Securing Intelligent Networks: Federated Learning Approaches for Privacy-Conscious Anomaly Detection

Habib Ullah Manzoor

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Engineering
College of Science and Engineering
University of Glasgow

August 2025

Abstract

As intelligent systems increasingly rely on distributed data generated at the edge, traditional centralized machine learning paradigms face critical limitations related to data privacy, bandwidth constraints, and vulnerability to single points of failure. Federated Learning (FL) has emerged as a promising distributed alternative that enables collaborative model training across devices without sharing raw data, preserving privacy while leveraging collective intelligence. In smart energy systems, Short-Term Load Forecasting (STLF) exemplifies a critical application where FL's privacy-preserving capabilities offer significant advantages, supporting efficient grid operation. However, deploying FL in dynamic and heterogeneous environments poses unique challenges, including security vulnerabilities from untrusted participants, communication bottlenecks in low-bandwidth networks, and managing data heterogeneity across clients. This research systematically addresses these hurdles by targeting four key challenges to advance the practical deployment of FL in energy-centric applications. The first challenge (**C1**) involves model attacks, where adversarial clients attempt to compromise the global model. Existing attacks do not fully exploit FL's vulnerabilities and can be captured with current defence frameworks. This necessitates the development of stealth attack strategies. This research introduces novel stealth model poisoning techniques, including the **Federated Communication Round Attack (Fed-CRA)**, which increases communication rounds without degrading model performance but at the cost of higher resource consumption. These vulnerabilities highlight the need for stronger defense mechanisms, driving the development of more robust frameworks. The second challenge (**C2**) focuses on robust aggregation, as traditional FL methods often struggle to filter out adversarial updates effectively. To address this challenge, we introduce four innovative frameworks: **(a). Federated Random Layer Aggregation (FedRLA)**, which enhances security by aggregating only a randomly selected layer during each round, which reduces the attack surface as attack can only attack single layer of local model thereby mitigating the impact of adversarial updates; **(b). Layer-Based Anomaly Aware Federated Averaging (LBAAFedAvg)**, which detects and isolates compromised layers while ensuring that valid updates are preserved with the help novel clustering criteria to identify good and back clients, improving the overall integrity of the aggregation process; **(c). Federated Incentivized Averaging (Fed-InA)**, specifically designed for Fed-CRA, which is based on game theory, it incentivizes honest clients by rewarding them and penalizes malicious ones, promoting a healthier collaborative environment;

and Decentralized Federated Learning (DFL), which distributes the aggregation process across multiple clients, minimizing the risk of single points of failure and eliminate the need of server. Furthermore, **(d). Decentralized Federated Random Layer Aggregation (DRLA)** combines DFL with FedRLA to significantly enhance robustness against adversarial attacks by aggregating a single layer in peer to peer communication manner. The third challenge (**C3**) concerns communication and computational efficiency, as FL's iterative updates can strain bandwidth and processing resources, especially in energy-constrained environments. The proposed frameworks optimize efficiency by minimizing transmitted data and computational overhead. **FedRLA** significantly reduces communication costs by limiting shared model information, while **Adaptive Single Layer Aggregation (ASLA)** leverages quantization and adaptive stopping criteria to ensure minimal resource usage. Other robust frameworks, LBAA-FedAvg, Fed-InA, and DFL, are designed to require minimal resources for model training. The fourth challenge (**C4**) addresses data heterogeneity, a fundamental issue in energy networks where clients possess diverse consumption patterns. Two frameworks tackle this problem: **(a). FedBranched**, which clusters clients based on data similarity to enhance local model convergence, and **(b). ASLA**, which selectively aggregates the most effective layer across clients, improving generalization across varied datasets. Through addressing these interconnected challenges, this research enhances the robustness of the model, communication and computational efficiency and fixes the issue of data heterogeneity between different clients, paving the way for more resilient and privacy-preserving intelligent systems.

Keywords

Federated Learning, Energy Efficiency, Robust Aggregation, Anomaly Detection, Data Heterogeneity, Communication Efficiency

Acknowledgements

All praise is due to **Allah Almighty**, the Most Merciful and Compassionate, who has granted me strength, perseverance, and wisdom throughout this challenging yet rewarding journey. I am deeply grateful for the teachings of our beloved **Prophet Muhammad (Peace Be Upon Him)**, whose exemplary character, guidance, and blessings have served as a profound source of inspiration and motivation in my life.

I extend my heartfelt thanks to my parents **Mr. and Mrs. Manzoor Ahmad**, whose unwavering faith, fervent prayers, and countless sacrifices have been instrumental in shaping my path toward this significant achievement. Their relentless support has provided me with the foundation I needed to pursue my dreams.

I express profound gratitude to my supervisor, **Dr. Ahmed Zoha**, for his steadfast support, insightful guidance, and remarkable patience throughout this research endeavor. His expertise and encouragement have been invaluable, steering me through challenges and helping me achieve clarity in my work. I am equally thankful to **Prof. Sajjad Hussain** for his dedicated mentorship and relentless inspiration during my PhD journey, whose wisdom has illuminated my academic path.

Special thanks also go to **Prof. Muhammad Imran**, my lab director and my source of inspiration, for his visionary leadership and invaluable guidance, which have significantly shaped the direction and scope of this work. His ability to foster an environment of innovation and inquiry has greatly enriched my research experience.

I acknowledge **Dr. Ahsan Raza Khan** for his invaluable guidance during the early stages of my PhD, whose insights helped me navigate the complexities of my research. Additionally, I am grateful to all my lab fellows, whose camaraderie and collaboration have made this experience not only enriching but also enjoyable.

My deepest appreciation goes to my siblings, **Dr. Tariq Manzoor**, **Dr. Muhammad Nasir Manzoor**, **Mr and Mrs. Saifullah Khan**, **Mr. Tahir Manzoor**, **Engr. Abaidullah Salfi**, and **Dr. Sanaullah Manzoor**, for their unwavering motivation, encouragement, and belief in my abilities. Their support has been a pillar of strength, inspiring me to overcome obstacles and strive for excellence. I also like thank my wife **Hira Habib** for her support in this journey.

All achievements come from **Allah Almighty** alone, and I remain humbly thankful for His countless blessings.

Declaration

University of Glasgow
College of Science & Engineering
Statement of Originality

Name: Habib Ullah Manzoor

I certify that the thesis presented here for examination for a PhD degree in the University of Glasgow is solely my own work except where I have indicated that it is the work of others (in which case the nature and extent of any work carried out jointly by me and another person is clearly identified in it) and that the thesis has not been edited by a third party permitted without permission by the University's PGR Code of Practice.

I declare that this thesis has been produced in accordance with the University of Glasgow's Code of Good Practice in Research.

I acknowledge that this thesis has been raised regarding good research practice because of the need to maintain the integrity of the research and to minimize the potential for any investigation of the issues.

Signature:

_____

Date: 19/08/2025

Contents

Abstract i

Acknowledgements iii

Declaration iv

Publications during PhD xiii

List of Abbreviations xv

List of Symbols xvii

1 Introduction 1

1.1 Motivation . . . 3
1.2 Problem Statement . . . 4
1.2.1 C1: Model Attacks . . . 5
1.2.2 C2: Robust Aggregation . . . 6
1.2.3 3: Communication and Computational Efficiency . . . 6
1.2.4 C4: Data Heterogeneity . . . 7
1.3 Aims and Objectives . . . 8
1.4 Contributions . . . 8
1.4.1 Adversarial Attacks . . . 8
1.4.2 Defence Frameworks . . . 9
1.4.3 Data Heterogeneity . . . 10
1.4.4 Communication Efficiency . . . 11
1.4.5 Adversarial Attack Mitigation with Decentralized FL . . . 11
1.5 Thesis Organization . . . 12

2 Literature Survey 14

2.1 Model Training Process of FL . . . 15
2.2 Categories of FL . . . 17
2.2.1 Data partitioning based categories . . . 17

2.2.2 System Architecture Based Categories . . . 18
2.2.3 Operation Strategies Based Categories . . . 19
2.3 Applications of FL . . . 20
2.3.1 FL Assisted Load Forecasting . . . 22
2.4 Limitations and Implementation Challenges in FL . . . 23
2.5 Security and Privacy Risks . . . 24
2.5.1 Threat Modelling in Distributed Systems . . . 25
2.5.2 Data FL Attacks . . . 26
2.5.3 Model FL Attacks . . . 27
2.5.4 Privacy FL Attacks . . . 28
2.5.5 Gap Analysis . . . 29
2.5.6 Future Directions and Proposed Solutions . . . 29
2.6 Defense Frameworks in FL . . . 30
2.6.1 Gap Analysis . . . 32
2.6.2 Future Directions and Proposed Solutions . . . 32
2.7 Communication and Computational Efficiency in FL for Distributed Applications 33
2.7.1 Model Compression Techniques . . . 33
2.7.2 Communication Optimization . . . 34
2.7.3 Gap Analysis . . . 34
2.7.4 Future Directions and Proposed Solutions . . . 34
2.8 Heterogeneity in FL for Distributed Applications . . . 35
2.8.1 Gap Analysis . . . 36
2.8.2 Future Directions and Proposed Solutions . . . 36
2.9 Performance metrics . . . 37
2.9.1 Loss functions . . . 37
2.9.2 Energy Consumption: . . . 37
2.9.3 Communication Cost . . . 38
2.9.4 Levene's Test . . . 38
2.10 Summary of Literature Review, Research Gap and Link with Challenges . . . 39

3 Attack Strategies in Distributed Systems 41
3.1 Performance Degrading Attacks . . . 42
3.1.1 Experiments and Results . . . 45
3.2 Stealth Communication Round Attack (Fed-CRA) . . . 48
3.2.1 Experimental Results . . . 52
3.3 Discussion . . . 57
3.4 Concluding Remarks . . . 57

4 Novel Attack Resolution Frameworks **59**
4.1 Federated Random Layer Aggregation 60
4.1.1 Experiments and Results 63
4.1.2 Computational Efficiency 65
4.1.3 Discussion 67
4.2 LBAA-FedAvg 69
4.2.1 Experiments and Results: 71
4.3 Federated Incentivized Averaging (Fed-InA) 73
4.3.1 Experiments and Results 75
4.3.2 Discussion 78
4.4 Concluding Remarks 81

5 Novel Framework for Data Heterogeneity in FL **83**
5.1 FedBranched 84
5.1.1 Motivation 84
5.1.2 Methodology 85
5.1.3 Simulation Setup 85
5.1.4 Experiments and Results 88
5.1.5 Design Insights: 89
5.2 Adaptive Single Layer Aggregation (ASLA) 91
5.2.1 Motivation 92
5.2.2 Architecture 92
5.2.3 Quantization Analysis 93
5.2.4 Experiments and Results 94
5.2.5 Findings and Insights 99
5.3 Comparison and Discussion of FedBranched and ASLA 101
5.3.1 FedBranched vs. ASLA 101
5.3.2 When to Use Each Framework 102
5.3.3 Summary Comparison 103
5.4 Concluding Remarks 104

6 DRLA: A Decentralised Defence Framework for FL **105**
6.1 Decentralized FL (DFL) 107
6.1.1 Communication Methods 107
6.1.2 Design Consideration 108
6.1.3 DFL with Line Network Communication Topology 108
6.1.4 DFL with Ring Network Communication Topology 110
6.1.5 DFL with Bus Network Communication Topology 110
6.2 Experiments and Results 110

6.2.1 Baseline Results 110
6.2.2 Communication Cost 111
6.2.3 Effect of Adversarial Attack 114
6.3 Decentralised Random Layer Aggregation (DRLA) 115
6.4 Concluding Remarks 117

7 Conclusion and Future Work **119**
7.1 Summary of Contributions 119
7.2 Limitations of the Research 122
7.2.1 Tradeoffs Made in the Research 123
7.3 Future Work 124
7.3.1 Combining Presented Techniques 124
7.3.2 Evaluation on Other Domains 124
7.3.3 Additional Areas for Future Work 124

List of Tables

2.1 A summary of state of the art stealth attacks in federated systems. 28
2.2 Defense Frameworks and Privacy Techniques in FL 31

