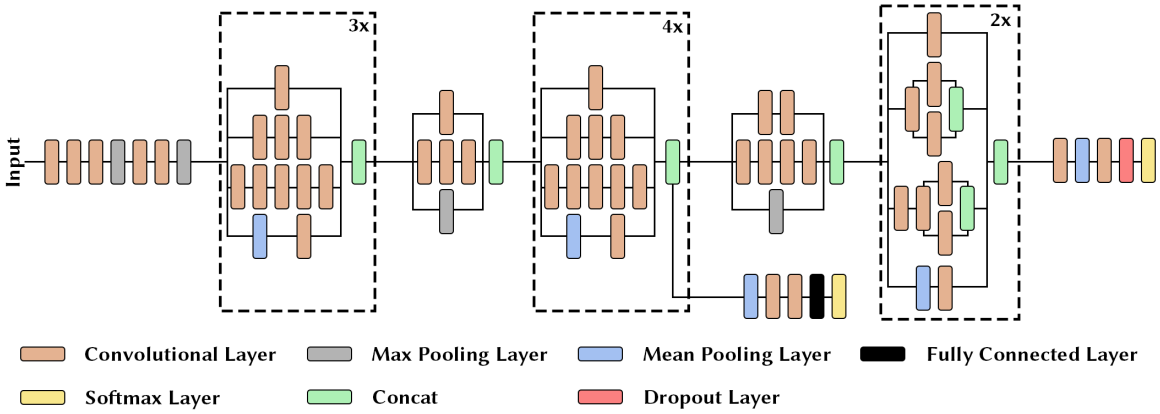


Figure 4.7: The architectures of the InceptionV3 model.

4.4.1.2 InceptionV3

The Inception v3 is an image classification model and the third iteration of the Inception model proposed by Szegedy et. al. [274]. It is a lightweight model compared to its first and second versions targeting constrained systems. The model comprises multiple symmetric and asymmetric blocks, including convolutions, average pooling, max pooling, concatenations, dropouts, and fully connected layers. Batch normalization is used extensively throughout the model and applied to activation inputs. Loss is computed using SoftMax. The architecture of the InceptionV3 is shown in Figure 4.7.

4.4.2 Experimental Setup for Fed-DLC Simulation

Our simulations are carried out on a system running Ubuntu 20.04 LTS equipped with an Intel (R) Xeon(R) Silver 4114 CPU @ 2.20GHz, 128GB DDR4 RAM, and two NVIDIA Titan RTX GPUs in having 24GB VRAM each. We implemented our models in Python 3.10.13 with Cuda 11.8 [253], Tensoflow 2.10 [275], and Numpy 1.24 [276] over Jupyter Notebook in the Conda virtual environment. 1000 devices are simulated simultaneously for the training federation to max out the resources available in the host machine to induce a resource-constrained environment for every simulated instance.

4.4.3 Dataset used for Fed-DLC Simulation

We tested our algorithm against the implementations discussed in [135, 277, 278] and standard FL over four openly available image datasets having dissimilar complexity, color space, and sample size (MNIST, Fashion-MNIST, EMNIST,FEMNIST) discussed in Chapter 3, Section 3.3.2.4.3. A brief description of the datasets and their composition is discussed in

Table 3.5 and samples from the datasets are shown in Figure 3.14.

4.4.4 Results of Simulations of Fed-DLC

We benchmarked Fed-DLC with ResNet-20,ResNet-50, and InceptionV3 models and compared them against standard FL, Fed-MSQE set to 5-bit parameter quantization, Turbo-Aggregate, and VerifyNet. The few techniques compared are accredited to the limited work done in secure FL for IoMT applications. The performance metrics used for the comparisons include accuracy, precision, recall, and f1-score.

Our Fed-DLC demonstrated improved accuracy over Fed-MSQE in all the comparisons while achieving on par results with standard FL. A summary of accuracy observed in our proposed framework is shown in table 4.1.

Table 4.1: Performance of Fed-DLC over Various Datasets

Metric/Dataset		**MNIST**	**Fashion-MNIST**	**FE-MNIST**	**Cifar-10**
Accuracy	RN-20	0.9962	0.9531	0.9934	0.9906
	RN-50	0.8303	0.8361	0.9970	0.9630
	IV3	0.9660	0.9051	0.9683	0.8943
Precision	RN-20	0.9962	0.9545	0.9933	0.9055
	RN-50	0.8330	0.8351	0.9970	0.9632
	IV3	0.9660	0.9048	0.9681	0.8949
Time	RN-20	29.8 Seconds			
	RN-50	68.5 Seconds			
	IV3	192.7 Seconds			

4.4.4.1 Performance Comparison with ResNet-20 Model

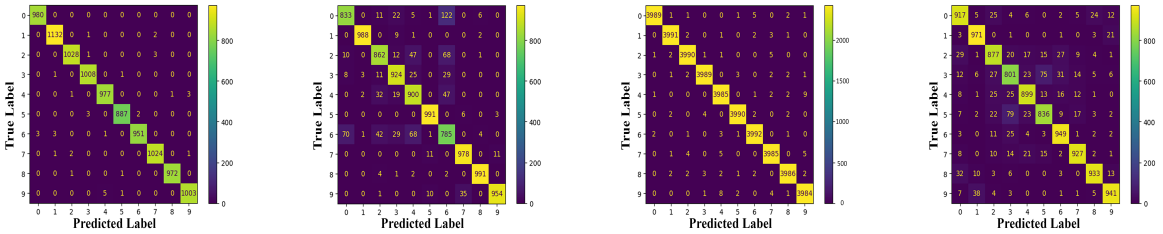
We first compare the performance achieved over the ResNet-20 model. The ResNet-20 is a lightweight model with 4,88,686 parameters, of which 4,88,236 are trainable in our test case over all datasets. The Fed-DLC does not suffer from the loss seen in the Fed-MSQE implementation and shows performance identical to Standard FL and Turbo-Aggregate regarding accuracy, precision, recall, and f1-score across all datasets with a variation of 0.1%, falling behind only VerifyNet by 0.97% on average. However, the authentication mechanism used in VerifyNet imbues a substantial computational overhead that may not be suitable for IoMT networks targeted by Fed-DLC. The summary of achieved results is presented in Table 4.2.

Figure 4.8 shows the confusion matrices generated after the experiments with the ResNet-

Table 4.2: Performance Comparison of Fed-DLC against state-of-the-art techniques in ResNet-20 Model

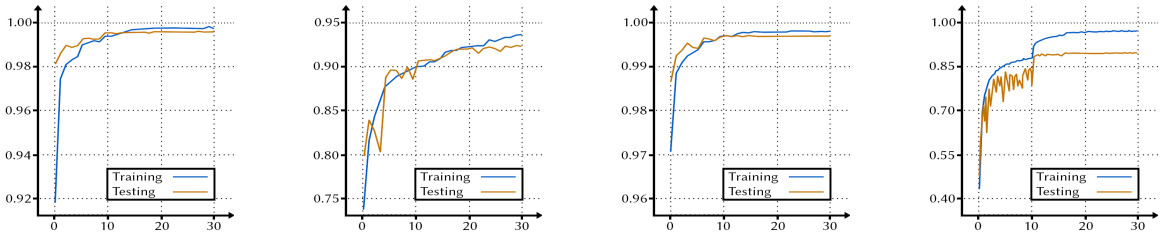
Model / Metric	**MNIST**		**Fashion-MNIST**		**FEMNIST**		**CIFAR-10**		**Security**	**Time**
	Acc.	Prec.	Acc.	Prec.	Acc.	Prec.	Acc.	Prec.		
Standard FL	0.9960	0.9960	0.9190	0.9201	0.9975	0.9974	0.9975	0.9054	Basic FL	28.5 s
Fed-MSQE (5-bit) [135]	0.9871	0.9871	0.9016	0.9016	0.9750	0.9751	0.8890	0.8892	Encryption	29.8 s
VerifyNet [278]	0.9991	0.9990	0.9480	0.9479	0.9985	0.9984	0.9376	0.9374	Authentication	29.3 s
Turbo-Aggregate [277]	0.9955	0.9955	0.9181	0.9185	0.9960	0.9959	0.8900	0.8900	Basic FL	26.2 s
Fed-DLC	0.9962	0.9962	0.9531	0.9545	0.9934	0.9933	0.9906	0.9055	Encryption	29.8 s

20 model over MNIST, Fashion-MNIST, E-MNIST, and Cifar-10 datasets. The accuracy curves for respective datasets are presented in Figure 4.9 The loss curves are presented in Figure 4.10.



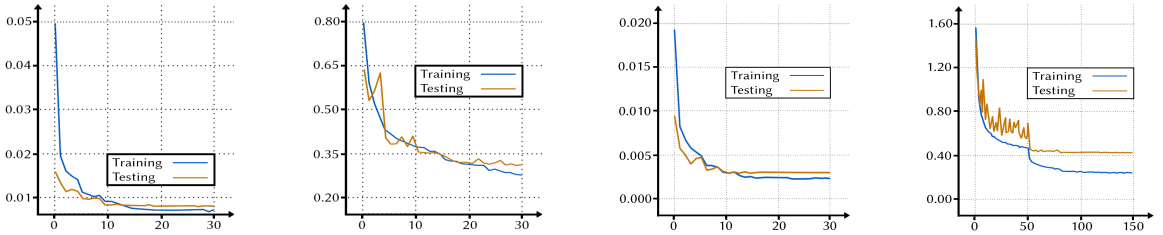
(a) MNIST **(b)** FMNIST **(c)** EMNIST **(d)** Cifar-10

Figure 4.8: Confusion Matrices for Fed-DLC applied over ResNet-20 over Various Datasets.



(a) MNIST **(b)** FMNIST **(c)** EMNIST **(d)** Cifar-10

Figure 4.9: Accuracy Over Epoch Graphs for Fed-DLC in ResNet-20 Model over Different Datasets.



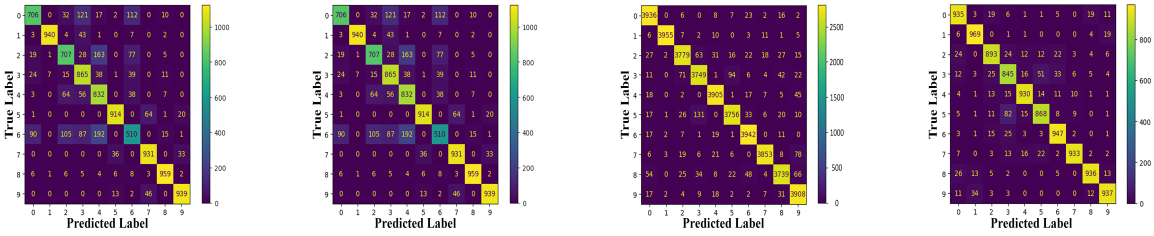
(a) MNIST **(b)** FMNIST **(c)** EMNIST **(d)** Cifar-10

Figure 4.10: Loss Over Epoch Graphs for Fed-DLC in ResNet-20 Model over Different Datasets.

4.4.4.2 Performance Comparison with ResNet-50 Model

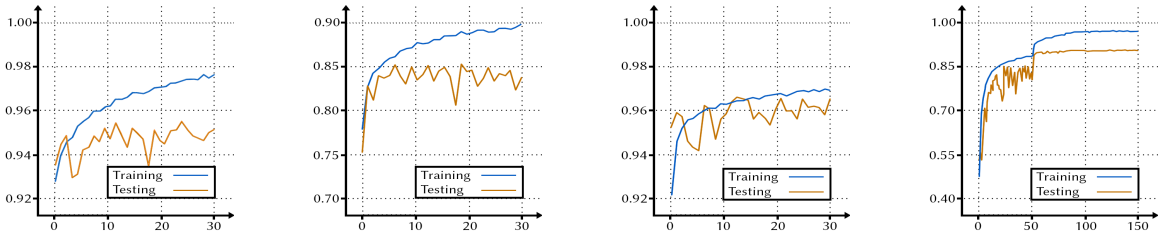
Moving to a heavier version of Residual Networks, we selected ResNet-50 as the second model in comparison. Transfer learning is applied using pre-trained weights of the "ImageNet" dataset. The transfered ResNet-50 has 23,608,202 parameters, of which 20,490 remain trainable in our test case over all datasets. Like ResNet-20, the Fed-DLC performs better than Fed-MSQE, showing identical results compared to Standard FL and Turbo-Aggregate

in all metrics across all datasets with a variation of 1%, again falling behind only VerifyNet by a margin of 3%. The summary of achieved results is presented in Table 4.3. Figure 4.11. shows the confusion matrices generated after the experiments with the ResNet-50 model over MNIST, Fashion-MNIST, E-MNIST, and Cifar-10 datasets. The accuracy curves for respective datasets are presented in Figure 4.12. The loss curves are presented in Figure 4.13.



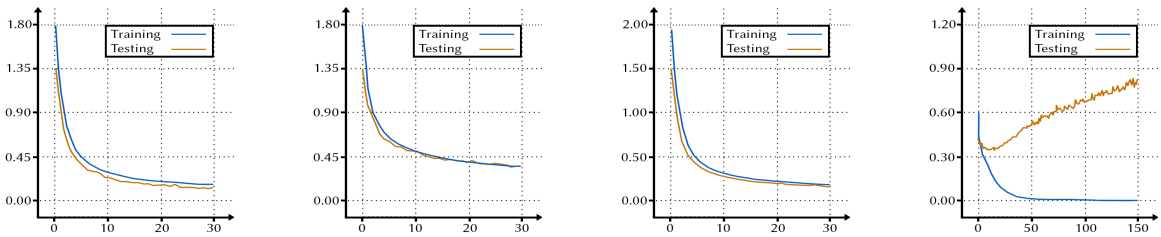
(a) MNIST **(b)** FMNIST **(c)** EMNIST **(d)** Cifar-10

Figure 4.11: Confusion Matrices for Fed-DLC applied over ResNet-50 over Various Datasets.



(a) MNIST **(b)** FMNIST **(c)** EMNIST **(d)** Cifar-10

Figure 4.12: Accuracy Over Epoch Graphs for Fed-DLC in ResNet-50 Model over Different Datasets.



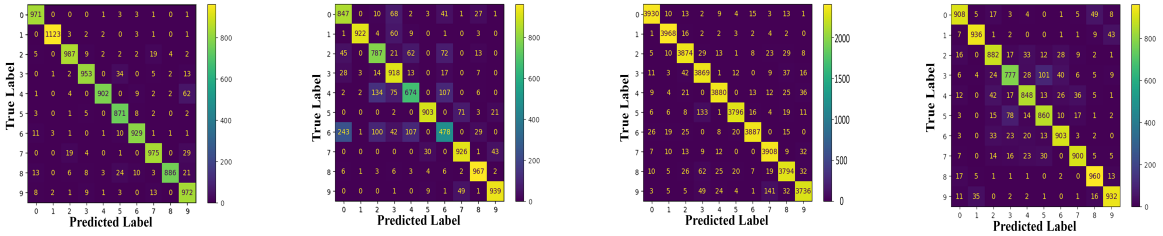
(a) MNIST **(b)** FMNIST **(c)** EMNIST **(d)** Cifar-10

Figure 4.13: Loss Over Epoch Graphs for Fed-DLC in ResNet-50 Model over Different Datasets.

4.4.4.3 Performance Comparison with InceptionV3 Model

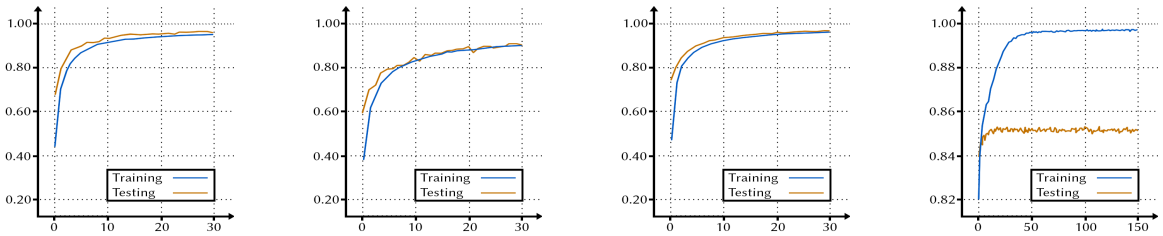
The final model tested in the experiment is Inception V3. Being the heaviest model with more than 22 million parameters, we applied transfer learning like that in the case of ResNet-

50 with weights pre-trained on ImageNet. Like the other tested models, the Fed-DLC outperforms Fed-MSQE and attains on-par results with Standard FL and Turbo-Aggregate across all datasets trailing behind VerifyNet. The summary of achieved results is presented in Table 4.4. Figure 4.14 shows the confusion matrices generated after the experiments with the Inception-V3 model over MNIST, Fashion-MNIST, E-MNIST, and Cifar-10 datasets. The accuracy curves for respective datasets are presented in Figure 4.15 and the loss curves are presented in Figure 4.16. As evidenced by data presented in Tables 4.2, 4.3, and 4.4,



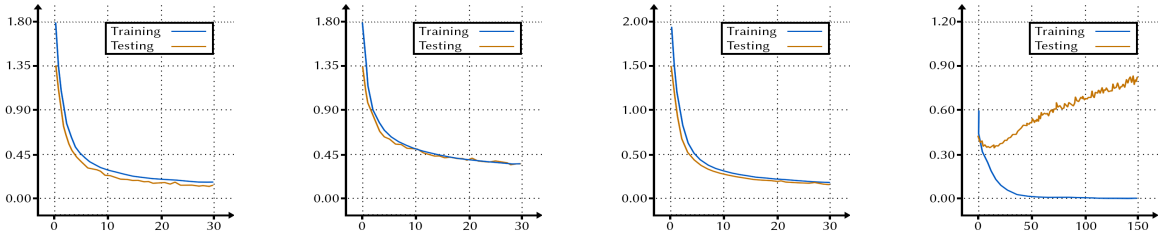
(a) MNIST **(b)** FMNIST **(c)** EMNIST **(d)** Cifar-10

Figure 4.14: Confusion Matrices for Fed-DLC applied over InceptionV3 over Various Datasets.



(a) MNIST **(b)** FMNIST **(c)** EMNIST **(d)** Cifar-10

Figure 4.15: Accuracy Over Epoch Graphs for Fed-DLC in InceptionV3 Model over Different Datasets.



(a) MNIST **(b)** FMNIST **(c)** EMNIST **(d)** Cifar-10

Figure 4.16: Loss Over Epoch Graphs for Fed-DLC in InceptionV3 Model over Different Datasets.

Fed-DLC's performance is identical to standard FL, and Turbo-Aggregate applied with the same model over the same Datasets. The differences seen in these readings are well within

the margin of error. Furthermore, compared to the Fed-MSQE, Fed-DLC provides similar or increased security without the performance loss observed in the Fed-MSQE.