5.1 Matrix of p-values of Levene's test for Data 1 87
5.2 Comparison of MAPE between Traditional FL and FedBranch. 89
5.3 Average client MAPE of step by step aggregation 97
5.4 Only a single layer aggregation 97
5.5 Sizes of Different Layers of Local Model 100
5.6 Summary Comparison of FedBranched and ASLA 103

7.1 Comparative Strengths of Frameworks 121

List of Figures

2.1 Overview of FL. 15
2.2 Types of FL. 19
2.3 Type of defense strategies. 30

3.1 Overview of dataset 1. 46
3.2 MAE during FL training process of global model on Dataset 1. 47
3.3 Examples of all attacks include: actual and attacked weights. 49
3.4 Effect of different model posing attacks when client 1 was subjected to attack. . 50
3.5 Sample of the dataset 2. 53
3.6 Baseline results for dataset 2. 55
3.7 A sample of the local model of client one's weight distribution, both as it is and as it was attacked. 55
3.8 Communication rounds (red) and MAPE (blue) resulting from Fed-CRA global model training. 56

4.1 Schematic diagram of FedRLA process. 61
4.2 MAE of every client following 20 rounds of communication. 64
4.3 FedRLA and FedAvg are compared under various hostile attacks. 66
4.4 Comparison of Different Robust FL Frameworks in Terms of Resource Utilization 68
4.5 Block diagram of LBAA-FedAVG). 70
4.6 Illustration of the clustering criterion for LBAA-FedAvg, where C_1 and C_2 denote two clusters, D_1 and D_2 represent the maximum Euclidean distances within each cluster, and D_3 is the distance between the centroids of C_1 and C_2. 70
4.7 Effect of LBAA-FedAvg on average client MAPE 72
4.8 Resource utilisation of FedAvg and LBAA-FedAVG. 73
4.9 Block diagram of Fed-InA. 74
4.10 Effect of Fed-InA on Fed-CRA. 77
4.11 Resource utilisation of Fed-InA. 78

4.12 Comparison of different state-of-the-art robust aggregation frameworks with Fed-CRA and Fed-InA. The figure illustrates the number of communication rounds required by each framework under adversarial conditions. Fed-InA significantly reduces the number of rounds compared to Fed-CRA, demonstrating its robustness and efficiency. 79
4.13 Visual representation of the heterogeneous dataset. The figure shows the data distribution across different clients, highlighting the significant deviation in client 10's data pattern. 80
4.14 Effect of Fed-InA on the heterogeneous dataset. The figure illustrates the stabilization of the global model's performance as Fed-InA adjusts its incentive mechanism to mitigate the influence of client 10. 81

5.1 Framework of FedBranched. 86
5.2 Dataset with nine substations. 88
5.3 Baseline results representing MAPE of all clients during training process of global model. 88
5.4 Graphical elaboration of FedBranch on considered example. 90
5.5 FL results of 2nd round of clustering after 30 communication rounds. 91
5.6 Adaptive framework of ASLA. 93
5.7 Block diagram of early stopping criteria. 94
5.8 A sample of used dateset containing ten different clients 95
5.9 MAPE of all local clients in the baseline simulation. The x-axis represents the MAPE of all clients, while the y-axis represents the communication rounds. . . 96
5.10 MAPE of all clients during training of global model when only first layers were aggregated. The x-axis represents the MAPE of all clients, while the y-axis represents the communication rounds. 98
5.11 The effect of quantization on communication rounds and on average client MAPE. The x-axis represents the MAPE of all clients, while the y-axis represents the communication rounds. 98
5.12 The effect of stopping criteria on communication rounds and on average client MAPE. The x-axis represents the MAPE of all clients, while the y-axis represents the communication rounds. 99
5.13 Communication costs for different layers of a three-layered neural network, when only single layer was used for aggregation 102

6.1 Line, Ring and Bus ring topologies used in DFL for load forecasting 109
6.2 Client-wise MAE for all clients during the global model's training phase. . . . 111
6.3 Client-wise MAE observed during the global model's training in the context of DFL. 112

6.4 Communication cost of CFL and DFL frameworks. 113
6.5 Comparison of actual and predicted curves in the CFL bus topology during a model flipping attack targeting Client 1. 114
6.6 Comparison of actual and predicted curves in the DFL line topology during a model flipping attack targeting Client 1. 114
6.7 Comparison of actual and predicted curves in the DFL ring topology during a model flipping attack on Client 1. 115
6.8 Comparison of actual and predicted curves in the DFL bus topology during a model flipping attack targeting Client 1. 115
6.9 Analysis of average client MAE under model flipping attack targeting Client 1: A comparison across CFL, DFL, and DRLA frameworks. 117

Publications during PhD

Journals

1. **Manzoor, Habib Ullah** Kamran Arshad, Khaled Assaleh, Ahmed Zoha,'Novel Stealth Communication Round Attack and Robust Incentivized Federated Averaging for Load Forecasting', in IEEE transactions sustainable computing, 2025
   https://ieeexplore.ieee.org/document/11004057

2. **Manzoor, Habib Ullah**, Atif Jafri, and Ahmed Zoha. "Adaptive single-layer aggregation framework for energy-efficient and privacy-preserving load forecasting in heterogeneous federated smart grids." Internet of Things 28 (2024): 101376.: 101376.
   https://doi.org/10.1016/j.iot.2024.101376

3. **Manzoor, Habib Ullah**, Attia Shabbir, Ao Chen, David Flynn, and Ahmed Zoha. "A survey of security strategies in federated learning: Defending models, data, and privacy." Future Internet 16, no. 10 (2024): 374.
   https://doi.org/10.3390/fi16100374

4. **Manzoor, Habib Ullah**, Sajjad Hussain, David Flynn, and Ahmed Zoha. "Centralised vs. decentralised federated load forecasting in smart buildings: Who holds the key to adversarial attack robustness?" Energy and Buildings 324 (2024): 114871.
   https://doi.org/10.1016/j.enbuild.2024.114871

5. **Manzoor, Habib Ullah**, Ahsan Raza Khan, David Flynn, Muhammad Mahtab Alam, Muhammad Akram, Muhammad Ali Imran, and Ahmed Zoha. "Fedbranched: Leveraging federated learning for anomaly-aware load forecasting in energy networks." Sensors 23, no. 7 (2023): 3570. https://doi.org/10.3390/s23073570

6. **Manzoor, Habib Ullah**, Naveed Rao, Sajjad Hussain, Ahmed Zoha.'Trustworthy Distributed Load Forecasting in Resource-Limited Smart Grids and Buildings via Random Layer Aggregation' Alexandria Engineering Journal (2025) (First Revision).

Conferences

1. **Manzoor, Habib Ullah**, Ahsan Raza Khan, Tahir Sher, Wasim Ahmad, and Ahmed Zoha. "Defending federated learning from backdoor attacks: Anomaly-aware fedavg with layer-based aggregation." In 2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1-6. IEEE, 2023. 10.1109/PIMRC56721.2023.10293950

List of Abbreviations

AI Artificial Intelligence

AMI Advanced Metering Infrastructure

ANN Artificial Neural Network

ARIMA Auto-Regressive Integrated Moving Average

ARMA Auto-Regressive Moving Average

ASLA Adaptive Single Layer Aggregation

CRA Completely Random Attack

DFL Decentralized Federated Learning

DDoS Distributed Denial-of-Service

DP Differential Privacy

FedAvg Federated Averaging

Fed-CRA Federated Communication Round Attack

Fed-InA Federated Incentivized Averaging

FedRLA Federated Random Layer Aggregation

FL Federated Learning

GDPR General Data Protection Regulation

HFL Horizontal Federated Learning

HMM Hidden Markov Model

IID Independent and Identically Distributed

IoT Internet of Things

LBAAFedAvg Layer-Based Anomaly Aware Federated Averaging

LSTM Long Short-Term Memory

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

MFA Model Flipping Attack

MITM Man-in-the-Middle

MSE Mean Squared Error

non-IID Non-Independent and Identically Distributed

PA Perturbed Attack

PRA Partially Random Attack

ReLU Rectified Linear Unit

STLF Short-Term Load Forecasting

UAV Unmanned Aerial Vehicle

VFL Vertical Federated Learning

5G 5th Generation Mobile Network

IEEE Institute of Electrical and Electronics Engineers

List of Symbols

R Number of communication rounds

α Energy consumption per second

t Transmission time

β Energy consumption per kilobyte

D Data size transmitted

E_{com} Energy consumption for transmitting local models

w_0 Initial global model weights

K Number of clients

n Total number of samples across selected clients

w_t^k Local model weights after training on client k

η Learning rate controlling the step size in weight updates

$\nabla F_k(w_t)$ Gradient of the local objective function on client k

w_{t+1}^L Aggregated weight for layer L in the global model

n_k Number of samples at client k

$w_{k,L}$ Weights of layer L of the local model on client k

Rand Random seed used in model updates

L Number of layers in the neural network model

$D1, D2, D3$ Euclidean distances used in clustering criteria

$C1, C2$ Clusters formed during the aggregation process

P Set of peers in decentralized communication

$w_{k,t,lrand}$ Weight of randomly selected layer $lrand$ from client k at time t

G_t Global model at time t

$w_{k,t+1,l}$ Updated local model weights for layer l at time $t+1$

$w_{k,t,l}$ Local model weights for layer l at time t

$w_{k,t+1,lrand}$ Aggregated weight for selected layer $lrand$ at time $t+1$

Chapter 1

Introduction

The rise of IoT (Internet of Things) devices has dramatically transformed the way people, businesses, and systems interact with technology, fostering a new era of connectivity and automation. While IoT encompasses a vast network of interconnected devices, sensors, and systems that communicate and exchange data over the internet with minimal human intervention [1], this increased connectivity introduces significant challenges related to data privacy, security, and the need for robust anomaly detection. The proliferation of IoT devices, including smart home appliances, wearable health monitors, industrial sensors, and autonomous vehicles, has led to the generation of massive amounts of sensitive data. This data often contains personal or proprietary information, making it a target for potential breaches and cyberattacks. Additionally, the distributed nature of IoT systems requires efficient and reliable mechanisms to detect and respond to anomalies in real-time, ensuring the integrity and reliability of these systems. As a result, the focus of this thesis is on developing advanced federated learning frameworks that address these critical concerns, particularly in the context of smart energy networks where privacy, security, and anomaly detection are paramount.

Despite the transformative potential of the IoT in domains such as energy management, healthcare delivery, and the development of smart cities, several critical challenges hinder its effective integration with machine learning systems:

1. **Data Privacy and Security Risks:** IoT devices frequently collect sensitive and high-frequency data, such as detailed energy consumption patterns and personal health metrics. This centralised accumulation of data poses significant risks, as it can become a target for cyberattacks, leading to potential breaches of privacy and violations of regulatory frameworks like GDPR. Moreover, the diversity of data sources exacerbates the complexity of implementing robust security measures, making it imperative for organizations to prioritize data encryption and secure transmission protocols to safeguard user information [2].

2. **Bandwidth and Resource Constraints:** The need for frequent model updates in centralized learning approaches leads to excessive communication overhead, particularly chal-

lenging for low-power and bandwidth-limited edge devices. For instance, transmitting large volumes of data for model training can strain network resources and result in latency issues that are detrimental to real-time applications, such as health monitoring or smart grid management. Therefore, developing lightweight algorithms that require minimal data transmission while maintaining predictive accuracy is critical for the viability of IoT systems [3].

3. **Data Heterogeneity:** IoT deployments often involve a wide array of devices that generate diverse, non-IID (Independent and Identically Distributed) data. Variations in usage patterns, sensor specifications, and environmental conditions contribute to this heterogeneity, complicating the process of training globally consistent machine learning models. The challenge lies in effectively aggregating and harmonizing data from these disparate sources to ensure that the models can learn from a comprehensive dataset while accounting for the unique characteristics of each data stream [4].

4. **Distributed Models and Aggregation:** Aggregating models in a distributed IoT environment introduces unique challenges. The need to combine model updates from numerous heterogeneous devices requires robust aggregation mechanisms that can handle non-IID data distributions and potential adversarial updates. Traditional aggregation methods like simple averaging may not be effective in such settings, as they can lead to model drift or poor generalization. Developing efficient and secure aggregation techniques that can operate under communication constraints while preserving model integrity is crucial. These techniques must also balance the trade-off between communication efficiency and model accuracy, ensuring that the aggregated model performs well across all participating devices despite their diversity.

Despite these drawbacks, the IoT landscape continues to expand, driven by innovations in AI, 5G technology, and edge computing. Addressing these challenges through robust security frameworks, standardized protocols, sustainable practices, and ethical guidelines will be essential to realising the full potential of IoT in creating a more connected and intelligent world.

The challenges faced by IoT devices, such as data privacy concerns, cybersecurity risks, and the need for personalised ML, highlight the limitations of traditional, centralised approaches to managing and analysing data. One application of IoT devices can be found in energy networks, where they monitor and optimize energy consumption, manage distributed generation, and facilitate demand response. However, these applications raise significant data privacy concerns, as sensitive information about energy usage patterns may be exposed.

Cybersecurity risks are particularly pronounced in energy networks, where interconnected IoT devices can create vulnerabilities; a breach in one device can compromise the entire system. Additionally, the need for personalised ML is critical to tailoring energy solutions to the unique

requirements of different consumers and environments. Traditional, centralised approaches often fall short in addressing these challenges, as they rely on aggregating large volumes of sensitive data in a single location, increasing the risk of breaches and limiting the ability to develop personalized models.

A promising solution to address these issues lies in federated learning (FL), a distributed approach that enables IoT devices to collaboratively train ML models without sharing raw data. By keeping data localized and transmitting only aggregated model updates, FL significantly enhances privacy and reduces the risk of breaches. It also supports the development of personalized models tailored to specific IoT environments while maintaining scalability and efficiency. Additionally, this approach fosters interoperability by allowing diverse devices to contribute to a shared, cohesive learning process, paving the way for a more secure and adaptive IoT ecosystem.

1.1 Motivation

Centralised ML has achieved impressive results across domains such as healthcare, finance, and energy, but it is increasingly challenged by concerns over data privacy, communication costs, and systemic vulnerabilities. These limitations are particularly evident in IoT applications, where vast amounts of distributed data are generated by edge devices with limited bandwidth, storage, and processing capabilities. Centralised training paradigms typically require raw data transmission to a central server, violating privacy constraints and creating single points of failure [5]. Recent attacks, such as those on Colonial Pipeline in 2021 [6], Indian State Load Dispatch Centres (SLDCs) in 2022 [7], a DDoS attack on a Lithuanian Energy Company in 2022 [8], and a ransomware attack on Encino Energy in 2022 [9], have exposed the vulnerabilities of centralised systems. The impact of these incidents would likely have been minimized with a distributed approach. For instance, the X-Force Threat Intelligence Index for 2023 reported that the energy sector experienced a notable increase in cyberattacks in 2022, accounting for 10.7% of all recorded incidents and making it the fourth most targeted industry [10].

FL has emerged as a promising alternative. It enables collaborative training across decentralized data sources without sharing raw data, thereby preserving privacy, improving scalability, and reducing network congestion [11]. However, FL also introduces new challenges such as vulnerability to adversarial attacks, communication bottlenecks, and difficulty handling non-IID data that must be addressed to ensure secure and reliable deployment.

To explore and address these challenges, this thesis applies FL to short-term load forecasting (STLF) in smart energy networks, a domain characterized by non-uniform data distributions, stringent privacy requirements, and adversarial threat exposure. STLF is essential for efficient energy resource planning, renewable integration, and grid stability [12]. Energy is a fundamental driver of economies and societies, but it also significantly contributes to global warming, being responsible for nearly two-thirds of greenhouse gas emissions [13]. Various nations and organi-

zations have established ambitious targets to address this pressing environmental concern, such as the European Union's aim to reduce emissions by 40% and improve energy efficiency by 27% by 2030 [14]. Despite these efforts, the growing demand for energy highlights the importance of effective energy management.

The implementation of advanced metering infrastructure and the widespread deployment of smart meters enable utility companies to collect and record energy usage data at intervals as short as one hour, covering individual buildings and households. For example, in the United Kingdom, over 15 million smart meters are currently in operation in residential and commercial properties [15]. Recent research has focused on predicting the short-term load of individual buildings to facilitate decentralized monitoring and control of power systems, driven by the integration of intermittent renewable energy sources [16]. This approach is particularly important given the dispersed nature of these resources. However, forecasting individual household load presents challenges due to the unpredictable behaviors of residents [17]. The need for customized ML models for each meter has increased computational demands, making this task more challenging and often impractical.

By demonstrating the effectiveness of novel FL-based security frameworks within this domain, the thesis aims to establish both the generalizability and practicality of the proposed solutions in real-world, high-stakes environments.

1.2 Problem Statement

ML systems deployed in large-scale IoT environments encounter a myriad of escalating challenges, particularly concerning data privacy, communication bottlenecks, and vulnerabilities to security breaches. These issues are critically pronounced in domains where sensitive and distributed data is generated at the edge, such as smart homes, healthcare, and industrial applications. FL has emerged as a promising privacy-preserving solution, enabling edge devices to collaboratively train ML models without the need to share raw data, thus mitigating privacy risks.

Despite its advantages, FL introduces several technical challenges that must be addressed to ensure its effective deployment:

Challenge C1: Need of new Adversarial Attacks. FL is susceptible to adversarial participants who may engage in model poisoning or stealth attacks. These malicious actions can manipulate the training process, leading to a gradual degradation of model performance without immediate detection. Most of the available attacks in the literature only damage the accuracy of the system; there is a lack of new types of attacks that can challenge FL systems in innovative ways.

Challenge C2: Need for Lightweight Aggregation Mechanisms. The design of robust and lightweight

aggregation algorithms is crucial, especially in edge environments where computational resources are limited. Efficient aggregation methods must balance the need for accurate model updates with the constraints imposed by the hardware capabilities of edge devices.

Challenge C3: Increased Communication and Computational Costs. Frequent synchronization among devices in a federated setting can lead to significant increases in communication and computational costs. This strain on network bandwidth and device energy budgets can hinder the scalability of FL systems, particularly in resource-constrained environments. Strategies to minimize communication overhead while maintaining model accuracy are essential.

Challenge C4: Data Heterogeneity Across Clients. The variability in data distribution among clients stemming from differences in usage patterns, device capabilities, and environmental conditions poses a fundamental challenge to achieving model convergence and fairness. Addressing data heterogeneity is vital to ensure that the federated model can generalize well across diverse user scenarios.

To tackle these challenges, this thesis delves into the foundational issues surrounding FL, with a keen focus on enhancing security, efficiency, and robustness in both adversarial and resource-constrained settings. The proposed methodologies are implemented and evaluated on STLF applications within smart energy networks. This domain exemplifies the critical need for FL's privacy and efficiency benefits, yet it remains highly susceptible to the aforementioned challenges. The overarching goal is to devise robust, communication-efficient FL frameworks that ensure consistent accuracy and scalability for IoT-based load forecasting applications, thereby advancing the field and contributing to secure and efficient energy management solutions.

1.2.1 C1: Model Attacks

FL enhances privacy by keeping data on local devices; however, it remains vulnerable to adversarial attacks, such as data poisoning and model poisoning. Malicious participants can subtly manipulate their local updates to corrupt the global model without detection [18]. Unlike traditional centralized learning, where anomaly detection techniques can monitor incoming data centrally, FL operates in a decentralized manner, making it difficult to track and mitigate such stealth attacks.

Stealth attacks can take multiple forms, including model poisoning and backdoor attacks. In model poisoning, adversaries introduce imperceptible modifications to their updates, causing the global model's accuracy to degrade gradually over time. Backdoor attacks allow adversaries to embed specific vulnerabilities that remain dormant until triggered by certain inputs, which is particularly concerning in high-stakes applications like healthcare and energy, where even minor

model deviations can lead to catastrophic failures. Chapter 3 presents five different model attacks, four of which are non-stealthy and one stealthy in nature, used to test the robustness of distributed STLF. Addressing these stealth attacks requires the development of adversary-resistant aggregation techniques, secure multiparty computation, and differential privacy mechanisms to detect and mitigate threats without compromising efficiency.

1.2.2 C2: Robust Aggregation

To counteract adversarial manipulation and unreliable updates, robust aggregation techniques are essential in FL. Traditional aggregation methods, such as FedAvg, assume that all participating clients provide honest and high-quality updates, an assumption that does not hold in real-world deployments [19]. The presence of adversarial clients necessitates aggregation strategies that can filter out anomalous updates while maintaining model convergence.

Current approaches to robust aggregation include trimmed mean, Krum, and adaptive federated averaging, which focus on detecting and excluding malicious updates [20]. However, these methods often introduce additional computational overhead, making them less suitable for resource-constrained IoT environments. Additionally, they may struggle against sophisticated adversaries who craft updates to bypass anomaly detection techniques. More advanced solutions, such as clustering-based aggregation, reputation-based weighting, and blockchain-enabled FL, are being explored to strike a balance between robustness and efficiency. These techniques need to be optimized to handle both Byzantine adversaries and system failures while ensuring fair participation from all clients. Chapter 4 presents three novel robust aggregation frameworks, Federated Random Layer Aggregation (FedRLA), Layer Based Anomaly Aware Federated Averaging (LBAA-FedAvg) and Federated Incentivised Averaging (Fed-InA) designed for different situations, depending upon the type of attack in distributed STLF.

1.2.3 3: Communication and Computational Efficiency

Despite reducing raw data transmission, FL still incurs significant communication costs due to frequent model updates between edge devices and the central server. This overhead is particularly problematic in resource-constrained IoT environments where bandwidth is limited [21]. Unlike traditional centralized learning, where all computation happens on a powerful server, FL requires edge devices to perform local training, which may exceed their computational capabilities. The cost of transmitting high-dimensional model updates can lead to network congestion, increased latency, and higher energy consumption.

Several techniques have been proposed to improve communication efficiency, including model compression, quantization, update sparsification, and local update accumulation [22]. However, these approaches introduce trade-offs: while reducing communication overhead, they may lead to loss of precision, slower convergence, or increased vulnerability to adversarial at-

tacks. Similarly, computational efficiency is a challenge since many IoT devices have limited processing power. Lightweight models, knowledge distillation, and edge-cloud hybrid FL architectures are promising solutions, but further optimization is needed to balance accuracy, latency, and resource consumption effectively.

In this work, two different frameworks, FedRLA (Chapter 4) and Adaptive Single Layer Aggregation (ASLA) (Chapter 5), were specifically designed for resource-constrained IoT environments where communication bandwidth and computational resources are limited. These frameworks aim to address the unique challenges faced in such scenarios, particularly in the context of short-term load forecasting (STLF) for smart energy networks. FedRLA reduces communication overhead by aggregating only a single randomly selected layer of the neural network during each communication round, making it suitable for environments with restricted bandwidth and high communication costs. ASLA further enhances efficiency by selectively aggregating a single layer based on client capabilities and incorporating quantization techniques to minimize data transmission sizes. Both frameworks are optimized to operate within the constraints of edge devices with limited processing power and memory, ensuring that they can be effectively deployed in real-world IoT settings for energy forecasting and management.

1.2.4 C4: Data Heterogeneity

IoT devices generate diverse, non-identically distributed (non-IID) data, posing a major challenge to FL. Unlike centralized ML, where data is pooled to create uniform training distributions, FL relies on local training with varying data distributions [23]. This heterogeneity can lead to biased models and uneven performance across devices [24]. Some clients may contribute disproportionately to model updates, while others may fail to generalize effectively. Data heterogeneity in FL can be attributed to several factors, including differences in sensor hardware, environmental conditions, and user behavior. Standard FL approaches assume IID data distributions, causing models to generalize poorly when trained on non-IID datasets. To address this issue, AI techniques such as federated multi-task learning, clustering-based FL, and personalized FL have been proposed. These methods leverage AI to adapt models to local data characteristics while maintaining global model performance. However, these AI-driven approaches introduce new challenges, including increased computation, storage, and coordination overhead. Moreover, fairness concerns arise when certain clients benefit more than others due to differences in data quality and quantity. Achieving fairness while maintaining overall model accuracy remains an open research problem. In Chapter 5, two AI-enhanced frameworks, FedBranched and ASLA, are presented to efficiently tackle the heterogeneous data in distributed STLF. These frameworks utilize advanced AI techniques to mitigate the impact of non-IID data distributions and improve model fairness and accuracy across all clients.