Fed-DLC leads over the quantization-based techniques as there is no quantization loss incurred in our proposed cipher. Fed-DLC only trails behind VerifyNet, which employs a homomorphic encryption-based authentication scheme that increases performance while providing substantial security. However, the computation overhead makes it unsuitable for resource-constrained networks, which are the focus of Fed-DLC, eliminating competition between these techniques.

Also inferred from Tables 4.2, 4.3, and 4.4, the time overhead presented by our proposed method is minute, ranging from 4.6% to 8%increase from naïve durations. The overhead is observed to be increasing with the increase in model size. The induced overhead remains negligible since the models applied over IoMT networks are on the lower end of the size spectrum.

Table 4.3: Performance Comparison of Fed-DLC against Standard FL, Fed-MSQE, VerifyNet, and Turbo-Aggregate in ResNet-50 Model

Model/Metric	MNIST		Fashion-MNIST		FEMNIST		Cifar-10		Security	Compute Time
	Accuracy	Precision	Accuracy	Precision	Accuracy	Precision	Accuracy	Precision		
Standard FL	0.9536	0.9550	0.8310	0.8332	0.9628	0.9627	0.9196	0.9198	Basic FL	64.2s
Fed-MSQE (5-bit) [135]	0.9325	0.9242	0.8100	0.8101	0.9540	0.9540	0.8812	0.8813	Encryption	68.6s
VerifyNet [278]	0.9850	0.9850	0.8621	0.8622	0.9912	0.9913	0.9300	0.9303	Authentication	66.4s
Turbo-Aggregate [277]	0.9530	0.9542	0.8206	0.8210	0.9500	0.9501	0.9015	0.9026	Basic FL	55.1s
Fed-DLC	0.8303	0.8330	0.8361	0.8351	0.9970	0.9970	0.9630	0.9632	Encryption	68.5s

Table 4.4: Performance Comparison of Fed-DLC against Standard FL, Fed-MSQE, VerifyNet, and Turbo-Aggregate in InceptionV3 Model

Model/Metric	MNIST		Fashion-MNIST		FEMNIST		Cifar-10		Security	Compute Time
	Accuracy	Precision	Accuracy	Precision	Accuracy	Precision	Accuracy	Precision		
Standard FL	0.9575	0.9575	0.8356	0.8360	0.9684	0.9684	0.8938	0.8944	Basic FL	178.5s
Fed-MSQE (5-bit) [135]	0.9445	0.9448	0.8064	0.8059	0.9167	0.9168	0.8447	0.8447	Encryption	192.5s
VerifyNet [278]	0.9865	0.9864	0.8566	0.8560	0.9884	0.9884	0.9231	0.9233	Authentication	180s
Turbo-Aggregate [277]	0.9570	0.9570	0.8225	0.8226	0.9571	0.9568	0.8821	0.8821	Basic FL	161.7s
Fed-DLC	0.9660	0.9660	0.9051	0.9048	0.9683	0.9681	0.8943	0.8949	Encryption	192.7s

4.5 Summary of this Chapter

In this chapter, we have proposed Fed-DLC, a secure FL framework with lightweight dynamic encryption focusing on IoMT and similar resource-constrained networks encountered frequently in DSAs. Through extensive experimentation with ResNet-20, ResNet-50, and InceptionV3 models over MNIST, Fashion-MNIST, E-MNIST, and Cifar-10 datasets, we observe that Fed-DLC achieves identical performance as that of native or standard FL and Turbo-Aggregate while providing an increased level of security and negligible induced overhead. In addition, we compare our framework with Fed-MSQE and VerifyNet, where Fed-DLC highlights a similar level of security without any performance loss observed in Fed-MSQE and falls behind the computationally expensive VerifyNet.

Chapter 5

Secure federated learning framework with differential privacy and homomorphic vector encryption

*The frameworks discussed so far focus on resource-constrained training networks, thus, have to balance stability with security. The purpose of this chapter * is to propose a secure hybrid framework (Fed-DHVE) for federated learning applied over DSA networks having sufficient resources like large hospital networks of banking clusters. The framework combines aspects of differential privacy, homomorphic encryption, and IFL to provide both security and efficiency to the training network.*

5.1 Introduction

A range of techniques have recently emerged that aim to enhance both the security and resource efficiency of Federated Learning (FL) systems. Among them, quantization is commonly employed to reduce communication overhead and protect sensitive data, albeit often at the expense of model accuracy [135, 154]. To address this trade-off, certain algorithms have been developed to recover performance lost due to quantization. For instance, Li et al. [155] proposed a highly effective method that extends Incentive-Aware FL by incorporating workload-awareness, thereby improving the overall training accuracy under quantized settings. Building upon this, Narula et al. [279] introduced a hybrid framework that integrates quantization with workload-aware FL. Their approach successfully reduces the impact of quantization loss while preserving data privacy. However, none of the existing methods have been able to fully eliminate the performance degradation.

*The research work covered in this chapter has been communicated in: Manu Narula, Jasraj Meena, and Dinesh Kumar Vishwakarma, "Federated Framework with Differential Privacy over Homomorphic Vector Encryption for Data-Sensitive Applications," *Communicated*

In this chapter, we propose a hybrid Federated framework with Differential Privacy (DP) over Homomorphic Vector Encryption for Data-Sensitive Applications (Fed-DHVE), corresponding to our third objective, which replaces quantization as the base security measure and provides a stable and private environment for DSA implementations

5.1.1 Differential Privacy

Differential Privacy (DP) provides a formal privacy guarantee by bounding the change in output distributions resulting from the inclusion or exclusion of an individual data record through the injection of carefully calibrated random noise. This mechanism limits the influence of any single participant, thereby preventing adversaries from inferring individual membership during data analysis or model training. In pure Local Differential Privacy (LDP), noise is independently applied at each client or edge device prior to sharing data or model updates, which often leads to high variance and degraded utility, particularly when local datasets are small [280]. Distributed Differential Privacy (DDP) effectively mitigates this limitation by enabling collaborative privacy preservation across multiple distributed nodes, allowing the aggregation of perturbed information to achieve a superior privacy–utility trade-off while maintaining rigorous DP guarantees. Consequently, DDP is well-suited for federated learning and edge intelligence scenarios, where accurate global model construction is required without compromising individual privacy [281].

To ensure clarity and reproducibility, the DP mechanism employed in the proposed Fed-DHVE framework is explicitly parameterized. In each training round, local gradients are clipped using an L_2-norm bound C, which limits the sensitivity of individual client updates and ensures that no single data point disproportionately influences the aggregated model. After clipping, Gaussian noise sampled from $\mathcal{N}(0, \sigma^2 C^2 I)$ is added to the gradients before encryption and transmission.

The privacy guarantee is defined in terms of (ε, δ)-Differential Privacy, where ε controls the privacy–utility trade-off and δ represents the probability of privacy failure. Smaller values of ε provide stronger privacy protection but introduce greater noise, potentially affecting model accuracy. The chosen clipping bound and noise multiplier σ are selected to balance formal privacy guarantees with practical model performance, ensuring meaningful protection against gradient reconstruction and membership inference attacks while maintaining acceptable convergence behaviour.

5.1.2 Homomorphic Encryption

Homomorphic Encryption (HE) is a cryptographic technique that enables arithmetic operations to be performed directly on encrypted data, such that decrypting the computed result yields the same output as if the operations had been performed on the plaintext. In federated learning, healthcare analytics, and edge AI systems, HE enables clients, medical institutions, or edge devices to securely transmit encrypted features, model updates, or inference requests to untrusted servers while ensuring that sensitive information—such as patient health records or proprietary data—remains confidential throughout the computation. By supporting operations on encrypted gradients or parameters, HE prevents information leakage during collaborative model training and aggregation, making it particularly suitable for privacy-critical healthcare applications and distributed intelligence at the network edge. Although Fully Homomorphic Encryption (FHE) provides strong security guarantees, its substantial computational and communication overhead often limits practical deployment; therefore, real-world systems typically rely on optimized or levelled HE schemes to balance privacy protection and efficiency in resource-constrained federated and edge learning environments [282].

We summarize the main contributions of this paper as follows:

- We introduce a hybrid framework that incorporates (DP) to strengthen defenses against inference and inversion attacks on intercepted parameter vectors in sensitive FL scenarios.
- To further reinforce security, the framework integrates HE during parameter transmission, ensuring confidentiality even in the presence of potential eavesdroppers.
- We validate the effectiveness of our approach through comprehensive experiments on multiple classification datasets, demonstrating that the proposed framework achieves robust and stable training with minimal degradation in accuracy, making it well-suited for deployment in resource-constrained environments.

5.1.3 Assumptions for Scalable Real-World Implementation of Fed-DHVE

The DP mechanism integrated into our framework necessitates a substantial number of active client devices to ensure accurate and reliable model training. To accommodate this,

we expanded our simulated client pool from the conventional 100 devices to 500 participating devices per training round, drawn from a larger simulated network of 1500 clients. This scaling approach reflects the requirement that, in real-world deployments, a similar number of active participants is essential to achieve optimal performance.

Our experimental results indicate that, in moderately sized networks comprising approximately 1000 to 1500 devices, the framework incurs minimal computational and communication overhead. However, as the network size surpasses 5000 nodes, the overhead increases significantly—almost exponentially—posing a challenge to the framework's scalability and real-time applicability in large-scale deployments.

5.2 Related Work in differential privacy and homomorphic encryption with federated learning

5.2.1 Differential Privacy with Federated Learning

FL enables collaborative model training without sharing raw data; however, prior studies have shown that gradients and model updates can still leak sensitive information through inference attacks. To address this issue, DP has been extensively integrated into FL frameworks. Early work extended differentially private stochastic gradient descent (DP-SGD) to the federated setting, providing formal client-level or record-level privacy guarantees by injecting calibrated noise into local updates or aggregated models [280]. Subsequent studies explored Local Differential Privacy (LDP), where noise is added directly at client devices, offering strong privacy protection at the cost of reduced model accuracy when local data are limited. More recent approaches, such as shuffled and distributed DP, improve the privacy–utility trade-off by leveraging secure shuffling or multi-client collaboration, making DP-based FL more practical for large-scale deployments [281].

5.2.2 Homomorphic Encryption with Federated Learning

HE has been proposed as a cryptographic solution to protect the confidentiality of model updates during communication and aggregation in FL. By enabling arithmetic operations on encrypted gradients or parameters, HE prevents an untrusted server or cloud service provider from accessing plaintext updates. Early HE-based FL systems focused on linear and logistic regression using additive HE schemes, demonstrating the feasibility of encrypted

aggregation. Later work extended these ideas to deep learning by employing levelled or somewhat homomorphic encryption to reduce computational overhead. Although HE offers strong confidentiality guarantees, its high computation and communication costs remain a major challenge, particularly in resource-constrained edge and mobile environments [282].

5.2.3 Hybrid Approaches (Differential Privacy and Homomorphic Encryption)

Recognizing that DP and HE address complementary threat models, recent research has investigated hybrid frameworks that combine both techniques in federated learning. In such systems, DP is typically applied at the client side to limit information leakage from individual updates, while HE is used to encrypt the noisy updates during transmission and aggregation, protecting against an honest-but-curious server. These hybrid approaches provide stronger end-to-end privacy guarantees than using either technique alone. However, balancing privacy strength, model utility, and system efficiency remains an open research problem, motivating ongoing work on adaptive noise injection, lightweight encryption schemes, and efficient secure aggregation protocols [283].

5.3 Proposed Framework (Fed-DHVE)

This section describes in detail our hybrid approach combining the security of differential privacy and homomorphic encryption. A detailed view of the complete framework, including all its sub-functions, is shown in Figure 5.1.

At the start of each round, the server selects a subset of random nodes from the available nodes in the network (step 1) and queries for their optimal workload capacities (step 2). The global model parameters are then encrypted via Conjugate Search Problem (CSP) based homomorphic encryption (Algorithm 5.2). and are shared over the subset (step 3). The client devices receive the update, and Local models are trained over the node's private data (step 4). The trained parameters are modified by adding noise as part of local differential privacy (step 5) and then encrypted using CSPH encryption (step 6) and returned to the server (step 6). Uploaded parameters are aggregated using FedAVG (step 7), and the global model is updated (step 8) at the end of the round.

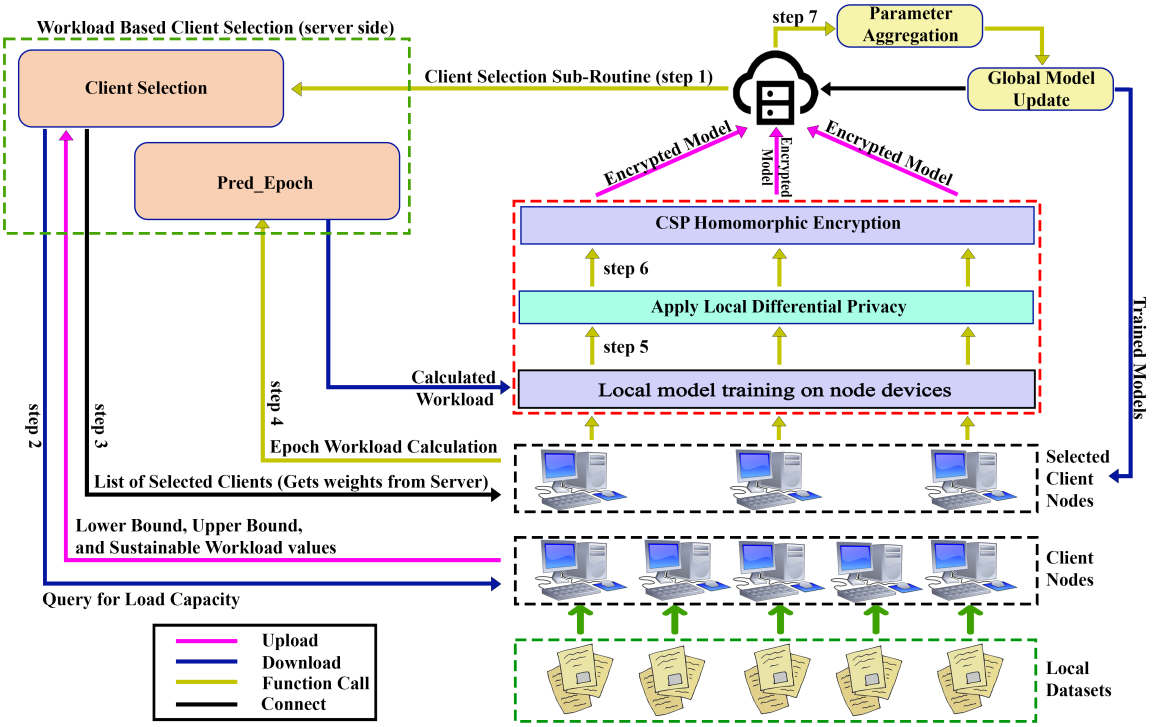


Figure 5.1: Proposed Architecture of Fed-DHVE.

5.3.1 Standard FL algorithm

FL, introduced by Google [8], enables collaborative model training without sharing raw data. In each training round, a randomly selected subset of clients, called the federation, receives the current global model from the server. Clients update the model locally using their private data and return the modified parameters. The server aggregates these updates to refine the global model. This process repeats over multiple rounds with different client subsets. Later enhancements replaced direct parameter sharing with gradient updates, ensuring sensitive data remains local to the device and remains confidential.

5.3.2 Differential Privacy-based security with Federated Learning

DP is a mathematical framework that protects individual data by ensuring that the inclusion or exclusion of a single record has minimal impact on the output of a computation. This is achieved by adding calibrated noise to queries or model updates, controlled by a privacy parameter ε. DP provides formal, quantifiable privacy guarantees, enabling secure analysis and learning from sensitive data without revealing personal information.

Algorithm 5.1, T denotes the total number of federated rounds, and S_t is the randomly

selected subset of K clients participating in round t (line 1-3). Each client computes its local gradient $g_i = \nabla\mathcal{L}(w_t; D_i)$ (line 5), clips it to a maximum norm C (line 6), and adds Gaussian noise $\tilde{g}_i = g_i + \mathcal{N}(0, \sigma^2 C^2 I)$ to ensure differential privacy (line 7). The server aggregates the noisy updates to produce $w_{t+1} = w_t - \eta \frac{1}{K} \sum_{i \in S_t} \tilde{g}_i$ (lines 8-9). This process repeats until the global model converges while protecting client-level privacy.

Algorithm 5.1 Differentially Private Federated Learning

Require: Number of rounds T, initial global model w_0, total clients N, subset size K, learning rate η, gradient clipping norm C, noise scale σ

Ensure: Trained global model w_T

1: **for** $t = 1$ to T **do**
2: **// Server Operations**
3: Randomly select a subset of clients $S_t \subseteq \{1, 2, \ldots, N\}$ of size K
4: Send current global model w_t to each client $i \in S_t$
5: **// Client Operations (in parallel)**
6: **for** each client i in S_t **do**
7: Compute local gradient: $g_i = \nabla\mathcal{L}(w_t; D_i)$
8: Clip gradient: $g_i \leftarrow \dfrac{g_i}{\max\left(1, \dfrac{\|g_i\|_2}{C}\right)}$
9: Add Gaussian noise: $\tilde{g}_i \leftarrow g_i + \mathcal{N}(0, \sigma^2 C^2 I)$
10: Send noisy gradient $\tilde{g}_i$ to the server
11: **end for**
12: **// Server Aggregation**
13: $w_{t+1} \leftarrow w_t - \eta \cdot \dfrac{1}{K} \sum_{i \in S_t} \tilde{g}_i$
14: **end for**

5.3.3 Conjugate Search Problem - Homomorphic Encryption

The Conjugate Search Problem (CSP) is a computational hardness assumption from non-abelian group theory. Given a non-commutative group G and elements $a, b \in G$, the CSP asks for an element $x \in G$ such that

$$b = x^{-1} a x, \tag{5.3.1}$$

assuming such an x exists. The difficulty of solving this problem underpins the security of several cryptographic schemes.

In the context of homomorphic encryption, CSP has been investigated as a foundation for constructing encryption schemes that allow algebraic operations on ciphertexts while relying on group-theoretic hardness rather than classical number-theoretic problems. By leverag-

ing the non-commutativity of the underlying group, CSP-based homomorphic encryption enables secure computation over encrypted data and provides resistance against certain algebraic and quantum attacks, making it relevant for privacy-preserving computation.

Algorithm 5.2 shows the CSP Homomorphic encryption used with FL. T denotes the total number of federated learning rounds, and S_t represents the randomly selected subset of K clients participating in round t. Each client $i \in S_t$ computes its local gradient $g_i = \nabla\mathcal{L}(w_t; D_i)$ on its private dataset D_i, and then encrypts it using a Conjugate Search Problem-based homomorphic encryption scheme: $\tilde{g}_i = E_{CSP}(g_i)$. The encrypted gradients are sent to the server, which performs a homomorphic aggregation $\tilde{G}_t = \text{Aggregate}(\tilde{g}_i, i \in S_t)$ without decrypting individual updates. The aggregated gradient is decrypted by the server or a trusted client $G_t = D_{CSP}(\tilde{G}_t)$ and used to update the global model $w_{t+1} = w_t - \eta \cdot G_t$. This process repeats for all rounds until convergence, ensuring that individual client data remains confidential while allowing collaborative model training.