1.3 Aims and Objectives

In light of the above discussion, the aims and objectives are as follows:

1. **Develop a Robust and Lightweight Defense Framework:** Develop anomaly detection and mitigation techniques capable of identifying malicious or erratic clients in FL environments. The framework should ensure minimal computational overhead and enable deployment on resource-constrained IoT devices.

2. **Construct an Energy and Communication-Efficient FL Framework:** Develop a FL architecture that maximizes resource efficiency by reducing energy usage and communication overhead. This encompasses the implementation of techniques such as model quantization, layer-wise aggregation, and early stopping criteria to improve the overall training process efficiency.

3. **Design a Lightweight Framework for Heterogenous Clients:** Create an adaptive and scalable FL framework that maintains reliable performance across non-IID datasets and heterogeneous clients. This includes developing personalized aggregation schemes and flexible model update protocols

1.4 Contributions

This thesis contributes to the field of secure FL through the design, implementation, and evaluation of multiple adversarial attack and defence frameworks. These contributions are validated within the context of privacy-conscious STLF in smart energy networks, a domain that exemplifies the challenges of data heterogeneity, constrained communication, and adversarial risk. The key contributions of this work are structured into the following categories.

1.4.1 Adversarial Attacks

This thesis presents a series of model poisoning strategies, including the Federated Communication Round Attack (Fed-CRA), aimed at inflating communication costs while evading traditional anomaly detection systems. These attacks expose previously overlooked vulnerabilities in FL systems, allowing adversaries to degrade efficiency without compromising model accuracy. To thoroughly evaluate the robustness of the FL framework, five distinct adversarial attacks were employed. These attacks target the training process, resulting in degraded model performance and compromised accuracy, and were generated at the client level to simulate real-world scenarios where malicious clients may seek to undermine the learning process.

Among these five attacks, four focus on reducing the efficiency of the machine learning model and are categorized as follows:

- **Partially Random Attack (PRA):** Randomly alters a subset of model parameters, introducing noise that diminishes the model's predictive capability.
- **Completely Random Attack (CRA):** Introduces random values across the entire model, significantly disrupting the training process.
- **Model Flipping Attack (MFA):** Flips the gradients sent by clients, effectively reversing the learning direction and leading to poor model performance.
- **Perturbed Attack (PA):** Adds small perturbations to model updates, which can accumulate over multiple rounds, resulting in substantial degradation of model performance.

The fifth attack, the **Fed-CRA**, is particularly innovative and strategic. Its objective is to increase the number of communication rounds between the server and clients while maintaining model accuracy. By manipulating communication frequency, this attack creates inefficiencies in the learning process, revealing vulnerabilities in communication protocols.

These attacks were implemented within a distributed energy network context, specifically for STLF. This setting not only illustrates the practical implications of the attacks but also offers insights into how FL can be enhanced to withstand such adversarial scenarios.

1.4.2 Defence Frameworks

The defense frameworks are designed to increase the robustness and resilience of the ML model trained under adversarial attacks, particularly when one or more clients are compromised. The following defense frameworks were developed to address various attack scenarios effectively:

1. **Federated Random Layer Aggregation (FedRLA):** In this framework, a novel aggregation method is introduced to mitigate the adversarial effects of anomalous clients. Instead of aggregating all layers of the model, this approach randomly selects only a single layer at each communication round for aggregation. This strategy not only reduces the impact of adversarial attacks by limiting the influence of potentially corrupted data but also improves communication efficiency by minimizing the amount of data transmitted during each round. By focusing on a single layer, the framework allows for quicker convergence and reduces the overall resource consumption during training.
2. **Layer-Based Anomaly Aware Federated Averaging (LBAAFedAvg):** This aggregation framework is specifically designed to address partial adversarial attacks, where only certain layers of a model may be compromised. It employs advanced ML techniques to detect which layers of the neural networks are being targeted by adversarial clients during the aggregation process. By identifying and isolating attacked layers, LBAAFedAvg can preserve the integrity of the overall model. The resource utilization of the LBAAFedAvg

framework is comparable to traditional frameworks that do not implement anomaly detection, thus ensuring that performance is not significantly compromised while enhancing security.

3. **Federated Incentivized Averaging (Fed-InA):** This framework targets stealth attacks in distributed systems, which are insidious as they do not create overt disturbances during training. If left undetected, these attacks can lead to significant degradation in model performance over time. Fed-InA introduces a novel scoring mechanism that evaluates clients based on their contribution to the model's accuracy and reliability. Good clients are rewarded, while bad clients are penalized and eventually removed from the aggregation process. This incentivization encourages clients to act honestly and contributes to the overall integrity and performance of the FL system.

1.4.3 Data Heterogeneity

Data heterogeneity in distributed systems refers to the variations in data distributions across different clients, which can significantly hinder the convergence of ML models during the training process. This issue arises because clients may possess data that is non-IID (Independent and Identically Distributed), leading to challenges in achieving a consensus model that accurately generalizes across diverse data sets. To mitigate the negative effects of data heterogeneity, the following two frameworks were presented:

1. **FedBranced:** This framework monitors the convergence of the trained model by assessing the performance metrics at each iteration. If the model is found to be not converging, FedBranced employs advanced ML techniques to categorize clients into two distinct branches based on their data characteristics. This allows for the training of two different models concurrently, tailored to the specific data distributions of each branch. If the models still do not converge after a predetermined number of iterations, the process is repeated, continuously adapting the branches until all models achieve convergence. This iterative approach ensures that the framework can effectively handle diverse data distributions, ultimately improving model performance and robustness.

2. **Adaptive Single Layer Aggregation:** This framework simplifies the aggregation process by utilizing only a single layer of neural networks for local clients. The selection of this layer is performed in an adaptive manner, taking into account the current performance of each layer across the clients. By focusing on a single layer, the framework reduces the complexity of model updates and accelerates the training process. Additionally, it incorporates quantization techniques to minimize the size of data transmitted, alongside stopping criteria that allow the system to halt training when certain performance thresholds are met. These features collectively enhance energy efficiency and reduce commu-

nication overhead, making the framework particularly suitable for resource-constrained environments.

1.4.4 Communication Efficiency

Communication efficiency is crucial in distributed ML systems, as it directly impacts the speed and resource consumption of the training process. Achieving communication efficiency is mainly accomplished by sending fewer parameters and limiting the number of iterations during model training. Two novel approaches were presented to effectively reduce communication costs:

1. **FedRLA:** In this framework, only a single layer of the neural network is aggregated from each client, and this layer is changed in each iteration. By restricting the communication to just one layer, FedRLA significantly reduces the number of parameters sent during the training process, minimizing bandwidth usage and communication latency. This method is particularly advantageous in scenarios where clients have limited connectivity or operate in environments with constrained resources. FedRLA has been shown to be 3.56 times more communication efficient than traditional methods that utilize all layers of the neural network, while still maintaining model accuracy and convergence speeds.

2. **ASLA:** The Adaptive Single Layer Aggregation (ASLA) framework utilizes only a single layer of the local model, which remains unchanged throughout the aggregation process. This stability allows for consistent updates and reduces the variability in communication overhead. Additionally, ASLA employs quantization techniques to decrease the size of the transmitted data and incorporates stopping criteria that enable the system to halt training when specific performance thresholds are met. This proactive approach further enhances efficiency by preventing unnecessary communication. ASLA has demonstrated to be 829.2 times more communication efficient than traditional methods, making it an ideal choice for resource-constrained environments and applications requiring rapid convergence.

1.4.5 Adversarial Attack Mitigation with Decentralized FL

Decentralized federated learning (DFL) leverages various peer-to-peer (P2P) communication topologies to train MLmodels without relying on a central server. This approach enhances privacy and reduces potential bottlenecks associated with centralized systems. In this framework, P2P topologies such as line, bus, and ring have shown distinct advantages when faced with adversarial attacks. Research indicates that these topologies limit the impact of attacks to only the clients directly involved, thereby isolating the threat. As the model transitions from line to

bus to ring configurations, the adversarial impact is significantly reduced. This progressive improvement highlights the resilience of decentralized systems compared to centralized FL, where adversarial attacks can compromise all clients simultaneously.

Moreover, further reduction in adversarial effects was achieved through the implementation of Decentralized Random Layer Aggregation (DRAL). This innovative method applies the principles of FedRLA in a decentralized manner, allowing for more robust aggregation of model updates while maintaining client confidentiality. By distributing the aggregation process across multiple clients, DRAL mitigates the risks associated with single points of failure and enhances the overall security of the learning process. The combination of P2P topologies and DRAL not only improves the model's resistance to adversarial attacks but also promotes a more efficient use of resources, making decentralized FL a compelling alternative for various applications in sensitive environments.

1.5 Thesis Organization

The rest of the thesis is organized into several chapters, each addressing a specific aspect of the research on securing intelligent networks using FL approaches for privacy-conscious anomaly detection. The following provides a brief overview of the structure and content of each chapter.

1. **Chapter 2: Literature Survey** This chapter provides a comprehensive review of the existing literature on FL, including its training process, categories, applications, limitations, and defence frameworks. It also discusses the specific challenges and opportunities in the context of load forecasting and highlights gaps in current research.

2. **Chapter 3: Attack Strategies in Distributed Systems** This chapter explores various attack strategies that can compromise FL systems, focusing on model poisoning attacks. It introduces several types of attacks, including Completely Random Attack (CRA), Partially Random Attack (PRA), Model Flipping Attack (MFA), Perturbed Attack (PA), and Federated Communication Round Attack (Fed-CRA). The chapter presents experimental results demonstrating the impact of these attacks on model performance.

3. **Chapter 4: Novel Attack Resolution Frameworks** This chapter proposes and evaluates several defense frameworks designed to mitigate the impact of adversarial attacks in FL systems. The frameworks include Federated Random Layer Aggregation (FedRLA), Layer-Based Anomaly Aware Federated Averaging (LBAAFedAvg), and Federated Incentivized Averaging (Fed-InA). The chapter presents experimental results demonstrating the effectiveness of these frameworks in enhancing security and model performance.

4. **Chapter 5: Novel Framework for Data Heterogeneity in FL** This chapter addresses the challenge of data heterogeneity in FL systems, particularly in the context of energy

networks. It proposes two frameworks, FedBranched and Adaptive Single Layer Aggregation (ASLA), to improve model convergence and performance by effectively handling diverse data distributions and optimizing communication efficiency. The chapter presents experimental results highlighting the effectiveness of these frameworks.

5. **Chapter 6: DRLA: A Decentralised Defence Framework for Robust and Efficient FL** This chapter explores the use of Decentralized FL (DFL) to mitigate adversarial attacks and enhance communication efficiency in FL systems. It compares DFL with traditional Centralized FL (CFL) and introduces the Decentralized Random Layer Aggregation (DRLA) framework. The chapter presents experimental results demonstrating the robustness and efficiency of DFL in various communication topologies.

6. **Chapter 7: Conclusion and Future Work** This chapter summarizes the key findings and contributions of the research. It discusses the implications of the proposed frameworks and methods for enhancing the security and efficiency of FL systems. The chapter concludes with suggestions for future work, including advanced anomaly detection techniques, dynamic data heterogeneity management, enhanced communication efficiency, and scalable decentralized FL.

Chapter 2

Literature Survey

FL has emerged as a transformative paradigm in machine learning, enabling collaborative model training while preserving data privacy. This chapter provides a comprehensive exploration of FL, detailing training processes, diverse applications, inherent challenges, and defense mechanisms. Through structured analysis, the potential of FL across various domains is highlighted, along with critical barriers to successful implementation. The contributions of this chapter are as follows:

1. The chapter begins with a comprehensive overview of the FL training pipeline in Section 2.1, which consists of five key phases: global model initialization, client selection and model distribution, local model training, layer-wise aggregation of local updates, and global model updates with convergence validation. The various categories of FL are discussed in Section 2.2, based on data partitioning (horizontal, vertical, transfer), system architecture (centralized, decentralized), and operational strategies (cross-device, cross-silo). These classifications highlight FL's adaptability across diverse deployment scenarios.

2. The applications of FL are extensive and particularly relevant to privacy-sensitive domains. Beyond healthcare, finance, and the Internet of Things (IoT), this chapter emphasizes FL's potential in energy systems, particularly for load forecasting, as discussed in Section 2.3. FL enables utilities to develop accurate load forecasting models without compromising consumer privacy, thereby addressing challenges in smart grid management and renewable energy integration. By collaboratively training models across distributed datasets, FL contributes to creating more responsive and efficient energy systems while maintaining data confidentiality.

3. Despite its advantages, FL faces several implementation challenges. Data heterogeneity (non-IID distributions) can introduce model bias, which is discussed in Section 2.8, targeting challenge C4. Additionally, the communication overhead from frequent model

updates increases bandwidth consumption. Limitations of edge devices, including computational power, memory, and energy constraints, further complicate deployment, as discussed in Section 2.7, targeting challenge C3. Privacy-security trade-offs remain a critical concern, with potential inference attacks threatening data integrity, as discussed in Section 2.5, targeting challenges C1 and C2. Balancing global model convergence with local personalization in non-IID settings presents additional complexities, as does managing client dynamics with transient connectivity and inconsistent participation patterns. Regulatory compliance and interoperability requirements also pose significant barriers to widespread adoption.

4. At the end, performance metrics are discussed in 2.9, which includes loss functions, energy consumption, communication cost, and Levene's test for data heterogeneity.

2.1 Model Training Process of FL

The FL training process involves the following five steps [18]: This process is graphically presented in Fig. 2.1

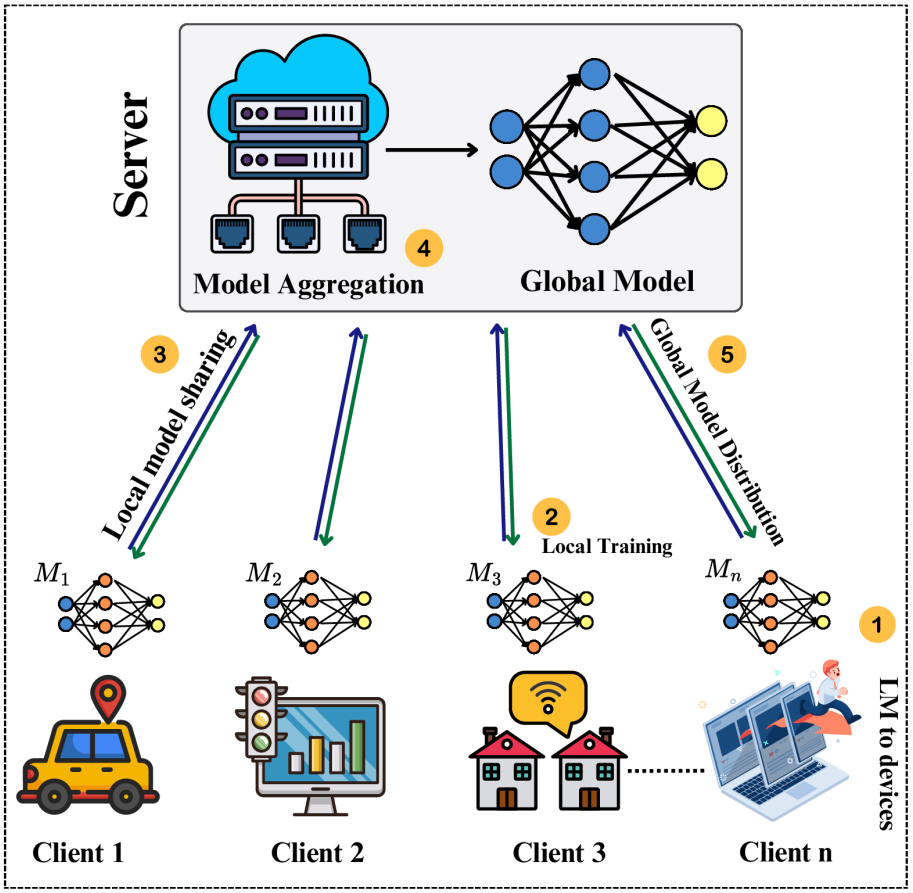


Figure 2.1: Overview of FL.

1. **Initialize Global Model:**
   The server initializes the global model w^0 with random weights or pre-trained values obtained from a similar task. This model serves as the starting point for all clients and will be updated iteratively through the aggregation of local model updates from each participating client in subsequent rounds.

$$w^0 = \text{InitializeModel}() \tag{2.1}$$

2. **Client Selection and Model Distribution:**
   In each training round t, the server randomly selects a subset of clients (e.g., 10% of the total) to participate in training. This selection process aims to ensure diversity and representativeness in the training data. The global model w^t is sent to each selected client, initializing their local training process.

3. **Local Model Training:**
   Each selected client trains the global model locally on its own dataset for a predetermined number of epochs. This training is done using local data, which ensures data privacy. Let w_k^t represent the local model weights after training on client k. This training step can be mathematically represented as:

$$w_k^t = w^t - \eta \nabla F_k(w^t) \tag{2.2}$$

   where η is the learning rate, controlling the step size in the weight update, and $\nabla F_k(w^t)$ is the gradient of the local objective function at client k with respect to the global model weights w^t. This local training allows clients to adapt the global model to their unique data distributions.

4. **Layer-by-Layer Aggregation of Local Updates:**
   After local training is completed, each client sends the weights of each layer L of its model, denoted by $w_{k,L}^t$, back to the server. The server performs a layer-wise aggregation of these weights, building the global model's layers. For a global model with m layers, the layer-by-layer aggregation at round $t+1$ for each layer L is represented as:

$$w_L^{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{k,L}^t \tag{2.3}$$

   where:

   - w_L^{t+1} is the aggregated weight for layer L in the global model at round $t+1$,
   - $w_{k,L}^t$ represents the weights for layer L of client k after local training,

- n_k and n are the number of samples at client k and the total number of samples across selected clients, respectively. This weighted aggregation ensures that clients with more data have a greater influence on the updated model.

Each layer L is aggregated across clients independently, allowing the server to construct each layer of the global model:

$$w^{t+1} = [w_1^{t+1}, w_2^{t+1}, \ldots, w_m^{t+1}] \tag{2.4}$$

5. **Global Model Update and Convergence Check:**
   The server updates the global model with the aggregated weights and performs a convergence check based on a predefined criterion (e.g., accuracy threshold or maximum number of rounds). If the convergence criterion is satisfied, the training process terminates; otherwise, it proceeds to the next training round. This step ensures that the model progressively improves and ultimately meets performance objectives.

$$w^{t+1} = \text{Aggregate}(w_1^t, w_2^t, \ldots, w_K^t) \tag{2.5}$$

This iterative process continues until the global model achieves the desired performance metrics, ensuring that the FL setup maintains data privacy while producing a robust and accurate global model. This training process is graphically illustrated in Fig. 2.1.

2.2 Categories of FL

There are various types of FL configurations, classified based on data distribution, communication architecture, and device connectivity. Here's an overview of each type, as depicted in Fig. 2.2:

2.2.1 Data partitioning based categories

These categories define how data is distributed across devices or organizations:

- **Horizontal FL (HFL)**
  HFL, also referred to as sample-based FL, is applicable when participant datasets have different samples but the same feature space. Each participant has data with the same features but unique instances [25–28]. Because it enables participants to train models collaboratively without sharing their raw data, this method is especially helpful in situations where privacy is an issue.

Use Case: Banks in different regions train a model using customer transaction data, where each bank has data on different customers but similar features. This enables the banks to develop a robust financial model while keeping sensitive customer data secure.

- **Vertical FL (VFL)**
  When participants have distinct characteristics but share the same sample space (i.e., the same people), VFL is employed. This setup enables learning from complementary data attributes across parties [29]. VFL is particularly advantageous in situations where data privacy regulations prevent data sharing, but organizations still want to gain insights from their combined datasets.

  Use Case: A bank and an e-commerce company collaborating on customer data where each has different features (e.g., purchase history vs. financial transactions). By leveraging VFL, both entities can enhance their predictive models without violating customer privacy.

- **Transfer FL (Transfer FL)**
  Transfer FL addresses scenarios where both sample and feature spaces have minimal overlap. It enables knowledge transfer between tasks to improve model performance across domains with limited shared data [30]. This approach is beneficial in situations where data collection is expensive or impractical, allowing for improved model training by utilizing insights from related tasks.

  Use Case: Medical institutions with non-overlapping datasets for different diseases sharing insights to improve prediction accuracy in related healthcare tasks. For example, knowledge gained from predicting outcomes for one disease could enhance the model's performance for another, even if the datasets are distinct.

2.2.2 System Architecture Based Categories

These categories define the communication and coordination setup in FL:

- **Centralized FL**
  A central server oversees the coordination and aggregation of models in centralized FL. The central server receives model updates from clients and compiles them into a global model. This configuration is widely used because it is simple to set up, but it depends on the central server for security and privacy. By serving as a coordinator, the central server makes sure that updates from different clients are gathered and combined into a single model. However, because the server manages sensitive data, this design may give rise to privacy issues and a single point of failure.

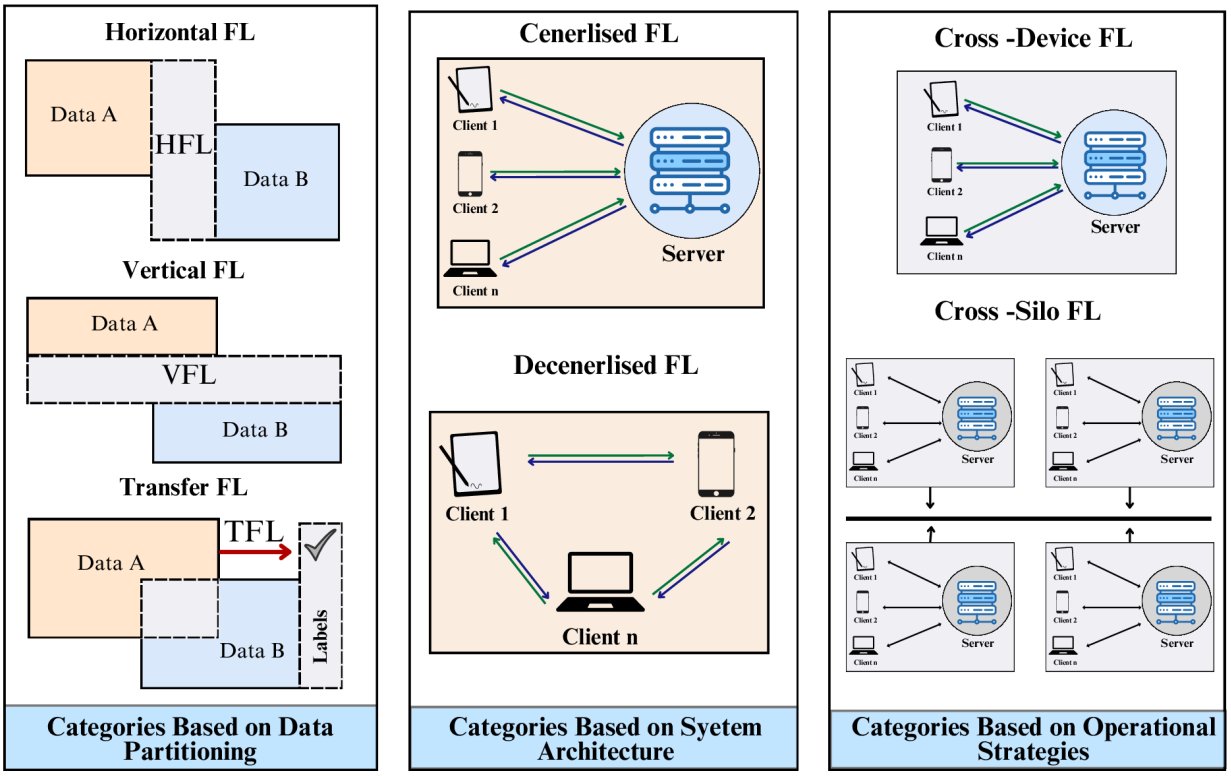


Figure 2.2: Types of FL.