Algorithm 5.2 CSP-based Homomorphic Encryption in Federated Learning

Require: Number of communication rounds T, initial global model w_0, total clients N, participating clients per round K, learning rate η, CSP public parameters (G, a, b), encryption function $E_{\text{CSP}}(\cdot)$, decryption function $D_{\text{CSP}}(\cdot)$

Ensure: Final trained global model w_T

for $t = 0$ to $T - 1$ **do**
  // **Server-side Operations**
  Randomly select a subset of clients $S_t \subseteq \{1, \ldots, N\}$ such that $|S_t| = K$
  Broadcast global model w_t and CSP public parameters (G, a, b) to all $i \in S_t$
  // **Client-side Operations (Executed in Parallel)**
  for each client $i \in S_t$ **do**
    Compute local gradient: $g_i \leftarrow \nabla\mathcal{L}(w_t; D_i)$
    Encrypt local gradient: $\tilde{g}_i \leftarrow E_{\text{CSP}}(g_i)$
    Send encrypted gradient $\tilde{g}_i$ to the server
  end for
  // **Server-side Homomorphic Aggregation**
  Compute encrypted aggregation:

$$\tilde{G}_t \leftarrow \sum_{i \in S_t} \tilde{g}_i$$

  Decrypt aggregated gradient: $G_t \leftarrow D_{\text{CSP}}(\tilde{G}_t)$
  // **Global Model Update**
  Update global model: $w_{t+1} \leftarrow w_t - \eta \cdot G_t$
end for

5.3.4 Fed-DHVE

Combining the paradigm of differential privacy, homomorphic encryption, and workload-aware client selection (discussed in Chapter 3, section 3.3.1.2 and algorithm 3.2), we propose a Federated Learning Framework with Differential Privacy over Homomorphic Vector Encryption (Fed-DHVE), shown in Algorithm 5.3.

T denotes the total number of federated rounds, and S_t is the subset of K clients selected for round t. Each client $i \in S_t$ computes its local gradient $g_i = \nabla\mathcal{L}(w_t; D_i)$, clips it to a maximum norm C, and adds Gaussian noise for differential privacy: $\hat{g}_i = g_i + \mathcal{N}(0, \sigma^2 C^2 I)$. The noisy gradient is then encrypted using CSP-based homomorphic encryption $\tilde{g}_i = E_{CSP}(\hat{g}_i)$ before being sent to the server. The server performs homomorphic aggregation on the encrypted gradients $\tilde{G}_t = \text{Aggregate}(\tilde{g}_i, i \in S_t)$, and the aggregated result is decrypted $G_t = D_{CSP}(\tilde{G}_t)$ to update the global model $w_{t+1} = w_t - \eta \cdot G_t$. This hybrid approach ensures that individual gradients remain encrypted during aggregation and are differentially obfuscated, providing strong privacy protection for federated learning applications.

Algorithm 5.3 Fed-DHVE

Require: Number of rounds T, initial global model w_0, total clients N, subset size K, learning rate η, gradient clipping norm C, DP noise scale σ, CSP public parameters (G, a, b), encryption E_{CSP}, decryption D_{CSP}

Ensure: Trained global model w_T

for $t = 1$ to T **do**
  Select a subset of clients based on their past performance, $S_t \subseteq \{1, 2, \ldots, N\}$ of size K **Algorithm 3.2**
  Server sends w_t and CSP public parameters (G, a, b) to clients in S_t
  for each client $i \in S_t$ **in parallel do**
    Compute local gradient: $g_i = \nabla\mathcal{L}(w_t; D_i)$
    Clip gradient: $g_i \leftarrow \dfrac{g_i}{\max\left(1, \dfrac{\|g_i\|_2}{C}\right)}$
    Add DP noise: $\hat{g}_i = g_i + \mathcal{N}(0, \sigma^2 C^2 I)$
    Encrypt noisy gradient: $\tilde{g}_i = E_{CSP}(\hat{g}_i)$
    Send encrypted noisy gradient $\tilde{g}_i$ to server
  end for
  Aggregate encrypted updates: $\tilde{G}_t = \sum_{i \in S_t} \tilde{g}_i$
  Decrypt aggregated gradient: $G_t = D_{CSP}(\tilde{G}_t)$
  Update global model: $w_{t+1} = w_t - \eta \cdot G_t$
end for

5.3.5 Privacy and Leakage Discussion

While the proposed framework evaluates model accuracy and computational overhead to demonstrate practicality, security and privacy properties require a different form of justification. In federated learning, privacy risks primarily arise from:

- Gradient leakage attacks (e.g., reconstruction of training data from gradients)
- Membership inference attacks
- Model inversion attacks
- Intermediate update interception during transmission

5.3.5.1. Protection Against Gradient Leakage

In the proposed framework, gradients are clipped to bound sensitivity:

$$g_i \leftarrow \frac{g_i}{\max\left(1, \frac{\|g_i\|_2}{C}\right)} \tag{5.3.2}$$

Gaussian noise is then added:

$$\tilde{g}_i = g_i + \mathcal{N}(0, \sigma^2 C^2 I) \tag{5.3.3}$$

This mechanism provides (ε, δ)-differential privacy guarantees under standard DP accounting assumptions. Clipping limits the maximum contribution of any single data point, while noise addition statistically masks sensitive gradient information, thereby reducing susceptibility to reconstruction attacks.

5.3.5.2. Encryption-Based Confidentiality

For the homomorphic encryption variants, local gradients are encrypted before transmission:

$$\tilde{g}_i = E_{\text{CSP}}(g_i) \tag{5.3.4}$$

Aggregation is performed in encrypted space:

$$\tilde{G}_t = \sum_{i \in S_t} \tilde{g}_i \tag{5.3.5}$$

Thus, raw gradients are never exposed to the central server or intermediate network adversaries. This mitigates eavesdropping attacks, honest-but-curious server threats, and man-in-the-middle interception risks.

5.3.5.3. Combined Privacy Effect

When Differential Privacy and Homomorphic Encryption are integrated, encryption protects gradients during transmission and aggregation, while DP protects against inference attacks even after decryption. This layered design reduces the risk of both communication-level leakage and post-aggregation inference attacks.

5.4 Experimental Results and Analysis for Fed-DHVE Simulation

In this section, we elaborate on our baseline models, experimental setup, the various datasets used, and the results obtained from the models discussed in the previous section, applied to those datasets.

5.4.1 Baseline Model for simulation of Fed-DHVE

We employed the MobileNetV2 architecture for all testing scenarios, utilizing the Stochastic Gradient Descent (SGD) optimizer in conjunction with the ReLU6 activation function. The initial learning rate was set to 0.1 and was programmatically reduced by a factor of 10 whenever the training loss plateaued, in order to enhance convergence and model stability.

MobileNetV2 is a lightweight deep convolutional neural network architecture optimized for efficient inference on mobile and embedded devices [284]. It improves upon its predecessor by introducing inverted residuals and linear bottleneck layers, which enable the model to maintain representational power while significantly reducing the number of parameters and computations. The architecture utilizes depthwise separable convolutions to minimize complexity, making it highly suitable for edge applications and federated learning environments where resources such as memory and computation are constrained.

5.4.2 Experimental setup and datasets used for Fed-DHVE Simulation

Our simulations are carried out on the same system described in section 4.4.2. 1,000 devices are simulated simultaneously for the training federation to utilise the maximum resources available on the host machine, thereby inducing a resource-constrained environment for every simulated instance. The dataset is divided into randomly sized chunks, with each simulated client receiving a chunk ranging from 10% to 100% of the data points. Each client trains the model for one epoch on their locally allotted dataset and sends the updated parameters to the simulated server at the end of each training round.

We tested our algorithm against the implementations discussed in [135, 277, 278], Fed-DLC, and standard FL over four openly available image datasets having dissimilar complexity, color space, and sample size (MNIST, Fashion-MNIST, EMNIST,FEMNIST) discussed in Chapter 3, Section 3.3.2.4.3. A brief description of the datasets and their composition is discussed in Table 3.5, and samples from the datasets are shown in Figure 3.14.

5.4.3 Results of Fed-DHVE Simulation

We benchmarked Fed-DHVE with the MobileNetV2 model over four distinct classification datasets and compared the results against the results of turbo-aggregate, verifynet, fed-MSQE, Fed-DLC, and Standard FL. Table 5.1 shows the results achieved by Fed-DHVE. It is observed that the average overhead on the client node is 2.76 seconds for a complete encryption-decryption cycle. The overhead is expected to increase with larger models as the number of parameters increases. The accuracy and precision achieved are similar to those of native FL implementation, as highlighted by Table 5.2.

Table 5.1: Performance of Fed-DHVE over Various Datasets

Metric/ Dataset	**MNIST**	**Fashion-MNIST**	**FE-MNIST**	**Cifar-10**
Accuracy	0.9533	0.8361	0.9970	0.9630
Precision	0.8330	0.8351	0.9970	0.9632
Avg Encryption Overhead (Client)			2.04 Seconds	
Avg Aggregation Time (Server Side)			0.16 Seconds	
Avg Decryption Time (Client)			0.72 Seconds	

Table 5.2 shows the performance comparison of Fed-DHVE with the state of the art security techniques and standard FL.

Table 5.2: Performance Comparison of Fed-DHVE against state-of-the-art techniques

Model / Metric	**MNIST**	**Fashion-MNIST**	**FEMNIST**	**CIFAR-10**	**Security**	**Avg Encryption Time**	**Avg Decryption Time**
Standard FL	0.9843	0.9537	0.9761	0.9552	Basic FL	–	–
Fed-MSQE (5-bit) [135]	0.9537	0.9016	0.9750	0.8890	Encryption	1.3 s	–
VerifyNet [278]	0.9791	0.9580	0.9785	0.9476	Authentication	0.8 s	–
Turbo-Aggregate [277]	0.9823	0.9525	0.9770	0.9554	Basic FL	–	–
Fed-DLC (Proposed in Chapter 4)	0.9850	0.9526	0.9764	0.9554	Encryption	1.3 s	1.3 s
Fed-DHVE	0.9783	0.9397	0.9681	0.9492	Encryption	2.04 s	0.72 s

It is observed that Fed-DHVE performs on par with standard FL in terms of accuracy, having a marginal loss induced due to differential noise. It is also observed that the encryption and decryption overheads induced are only slightly above those of state-of-the-art techniques, while providing a dual layer of security and trust for highly sensitive applications such as DSA.

5.5 Summary of the Chapter

In this chapter, we have proposed Fed-DHVE, a secure FL framework that combines differential privacy, homomorphic encryption, and workload-aware optimization to provide high level security and trust in applications dealing with sensitive data having sufficient resources in the training network to counter the overhead induced. Through extensive experimentation with the MobileNetV2 model over MNIST, Fashion-MNIST, E-MNIST, and Cifar-10 datasets, we observe that Fed-DHVE achieves identical performance as that of native or standard FL and Turbo-Aggregate while providing an increased level of security with slightly increased computational overhead compared to similar state-of-the-art methods.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

Artificial intelligence has made significant advancements in recent years, affecting nearly every field of human endeavour. This advancement has come with an equal threat to privacy and the security of sensitive data, often used to train the underlying model. Federated Learning (FL) was developed to improve data security in data-sensitive applications. There are many types of FL, each having its own advantages and shortcomings.

This first chapter of the thesis delves into the many types of FL, their utility, and their shortcomings, and examines the application of FL technology to improve security, privacy, and performance in DSA-based resource-constrained networks, including IoMT, IoT, AdHoc, and Edge.

In chapter Two, we present a detailed study of the tools, datasets, and implementation approaches available for FL to guide and motivate budding researchers in the field. We also propose a comprehensive taxonomy of FL based on its basics, types, supporting technologies, DSA prospects and adoptable techniques from other domains.

Chapter Three proposed multiple frameworks based on quantization and workload-aware FL aimed at bandwidth optimization, efficient resource distribution, and most importantly, data privacy at minimal computational cost, recovered quantization loss, and maximum stability. The frameworks are tested across a variety of datasets and prove to be stable and efficient. The results highlight a 1-5% improvement in accuracy (**99.04** vs 98.80 / **74.60** vs 72.06) and a 1-50% decline in straggler node percentages (**3.0** vs 3.1 / **15.0** vs 45.0) depending on the level of quantization deployed compared to the state of the art.

In Chapter Four, we propose Fed-DLC, an FL framework that employs a lightweight crypto-

graphic algorithm with a key tied to both the client node and the training network, enabling dynamic updates in each training round. The proposed framework incurs a negligible overhead of 1.3 to 14.2 seconds over 28.5 and 178.5 seconds of native FL rounds, respectively, during the encryption-decryption cycle. Overhead can increase with model size. The framework does not interfere with the underlying FL algorithm, thereby achieving results similar to those of the underlying FL algorithm.

Finally, in Chapter Five, we propose Fed-DHVE, a framework that combines differential privacy and homomorphic encryption to provide a double-layer security scheme for DSA networks having sufficient resources. The results show an increase of 0.7 to 1.96 seconds in encryption-decryption time compared to state-of-the-art, along with a substantial increase in resistance to various attacks, as the encryption and noise together make inference potential negligible.

Limitations and Trade-offs Despite the promising results achieved by the proposed frameworks, certain limitations must be acknowledged to provide a balanced perspective. First, integrating additional security and privacy mechanisms inevitably introduces computational and communication overhead. Although the proposed lightweight cipher and hybrid DP–HE schemes were designed to minimize this cost, a trade-off between security strength and system efficiency still exists, particularly in large-scale deployments. Second, while workload-aware quantization effectively reduces straggler nodes and bandwidth requirements, aggressive quantization may lead to information loss and potential degradation in model accuracy for highly complex datasets. Third, the scalability of homomorphic encryption–based aggregation remains constrained by computational complexity and key management challenges when the number of participating clients becomes extremely large. Finally, the evaluation has been conducted in simulated federated environments; real-world deployments may introduce additional factors such as network instability, device churn, and heterogeneous privacy policies. Addressing these challenges presents promising directions for future research toward building more scalable, adaptive, and practically deployable secure federated learning systems.

6.2 Future Directions

The future work may include:

- **Formal Security Validation:** Develop rigorous, provable security proofs and employ formal verification tools to mathematically validate robustness against adaptive, inversion-based, and collusion attacks.

- **Post-Quantum Resilience:** Integrate and evaluate post-quantum cryptographic primitives to ensure long-term security against quantum-enabled adversaries.

- **Scalable Real-World Deployment:** Implement and benchmark the framework in large-scale, heterogeneous, and resource-constrained environments to assess scalability, latency, communication overhead, and energy efficiency.

- **Robustness Under Adversity:** Conduct extensive experimentation under highly non-IID data distributions, adversarial settings, and realistic threat models to validate stability and generalization.

- **Hardware and Edge Integration:** Explore hardware-level and edge-device implementations to identify practical deployment constraints and system-level trade-offs.

- **Security-Oriented Evaluation:** Conduct attack simulations and leakage analysis, including eavesdropping, model interception, and known/chosen-plaintext scenarios, to quantitatively assess the robustness of the proposed lightweight cryptographic federated learning framework.

List of Publications

Journals

[1] **Manu Narula**, Jasraj Meena, and Dinesh Kumar Vishwakarma, "A comprehensive review on Federated Learning for Data-Sensitive Application: Open issues & challenges," *Engineering Applications of Artificial Intelligence*, Elsevier, Volume 133, Part B, pp. 108128, 2024, DOI:https://doi.org/10.1016/j.engappai.2024.108128. **(SCIE, IF=8.0)**.

[2] **Manu Narula**, Jasraj Meena, and Dinesh Kumar Vishwakarma, "Federated workload-aware quantized framework for secure learning in data-sensitive applications," *Future Generation Computer Systems*, Elsevier, vol. 168, pp.107772, 2025, DOI: https://doi.org/10.1016/j.future.2025.107772. **(SCIE, IF=6.1)**.

[3] **Manu Narula**, Jasraj Meena, and Dinesh Kumar Vishwakarma, "Secure Federated Learning Framework with Dynamic Lightweight Cipher (Fed-DLC) for Resource-Constrained Networks in Data Sensitive Applications," -*Communicated*

[4] **Manu Narula**, Jasraj Meena, and Dinesh Kumar Vishwakarma, "Federated Framework with Differential Privacy over Homomorphic Vector Encryption for Data-Sensitive Applications," -*Communicated*

Conferences

[1] **Manu Narula**, Jasraj Meena and Dinesh Kumar Vishwakarma, "A Comprehensive Study of Federated Learning Tools and Frameworks for Data-Sensitive Applications",*Puneet Kumar Gupta (eds), Computational Models for Intelligence and Automation*, SCRS, India, 2025, pp. 127-135. DOI: https://doi.org/10.56155/978-81-975670-1-8-12

[2] **Manu Narula**, Jasraj Meena, and Dinesh Kumar Vishwakarma, "Dynamic Resource-Aware Federated Framework for Secure and Sustainable Learning in Data Sensitive Applications," 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), Pune, India, 2024, pp. 1-8, 2025,https://doi.org/10.1109/I2CT61223.2024.10543390

Bibliography

[1] I. Ahmed, G. Jeon, and F. Piccialli, "From artificial intelligence to explainable artificial intelligence in industry 4.0: A survey on what, how, and where," *IEEE Transactions on Industrial Informatics*, vol. 18, pp. 5031–5042, Aug. 2022. https://doi.org/10.1109/TII.2022.3146552.

[2] H. Lin, K. Kaur, X. Wang, G. Kaddoum, J. Hu, and M. M. Hassan, "Privacy-aware access control in iot-enabled healthcare: A federated deep learning approach," *IEEE Internet of Things Journal*, 2021. https://doi.org/10.1109/JIOT.2021.3112686.

[3] T. Ngo *et al.*, "Federated deep learning for the diagnosis of cerebellar ataxia: Privacy preservation and auto-crafted feature extractor," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 803–811, 2022. https://doi.org/10.1109/TNSRE.2022.3161272.

[4] D. Myalil, M. A. Rajan, M. Apte, and S. Lodha, "Robust collaborative fraudulent transaction detection using federated learning," in *Proceedings - 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 373–378, IEEE, 2021. https://doi.org/10.1109/ICMLA52953.2021.00064.

[5] B. Lv, P. Cheng, C. Zhang, H. Ye, X. Meng, and X. Wang, "Research on modeling of e-banking fraud account identification based on federated learning," in *2021 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pp. 611–618, IEEE, Oct. 2021. https://doi.org/10.1109/DASC-PICom-CBDCom-CyberSciTech52372.2021.00105.

[6] W. Li *et al.*, "Privacy-preserving federated brain tumour segmentation," 2019. http://arxiv.org/abs/1910.00962.