Use Case: Smartphones updating a central server for predictive text input, where the central server aggregates updates from multiple devices. This allows for improved text prediction algorithms while keeping individual user data private, as only model updates rather than raw data are transmitted.

- **Decentralized FL**

  Decentralized FL uses P2P communication amongst clients in place of a central server. This setup provides enhanced privacy and resilience against central failures [31, 32]. In this approach, clients exchange model changes directly with one another, distributing the computing load and lowering latency. It is appropriate for applications where privacy is crucial since the lack of a central server reduces privacy threats and improves robustness.

  Use Case: Smart home devices developing energy-saving models through local communication without involving a central server. By sharing insights locally, these devices can learn from each other while maintaining user privacy and reducing reliance on external infrastructure.

2.2.3 Operation Strategies Based Categories

These categories define the scale and type of devices or entities involved in FL, as well as strategies for handling data heterogeneity:

- **Cross-Device FL**
  Inter-device FL generally uses smaller, frequently heterogeneous datasets and a large number of edge devices, such as smartphones and Internet of Things sensors. To support low-power devices, this type places an emphasis on scalability, effective communication, and lightweight processing. [33, 34]. The architecture is perfect for applications where user data is diverse and decentralized since it is made to handle the diversity in data quantity and quality across various devices.

  Use Case: Training language models across smartphones for keyboard suggestions, where each device only contributes small amounts of data but is part of a vast network. This technique guarantees that private user information stays on the device while enabling tailored recommendations.

- **Cross-Silo FL (Cross-Organization FL)**
  Less people participate in cross-silo FL; these participants are usually institutions or organizations with bigger, more reliable datasets. Cross-silo FL is suited for collaboration among enterprises, research institutes, or hospitals, where data consistency is crucial, and models are developed collaboratively [35]. This approach facilitates the sharing of insights and resources while maintaining data privacy and security, enabling organizations to benefit from each other's datasets without compromising sensitive information.

  Use Case: Hospitals collaborating on a disease prediction model, where each institution has substantial patient data, facilitating robust and reliable model training. This collaboration can lead to improved diagnostic capabilities and better patient outcomes while adhering to privacy regulations.

2.3 Applications of FL

FL is widely applied across various domains. In healthcare, FL facilitates collaborative medical image analysis, disease prediction, and drug discovery while preserving patient confidentiality. Hospitals and research institutions can jointly train models on brain tumor detection [36], diabetic retinopathy classification [37], and COVID-19 diagnosis without exposing sensitive patient data [38]. Similarly, in wearable healthcare devices [39], FL enables personalized predictive analytics for heart rate monitoring, sleep pattern analysis, and early disease detection. FL also enhances hospital management systems by improving patient flow predictions, optimizing resource allocation, and supporting real-time decision-making for personalized treatments.

In the financial sector, FL is employed for fraud detection [40], credit risk assessment [41], and anti-money laundering by enabling banks and financial institutions to collaboratively train models on distributed transactional data [42]. This approach ensures compliance with strict

privacy regulations while improving the accuracy of fraud detection systems. Additionally, FL assists in algorithmic trading [43], where different financial institutions can refine their trading strategies based on shared insights without exposing proprietary information.

FL is also revolutionizing edge computing and IoT. Smart devices, including autonomous vehicles [44], industrial sensors [45], and smart home appliances [46], leverage FL to enhance predictive maintenance, intrusion detection, and personalized user experiences. Autonomous vehicles, for instance, use FL to improve self-driving algorithms without centralizing vast amounts of sensor data, reducing latency and security risks. Industrial IoT benefits from FL in predictive maintenance for manufacturing systems, minimizing downtime and increasing operational efficiency.

In mobile devices, FL enables on-device learning for applications such as predictive text, speech recognition, and image processing while ensuring user privacy [47]. Virtual assistants [48] and recommendation systems benefit from FL by adapting to individual user preferences without requiring centralized data storage [49]. FL also supports federated analytics for mobile networks, allowing telecommunications providers to enhance service quality by optimizing network performance based on distributed data from users.

FL is also transforming smart cities by enhancing traffic management, environmental monitoring, and public safety [50]. By enabling collaborative learning across distributed sensors and infrastructure, FL facilitates real-time decision-making while preserving data privacy. Intelligent transportation systems leverage FL for congestion prediction, while energy-efficient urban planning benefits from FL-based analytics [51]. FL is used in smart surveillance systems to enhance security while ensuring compliance with data privacy regulations [52].

FL is also increasingly used in the defense and aerospace industries, where it facilitates secure intelligence analysis, predictive maintenance for aircraft, and coordination of unmanned aerial vehicles (UAVs) without exposing sensitive military data [53]. Similarly, in supply chain and logistics, FL optimizes inventory management, demand forecasting, and delivery route optimization while preserving confidentiality across different stakeholders [54].

In energy networks, FL is playing a critical role in demand-side management [55], energy load forecasting [56], and grid anomaly detection [57]. By leveraging FL, utilities can optimize energy distribution, enhance grid resilience, and improve fault detection while maintaining data privacy. Smart grids benefit from FL by enabling decentralized energy trading and demand response optimization. FL supports distributed renewable energy management by allowing solar and wind energy producers to collaboratively predict energy generation patterns without revealing sensitive operational data. Additionally, FL enhances cybersecurity in energy networks by detecting and mitigating cyber threats through collaborative anomaly detection models trained across multiple grid operators.

In my research, the key application area is the utilization of FL in energy networks, particularly for short-term load forecasting (STLF). I evaluate the effectiveness of FL in enhancing

forecasting accuracy while preserving data privacy and reducing communication overhead. My work focuses on developing robust aggregation mechanisms and addressing data heterogeneity to improve model performance in realistic energy forecasting scenarios. Experimental evaluations are performed in simulated energy network environments with multiple clients, where the proposed FL frameworks are compared against traditional centralized approaches. The results demonstrate the superiority of FL in terms of forecasting accuracy, privacy preservation, and communication efficiency, highlighting its potential for practical implementation in energy management systems.

The upcoming section will explore the need for FL in energy networks and identify existing research gaps in this domain.

2.3.1 FL Assisted Load Forecasting

STLF is essential for ensuring the stability and operational efficiency of modern power systems. It enables utility companies to enhance the integration of renewable energy sources, optimize generation scheduling, and improve demand-side management strategies. The complexities of today's electricity market, marked by deregulation, competition among various stakeholders, and the integration of advanced metering infrastructure (AMI), have rendered accurate STLF even more critical [56, 58]. A variety of STLF techniques have been created for both macro-scale (substation/grid level) and micro-scale (household level) forecasting [59–61]. Traditional statistical methods, such as linear regression, auto-regressive moving average (ARMA), and auto-regressive integrated moving average (ARIMA), have been widely used [62]. However, the rise of big data and artificial intelligence (AI) has propelled the adoption of deep learning (DL) models, which are particularly adept at recognizing complex, non-linear patterns in load data [63]. Despite their considerable potential, these models often require large amounts of historical data for training. This necessity typically leads to centralized data aggregation, creating challenges related to data privacy, high communication costs, and limited access to secure data repositories [60]. For instance, residential energy data collected at the micro-scale is highly sensitive and could be exploited to infer user behaviors, raising security concerns [64]. Similarly, utility providers at the macro-scale level often hesitate to share data due to competitive and privacy issues. FL is increasingly applied for residential-level STLF, where smart meter data is inherently diverse and exhibits significant variations among households. Numerous studies have employed FL on residential data using clustering techniques to tackle data heterogeneity and categorize clients based on their consumption behaviors. These clustering-based methods generate multiple federated models tailored for different groups, as demonstrated by Singh et al. [65], who combined FL and transfer learning to enhance forecasting accuracy by grouping households with similar electricity usage patterns. Although effective, the accuracy of these models heavily relies on the quality of the clustering and is sensitive to data anomalies, which can negatively impact model stability. Other clustering-based FL models for STLF have utilized various

data attributes. For example, researchers have developed federated LSTM models that apply socioeconomic clustering to classify users based on their load characteristics [66]. Additionally, studies such as [5] have explored non-clustered LSTM training for individual households but faced challenges related to data variability affecting model stability. Likewise, methods combining bidirectional LSTM models with optics-based clustering have been created to group users by region and heating type [67]. While clustering can help mitigate data diversity by organizing clients into more homogeneous groups, it has its limitations. Static, predefined clusters often fail to capture dynamic changes in data, leading to increased communication and computational overhead. Furthermore, clustering frameworks may exclude clients exhibiting different behaviors, disrupting the collaborative process. Despite advancements in clustering-based FL for residential STLF, research on applying FL at the substation level remains limited, even though this level exhibits greater stability and consistency compared to residential smart meter data. Consumption patterns at the substation level are aggregated and less affected by individual behavioral variations. Consequently, the need for clustering is diminished, making alternative strategies for addressing data heterogeneity more appropriate. Additionally, current studies often overlook the importance of effective aggregation methods capable of managing diverse data without relying on clustering. Utilizing clustering-based approaches for substation-level data can result in unnecessary segmentation and increased complexity, where a single, unified model would be sufficient.

2.4 Limitations and Implementation Challenges in FL

Despite its advantages in privacy preservation and decentralized learning, FL faces significant limitations and challenges that hinder its practical adoption. These issues span technical, regulatory, and operational domains, as outlined below.

1. **Data Heterogeneity and Quality**
   FL systems must handle non-IID data distributions across clients, including feature/label skew [68], data quality imbalances [69], and class imbalance [70]. Extreme heterogeneity can degrade global model convergence and generalization, particularly when combined with quantity skew [71, 72]. Prototype-based methods [73] and clustering techniques [74] partially address this but struggle with dynamic data evolution and multi-dimensional skew.

2. **Communication Overhead**
   Frequent model updates between resource-constrained edge devices and servers create bandwidth bottlenecks [75], exacerbated by inefficient protocols and large model sizes. While compression techniques [76] and adaptive communication strategies [77] help mitigate costs, they often trade off update granularity against convergence speed [78].

3. **Resource Constraints**
   Edge devices face inherent limitations in computation, memory, and energy [79]. Heterogeneous hardware capabilities compound these challenges, requiring adaptive algorithms [80] to prevent resource exhaustion during training. This heterogeneity also complicates client selection strategies, as dropout-prone devices [81] may destabilize model convergence.

4. **Privacy-Security Trade-offs**
   While FL avoids raw data sharing, model updates remain vulnerable to inference attacks [82] and poisoning [83]. Defense mechanisms like differential privacy [18] and homomorphic encryption increase computational overhead, creating tension between security and performance. Real-time adaptation to evolving threats remains an open challenge [84].

5. **Model Convergence and Personalization**
   Non-IID data distributions lead to client model divergence [24], requiring careful balance between global consistency and local personalization [85]. Parameter decoupling methods [86] demand manual tuning of shared/personalized components, limiting scalability for clients with sparse data.

6. **Client Dynamics**
   Transient connectivity and inconsistent participation patterns create partial updates and training delays. Current client selection strategies [87] often assume static availability, performing poorly in real-world scenarios with unpredictable dropout rates [81].

7. **Regulatory and Interoperability Barriers**
   Conflicting data privacy regulations (e.g., GDPR) complicate cross-border FL deployments [88]. Lack of standardized protocols [77] hinders integration across diverse hardware/software ecosystems, increasing deployment complexity.

8. **Scalability and Usability**
   Current FL frameworks require significant technical expertise for configuration, limiting accessibility [79]. Scaling defense mechanisms [83] and prototype-based methods [89] to large client populations remains challenging, with most approaches assuming ideal network conditions.

2.5 Security and Privacy Risks

FL is intrinsically susceptible to a variety of assaults that could jeopardize model confidentiality, integrity, and overall performance. Data FL attacks, Model FL Attacks, and Privacy FL Attacks are the three basic categories into which these assaults can be generally divided. Each

category poses unique challenges and risks to the FL process, impacting not only the immediate functionality of the system but also its broader implications for security and trustworthiness. The cascading effects of these attacks can lead to significant disruptions within the entire federated framework, particularly in load forecasting applications where accurate energy demand prediction is critical for grid stability and operational planning.

1. Data FL Attacks
2. Model FL Attacks
3. Privacy FL Attacks

The type of attack an adversary can do, depends on the capabilities of that adversary. These capabilities can be explained by threat modelling.

2.5.1 Threat Modelling in Distributed Systems

Threat modelling is a methodical technique that involves closely analysing a system's essential elements and operational procedures in order to detect and evaluate any threats and vulnerabilities [90, 91]. This methodology involves a comprehensive analysis of potential adversaries, including their objectives, the attack vectors they may exploit, and the potential repercussions on system security, particularly in terms of confidentiality, integrity, and availability. Within FL systems deployed in the real world, three primary categories of attacks exist: data poisoning, where malicious data inputs compromise model performance [92]; model poisoning, which introduces adversarial alterations directly into model updates [93, 94]; and privacy attacks, which aim to extract sensitive information from model parameters [95]. Previous research has demonstrated the resilience of FL to data poisoning attacks in energy forecasting applications [96].

Attacker's Objective: In FL, adversaries typically pursue three primary goals, each shaping the nature of the attack. First, compromising system security may affect either the integrity or availability of the model. Integrity breaches target the accuracy and reliability of the model's outputs, whereas availability attacks attempt to disrupt model functionality or cause system downtime. Second, attackers may choose either discriminatory or indiscriminate targeting. Discriminatory attacks focus on specific aspects of the model or specific classes within the data, aiming to manipulate the model's performance selectively. Indiscriminate attacks, in contrast, do not target specific components but instead aim to degrade overall model performance. Finally, the type of error introduced can be either specific or general. Specific errors attempt to manipulate predictions in a certain direction, such as favouring particular outcomes, while general errors lead to a broader degradation in accuracy and model reliability.

Attacker's Knowledge: The knowledge of an attacker in FL can be roughly divided into three categories: white-box, grey-box, and black-box scenarios. Each of these scenarios has

varying degrees of access to system data and models. The parameters, structure, and prediction outputs of the model are all fully accessible to attackers in a white-box scenario. With this thorough understanding, complex model poisoning attacks can be carried out, precisely modifying weights or gradients to cause a particular harmful impact on the global model [97]. In a grey-box scenario, attackers have partial knowledge, such as access to some but not all model parameters, or limited visibility into the data or model structure. This level of access enables targeted attacks, where adversaries may not know the complete architecture but can still conduct effective poisoning by leveraging known aspects of the model to influence specific behaviors [98]. In contrast, black-box attackers have minimal information about the model and lack access to its internal parameters, thus relying on observed model outputs or general system behavior to craft attacks. This scenario limits the attacker's control but still allows for certain types of malicious activity, such as inference attacks that infer information about the training data without direct access [99]. Moreover, the attacker's knowledge about data distribution can vary, as they may possess complete or limited knowledge about the dataset, further influencing attack strategies.

Attacker's Capabilities: In FL, attackers exhibit a range of capabilities that can be categorized into passive and active roles, each entailing distinct methods and impacts on the FL environment [100]. Passive attacks involve adversaries who monitor communication channels between clients and the server, collecting information on model updates or data exchange patterns without interfering. Such passive attacks often include eavesdropping, which enables attackers to observe system behavior covertly. While these attacks do not disrupt the system, they may lead to privacy violations, as attackers gather insights about model updates or training data, which can compromise confidentiality. Active assaults, on the other hand, entail direct contact with the model or data, whereby attackers change model parameters or manipulate training data at the client or server level. Active attacks include a variety of strategies, such as model poisoning, in which attackers alter the model parameters that are provided to the server in order to reduce the accuracy of the model, and data poisoning, in which attackers add modified data to affect the model's behavior. Furthermore, adversaries use updates to infer private information about training data in FL inference attacks, which could compromise privacy [101].

2.5.2 Data FL Attacks

In order to adversely affect the learning process, data attacks target the training data that is kept on client devices. These attacks can take several prominent forms, each with unique processes and outcomes:

Data Poisoning: One of the most alarming risks of FL is this. In order to skew the model's training process and ultimately produce a degraded global model, malicious clients may introduce inaccurate or misleading data into their local datasets [102, 103]. In the context of load forecasting, attackers could inject false energy consumption records, leading to unreliable demand predictions that disrupt energy supply planning and increase operational costs.

Label Flipping: a particular and sneaky type of data poisoning in which the labels of individual training instances are purposefully changed [104]. However, it is important to note that label flipping attacks are not applicable in load forecasting applications, as energy consumption data does not have discrete class labels but rather continuous numerical values. Consequently, such attacks do not pose a realistic threat in this domain.

Backdoor Attacks: In these types of attacks, adversaries establish a covert backdoor by altering the training data. Certain triggers have the ability to activate this backdoor, which causes the model to generate inaccurate outputs when it encounters them [105]. In load forecasting, such an attack could introduce systematic biases in energy predictions, leading to resource misallocation and inefficiencies in power generation and distribution [106].

Local data corruption at the client level is usually the first sign of the impact of data attacks. The corresponding local model updates are influenced by these compromised local datasets and subsequently transmitted to a central server for aggregation. Consequently, the aggregation process incorporates these poisoned updates, leading to a compromised global model. The ramifications can be severe, resulting in degraded performance across the system. However, it is important to note that in load forecasting applications, data attacks are often ineffective due to the continuous and structured nature of energy consumption data. Unlike classification tasks, where mislabeling can directly impact model accuracy, dat attacks are usually ineffective in load forecasting applications [96].

2.5.3 Model FL Attacks

A serious risk to FL is model attacks, in which malicious clients deliberately alter their local model updates before sending them to the central server or man in the middle can create malicious updates [107, 108]. The primary objective of these attacks is to compromise the integrity of the global model through a technique known as model poisoning. This process involves the introduction of harmful alterations to the local updates, which can significantly disrupt the learning process.

One of the common strategies employed in model poisoning is the modification of gradients or model parameters. By intentionally altering these updates, malicious clients can steer the global model towards producing incorrect or biased outputs [94]. In load forecasting, such manipulation could lead to inaccurate energy demand predictions, which can cause over- or under-supply of electricity, leading to economic losses and instability in grid operations [107].

Model attacks start to have an effect at the client level, where the fraudulently modified local model updates are produced. The aggregation procedure starts as soon as the central server receives these erroneous updates. All participating clients, both malicious and honest, provide updates to the server. Sadly, most of the time, the server is unable to distinguish between authentic updates and ones that have been altered. Consequently, it can inadvertently add these tainted changes to the global model.

Table 2.1: A summary of state of the art stealth attacks in federated systems.

Sr. No.	Problem Type	Dataset	Ref.
1	Classification	CIRF10	[113]
2	NLP	Reddit, IMDB, Sentiments140	[114]
3	Classification	MNIST	[115]
4	Classification	FMNIST, Adult Census	[116]
5	NLP	20Newsgoups, DistilBert	[117]
6	Classification	MNIST, CIRF10	[114]
7	Classification	MNIST, CIRF10 and 100	[118]

This aggregation presents a serious danger since the compromised updates' influence may distort the model's performance and result in less-than-ideal choices or actions throughout the FL network. Such model attacks can have serious repercussions in critical domains such as energy forecasting, where compromised predictions could lead to inefficient power distribution strategies, increased carbon emissions, and economic losses.

In FL, model attacks can be categorized into two main types: partial and fully model poisoning attacks, as well as blunt and stealthy attacks.

- **Partial Model Poisoning Attacks:** These attacks involve altering only a subset of model parameters, which can introduce noise and degrade the model's performance without being immediately detectable [94].
- **Fully Model Poisoning Attacks:** In contrast, these attacks modify the entire model, leading to severe disruptions in the training process and significant degradation in model accuracy [31, 93, 107, 109–112].
- **Blunt Attacks:** These are overt and easily detectable, often resulting in noticeable performance drops in the model [31, 107, 112].
- **Stealthy Attacks:** These attacks are designed to evade detection, subtly degrading model performance while maintaining overall accuracy in the short term [113, 114, 114–117].

Literature surveys suggest that most stealth attacks are implemented on classification models and natural language processing applications. A summary of stealth attacks is presented in a Table 2.1. It suggests that no one has yet implemented stealth attacks in regression problems such as STLF.

2.5.4 Privacy FL Attacks

As privacy attacks are explicitly made to extract sensitive information from model updates or the aggregated model, they could compromise the confidentiality of training data, which is a major concern in the field of FL [88]. These attacks take advantage of the information contained in

model updates to infer private information about the underlying data, even while FL is designed to protect privacy by keeping data localized on client devices [119, 120].

The transmission of model changes from clients to the central server is where privacy threats start to have an impact. Attackers may intercept these updates throughout this procedure and use advanced analytic methods to retrieve potentially private data. In load forecasting, attackers could infer energy consumption patterns of individual households, potentially exposing sensitive details about users' daily routines, appliance usage, or even occupancy status [121].

The severity of a privacy breach largely depends on the attacker's capabilities and the sophistication of their inference methods. Advanced techniques may enable attackers to reverse-engineer model updates, revealing confidential information such as individual energy consumption trends. This undermines consumer privacy and can significantly erode trust in FL-based load forecasting systems. Effective privacy-preserving mechanisms must be implemented to ensure the confidentiality of users' energy data while maintaining accurate demand predictions.

2.5.5 Gap Analysis

Despite the existing research on FL security, a significant gap remains in addressing stealth attacks that subtly degrade model performance over time without triggering conventional detection mechanisms. Additionally, while differential privacy is often employed to enhance security, it can negatively impact forecasting accuracy, particularly in load forecasting applications where precise numerical predictions are required. Forecasting accuracy is paramount for ensuring efficient energy management, and any reduction in precision due to privacy-enhancing techniques can lead to suboptimal resource allocation. Therefore, a more effective solution is necessary to balance privacy preservation and forecasting accuracy while ensuring resilience against stealth attacks in load forecasting scenarios.

2.5.6 Future Directions and Proposed Solutions

While existing defense mechanisms can effectively detect traditional model attacks in FL, there is a growing need to explore stealth attacks that remain undetected by current security frameworks. Conventional model poisoning attacks introduce abrupt changes in model updates, making them easier to identify. However, a carefully designed stealth attack could gradually manipulate model parameters in a way that subtly skews load forecasting predictions over time without raising suspicion. Such an attack could exploit the natural variability in energy consumption patterns, making it indistinguishable from normal fluctuations.

2.6 Defense Frameworks in FL

Deploying customized security frameworks that take into account device configurations, FL architecture, and resources is crucial to combating the many threats on FL. As seen in Fig. 2.3, several attack types, including data, model, and privacy, exploit unique weaknesses and call for different defensive tactics [122]. For instance, secure aggregation and Byzantine fault tolerance successfully combat model poisoning [123], while strong data validation and anomaly detection can reduce data poisoning [124]. To defend against inference attacks, methods such as homomorphic encryption and differential privacy are essential. To improve the security and resilience of FL systems, a layered defense strategy encompassing many frameworks is required; a one-size-fits-all approach is not feasible [125].