[7] S. Hussein, P. Kandel, C. W. Bolan, M. B. Wallace, and U. Bagci, "Lung and pancreatic tumor characterization in the deep learning era: Novel supervised and unsupervised learning approaches," *IEEE Transactions on Medical Imaging*, vol. 38, pp. 1777–1787, Aug. 2019. https://doi.org/10.1109/TMI.2019.2894349.

[8] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Feb. 2017.

[9] A. Hard *et al.*, "Federated learning for mobile keyboard prediction," 2018. http://arxiv.org/abs/1811.03604.

[10] T. Yang *et al.*, "Applied federated learning: Improving google keyboard query suggestions," 2018. http://arxiv.org/abs/1812.02903.

[11] S. Hu, X. Chen, W. Ni, E. Hossain, and X. Wang, "Distributed machine learning for wireless communication networks: Techniques, architectures, and applications," *IEEE Communications Surveys and Tutorials*, vol. 23, pp. 1458–1493, July 2021. https://doi.org/10.1109/COMST.2021.3086014.

[12] N. Bouacida and P. Mohapatra, "Vulnerabilities in federated learning," *IEEE Access*, vol. 9, pp. 63229–63249, 2021. https://doi.org/10.1109/ACCESS.2021.3075203.

[13] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, pp. 50–60, Aug. 2019. https://doi.org/10.1109/MSP.2020.2975749.

[14] K. Zhang, X. Song, C. Zhang, and S. Yu, "Challenges and future directions of secure federated learning: A survey," *Frontiers of Computer Science*, vol. 16, Dec. 2021. https://doi.org/10.1007/s11704-021-0598-z.

[15] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Communications Surveys and Tutorials*, vol. 23, pp. 1759–1799, July 2021. https://doi.org/10.1109/COMST.2021.3090430.

[16] K. Wei *et al.*, "Vertical federated learning: Challenges, methodologies and experiments," 2022. http://arxiv.org/abs/2202.04309.

[17] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," 2020. http://arxiv.org/abs/2003.02133.

[18] M. Zhang, L. Qu, P. Singh, J. Kalpathy-Cramer, and D. L. Rubin, "Splitavg: A heterogeneity-aware federated deep learning method for medical imaging," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–9, June 2022. https://doi.org/10.1109/JBHI.2022.3185956.

[19] X. Wang, W. Chen, J. Xia, Z. Wen, R. Zhu, and T. Schreck, "Hetvis: A visual analysis approach for identifying data heterogeneity in horizontal federated learning," 2022. http://arxiv.org/abs/2208.07491.

[20] J. Aquilante and A. Jicha, "Heterogeneity aware federated learning major qualifying project," tech. rep., Worcester Polytechnic Institute, 2021.

[21] C. Yang *et al.*, "Characterizing impacts of heterogeneity in federated learning upon large-scale smartphone data," in *The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021*, pp. 935–946, Association for Computing Machinery, Inc, Apr. 2021.

[22] G. K. Gudur and S. K. Perepu, "Federated learning with heterogeneous labels and models for mobile activity monitoring," Dec. 2020. http://arxiv.org/abs/2012.02539.

[23] W. Zhu, J. Luo, and A. D. White, "Federated learning of molecular properties with graph neural networks in a heterogeneous setting," *Patterns*, vol. 3, Jun. 2022.

[24] W. Issa, N. Moustafa, B. Turnbull, N. Sohrabi, and Z. Tari, "Blockchain-based federated learning for securing internet of things: A comprehensive survey," *ACM Computing Surveys*, Sep. 2022.

[25] Z. Sun, J. Wan, L. Yin, Z. Cao, T. Luo, and B. Wang, "A blockchain-based audit approach for encrypted data in federated learning," *Digital Communications and Networks*, vol. 8, pp. 614–624, May 2022.

[26] Y. Qu, M. P. Uddin, C. Gan, Y. Xiang, L. Gao, and J. Yearwood, "Blockchain-enabled federated learning: A survey," *ACM Computing Surveys*, vol. 55, Mar. 2022. https://doi.org/10.1145/3524104.

[27] Y. He, K. Huang, G. Zhang, F. R. Yu, J. Chen, and J. Li, "Bift: A blockchain-based federated learning system for connected and autonomous vehicles," *IEEE Internet of Things Journal*, vol. 9, pp. 12311–12322, July 2021. https://doi.org/10.1109/JIOT.2021.3135342.

[28] B. Jia, X. Zhang, J. Liu, Y. Zhang, K. Huang, and Y. Liang, "Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in iiot," *IEEE Transactions on Industrial Informatics*, vol. 18, pp. 4049–4058, June 2022. https://doi.org/10.1109/TII.2021.3085960.

[29] C. Korkmaz, H. E. Kocas, A. Uysal, A. Masry, O. Ozkasap, and B. Akgun, "Chain fl: Decentralized federated machine learning via blockchain," in *2020 2nd International Conference on Blockchain Computing and Applications (BCCA)*, pp. 140–146, Institute of Electrical and Electronics Engineers Inc., Nov. 2020. https://doi.org/10.1109/BCCA50787.2020.9274451.

[30] J. Sun, Y. Wu, S. Wang, Y. Fu, and X. Chang, "Permissioned blockchain frame for secure federated learning," *IEEE Communications Letters*, vol. 26, pp. 13–17, Jan. 2022. https://doi.org/10.1109/LCOMM.2021.3121297.

[31] J. Li *et al.*, "Blockchain assisted decentralized federated learning (blade-fl): Performance analysis and resource allocation," Jan. 2021. http://arxiv.org/abs/2101.06905.

[32] O. Samuel *et al.*, "Iomt: A covid-19 healthcare system driven by federated learning and blockchain," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–1, 2022. https://doi.org/10.1109/JBHI.2022.3143576.

[33] P. Zhang, H. Sun, J. Situ, C. Jiang, and D. Xie, "Federated transfer learning for iiot devices with low computing power based on blockchain and edge computing," *IEEE Access*, vol. 9, pp. 98630–98638, 2021. https://doi.org/10.1109/ACCESS.2021.3095078.

[34] Z. Xue *et al.*, "A resource-constrained and privacy-preserving edge-computing-enabled clinical decision system: A federated reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 8, pp. 9122–9138, June 2021. https://doi.org/10.1109/JIOT.2021.3057653.

[35] 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, pp. 156–165, Sept. 2019. https://doi.org/10.1109/MNET.2019.1800286.

[36] W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys and Tutorials*, vol. 22, pp. 2031–2063, July 2020. https://doi.org/10.1109/COMST.2020.2986024.

[37] M. N. H. Nguyen, N. H. Tran, Y. K. Tun, Z. Han, and C. S. Hong, "Toward multiple federated learning services resource sharing in mobile edge networks," *IEEE Transactions on Mobile Computing*, pp. 1–1, June 2021. https://doi.org/10.1109/TMC.2021.3085979.

[38] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, and S. Yu, "Poisongan: Generative poisoning attacks against federated learning in edge computing systems," *IEEE Internet of Things Journal*, vol. 8, pp. 3310–3322, Mar. 2021. https://doi.org/10.1109/JIOT.2020.3023126.

[39] Y. Li, X. Qin, H. Chen, K. Han, and P. Zhang, "Energy-aware edge association for cluster-based personalized federated learning," *IEEE Transactions on Vehicular Technology*, June 2022. https://doi.org/10.1109/TVT.2022.3161503.

[40] X. Mo, J. Xu, X. P. Mo, and J. Xu, "Energy-efficient federated edge learning with joint communication and computation design," *Journal of Communications and Information Networks*, vol. 6, pp. 110–124, June 2021. https://doi.org/10.23919/JCIN.2021.9475121.

[41] J. Á. Morell and E. Alba, "Dynamic and adaptive fault-tolerant asynchronous federated learning using volunteer edge devices," *Future Generation Computer Systems*, vol. 133, pp. 53–67, Aug. 2022. https://doi.org/10.1016/j.future.2022.02.024.

[42] 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*, Aug. 2020. https://doi.org/10.1109/TMC.2020.3045266.

[43] N. Harth, H.-J. Voegel, K. Kolomvatsos, and C. Anagnostopoulos, "Local learning at the network edge for efficient & secure real-time predictive analytics," Sept. 2021. http://arxiv.org/abs/2109.12375.

[44] L. Zhao, J. Li, Q. Li, and F. Li, "A federated learning framework for detecting false data injection attacks in solar farms," *IEEE Transactions on Power Electronics*, vol. 37, pp. 2496–2501, Mar. 2022. https://doi.org/10.1109/TPEL.2021.3114671.

[45] B. Xu, W. Xia, W. Wen, P. Liu, H. Zhao, and H. Zhu, "Adaptive hierarchical federated learning over wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 71, pp. 2070–2083, Feb. 2022. https://doi.org/10.1109/TVT.2021.3135541.

[46] H. Xing, Z. Xiao, R. Qu, Z. Zhu, and B. Zhao, "An efficient federated distillation learning system for multi-task time series classification," *IEEE Transactions on Instrumentation and Measurement*, 2022. https://doi.org/10.1109/TIM.2022.3201203.

[47] A. Renda *et al.*, "Federated learning of explainable ai models in 6g systems: Towards secure and automated vehicle networking," *Information*, vol. 13, Aug. 2022. https://doi.org/10.3390/info13080395.

[48] J. Luis *et al.*, "Fed-xai: Federated learning of explainable artificial intelligence models ," 2022. http://ceur-ws.org.

[49] Z. Tian, L. Cui, J. Liang, and S. Yu, "A comprehensive survey on poisoning attacks and countermeasures in machine learning," *ACM Computing Surveys*, vol. 55, p. 171, July 2022. https://doi.org/10.1145/3551636.

[50] K. Hsieh, "Machine learning systems for highly-distributed and rapidly-growing data." arXiv preprint, Oct. 2019. https://arxiv.org/abs/1910.08663.

[51] K. S. Bohnsack, M. Kaden, J. Abel, and T. Villmann, "Alignment-free sequence comparison: A systematic survey from a machine learning perspective," *IEEE/ACM*

Transactions on Computational Biology and Bioinformatics, 2022. https://doi.org/10.1109/TCBB.2022.3140873.

[52] R. Tao *et al.*, "Magnetocardiography-based ischemic heart disease detection and localization using machine learning methods," *IEEE Transactions on Biomedical Engineering*, vol. 66, pp. 1658–1667, June 2019. https://doi.org/10.1109/TBME.2018.2877649.

[53] M. P. Hosseini, A. Hosseini, and K. Ahi, "A review on machine learning for eeg signal processing in bioengineering," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 204–218, 2021. https://doi.org/10.1109/RBME.2020.2969915.

[54] M. Dol and A. Geetha, "A learning transition from machine learning to deep learning: A survey," in *Proceedings of the 2021 International Conference on Emerging Techniques in Computational Intelligence (ICETCI)*, pp. 89–94, IEEE, Aug. 2021. https://doi.org/10.1109/ICETCI51973.2021.9574066.

[55] T. T. Nguyen and V. J. Reddi, "Deep reinforcement learning for cyber security," *IEEE Transactions on Neural Networks and Learning Systems*, 2021. https://doi.org/10.1109/TNNLS.2021.3121870.

[56] P. Ren *et al.*, "A survey of deep active learning," *ACM Computing Surveys*, vol. 54, Dec. 2022. https://doi.org/10.1145/3472291.

[57] T. Zhu, K. Li, P. Herrero, and P. Georgiou, "Deep learning for diabetes: A systematic review," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, pp. 2744–2757, July 2021. https://doi.org/10.1109/JBHI.2020.3040225.

[58] J. Zhang, T. Liu, and D. Tao, "An optimal transport analysis on generalization in deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2021. https://doi.org/10.1109/TNNLS.2021.3109942.

[59] P. Vepakomma, T. Swedish, R. Raskar, O. Gupta, and A. Dubey, "No peek: A survey of private distributed deep learning." arXiv preprint, Dec. 2018. https://arxiv.org/abs/1812.03288.

[60] H. Song, M. Kim, D. Park, Y. Shin, and J. G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022. https://doi.org/10.1109/TNNLS.2022.3152527.

[61] J. Chen, S. Chen, X. Chen, Y. Yang, and Y. Rao, "Statenet: Deep state learning for robust feature matching of remote sensing images," *IEEE Transactions on Neural Networks and Learning Systems*, 2021. https://doi.org/10.1109/TNNLS.2021.3120768.

[62] C. H. Yeh *et al.*, "Lightweight deep neural network for joint learning of underwater object detection and color conversion," *IEEE Transactions on Neural Networks and Learning Systems*, 2021. https://doi.org/10.1109/TNNLS.2021.3072414.

[63] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Machine Learning*, vol. 81, pp. 121–148, Nov. 2010. https://doi.org/10.1007/s10994-010-5188-5.

[64] T. Ben-Nun and T. Hoefler, "Demystifying parallel and distributed deep learning: An in-depth concurrency analysis," *ACM Computing Surveys*, vol. 52, Aug. 2019. https://doi.org/10.1145/3320060.

[65] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data." arXiv preprint, June 2018. https://doi.org/10.48550/arXiv.1806.00582.

[66] M. F. Criado, F. E. Casado, R. Iglesias, C. v. Regueiro, and S. Barro, "Non-iid data and continual learning processes in federated learning: A long road ahead," *Information Fusion*, vol. 88, pp. 263–280, Dec. 2022. https://doi.org/10.1016/j.inffus.2022.07.024.

[67] S. Chakraborty, V. Balasubramanian, and S. Panchanathan, "Adaptive batch mode active learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, pp. 1747–1760, Aug. 2015. https://doi.org/10.1109/TNNLS.2014.2356470.

[68] A. Shahraki, M. Abbasi, A. Taherkordi, and A. D. Jurcut, "Active learning for network traffic classification: A technical study," *IEEE Transactions on Cogni-*

tive Communications and Networking, vol. 8, pp. 422–439, Mar. 2022. https://doi.org/10.1109/TCCN.2021.3119062.

[69] J. Eickholt, M. R. Johnson, and P. Seeling, "Practical active learning stations to transform existing learning environments into flexible, active learning classrooms," *IEEE Transactions on Education*, vol. 64, pp. 95–102, May 2021. https://doi.org/10.1109/TE.2020.3009919.

[70] H. Yu, X. Yang, S. Zheng, and C. Sun, "Active learning from imbalanced data: A solution of online weighted extreme learning machine," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, pp. 1088–1103, Apr. 2019. https://doi.org/10.1109/TNNLS.2018.2855446.

[71] L. Ahmed, K. Ahmad, N. Said, B. Qolomany, J. Qadir, and A. Al-Fuqaha, "Active learning based federated learning for waste and natural disaster image classification," *IEEE Access*, vol. 8, pp. 208518–208531, 2020. https://doi.org/10.1109/ACCESS.2020.3038676.

[72] Z. Zhao, Z. Zeng, K. Xu, C. Chen, and C. Guan, "Dsal: Deeply supervised active learning from strong and weak labelers for biomedical image segmentation," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, pp. 3744–3751, Oct. 2021. https://doi.org/10.1109/JBHI.2021.3052320.

[73] X. Y. Zhang, S. Wang, and X. Yun, "Bidirectional active learning: A two-way exploration into unlabeled and labeled data set," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, pp. 3034–3044, Dec. 2015. https://doi.org/10.1109/TNNLS.2015.2401595.

[74] B. Gu, Z. Zhai, C. Deng, and H. Huang, "Efficient active learning by querying discriminative and representative samples and fully exploiting unlabeled data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 4111–4122, Sept. 2021. https://doi.org/10.1109/TNNLS.2020.3016928.

[75] W. Huang, T. Li, D. Wang, S. Du, J. Zhang, and T. Huang, "Fairness and accuracy in horizontal federated learning," *Information Sciences*, vol. 589, pp. 170–185, Apr. 2022. https://doi.org/10.1016/j.ins.2021.12.102.

[76] D. Liu, L. Bai, T. Yu, and A. Zhang, "Towards method of horizontal federated learning: A survey," in *Proceedings of the 2022 8th International Conference on Big Data and Information Analytics (BigDIA)*, pp. 259–266, IEEE, Aug. 2022. https://doi.org/10.1109/BigDIA56350.2022.9874186.

[77] J. Zhao, X. Zhu, J. Wang, and J. Xiao, "Efficient client contribution evaluation for horizontal federated learning," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3060–3064, IEEE, 2021. https://doi.org/10.1109/ICASSP39728.2021.9413377.

[78] J. Wang, L. Zhang, A. Li, X. You, and H. Cheng, "Efficient participant contribution evaluation for horizontal and vertical federated learning," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pp. 911–923, IEEE, 2022. https://doi.org/10.1109/ICDE53745.2022.00073.

[79] B. Gu, A. Xu, Z. Huo, C. Deng, and H. Huang, "Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2021. https://doi.org/10.1109/TNNLS.2021.3072238.

[80] S. Feng, "Vertical federated learning-based feature selection with non-overlapping sample utilization," *Expert Systems with Applications*, vol. 208, Dec. 2022. https://doi.org/10.1016/j.eswa.2022.118097.

[81] H. Gupta, D. Patel, A. Makade, K. Gupta, O. P. Vyas, and A. Puliafito, "Risk prediction in the life insurance industry using federated learning approach," in *Proceedings of the 2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON)*, pp. 948–953, IEEE, June 2022. https://doi.org/10.1109/MELECON53508.2022.9842869.

[82] K. Nomura, Y. Shiraishi, M. Mohri, and M. Morii, "Secure association rule mining on vertically partitioned data using private-set intersection," *IEEE Access*, vol. 8, pp. 144458–144467, 2020. https://doi.org/10.1109/ACCESS.2020.3014330.

[83] A. Abadi, S. Terzis, R. Metere, and C. Dong, "Efficient delegated private set intersection on outsourced private datasets," *IEEE Transactions on Dependable and Secure*

Computing, vol. 16, pp. 608–624, July 2019. https://doi.org/10.1109/TDSC.2017.2708710.

[84] Z. Wang, K. Banawan, and S. Ulukus, "Multi-party private set intersection: An information-theoretic approach," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, pp. 366–379, Feb. 2021. https://doi.org/10.1109/JSAIT.2021.3057597.

[85] Z. Wang, K. Banawan, and S. Ulukus, "Private set intersection using multi-message symmetric private information retrieval," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, pp. 1035–1040, IEEE, June 2020. https://doi.org/10.1109/ISIT44484.2020.9174221.

[86] A. Tajima, H. Sato, and H. Yamana, "Outsourced private set intersection cardinality with fully homomorphic encryption," in *Proceedings of the 2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*, pp. 1–8, 2018. https://doi.org/10.1109/ICMCS.2018.8525881.

[87] S. Mishima, K. Nakasho, Y. Takano, and A. Miyaji, "A practical parallel computation in a scalable multiparty private set intersection," in *Proceedings of the 2021 9th International Symposium on Computing and Networking Workshops (CANDARW)*, pp. 332–338, IEEE, 2021. https://doi.org/10.1109/CANDARW53999.2021.00063.

[88] L. Lu and N. Ding, "Multi-party private set intersection in vertical federated learning," in *Proceedings of the 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, 2020. https://doi.org/10.1109/TrustCom50675.2020.00098.

[89] M. Long, J. Wang, Y. Cao, J. Sun, and P. S. Yu, "Deep learning of transferable representation for scalable domain adaptation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, pp. 2027–2040, Aug. 2016. https://doi.org/10.1109/TKDE.2016.2554549.

[90] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*

(CVPR), pp. 1855–1862, IEEE, June 2010. https://doi.org/10.1109/CVPR.2010.5539857.

[91] B. Cheng, M. Liu, H.-I. Suk, D. Shen, and D. Zhang, "Multimodal manifold-regularized transfer learning for mci conversion prediction," *Brain Imaging and Behavior*, vol. 9, pp. 913–926, Dec. 2015. https://doi.org/10.1007/s11682-015-9356-x.

[92] F. Zhuang, P. Luo, Q. He, and Z. Shi, "Inductive transfer learning for unlabeled target-domain via hybrid regularization," *Chinese Science Bulletin*, vol. 54, pp. 2470–2478, July 2009. https://doi.org/10.1007/s11434-009-0171-x.

[93] Y. E. Haj, A. H. El-Hag, and R. A. Ghunem, "Application of deep-learning via transfer learning to evaluate silicone rubber material surface erosion," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 28, pp. 1465–1467, Aug. 2021. https://doi.org/10.1109/TDEI.2021.009617.

[94] M. Srivastava and J. Meena, "Plant leaf disease detection and classification using modified transfer learning models," *Multimedia Tools and Applications*, 2023. https://doi.org/10.1007/s11042-023-16929-y.

[95] S. Feng, B. Li, H. Yu, Y. Liu, and Q. Yang, "Semi-supervised federated heterogeneous transfer learning," *Knowledge-Based Systems*, vol. 252, Sept. 2022. https://doi.org/10.1016/j.knosys.2022.109384.

[96] 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, pp. 83–93, July 2020. https://doi.org/10.1109/MIS.2020.2988604.

[97] H. Yang, H. He, W. Zhang, and X. Cao, "Fedsteg: A federated transfer learning framework for secure image steganalysis," *IEEE Transactions on Network Science and Engineering*, vol. 8, pp. 1084–1094, Apr. 2021. https://doi.org/10.1109/TNSE.2020.2996612.

[98] W. Zhang, Z. Wang, and X. Li, "Blockchain-based decentralized federated transfer learning methodology for collaborative machinery fault diagnosis," *Reliability Engineering & System Safety*, vol. 229, p. 108885, Jan. 2023. https://doi.org/10.1016/j.ress.2022.108885.

[99] W. Zhang and X. Li, "Federated transfer learning for intelligent fault diagnostics using deep adversarial networks with data privacy," *IEEE/ASME Transactions on Mechatronics*, vol. 27, pp. 430–439, Feb. 2022. https://doi.org/10.1109/TMECH.2021.3065522.

[100] D. Liu, K. Fox, G. Weber, and T. Miller, "Confederated learning in healthcare: training machine learning models using disconnected data separated by individual, data type and identity for large-scale health system intelligence," *Journal of Biomedical Informatics*, vol. 134, p. 104151, 2022. https://doi.org/10.1016/j.jbi.2022.104151.

[101] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–17, Mar. 2022. https://doi.org/10.1109/TNNLS.2022.3160699.

[102] S. Baghersalimi, T. Teijeiro, D. Atienza, and A. Aminifar, "Personalized real-time federated learning for epileptic seizure detection," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, pp. 898–909, Feb. 2022. https://doi.org/10.1109/JBHI.2021.3096127.

[103] C. Wu, F. Wu, L. Lyu, T. Qi, Y. Huang, and X. Xie, "A federated graph neural network framework for privacy-preserving personalization," *Nature Communications*, vol. 13, Dec. 2022. https://doi.org/10.1038/s41467-022-30714-9.

[104] C. Huang, G. Xu, S. Chen, W. Zhou, E. Y. K. Ng, and V. H. C. de Albuquerque, "An improved federated learning approach enhanced internet of health things framework for private decentralized distributed data," *Information Sciences*, vol. 614, pp. 138–152, Oct. 2022. https://doi.org/10.1016/j.ins.2022.10.011.

[105] C. Brunetta, G. Tsaloli, B. Liang, G. Banegas, and A. Mitrokotsa, "Non-interactive, secure verifiable aggregation for decentralized, privacy-preserving learning," in *Lecture Notes in Computer Science*, vol. 13083, Springer, Cham, 2021. https://doi.org/10.1007/978-3-030-90567-5_26.

[106] S. T. Ahmed, V. V. Kumar, K. K. Singh, A. Singh, V. Muthukumaran, and D. Gupta, "6g enabled federated learning for secure iomt resource recommendation

and propagation analysis," *Computers and Electrical Engineering*, vol. 102, Sept. 2022. https://doi.org/10.1016/j.compeleceng.2022.108210.

[107] Z. Wang, B. Yan, and A. Dong, "Blockchain empowered federated learning for data sharing incentive mechanism," in *Procedia Computer Science*, pp. 348–353, Elsevier B.V., 2022. https://doi.org/10.1016/j.procs.2022.04.047.

[108] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, pp. 10700–10714, Dec. 2019. https://doi.org/10.1109/JIOT.2019.2940820.

[109] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, Nov. 2019. https://doi.org/10.1109/TDSC.2019.2952332.

[110] Y. Zhu, Z. Liu, P. Wang, and C. Du, "A dynamic incentive and reputation mechanism for energy-efficient federated learning in 6g," *Digital Communications and Networks*, Apr. 2022. https://doi.org/10.1016/j.dcan.2022.04.005.

[111] W. Lee, "Reward-based participant selection for improving federated reinforcement learning," *ICT Express*, 2022. https://doi.org/10.1016/j.icte.2022.08.008.

[112] G. de Carvalho Bertoli, L. A. P. Junior, O. Saotome, and A. L. dos Santos, "Generalizing intrusion detection for heterogeneous networks: A stacked-unsupervised federated learning approach," *Computers & Security*, vol. 127, p. 103106, Apr. 2023. https://doi.org/10.1016/j.cose.2023.103106.

[113] A. K. Singh and N. Goyal, "Android web security solution using cross-device federated learning," in *2022 14th International Conference on COMmunication Systems and NETworkS (COMSNETS)*, pp. 473–481, Institute of Electrical and Electronics Engineers Inc., 2022. https://doi.org/10.1109/COMSNETS53615.2022.9668449.

[114] H. Q. Le, L. X. Nguyen, S. B. Park, and C. S. Hong, "Layer-wise knowledge distillation for cross-device federated learning," in *International Conference on Infor-*

mation Networking (ICOIN), pp. 526–529, IEEE Computer Society, 2023. https://doi.org/10.1109/ICOIN56518.2023.10049011.

[115] P. M. S. Sánchez, A. H. Celdrán, N. Xie, G. Bovet, G. M. Pérez, and B. Stiller, "Federatedtrust: A solution for trustworthy federated learning," *Future Generation Computer Systems*, Feb. 2023. https://doi.org/10.1016/j.future.2023.10.013.

[116] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated meta-learning with fast convergence and efficient communication." arXiv preprint, Feb. 2018. [Online]. Available: http://arxiv.org/abs/1802.07876.

[117] M. N. Hossen, V. Panneerselvam, D. Koundal, K. Ahmed, F. M. Bui, and S. M. Ibrahim, "Federated machine learning for detection of skin diseases and enhancement of internet of medical things (iomt) security," *IEEE Journal of Biomedical and Health Informatics*, 2022. https://doi.org/10.1109/JBHI.2022.3149288.

[118] B. G. A. Tekgul, Y. Xia, S. Marchal, and N. Asokan, "Waffle: Watermarking in federated learning," in *Proceedings of the IEEE Symposium on Reliable Distributed Systems (SRDS)*, pp. 310–320, IEEE Computer Society, 2021. https://doi.org/10.1109/SRDS53918.2021.00038.

[119] F. A. KhoKhar, J. H. Shah, M. A. Khan, M. Sharif, U. Tariq, and S. Kadry, "A review on federated learning towards image processing," *Computers and Electrical Engineering*, vol. 99, Apr. 2022. https://doi.org/10.1016/j.compeleceng.2022.107818.

[120] J. D. S. Quilis, S. López-Huguet, P. Lozano, and I. Blanquer, "A federated cloud architecture for processing of cancer images on a distributed storage," *Future Generation Computer Systems*, vol. 139, pp. 38–52, Feb. 2023. https://doi.org/10.1016/j.future.2022.09.019.

[121] Z. Yan, J. Wicaksana, Z. Wang, X. Yang, and K. T. Cheng, "Variation-aware federated learning with multi-source decentralized medical image data," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, pp. 2615–2628, July 2021. https://doi.org/10.1109/JBHI.2020.3040015.

[122] P. Pinyoanuntapong, P. Janakaraj, R. Balakrishnan, M. Lee, C. Chen, and P. Wang, "Edgeml: Towards network-accelerated federated learning over wireless edge," *Computer Networks*, vol. 219, Dec. 2022. https://doi.org/10.1016/j.comnet.2022.109396.

[123] H. Du, M. Shen, R. Sun, J. Jia, L. Zhu, and Y. Zhai, "Malicious transaction identification in digital currency via federated graph deep learning," in *IEEE International Workshop on Intelligent Cloud Computing and Networking*, pp. 1–6, Institute of Electrical and Electronics Engineers (IEEE), June 2022. https://doi.org/10.1109/infocomwkshps54753.2022.9797992.

[124] D. C. Nguyen and et al., "Federated learning for smart healthcare: A survey," *ACM Computing Surveys*, vol. 55, pp. 1–37, Apr. 2023. https://doi.org/10.1145/3501296.

[125] 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, Mar. 2021. https://doi.org/10.1007/s41666-020-00082-4.

[126] M. G. Poirot, P. Vepakomma, K. Chang, J. Kalpathy-Cramer, R. Gupta, and R. Raskar, "Split learning for collaborative deep learning in healthcare." arXiv preprint, Dec. 2019. [Online]. Available: http://arxiv.org/abs/1912.12115.

[127] A. Rehman, S. Abbas, M. A. Khan, T. M. Ghazal, K. M. Adnan, and A. Mosavi, "A secure healthcare 5.0 system based on blockchain technology entangled with federated learning technique," *Computers in Biology and Medicine*, vol. 150, Nov. 2022. https://doi.org/10.1016/j.compbiomed.2022.106019.

[128] C. R. Hansen and et al., "Larynx cancer survival model developed through open-source federated learning," *Radiotherapy and Oncology*, vol. 176, pp. 179–186, Nov. 2022. https://doi.org/10.1016/j.radonc.2022.09.023.

[129] D. G. Nair, J. J. Nair, K. J. Reddy, and C. V. A. Narayana, "A privacy preserving diagnostic collaboration framework for facial paralysis using federated learning," *Engineering Applications of Artificial Intelligence*, vol. 116, Nov. 2022. https://doi.org/10.1016/j.engappai.2022.105476.

[130] U. C. Srivastava, A. Singh, and K. S. Kumar, "Intracranial hemorrhage detection using neural network based methods with federated learning." arXiv preprint, May 2020. [Online]. Available: http://arxiv.org/abs/2005.08644.

[131] I. Dayan and et al., "Federated learning for predicting clinical outcomes in patients with covid-19," *Nature Medicine*, vol. 27, pp. 1735–1743, Oct. 2021. https://doi.org/10.1038/s41591-021-01506-3.

[132] Q. Dou and et al., "Federated deep learning for detecting covid-19 lung abnormalities in ct: a privacy-preserving multinational validation study," *NPJ Digital Medicine*, vol. 4, Dec. 2021. https://doi.org/10.1038/s41746-021-00431-6.

[133] E. Lopez, A. Elmir, and S. Axelsson, "Paysim: A financial mobile money simulator for fraud detection," in *28th European Modeling and Simulation Symposium (EMSS 2016)*, (Larnaca, Cyprus), 2016. [Online]. Available: https://www.researchgate.net/publication/313138956.

[134] C. Dwork, "Differential privacy," in *Lecture Notes in Computer Science*, vol. 4052, pp. 1–12, Springer, Berlin, Heidelberg, 2006. https://doi.org/10.1007/11787006_1.

[135] Z. Xu, Y. Guo, C. Chakraborty, Q. Hua, S. Chen, and K. Yu, "A simple federated learning-based scheme for security enhancement over internet of medical things," *IEEE Journal of Biomedical and Health Informatics*, 2022. https://doi.org/10.1109/JBHI.2022.3187471.

[136] N. Xu, X. Chen, Y. Cao, and W. Zhang, "Hybrid post-training quantization for super-resolution neural network compression," *IEEE Signal Processing Letters*, 2023. https://doi.org/10.1109/LSP.2023.3264558.

[137] L. Lavaur, M. O. Pahl, Y. Busnel, and F. Autrel, "The evolution of federated learning-based intrusion detection and mitigation: a survey," *IEEE Transactions on Network and Service Management*, 2022. https://doi.org/10.1109/TNSM.2022.3177512.

[138] G. Sun, Y. Cong, J. Dong, Q. Wang, L. Lyu, and J. Liu, "Data poisoning attacks on federated machine learning," *IEEE Internet of Things Journal*, vol. 9, pp. 11365–11375, July 2022. https://doi.org/10.1109/JIOT.2021.3128646.

[139] J. Wen, B. Z. H. Zhao, M. Xue, A. Oprea, and H. Qian, "With great dispersion comes greater resilience: Efficient poisoning attacks and defenses for linear regression models," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3709–3723, 2021. https://doi.org/10.1109/TIFS.2021.3087332.

[140] Z. Ma, J. Ma, Y. Miao, Y. Li, and R. H. Deng, "Shieldfl: Mitigating model poisoning attacks in privacy-preserving federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1639–1654, 2022. https://doi.org/10.1109/TIFS.2022.3169918.

[141] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to byzantine-robust federated learning." arXiv preprint, Nov. 2019. [Online]. Available: http://arxiv.org/abs/1911.11815.

[142] X. Zhang, J. Wang, M. Shu, Y. Wang, M. Pan, and Z. Han, "Tpp: Trajectory privacy preservation against tensor voting based inference attacks," *IEEE Access*, vol. 6, pp. 77975–77985, 2018. https://doi.org/10.1109/ACCESS.2018.2884516.

[143] G. Zhang, B. Liu, T. Zhu, M. Ding, and W. Zhou, "Label-only membership inference attacks and defenses in semantic segmentation models," *IEEE Transactions on Dependable and Secure Computing*, 2022. https://doi.org/10.1109/TDSC.2022.3154029.

[144] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, pp. 557–570, Oct. 2002. https://doi.org/10.1142/S0218488502001648.

[145] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, p. 3, Mar. 2007. https://doi.org/10.1145/1217299.1217302.

[146] H. Lee, J. Kim, S. Ahn, R. Hussain, S. Cho, and J. Son, "Digestive neural networks: A novel defense strategy against inference attacks in federated learning," *Computers & Security*, vol. 109, Oct. 2021. https://doi.org/10.1016/j.cose.2021.102378.

[147] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable*

and Secure Computing, vol. 15, pp. 577–590, July 2018. https://doi.org/10.1109/TDSC.2016.2613521.

[148] J. Lin, R. Luley, and K. Xiong, "From adversarial examples to data poisoning instances: Utilizing an adversarial attack method to poison a transfer learning model," in *ICC 2022 - IEEE International Conference on Communications*, pp. 2351–2356, Institute of Electrical and Electronics Engineers (IEEE), Aug. 2022. https://doi.org/10.1109/ICC45855.2022.9839219.

[149] N. Rodríguez-Barroso, E. Martínez-Cámara, M. V. Luzón, and F. Herrera, "Dynamic defense against byzantine poisoning attacks in federated learning," *Future Generation Computer Systems*, vol. 133, pp. 1–9, July 2022. https://doi.org/10.1016/j.future.2022.03.003.

[150] A. Krall, D. Finke, and H. Yang, "Gradient mechanism to preserve differential privacy and deter against model inversion attacks in healthcare analytics," in *Proceedings of the 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2020. https://doi.org/10.1109/EMBC44109.2020.9176834.

[151] Y. Alufaisan, M. Kantarcioglu, and Y. Zhou, "Robust transparency against model inversion attacks," *IEEE Transactions on Dependable and Secure Computing*, 2020. https://doi.org/10.1109/TDSC.2020.3019508.

[152] H. Masuda, K. Kita, Y. Koizumi, J. Takemasa, and T. Hasegawa, "Model fragmentation, shuffle and aggregation to mitigate model inversion in federated learning," in *IEEE Workshop on Local and Metropolitan Area Networks (LANMAN)*, IEEE Computer Society, July 2021. https://doi.org/10.1109/LANMAN52105.2021.9478813.

[153] S. A. Khowaja, I. H. Lee, K. Dev, M. A. Jarwar, and N. M. F. Qureshi, "Get your foes fooled: Proximal gradient split learning for defense against model inversion attacks on iomt data," *IEEE Transactions on Network Science and Engineering*, pp. 1–10, 2022. https://doi.org/10.1109/TNSE.2022.3188575.

[154] R. Kerkouche, G. Ács, C. Castelluccia, and P. Genevès, "Privacy-preserving and bandwidth-efficient federated learning: An application to in-hospital mortality predic-

tion," in *Proceedings of the 2021 ACM Conference on Health, Inference, and Learning (ACM CHIL 2021)*, pp. 25–35, Association for Computing Machinery, Apr. 2021. https://doi.org/10.1145/3450439.3451859.

[155] L. Li *et al.*, "Federated learning with workload-aware client scheduling in heterogeneous systems," *Neural Networks*, vol. 154, pp. 560–573, Oct. 2022. https://doi.org/10.1016/j.neunet.2022.07.030.

[156] O. A. Wahab, A. Mourad, H. Otrok, and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *IEEE Communications Surveys and Tutorials*, vol. 23, pp. 1342–1397, Apr. 2021. https://doi.org/10.1109/COMST.2021.3058573.

[157] Q. Li *et al.*, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, 2021. https://doi.org/10.1109/TKDE.2021.3124599.

[158] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140699–140725, 2020. https://doi.org/10.1109/ACCESS.2020.3013541.

[159] K. Bonawitz, P. Kairouz, B. McMahan, and D. Ramage, "Federated learning and privacy," *Communications of the ACM*, vol. 65, pp. 90–97, Apr. 2022. https://doi.org/10.1145/3500240.

[160] N. Rodríguez-Barroso, D. Jiménez-López, M. V. Luzón, F. Herrera, and E. Martínez-Cámara, "Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges," *Information Fusion*, vol. 90, pp. 148–173, Feb. 2023. https://doi.org/10.1016/j.inffus.2022.09.011.

[161] S. Abdulrahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet of Things Journal*, vol. 8, pp. 5476–5497, Apr. 2021. https://doi.org/10.1109/JIOT.2020.3030072.

[162] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, S. Ermon, C. Sitawarin, A. D. Sarma, S. Seth, M. Singh, K. T. Suresh, P. Sylla, C.-S.

Yang, Z. Wang, B. Xu, and H. Yu, "Advances and open problems in federated learning." arXiv preprint arXiv:1912.04977, Dec. 2019. http://arxiv.org/abs/1912.04977.

[163] N. Truong, K. Sun, S. Wang, F. Guitton, and Y. Guo, "Privacy preservation in federated learning: An insightful survey from the gdpr perspective," *Computers Security*, vol. 110, p. 102402, Nov. 2021. https://doi.org/10.1016/j.cose.2021.102402.

[164] R. Gupta and T. Alam, "Survey on federated-learning approaches in distributed environment," *Wireless Personal Communications*, vol. 125, pp. 1631–1652, July 2022. https://doi.org/10.1007/s11277-022-09624-y.

[165] P. Bellavista, L. Foschini, and A. Mora, "Decentralised learning in federated deployment environments," *ACM Computing Surveys*, vol. 54, pp. 1–38, Apr. 2021. https://doi.org/10.1145/3429252.

[166] A. Sadilek and et al., "Privacy-first health research with federated learning," *NPJ Digital Medicine*, vol. 4, Dec. 2021. https://doi.org/10.1038/s41746-021-00489-2.

[167] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence*, vol. 2, pp. 305–311, June 2020. https://doi.org/10.1038/s42256-020-0186-1.

[168] G. Long, Y. Tan, J. Jiang, and C. Zhang, "Federated learning for open banking." arXiv preprint arXiv:2108.10749, Aug. 2021. http://arxiv.org/abs/2108.10749.

[169] M. Narula, J. Meena, and D. K. Vishwakarma, "A comprehensive review on federated learning for data-sensitive application: Open issues & challenges," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108128, 2024. https://doi.org/10.1016/j.engappai.2024.108128.

[170] Q. V. Pham, M. Zeng, R. Ruby, T. Huynh-The, and W. J. Hwang, "Uav communications for sustainable federated learning," *IEEE Transactions on Vehicular Technology*, vol. 70, pp. 3944–3948, Apr. 2021. https://doi.org/10.1109/TVT.2021.3065084.

[171] S. Dara, A. Kanapala, A. R. Babu, S. Dhamercherala, A. Vidyarthi, and R. Agarwal, "Scalable federated-learning and internet-of-things enabled architecture for chest computer tomography image classification," *Computers and Electrical Engineering*, vol. 102, Sept. 2022. https://doi.org/10.1016/j.compeleceng.2022.108266.

[172] S. K. Singh, L. T. Yang, and J. H. Park, "Fusionfedblock: Fusion of blockchain and federated learning to preserve privacy in industry 5.0," *Information Fusion*, Feb. 2022. https://doi.org/10.1016/j.inffus.2022.09.027.

[173] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," Dec. 2018. [Online]. Available: http://arxiv.org/abs/1812.00564.

[174] S. Boughorbel, F. Jarray, N. Venugopal, S. Moosa, H. Elhadi, and M. Makhlouf, "Federated uncertainty-aware learning for distributed hospital EHR data," Oct. 2019. [Online]. Available: http://arxiv.org/abs/1910.12191.

[175] M. Adnan, S. Kalra, J. C. Cresswell, G. W. Taylor, and H. R. Tizhoosh, "Federated learning and differential privacy for medical image analysis," *Scientific Reports*, vol. 12, Dec. 2022. https://doi.org/10.1038/s41598-022-05539-7.

[176] A. Linardos, K. Kushibar, S. Walsh, P. Gkontra, and K. Lekadir, "Federated learning for multi-center imaging diagnostics: A simulation study in cardiovascular disease," *Scientific Reports*, vol. 12, Dec. 2022. https://doi.org/10.1038/s41598-022-07186-4.

[177] "Diabetic retinopathy detection — kaggle," Dec. 2022. [Online]. Available: https://www.kaggle.com/c/diabetic-retinopathy-detection.

[178] "Chexpert dataset — papers with code," Dec. 2022. [Online]. Available: https://paperswithcode.com/dataset/chexpert.

[179] "Prostatex - grand challenge," Dec. 2022. [Online]. Available: https://prostatex.grand-challenge.org/.

[180] "Rsna bone age — kaggle," Dec. 2022. [Online]. Available: https://www.kaggle.com/datasets/kmader/rsna-bone-age.

[181] "Brats 2017 dataset — papers with code," Dec. 2022. [Online]. Available: https://paperswithcode.com/dataset/brats-2017-1.

[182] "TCGA-LUAD," Dec. 2022. [Online]. Available: https://portal.gdc.cancer.gov/projects/TCGA-LUAD.

[183] "TCGA-LUSC," Dec. 2022. [Online]. Available: https://portal.gdc.cancer.gov/projects/TCGA-LUSC.

[184] "Multi-centre, multi-vendor & multi-disease cardiac image segmentation challenge – eucanshare," Dec. 2022. [Online]. Available: http://www.eucanshare.eu/multi-centre-multi-vendor-multi-disease-cardiac-image-segmentation-chall

[185] "Acdc dataset — papers with code," Dec. 2022. [Online]. Available: https://paperswithcode.com/dataset/acdc.

[186] A. Rehman, I. Razzak, and G. Xu, "Federated learning for privacy preservation of healthcare data from smartphone-based side-channel attacks," *IEEE Journal of Biomedical and Health Informatics*, 2022. https://doi.org/10.1109/JBHI.2022.3171852.

[187] Tore Opsahl, "Datasets," Dec. 2022. Accessed: Dec. 06, 2022. [Online]. Available: https://toreopsahl.com/datasets/#online_social_network.

[188] UCI Machine Learning Repository, "Parkinsons data set," Dec. 2022. Accessed: Dec. 07, 2022. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/parkinsons.

[189] Canadian Institute for Cybersecurity, UNB, "Nsl-kdd dataset," Dec. 2022. Accessed: Dec. 07, 2022. [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html.

[190] Z. Xiao, X. Xu, H. Xing, F. Song, X. Wang, and B. Zhao, "A federated learning system with enhanced feature extraction for human activity recognition," *Knowledge-Based Systems*, vol. 229, Oct. 2021. https://doi.org/10.1016/j.knosys.2021.107338.

[191] BMI, Hellenic Mediterranean University, "The mobifall and mobiact datasets," Dec. 2022. Accessed: Dec. 06, 2022. [Online]. Available: https://bmi.hmu.gr/the-mobifall-and-mobiact-datasets-2/.

[192] UCI Machine Learning Repository, "Heterogeneity activity recognition data set," Dec. 2022. Accessed: Dec. 06, 2022. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/heterogeneity+activity+recognition.

[193] S. Silva, B. A. Gutman, E. Romero, P. M. Thompson, A. Altmann, and M. Lorenzi, "Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data," in *Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, (Venice, Italy), pp. 270–274, July 2019. https://doi.org/10.1109/ISBI.2019.8759317.

[194] U. Ahmed, J. C. W. Lin, and G. Srivastava, "Hyper-graph attention based federated learning method for mental health detection," *IEEE Journal of Biomedical and Health Informatics*, 2022. https://doi.org/10.1109/JBHI.2022.3172269.

[195] DermNet, "Image library," Dec. 2022. Accessed: Dec. 06, 2022. [Online]. Available: https://dermnetnz.org/image-library.

[196] UCI Machine Learning Repository, "Epileptic seizure recognition data set," Dec. 2022. Accessed: Dec. 06, 2022. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Epileptic+Seizure+Recognition.

[197] Kaggle, "Chest x-ray (covid-19 & pneumonia)," Dec. 2022. Accessed: Dec. 07, 2022. [Online]. Available: https://www.kaggle.com/datasets/prashant268/chest-xray-covid19-pneumonia.

[198] NVIDIA, "Federated learning powered by nvidia clara — nvidia technical blog." Online, Dec. 2023. Accessed: Dec. 01, 2023. [Online]. Available: https://developer.nvidia.com/blog/federated-learning-clara/.

[199] OpenFL, "Welcome to the open federated learning (openfl) documentation! — openfl 2023.9 documentation." Online, Dec. 2023. Accessed: Dec. 01, 2023. [Online]. Available: https://openfl.readthedocs.io/en/latest/#.

[200] J. Liu *et al.*, "Federated learning-based vertebral body segmentation," *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105451, Nov. 2022. https://doi.org/10.1016/j.engappai.2022.105451.

[201] Kaggle, "Spine segmentation in 3d mr," Dec. 2022. Accessed: Dec. 07, 2022. [Online]. Available: https://www.kaggle.com/c/testzl.

[202] M. M. Rahman, D. Kundu, S. A. Suha, U. R. Siddiqi, and S. K. Dey, "Hospital patients' length of stay prediction: A federated learning approach," *Journal of King Saud University - Computer and Information Sciences*, 2022. https://doi.org/10.1016/j.jksuci.2022.07.006.

[203] New York State Department of Health, "Statewide planning and research cooperative system (sparcs)," Dec. 2022. Accessed: Dec. 07, 2022. [Online]. Available: https://www.health.ny.gov/statistics/sparcs/.

[204] MoleculeNet Project, "Moleculenet - datasets," Dec. 2022. Accessed: Dec. 06, 2022. [Online]. Available: https://moleculenet.org/datasets-1.

[205] D. Byrd and A. Polychroniadou, "Differentially private secure multi-party computation for federated learning in financial applications," in *Proceedings of the 1st ACM International Conference on AI in Finance (ICAIF 2020)*, Association for Computing Machinery, Oct. 2020. https://doi.org/10.1145/3383455.3422562.

[206] M. Schreyer, T. Sattarov, and D. Borth, "Federated and privacy-preserving learning of accounting data in financial statement audits," in *Proceedings of the 3rd ACM International Conference on AI in Finance*, (New York, NY, USA), pp. 105–113, Association for Computing Machinery, Nov. 2022. https://doi.org/10.1145/3533271.3561674.

[207] Kaggle, "Credit card fraud detection," Dec. 2022. Accessed: Dec. 07, 2022. [Online]. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud.

[208] City of Philadelphia, "Philadelphia's initial release of city payments data," Dec. 2022. Accessed: Dec. 07, 2022. [Online]. Available: https://www.phila.gov/2019-03-29-philadelphias-initial-release-of-city-payments-data/.

[209] City of Chicago, "Payments — city of chicago — data portal," Dec. 2022. Accessed: Dec. 07, 2022. [Online]. Available: https://data.cityofchicago.org/Administration-Finance/Payments/s4vu-giwb.

[210] City of York Council, "All payments to suppliers - datasets - york open data," Dec. 2022. Accessed: Dec. 07, 2022. [Online]. Available: https://data.yorkopendata.org/dataset/all-payments-to-suppliers.

[211] Kaggle, "Elliptic data set," Dec. 2022. Accessed: Dec. 07, 2022. [Online]. Available: https://www.kaggle.com/datasets/ellipticco/elliptic-data-set.

[212] G. Shingi, "A federated learning based approach for loan defaults prediction," in *Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 362–368, IEEE Computer Society, Nov. 2020. https://doi.org/10.1109/ICDMW51313.2020.00057.

[213] Kaggle, "Ltfs data science finhack 3 (analytics vidhya)," Dec. 2022. Accessed: Dec. 07, 2022. [Online]. Available: https://www.kaggle.com/datasets/parv619/ltfs-data-science-finhack-3analytics-vidhya.

[214] Kaggle, "Prudential life insurance assessment," Dec. 2022. Accessed: Dec. 07, 2022. [Online]. Available: https://www.kaggle.com/c/prudential-life-insurance-assessment.

[215] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Transactions on Communications*, vol. 68, pp. 317–333, Jan. 2020. https://doi.org/10.1109/TCOMM.2019.2944169.

[216] X. Zeng, S. Xia, K. Yang, Y. Wu, and Y. Shi, "Over-the-air computation for vertical federated learning," in *Proceedings of the 2022 IEEE International Conference on Communications Workshops (ICC Workshops 2022)*, pp. 788–793, Institute of Electrical and Electronics Engineers (IEEE), 2022. https://doi.org/10.1109/ICCWorkshops53468.2022.9814484.

[217] X. Zheng, S. B. H. Shah, A. K. Bashir, R. Nawaz, and U. Rana, "Distributed hierarchical deep optimization for federated learning in mobile edge computing," *Computer Communications*, vol. 194, pp. 321–328, Oct. 2022. https://doi.org/10.1016/j.comcom.2022.07.028.

[218] K. Yang, Z. Song, Y. Zhang, Y. Zhou, X. Sun, and J. Wang, "Model optimization method based on vertical federated learning," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, Institute of Electrical and Electronics Engineers Inc., 2021. https://doi.org/10.1109/ISCAS51556.2021.9401521.

[219] J. J. Gamboa-Montero, F. Alonso-Martin, S. Marques-Villarroya, J. Sequeira, and M. A. Salichs, "Asynchronous federated learning system for human–robot touch interaction," *Expert Systems with Applications*, vol. 211, p. 118510, Jan. 2023. https://doi.org/10.1016/j.eswa.2022.118510.

[220] O. Choudhury *et al.*, "Differential privacy-enabled federated learning for sensitive health data," *arXiv preprint arXiv:1910.02578*, Oct. 2019. http://arxiv.org/abs/1910.02578.

[221] M. Malekzadeh, B. Hasircioglu, N. Mital, K. Katarya, M. E. Ozfatura, and D. Gündüz, "Dopamine: Differentially private federated learning on medical data," *arXiv preprint arXiv:2101.11693*, Jan. 2021. http://arxiv.org/abs/2101.11693.

[222] H. Zhu, R. Wang, Y. Jin, and K. Liang, "Pivodl: Privacy-preserving vertical federated learning over distributed labels," *IEEE Transactions on Artificial Intelligence*, pp. 1–1, Dec. 2021. https://doi.org/10.1109/TAI.2021.3139055.

[223] J. So, B. Guler, and A. S. Avestimehr, "Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, pp. 479–489, Jan. 2021. https://doi.org/10.1109/JSAIT.2021.3054610.

[224] V. Rey, P. M. S. Sánchez, A. H. Celdrán, and G. Bovet, "Federated learning for malware detection in iot devices," *Computer Networks*, vol. 204, Feb. 2022. https://doi.org/10.1016/j.comnet.2021.108693.

[225] T. Ryffel and et al., "A generic framework for privacy preserving deep learning." arXiv preprint, Nov. 2018. [Online]. Available: http://arxiv.org/abs/1811.04017.

[226] M. Abadi and et al., "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, (Savannah, GA, USA), pp. 265–283, USENIX Association,

Nov. 2016. Accessed: Jul. 23, 2023. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

[227] S. Imambi, K. B. Prakash, and G. R. Kanagachidambaresan, "Pytorch," in *Programming with TensorFlow*, pp. 87–104, Springer, Cham, 2021. https://doi.org/10.1007/978-3-030-57077-4_10.

[228] The TensorFlow Federated Authors, "Tensorflow federated: Machine learning on decentralized data." https://www.tensorflow.org/federated, 2021. Accessed: April 20, 2025.

[229] FATE, "Fate." Online, Dec. 2023. Accessed: Dec. 01, 2023. [Online]. Available: https://fate.fedai.org/.

[230] Owkin, "Substra - open source federated learning software for healthcare research — owkin." Online, Dec. 2023. Accessed: Dec. 01, 2023. [Online]. Available: https://www.owkin.com/substra.

[231] IBM, "Ibm federated learning - ibm documentation." Online, Dec. 2023. Accessed: Dec. 01, 2023. [Online]. Available: https://www.ibm.com/docs/en/cloud-paks/cp-data/4.8.x?topic=models-federated-learning.

[232] PaddlePaddle Developers, "PaddlePaddle: An Open-Source Deep Learning Platform." https://github.com/PaddlePaddle/Paddle, 2024. Accessed: April 20, 2025.

[233] C. He, S. Li, J. So, X. Zeng, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu, *et al.*, "Fedml: A research library and benchmark for federated machine learning," *arXiv preprint arXiv:2007.13518*, 2020. https://doi.org/10.48550/arXiv.2007.13518.

[234] "Secure federated learning for a better world," 2022. Accessed: Jan. 18, 2024. [Online]. Available: https://download.intel.com/newsroom/2022/security/secure-federated-learning_tech-brief.pdf.

[235] K. C. Jeon, G. S. Han, C. Y. Han, and I. Chong, "Federated learning model for contextual sensitive data quality applications: Healthcare use case," in *Proceedings of the 31st IEEE Conference on Signal Processing and Communications Applications*

(SIU 2023), Institute of Electrical and Electronics Engineers Inc., 2023. https://doi.org/10.1109/SIU59756.2023.10223768.

[236] W. Sun, S. Lian, H. Zhang, and Y. Zhang, "Lightweight digital twin and federated learning with distributed incentive in air-ground 6g networks," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 3, pp. 1214–1227, 2022. https://doi.org/10.1109/TNSE.2022.3217923.

[237] H. Cai, L. Gao, J. Wang, and F. Li, "Reliable incentive mechanism in hierarchical federated learning based on two-way reputation and contract theory," *Future Generation Computer Systems*, 2024.

[238] Z. Li, L. Wang, G. Chen, Z. Zhang, M. Shafiq, and Z. Gu, "E2egi: End-to-end gradient inversion in federated learning," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 2, pp. 756–767, 2022. https://doi.org/10.1109/JBHI.2022.3204455.

[239] Z. Li, J. Chen, L. Wang, B. Cheng, J. Yu, and S. Jiang, "Cnn weight parameter quantization method for fpga," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, pp. 1548–1553, IEEE, 2020. https://doi.org/ICCC51575.2020.9345248.

[240] C. Shu, W. Pang, H. Liu, and S. Lu, "High energy efficiency fpga-based accelerator for convolutional neural networks using weight combination," in *2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)*, pp. 578–582, IEEE, 2019. https://doi.org/10.1109/SIPROCESS.2019.8868502.

[241] Z. Bao, K. Zhan, W. Zhang, and J. Guo, "Lsfq: A low precision full integer quantization for high-performance fpga-based cnn acceleration," in *2021 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS)*, pp. 1–6, IEEE, 2021. https://doi.org/10.1109/COOLCHIPS52128.2021.9410327.

[242] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, "Pact: Parameterized clipping activation for quantized neural networks," *arXiv preprint arXiv:1805.06085*, 2018. http://arxiv.org/abs/1805.06085.

[243] H. Li, C. Yan, S. Lin, X. Zheng, B. Zhang, F. Yang, and R. Ji, "Pams: Quantized super-resolution via parameterized max scale," in *Computer Vision–ECCV*

2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16, pp. 564–580, Springer, 2020. https://doi.org/10.1007/978-3-030-58595-2_34.

[244] J. Fang, A. Shafiee, H. Abdel-Aziz, D. Thorsley, G. Georgiadis, and J. H. Hassoun, "Post-training piecewise linear quantization for deep neural networks," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 69–86, Springer, 2020. doi.org/10.1007/978-3-030-58536-5_5.

[245] Z. Dong, Z. Yao, A. Gholami, M. W. Mahoney, and K. Keutzer, "Hawq: Hessian aware quantization of neural networks with mixed-precision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 293–302, 2019. http://arxiv.org/abs/1905.03696.

[246] A. Thakur, D. V. Ranga, and R. Agarwal, "Performance benchmarking and analysis of blockchain platforms," in *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*, 2022. https://doi.org/10.2139/ssrn.4385643.

[247] M. Shayan, C. Fung, C. J. Yoon, and I. Beschastnikh, "Biscotti: A blockchain system for private and secure federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 7, pp. 1513–1525, 2020. https://doi.org/10.1109/TPDS.2020.3044223.

[248] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *ICC 2019-2019 IEEE international conference on communications (ICC)*, pp. 1–7, IEEE, 2019. https://doi.org/10.1109/ICC.2019.8761315.

[249] S. AbdulRahman, H. Tout, A. Mourad, and C. Talhi, "Fedmccs: Multicriteria client selection model for optimal iot federated learning," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4723–4735, 2020. https://doi.org/10.1109/JIOT.2020.3028742.

[250] N. Ferdinand, H. Al-Lawati, S. C. Draper, and M. Nokleby, "Anytime minibatch: Exploiting stragglers in online distributed optimization," *arXiv preprint*

arXiv:2006.05752, 2020. http://arxiv.org/abs/2006.05752.

[251] A. Reisizadeh, H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani, "Robust and communication-efficient collaborative learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019. https://doi.org/10.48550/arXiv.1907.10595.

[252] 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*, pp. 770–778, 2016.

[253] R. S. Dehal, C. Munjal, A. A. Ansari, and A. S. Kushwaha, "Gpu computing revolution: Cuda," in *Proceedings - IEEE 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pp. 197–201, 2018. https://doi.org/10.1109/ICACCCN.2018.8748495.

[254] A. Ziller *et al.*, "Pysyft: A library for easy federated learning," in *Federated Learning Systems*, pp. 111–139, Springer, Cham, 2021. https://doi.org/10.1007/978-3-030-70604-3_5.

[255] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. https://doi.org/10.1109/5.726791.

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

[257] "Emnist (extended mnist) — kaggle." Accessed on June 07, 2023.

[258] "Leaf: A benchmark for federated settings." Accessed on June 07, 2023.

[259] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008. https://doi.org/10.1109/TPAMI.2008.128.

[260] S. Rani, A. Kataria, S. Kumar, and P. Tiwari, "Federated learning for secure iomt-applications in smart healthcare systems: A comprehensive review," *Knowledge-Based Systems*, p. 110658, 2023.

[261] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," *arXiv preprint arXiv:1905.10497*, 2019.

[262] X. Zhang, Y. Kang, K. Chen, L. Fan, and Q. Yang, "Trading off privacy, utility, and efficiency in federated learning," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 6, pp. 1–32, 2023.

[263] A. Farhad, S. Woolley, and P. Andras, "Federated learning for ai to improve patient care using wearable and iomt sensors," in *2021 IEEE 9th International Conference on Healthcare Informatics (ICHI)*, pp. 434–434, IEEE, 2021.

[264] M. Arya and H. Sastry, "Ensemble federated learning for classifying iomt data streams," in *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*, pp. 1–5, IEEE, 2022.

[265] A. Lakhan, M. A. Mohammed, J. Nedoma, R. Martinek, P. Tiwari, A. Vidyarthi, A. Alkhayyat, and W. Wang, "Federated-learning based privacy preservation and fraud-enabled blockchain iomt system for healthcare," *IEEE journal of biomedical and health informatics*, vol. 27, no. 2, pp. 664–672, 2022.

[266] R. Chandramouli, S. Bapatla, K. Subbalakshmi, and R. Uma, "Battery power-aware encryption," *ACM Transactions on Information and System Security (TISSEC)*, vol. 9, no. 2, pp. 162–180, 2006.

[267] G. Dini and I. M. Savino, "Lark: a lightweight authenticated rekeying scheme for clustered wireless sensor networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 10, no. 4, pp. 1–35, 2011.

[268] L. Harn, C.-F. Hsu, Z. Xia, and Z. He, "Lightweight aggregated data encryption for wireless sensor networks (wsns)," *IEEE Sensors Letters*, vol. 5, no. 4, pp. 1–4, 2021.

[269] K. N. Singh, O. Singh, A. K. Singh, and A. K. Agrawal, "Eimol: a secure medical image encryption algorithm based on optimization and the lorenz system," *ACM*

Transactions on Multimedia Computing, Communications and Applications, vol. 19, no. 2s, pp. 1–19, 2023.

[270] S. Wang, K. Huang, K. Ma, X. Xu, and X. Hu, "A lightweight encryption and message authentication framework for wireless communication," *IET Communications*, vol. 17, no. 3, pp. 265–278, 2023.

[271] M. Adil, M. K. Khan, M. M. Jadoon, M. Attique, H. Song, and A. Farouk, "An ai-enabled hybrid lightweight authentication scheme for intelligent iomt based cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, 2022.

[272] S. Roy and K. L. Baishnab, "Spise: A tiny, cost effective, speedy block cipher for low resourced devices," *Journal of Engineering Research*, vol. 11, no. 3, pp. 106–113, 2023.

[273] L. Heim, A. Biri, Z. Qu, and L. Thiele, "Measuring what really matters: Optimizing neural networks for tinyml. arxiv 2021," *arXiv preprint arXiv:2104.10645*, 2021.

[274] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[275] S. Imambi, K. B. Prakash, and G. R. Kanagachidambaresan, "Pytorch," in *Programming with TensorFlow: Solution for Edge Computing Applications*, pp. 87–104, 2021.

[276] T. E. Oliphant *et al.*, *Guide to numpy*, vol. 1. Trelgol Publishing USA, 2006.

[277] J. So, B. Güler, and A. S. Avestimehr, "Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 1, pp. 479–489, 2021.

[278] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "Verifynet: Secure and verifiable federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 911–926, 2019.

[279] M. Narula, J. Meena, and D. K. Vishwakarma, "Federated workload-aware quantized framework for secure learning in data-sensitive applications," *Future Generation Computer Systems*, vol. 168, p. 107772, 2025. `https://doi.org/10.1016/j.future.2025.107772`.

[280] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016. https://doi.org/10.1145/2976749.2978318.

[281] A. Cheu, A. Smith, J. Ullman, and D. Zeber, "Distributed differential privacy via shuffling," in *Advances in Cryptology–EUROCRYPT 2019*, pp. 375–403, Springer International Publishing, 2019. https://doi.org/10.1007/978-3-030-17653-2_13.

[282] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additive homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2018. https://doi.org/10.1109/TIFS.2017.2787987.

[283] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, "Privacy-preserving federated learning with differential privacy and homomorphic encryption," *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 2455–2458, 2019. https://doi.org/10.1145/3357384.3358120.

[284] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, 2018.

Proof of Publications

Refereed journals:

Manu Narula, Jasraj Meena, and Dinesh Kumar Vishwakarma, "A comprehensive review on Federated Learning for Data-Sensitive Application: Open issues & challenges," *Engineering Applications of Artificial Intelligence*, Elsevier, Volume 133, Part B, pp. 108128, 2024, DOI:https://doi.org/10.1016/j.engappai.2024.108128. **(SCIE, IF=8.0)**.

Engineering Applications of Artificial Intelligence 133 (2024) 108128

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Survey Paper

A comprehensive review on Federated Learning for Data-Sensitive Application: Open issues & challenges

Manu Narula [a], Jasraj Meena [b], Dinesh Kumar Vishwakarma [a,*]

[a] Department of Information Technology, Delhi Technological University, India
[b] School of Computer & Systems Sciences, Jawaharlal Nehru University, India

ARTICLE INFO

CCS CONCEPTS:
Federated learning
Data-sensitive applications
Security and privacy
Distributed learning
Data security
Internet of things
Edge networks

ABSTRACT

Artificial intelligence employs Machine Learning (ML) and Deep Learning (DL) to analyze data. In both, the data is stored centrally. The data involved may be sensitive and leakage may incur consequences. Applications dealing with intimate data, with critical results, cannot afford this risk and are termed Data-Sensitive Applications (DSA). Some examples are healthcare, finance, etc. The data required for DSA cannot be stored centrally due to large amounts, or isolated data islands. The ML and DL techniques following a data-centralized approach have difficulties in handling the scattered data frequently associated with DSA. Federated Learning (FL) acknowledges the scattered data and provides a more secure and efficient way to analyze such data. This motivates previously reluctant entities like banks to collaborate for variety and quantity of data. Most DSA transitioned to FL, but the migration is not without concerns. These include communication costs, heterogeneity, and malicious attacks. In this paper, we deeply analyze the role of FL in DSA and provide a taxonomy for the studies and implementations of FL. Then we provide an insight into DSA covering works in healthcare and finance. A glance is provided at attempts in non-DSA with possible DSA applications. Finally, we discuss FL's open issues and challenges with their possible solutions.

1. Introduction

With the emergence of Artificial Intelligence (AI), there has been a growth of in-conceivable standards in technology affecting nearly every field, however minuscule or vast it may be (Ahmed et al., 2022a). AI applications require significant data to train Machine Learning (ML) and Deep Learning (DL) models for accurate results. In both, the data is stored centrally in one place or shared/copied to support parallel training. This meant that all the data, directly or indirectly, was controlled by a third party which may or may not be trusted by the data owners. This was not a major concern for applications dealing with general or casual data like e-commerce, recommender systems, etc. However, it became a respectable concern for the applications that dealt with sensitive data like finance, healthcare, surveillance systems, etc. Moreover, the data involved with these applications are either in much more significant amounts that cannot be stored centrally or in the form of isolated data islands. It became a pivotal issue and limited the acceptance of ML and DL in these applications. This is because when ML or DL is applied over scattered data, the amount of data available for the training process at one location is insufficient for a good accuracy output. Since Data-Sensitive Applications (DSA) frequently deal with scattered data islands, ML and DL are of limited utility.

A data-sensitive application or data-sensitive field is conceptualized as a set of implementations that depend on data of a highly confidential nature, such as data related to the identity or social footprint of individuals (Lin et al., 2021). In addition to this, in most of these deployments, accuracy takes center stage, and the slightest variation in data processing can result in invalid outcomes. Some of the works in DSA are Cerebral ataxia diagnosis (Ngo et al., 2022), Fraudulent transaction detection (Myalil et al., 2021), (Lv et al., 2021), Tumor characterization (Li et al., 2019a), (Hussein et al., 2019), etc. A major challenge when dealing with DSA is that it is difficult to apply direct encryptions within the training and analysis process without affecting performance, which renders such measures inappropriate. Another challenge in these applications is that one entity cannot have sufficient data for good-quality training in practical environments. The concerned institutions are largely resistant to collaborating, given the nature of the data involved and the competitive onset amongst them.

To cater to the demands of DSA, Federated Learning (FL), came out as the top contender for private training and analysis. The existence of

* Corresponding author.
E-mail addresses: manunarula1996@gmail.com (M. Narula), jasrajengg@gmail.com (J. Meena), dvishwakarma@gmail.com (D.K. Vishwakarma).

https://doi.org/10.1016/j.engappai.2024.108128
Received 7 October 2023; Received in revised form 16 February 2024; Accepted 19 February 2024
0952-1976/© 2024 Elsevier Ltd. All rights reserved.

Manu Narula, Jasraj Meena, and Dinesh Kumar Vishwakarma, "Federated workload-aware quantized framework for secure learning in data-sensitive applications," *Future Generation Computer Systems*, Elsevier, vol. 168, pp.107772, 2025, DOI: https://doi.org/10.1016/j.future.2025.107772. **(SCIE, IF=6.1)**

Future Generation Computer Systems 168 (2025) 107772

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Federated workload-aware quantized framework for secure learning in data-sensitive applications

Manu Narula [a], Jasraj Meena [b], Dinesh Kumar Vishwakarma [a,*]

[a] Delhi Technological University, Delhi, India
[b] Jawaharlal Nehru University, Delhi, India

ARTICLE INFO

Keywords:
Federated learning
Internet of Things
Security
Edge networks

ABSTRACT

Federated Learning (FL) emerged as a leading secure, distributed learning technology based on sharing insights instead of data. The privacy-ensuring capability of FL has enabled its extensive use in Data-Sensitive Applications like healthcare and finance. However, the transmitted insights are at risk of leakage as the security of the medium cannot be guaranteed and can lead to the inference of the user data. Quantization is sometimes used to change these transmitted values to provide security but at the cost of accuracy loss in global models. Coupled with client dropouts, this increases performance loss. In this paper, we propose a Federated Workload-Aware Framework with Linear Quantization (Fed-WALQ), which layers the quantization process with an active client-selection technique based on the sustainable workload of the clients. The framework minimizes the dropout rates and compensates for the loss due to quantization. Through numerical experiments compared against traditional FL and Quantization-enabled FL over multiple datasets, the Fed-WALQ shows improvements in security over the former and accuracy over the latter. The accuracy improvement varies with the complexities of the involved datasets, while a substantial drop in straggler node percentages is seen in all cases (up to 91.8% drop).

1. Introduction

Artificial Intelligence (AI) has affected every sphere of day-to-day life in some capacity over the years. Every industry has incorporated AI in its production or planning section at some level, either directly or indirectly [1]. AI models deployed for any application demand enormous amounts of data for appropriate training. In many cases, the required data can be sensitive or may be used to deduce crucial information about users via analytical approaches. Disclosure of such information can lead to many issues, including but not limited to several cybercriminal activities. The growing concern regarding data privacy led to the inception of Federated Learning (FL) [2]. FL provided a way to enable researchers and data scientists to work on private datasets for insights without accessing the data. This is achieved by replacing the data sharing with parameter/gradient sharing among an extensive network of computing nodes working over their local private data [3]. The insights/gradients shared can then modify the global results as required.

The various abbreviations and symbols used throughout the manuscript are summarized in Tables 1 and 2 respectively.

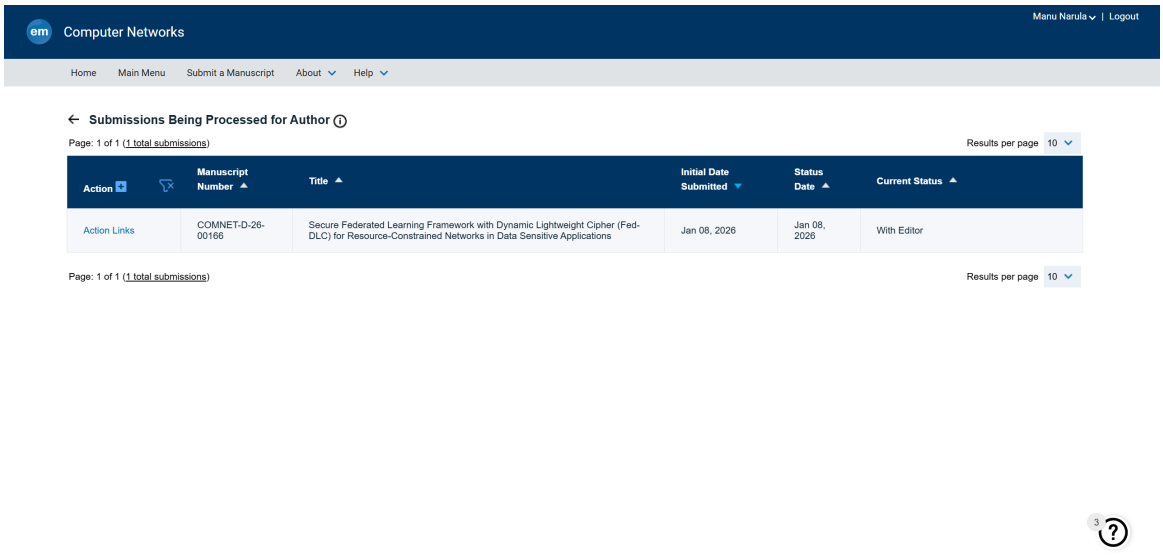
1.1. Federated learning

FL was seeded in 2016 through the work of McMahan et al. [3] proposing parameter transmission instead of data transmission in a distributed learning network. The new approach increased the security quotient and gave the data owners more control over what is to be shared publicly and to what extent. FL has found applications in various IoT areas, such as IoMT [4], IIoT [5,6], and Vehicular Networks [7]. Broadly, FL is divided into Horizontal or homogeneous FL [8,9] and Vertical or heterogeneous FL [10,11]. The former works upon the assumption that the training data available to different devices are identical in terms of feature set and differ only in sampling method and sample size. For example, sales records of multiple medical shops. The number of customers for each shop can vary, but the features will be the same. The latter focuses on data partitioning, which has no exact attributes throughout the subsets. A single controller (Host) handles the data features' dissimilarities. The standard identification keys for entries that contain data from different owners in separate columns in the same row are calculated via Private Set Intersection (PSI) schemes [12–14].

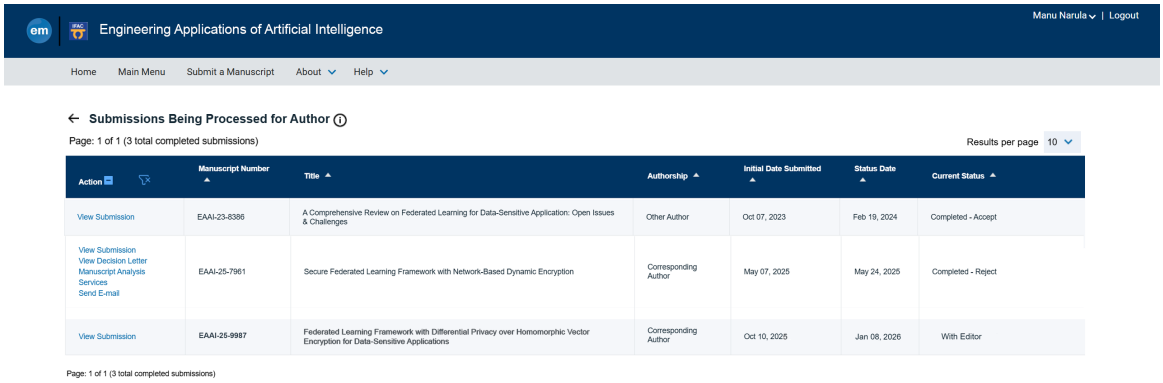
* Corresponding author.
E-mail address: dinesh@dtu.ac.in (D.K. Vishwakarma).

https://doi.org/10.1016/j.future.2025.107772
Received 6 June 2024; Received in revised form 17 January 2025; Accepted 18 February 2025
Available online 28 February 2025
0167-739X/© 2025 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

Manu Narula, Jasraj Meena, and Dinesh Kumar Vishwakarma, "Secure Federated Learning Framework with Dynamic Lightweight Cipher (Fed-DLC) for Resource-Constrained Networks in Data Sensitive Applications," *Computer Networks*, Elsevier. -*Communicated*



Manu Narula, Jasraj Meena, and Dinesh Kumar Vishwakarma, "Federated Framework with Differential Privacy over Homomorphic Vector Encryption for Data-Sensitive Applications,"*Engineering Applications of Artificial Intelligence*, Elsevier. -*Communicated*



International conferences:

Manu Narula, Jasraj Meena and Dinesh Kumar Vishwakarma, "A Comprehensive Study of Federated Learning Tools and Frameworks for Data-Sensitive Applications",*Puneet Kumar Gupta (eds), Computational Models for Intelligence and Automation*, SCRS, India, 2025, pp. 127-135. DOI: https://doi.org/10.56155/978-81-975670-1-8-12

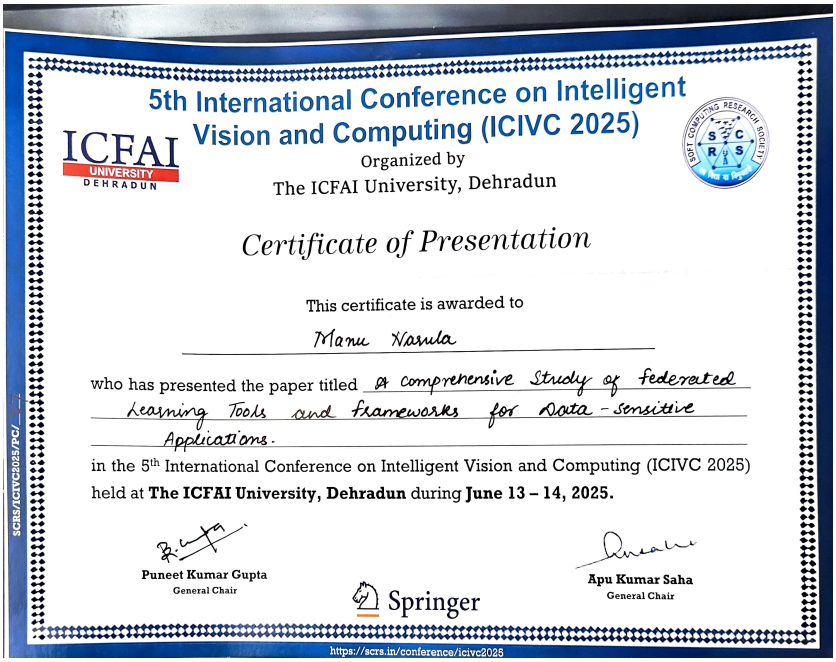
A Comprehensive Study of Federated Learning Tools and Frameworks for Data-Sensitive Applications

Manu Narula[1], Jasraj Meena[2], Dinesh Kumar Vishwakarma[1]
Delhi Technological University, India[1]
Jawaharlal Nehru University, India[2]
Corresponding author: Manu Narula, Email: manunarula1996@gmail.com

Artificial intelligence depends upon various learning techniques to analyze and process data. Most of these techniques store data centrally or clone it to all training devices. While it presents no issue in standard applications, it quickly becomes a potential security risk when sensitive information gets involved in training, including a range of healthcare and finance utilities. Additionally, the data available for training of a Data Sensitive Application is often scattered across isolated data islands, either due to collaborative or competitive factors. Federated Learning (FL) provides a secure and efficient way to analyze such data. Due to the infancy of the technology, many tools and frameworks of FL are not well known and hence may not be used where their potential utility is maximum. This paper analyzes the various implementations and tools of FL and highlights their suitability for different scenarios with the DSAs.

Keywords: Federated Learning, Tools and Frameworks, Internet of Things, Distributed Learning

2025. In Puneet Kumar Gupta (ed.), *Computational Models for Intelligence and Automation*, 127–135. Computing & Intelligent Systems, SCRS, India. DOI: https://doi.org/10.56155/978-81-975670-1-8-12


5th International Conference on Intelligent Vision and Computing (ICIVC 2025)

ICFAI UNIVERSITY DEHRADUN

Organized by
The ICFAI University, Dehradun

Certificate of Presentation

This certificate is awarded to

Manu Narula

who has presented the paper titled A comprehensive Study of federated learning Tools and frameworks for Data-sensitive Applications.

in the 5th International Conference on Intelligent Vision and Computing (ICIVC 2025) held at **The ICFAI University, Dehradun** during **June 13 – 14, 2025.**

Puneet Kumar Gupta
General Chair

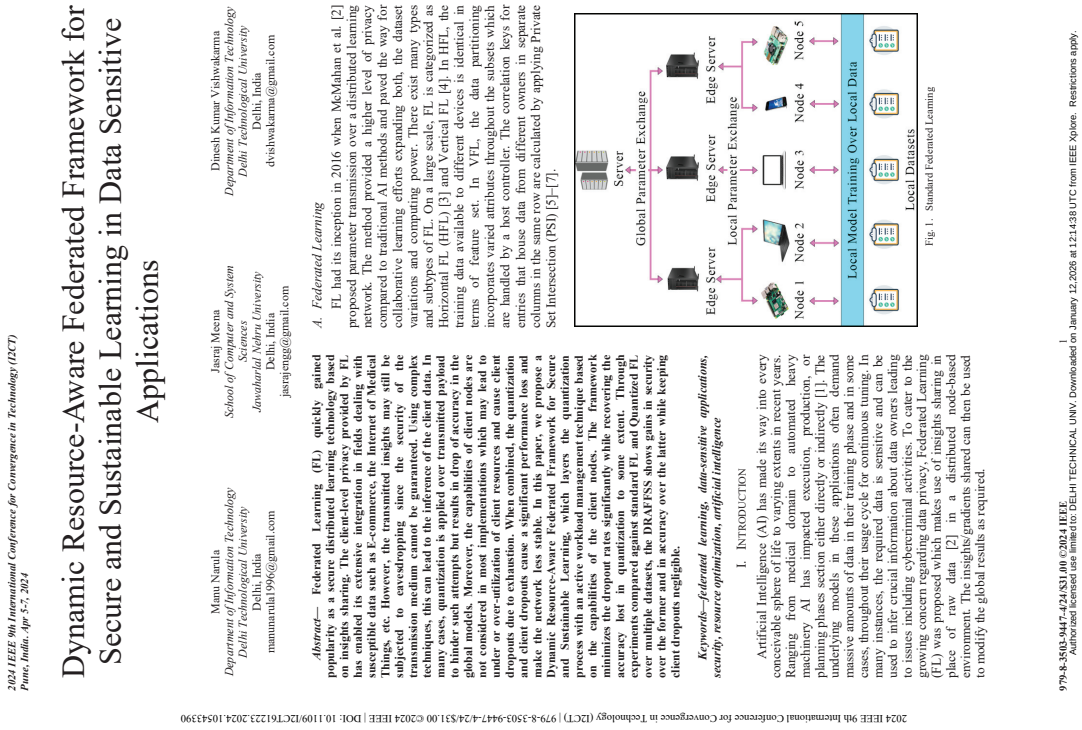
Springer

Apu Kumar Saha
General Chair

https://scrs.in/conference/icivc2025

SCRS/ICIVC2025/PC/

Manu Narula, Jasraj Meena, and Dinesh Kumar Vishwakarma, "Dynamic Resource-Aware Federated Framework for Secure and Sustainable Learning in Data Sensitive Applications," 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), Pune, India, 2024, pp. 1-8, 2025,https://doi.org/10.1109/I2CT61223.2024.10543390



2024 IEEE 9th International Conference for Convergence in Technology (I2CT)
Pune, India. Apr 5-7, 2024

Dynamic Resource-Aware Federated Framework for Secure and Sustainable Learning in Data Sensitive Applications

Manu Narula
Department of Information Technology
Delhi Technological University
Delhi, India
manunarula1996@gmail.com

Jasraj Meena
School of Computer and System Sciences
Jawaharlal Nehru University
Delhi, India
jasrajmeg@gmail.com

Dinesh Kumar Vishwakarma
Department of Information Technology
Delhi Technological University
Delhi, India
dvishwakarma@gmail.com

***Abstract*— Federated Learning (FL) quickly gained popularity as a secure distributed learning technology based on insights sharing. The client-level privacy provided by FL has enabled its extensive integration in fields dealing with susceptible data such as E-commerce, the internet of Medical Things, etc. However, the transmitted insights may still be subjected to eavesdropping since the security of the transmission medium cannot be guaranteed. Using complex techniques, this can lead to the inference of the client data. In many cases, quantization is applied over transmitted payload to hinder such attempts but results in drop of accuracy in the global models. Moreover, the capabilities of client nodes are not considered in most implementations which may lead to under or over-utilization of client resources and cause client dropouts due to exhaustion. When combined, the quantization and client dropouts cause a significant performance loss and make the network less stable. In this paper, we propose a Dynamic Resource-Aware Federated Framework for Secure and Sustainable Learning which layers the quantization process with an active workload management technique based on the capabilities of the client nodes. The framework minimizes the dropout rates significantly while recovering the accuracy lost in quantization to some extent. Through experiments compared against standard FL and Quantized FL over multiple datasets, the DRAFFSS shows gains in security over the former and in accuracy over the latter while keeping client dropouts negligible.**

Keywords*—*federated learning, data-sensitive applications, security, resource optimization, artificial intelligence

I. Introduction

Artificial Intelligence (AI) has made its way into every conceivable sphere of life to varying extents in recent years. Ranging from medical domain to automated heavy machinery, AI has impacted execution, production, or planning phases section either directly or indirectly [1]. The underlying models in these applications often demand massive amounts of data in their training phase and in some cases, throughout their usage cycle for continuous tuning. In many instances, the required data is sensitive and can be used to infer crucial information about data owners leading to issues including cybercriminal activities. To cater to the growing concern regarding data privacy, Federated Learning (FL) was proposed which makes use of insights sharing in place of raw data [2] in a distributed node-based environment. The insights/gradients shared can then be used to modify the global results as required.

A. Federated Learning

FL had its inception in 2016 when McMahan et al. [2] proposed parameter transmission over a distributed learning network. The method provided a higher level of privacy compared to traditional AI methods and paved the way for collaborative learning efforts expanding both, the dataset variations and computing power. There exist many types and subtypes of FL. On a large scale, FL is categorized as Horizontal FL (HFL) [3] and Vertical FL [4]. In HFL, the training data available to different devices is identical in terms of feature set. In VFL, the data partitioning incorporates varied attributes throughout the subsets which are handled by a host controller. The correlation keys for entries that house data from different owners in separate columns in the same row are calculated by applying Private Set Intersection (PSI) [5]–[7].

Fig. 1. Standard Federated Learning

2024 IEEE 9th International Conference for Convergence in Technology (I2CT) | 979-8-3503-9447-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/I2CT61223.2024.10543390

979-8-3503-9447-4/24/$31.00 ©2024 IEEE
Authorized licensed use limited to: DELHI TECHNICAL UNIV. Downloaded on January 12,2026 at 12:14:38 UTC from IEEE Xplore. Restrictions apply.

9th International Conference for Convergence in Technology (I2CT)

5th – 7th April 2024

Certificate

This is to certify that Dr./Prof./Mr./Ms. ***Manu Narula*** *has presented paper entitled* ***Dynamic Resource-Aware Federated Framework for Secure and Sustainable Learning in Data Sensitive Applications*** *in 9th International Conference for Convergence in Technology (I2CT) during 5th & 7th April 2024.*

Dr. Chanakya Kumar Jha
General Chair -I2CT

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Bawana Road-Delhi-42

Plagiarism Verification

Title of the Thesis: Design and Development of Secure Federated Learning Framework for Data-Sensitive Applications

Total Pages:

Name of the Scholar: Manu Narula

Supervisors: Prof. Dinesh Kumar Vishwakarma

Joint-Supervisor: Dr. Jasraj Meena

Department: Information Technology

This is to report that the above thesis has been scanned for similarity detection. The process and outcome are given below:

Software used: Turnitin Similarity Index: 4 % Word Count: 59059

(**Manu Narula**)
(Regn. No. 2K21/PHDIT/02)

Prof. Dinesh K. Vishwakarma
(Supervisor)
Professor
Department of Information Technology
Delhi Technological University, Delhi, India

Dr. Jasraj Meena
(Joint-Supervisor)
Associate Professor
School of Computer and System Sciences
Jawaharlal Nehru University, Delhi, India

Place: Delhi, INDIA
Date:

Curriculum Vitae

MANU NARULA

MALE, 29 | NEW DELHI

manunarula1996@gmail.com | +91-9212158133 | https://in.linkedin.com/in/manu-narula-876267164

~4 yrs. exp. *In Research: Machine Learning, Deep Learning, Distributed Learning, Data Science, & Data Security.*

EDUCATION

PH.D. (Information Technology), **Department of IT (DTU), 2021-Present**
M.TECH. (Computer Science & Engineering), **USICT (GGSIPU), 2019-2021** | **CGPA: 8.23** | Specialization: **Data Security**
B. Tech. (Information Technology), **ADGITM formerly NIEC (GGSIPU), 2015-2019** | **76.02 %**
XII (CBSE), **D.L.D.A.V. MODEL SCHOOL, NEW DELHI, 2015** | **88.2 %**
X (CBSE), **D.L.D.A.V. MODEL SCHOOL, NEW DELHI, 2013** | **CGPA: 9.0**

EXPERIENCE

Research Scholar, **DEPARTMENT OF IT, DTU, Aug 21 – Present** Teaching Assistant, **DEPARTMENT OF IT, DTU, Aug 21 – Present**
Teaching Assistant, **APTECH INSTITUTE, Jun 18 – Aug-18**

PROJECTS & INTERNSHIPS

- **Anti-Spoof Detection System for facial spoofs –** implemented code to identify fake or spoof faces inputted to facial recognition systems using video vision transformers, 2023
- **Secure Distributed Learning Framework for Low Powered Networks –** implemented code to strengthen security for federated learning in low-powered networks such as IoMT using lightweight encryptions, 2023
- **Secure Hybrid Framework for Sensitive Applications –** implemented code to strengthen security with minimal performance loss for federated learning in sensitive applications (Healthcare, Finance), 2023
- **Detailed Comprehensive Analysis of Secure Distributed Learning for Sensitive Applications –** conducted detailed analytical study on efforts and accomplishments in federated learning setups for sensitive applications (Healthcare, Finance, Military), 2022
- **Hybrid Steganography Framework against Analytical Attacks –** implemented code to fortify multi-channel stego-images against analytical and brute force attacks, 2021
- **Realtime Driver Drowsiness Detection System** – implemented real-time detection system for measuring drowsiness of a driver and trigger alarm, if necessary, 2019
- **Realtime Dynamic Aptitude Exam and Recommender System** – implemented code to conduct student specific dynamic aptitude exams in real-time for in-depth recommendations, 2018

PUBLICATIONS

- M. Narula, J. Meena and D.K. Vishwakarma, " A Comprehensive Study of Federated Learning Tools and Frameworks for Data-Sensitive Applications," 5th International Conference on Intelligent Vision and Computing (ICIVC 2025), Dehradun, India, 2025
- M. Narula, J. Meena and D.K. Vishwakarma, "Federated workload-aware quantized framework for secure learning in data-sensitive applications," Future Generation Computer Systems 168 (2025): 107772., Elsevier, doi: 10.1016/j.future.2025.107772
- M. Narula, J. Meena and D.K. Vishwakarma, "A comprehensive review on Federated Learning for Data-Sensitive Application: Open issues & challenges," Engineering Applications of Artificial Intelligence, Volume 133, Part B, pp. 108128, 2024, Elsevier, doi: 10.1016/j.engappai.2024.108128.
- M. Narula, J. Meena, & D. K. Vishwakarma, "A Comprehensive Study of Federated Learning Tools and Frameworks for Data-Sensitive Applications." in Puneet Kumar Gupta (ed.), Computational Models for Intelligence and Automation, 127–135. Computing & Intelligent Systems, SCRS, India. DOI: https://doi.org/10.56155/978-81-975670-1-8-12.
- M. Narula, J. Meena and D. K. Vishwakarma, "Dynamic Resource-Aware Federated Framework for Secure and Sustainable Learning in Data Sensitive Applications," 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), Pune, India, 2024, pp. 1-8, doi: 10.1109/I2CT61223.2024.10543390.

- M. Narula and D. Tyagi, "Autonomous Cars: A Comprehensive Survey," *2023 Seventh International Conference on Image Information Processing (ICIIP)*, Solan, India, 2023, pp. 586-590, doi: 10.1109/ICIIP61524.2023.10537704.
- M. Narula, M. Gupta and M. Garg, "Implementation of hybrid technique from spatial and frequency domain steganography: Along with cryptography to withstand statistical attacks," 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2020, pp. 803-808, doi: 10.1109/ICACCCN51052.2020.9362754.
- M. Narula, C. Goyal and N. Sinha, "Security Through Steganography- A Survey," International Journal of Science and Research (IJSR), pp. 1796-1803, 2019, doi: 10.21275/ART20203700.
- G. Chugh, M. Narula, S. Prajapati, S. Jain and S. Singh,"A Survey on Application of Automobile Safety Features," International Research Journal of Engineering and Technology (IRJET), vol. 6, no. 4, pp. 5055-5059, 2019.
- V. Mahavar, S. Iskon, M. Narula and T. Dhingra," Analysis of Machine Learning Methodologies in Big Data Applications," International Journal of Recent Research Aspects, vol. 5, no. 1, pp. 157-159, 2018.
- M. Narula, N. Chawla, A. Yadav and S. Prajapati," Text Extraction From Digital Images," International Journal of Recent Research Aspects, vol. 5, no. 1, pp. 178-180, 2018.

INTERESTS, SKILLS & TOOLS

- **Areas of Interest: Software Development**, Data Structures & Algorithms, **Data Science** & Deep Learning, Data Analysis, and Data Security
- **Programming Languages: Java, Python**, Kotlin, C/C++, Java, HTML/CSS
- **Tools & Technologies**: **Eclipse, MySQL, Jupyter Notebook**, Anaconda, Google Colab, MS-Office, Python libraries such as Pytorch, Numpy, Pandas, Matplotlib, etc.
- **Others**: **Comperer, Creative Writer, Motion Designer**

ACHIEVEMENTS, TRAINING & CERTIFICATIONS

- **PyTorch for Deep Learning:** Udemy | 2023
- **Tensorflow for Deep Learning:** Udemy | 2023
- **Organized student outreach program in DTU** | 2023
- **Organized Short Term Course on "Recent Trends in Machine Learning and Deep Learning for AI Applications"** | 2023
- **Attended Workshop on "Intellectual Property Rights and IP Management for Startups"** | 2023
- **Organized student outreach program in DTU** | 2022
- **Hosted multiple Interviews with renowned individuals in DTU** | 2022-2023
- Secured position in top 100 in **HackwithInfy Hackathon** | 2019
- **Python Fundamentals**: course by APTECH | 2019