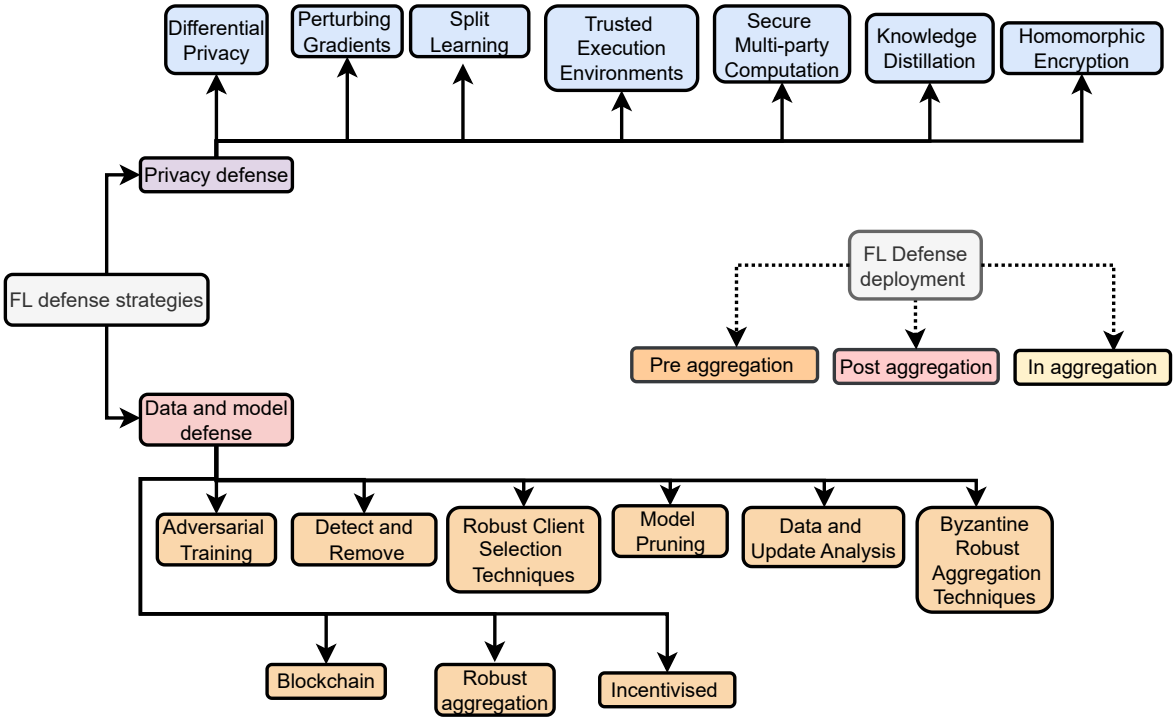


Figure 2.3: Type of defense strategies.

Scholars have put forth a number of defensive tactics to combat new FL risks [115,126,127], concentrating on pre-, in-, and post-aggregation stages of the learning process. Identifying and mitigating harmful updates early is the goal of pre-aggregation protections [126–128]. During global model updates, robust operators are used in in-aggregation approaches [94,115,129–131]. The goal of post-aggregation strategies [93,132,133] is to ensure the integrity of the final model by fixing adversarial models after training. The security of FL systems against backdoor assaults is improved by these phase-specific countermeasures. Defenses are further arranged according to their operational principles and classified into data, model, and privacy frameworks. Table 2.2 provides a summary of these frameworks.

Table 2.2: Defense Frameworks and Privacy Techniques in FL

No.	Defense Strategies	Description	References
1	Detect and Remove	Employs statistical analysis or anomaly detection techniques to identify and eliminate harmful updates or corrupted clients, thus maintaining the integrity of the global model.	[104, 105, 134–140]
2	Adversarial Training	Enhances robustness by incorporating adversarial examples during training, which helps defend against potential threats to both data integrity and model updates.	[141, 142]
3	Model Pruning	Optimizes the model by eliminating non-essential parameters, thereby improving efficiency and complicating efforts for attackers to introduce malicious updates.	[143, 144]
4	Byzantine/Reliable Aggregation	Combines updates that are consistent with the majority, effectively filtering out anomalies that may originate from compromised clients.	[111, 130, 131, 145–149]
5	Robust Client Selection	Chooses clients based on their reliability, minimizing the impact of malicious or unreliable participants on the global model's performance.	[150–152]
6	Data and Update Analysis	Conducts thorough analysis of local data and model updates to pinpoint inconsistencies or suspicious activities that may signal adversarial attacks.	[124, 153–155]
7	Blockchain	Integrates blockchain technology to provide transparency, immutability, and auditability in FL, thereby preventing tampering and ensuring trustworthy aggregation of updates.	[156–158]
8	Incentive FL	Implements reward systems for client participation, fostering honest contributions and deterring malicious behavior within the network.	[159, 160]
9	Regularization	Aims to prevent overfitting by placing constraints on model updates, which enhances generalization and mitigates vulnerabilities to various attacks.	[161, 162]
10	Homomorphic Encryption	Utilizes a cryptographic approach that safeguards privacy in FL by enabling computations on encrypted data without requiring decryption.	[163, 164]
11	Knowledge Distillation	Facilitates privacy-preserving model compression, where a smaller "student" model learns from the outputs of a larger "teacher" model.	[165–167]
12	Secure Multi-party Computation	Establishes a protocol that upholds input privacy while allowing multiple parties to compute a function based on those inputs.	[168–170]
13	Split Learning	Segments the model between the client and server, ensuring only intermediate activations are shared, which keeps the data localized and secure.	[171–173]
14	Perturbing Gradients	Lowers the risk of exposing private information by introducing noise into gradients before they are shared with the server.	[174–177]
15	Differential Privacy	Reduces the likelihood of individual identification by injecting calibrated noise into data or gradients, thus enhancing privacy.	[178, 179]
16	Trusted Execution Environments	Employs secure hardware that processes data within an isolated, tamper-proof environment, thereby ensuring data confidentiality and security.	[180, 181]

2.6.1 Gap Analysis

Despite the breadth of existing defense frameworks (summarized in Table 2.2), critical gaps remain in addressing emerging challenges specific to resource-constrained FL environments like load forecasting. First, most defences prioritize robustness over efficiency, resulting in computationally heavy frameworks ill-suited for lightweight edge devices common in smart grids or IoT-based forecasting systems [125]. Second, stealth attacks—such as adaptive model poisoning that mimics benign gradient patterns—often evade traditional anomaly detection and Byzantine aggregation methods [123]. These attacks require dynamic detection mechanisms that analyze temporal behavioral shifts in client updates, which existing tools lack. Third, while DP mitigates inference risks, its noise injection severely degrades load forecasting accuracy due to the sensitivity of time-series patterns [178]. A balance between DP's privacy guarantees and utility preservation remains unresolved, particularly in scenarios requiring high-frequency model updates. Finally, load forecasting's unique attack surface—where subtle model manipulations can propagate grid instability—demands domain-specific defenses that integrate physical constraints (e.g., energy conservation laws) into adversarial model validation [124]. Current frameworks treat FL security generically, overlooking these application-critical nuances.

2.6.2 Future Directions and Proposed Solutions

To address these gaps, future defence frameworks must adopt three principles:

1. *Stealth-aware Detection*: In an increasingly sophisticated threat environment, detection mechanisms must be aware of stealth tactics employed by adversaries. This principle involves developing detection methods that can identify subtle and covert attacks without generating excessive false positives. Techniques such as behavioral analysis, anomaly detection, and ML can play a crucial role in recognizing patterns indicative of stealthy intrusions, allowing for timely and effective responses.

2. *Lightweight Defence Framework*: This principle emphasizes the importance of designing security measures that do not impose significant overhead on system resources. A lightweight framework should efficiently operate in resource-constrained environments, ensuring that performance is not sacrificed for security. This can involve the use of simplified algorithms, minimalistic architectures, and adaptive mechanisms that dynamically adjust resource allocation based on the threat landscape.

2.7 Communication and Computational Efficiency in FL for Distributed Applications

Effective FL requires balancing model performance with resource constraints across distributed environments. Key efficiency optimization strategies include model compression, communication optimization, and topology design, which address challenges in distributed training scenarios like resource-constrained edge devices and heterogeneous network conditions [94].

2.7.1 Model Compression Techniques

Model compression techniques reduce neural network size while preserving performance, critical for edge deployment in distributed systems.

Pruning

Pruning eliminates unnecessary parameters to simplify networks. **Unstructured pruning** removes small-magnitude weights using magnitude thresholds [182] or Bayesian methods [183], though requiring specialized hardware for full benefits. **Structured pruning** removes entire network components via sensitivity analysis [184] or soft thresholding [185], achieving 2-4× CPU speedups [186] crucial for large-scale distributed applications. Post-training pruning can reduce model size by 90% without accuracy loss [187].

Sparsification

Sparsification minimizes communication by transmitting only essential parameters. **Weight sparsification** zeros small weights during training [188], while **gradient sparsification** applies similar principles to gradients, achieving 10-100× compression [189]. This technique reduces bandwidth by up to 95% in cross-silo FL [189], beneficial for time-sensitive applications like energy forecasting.

Gradient Compression

Gradient compression reduces update sizes through parameter-efficient strategies. **Quantization** lowers gradient precision (e.g., 8-bit vs 32-bit), achieving 4× compression with <1% accuracy loss [190, 191]. **Selective transmission** prioritizes high-magnitude gradients using error feedback [192], combining with Huffman coding for up to 89% bandwidth reduction [193].

Quantization

Quantization techniques enhance deployment efficiency in distributed systems. **Fixed-point arithmetic** reduces memory footprint by 4-8× [194], lowering energy consumption in edge

devices [195, 196]. **Dynamic quantization** adapts bit-widths per layer for better embedded performance [197, 198]. Medical IoT applications show 63% energy reduction with 8-bit quantization [199], applicable to smart grid deployments.

2.7.2 Communication Optimization

Optimizing communication improves FL efficiency in distributed environments.

Topology Design

Topology design optimizes device-server interactions in distributed systems. **Hierarchical topologies** use edge intermediaries, reducing direct transmissions by 70% [200]. **Adaptive connectivity** adjusts participation based on channel quality and energy levels, accelerating convergence by 2× [201, 202].

Communication Compression

Balancing update quality and resource usage is critical. **Early stopping** terminates local training at validation plateaus, reducing compute time by 40-60% [94]. The FL-RCE framework automates this process [203]. **Adaptive compression** adjusts sparsity/quantization based on network conditions, achieving 82% communication reduction [204]. **Lossless methods** like Huffman coding recover exact gradients with 15-30% additional compression [205].

Hardware-Software Co-Design

Emerging co-design approaches improve FL efficiency. **FPGA accelerators** offer 8× energy efficiency over GPUs [206]. **In-memory computing** reduces data movement overhead by 90% [196], enhancing prediction speed in distributed systems.

2.7.3 Gap Analysis

While existing FL efficiency techniques show promise, critical gaps remain. First, most compression methods rely on static heuristics that fail to adapt to dynamic network conditions or data distribution shifts. Second, topology optimization lacks integration with adaptive compression mechanisms. Third, hardware-specific optimizations sacrifice cross-platform generality.

2.7.4 Future Directions and Proposed Solutions

Current frameworks often treat communication and computation as isolated optimizations. Co-designing pruning and topology-aware aggregation could reinforce efficiency gains. Developing lightweight self-adaptive mechanisms for dynamic resource allocation represents an important

research direction. Our proposed framework addresses these gaps by combining quantization, early stopping criteria, and partial layer aggregation to enhance distributed learning efficiency in applications like energy load forecasting.

2.8 Heterogeneity in FL for Distributed Applications

FL enables distributed model training by allowing multiple clients to learn from their local datasets without directly sharing data. Despite this advantage, achieving consistent model performance across heterogeneous data sources remains a critical challenge. In distributed applications like energy load forecasting, heterogeneity is influenced by variations in data patterns, which stem from factors such as geographical location, consumer behavior, seasonal changes, and sensor accuracy. These variations lead to diverse data distributions among clients, affecting both model convergence and generalization capabilities. Data heterogeneity in distributed learning can be divided into several categories. *Distribution Skew* occurs when data patterns across regions vary significantly due to factors like climate and economic conditions, which can lead to biased models [207]. *Label Skew* arises when certain clients lack data for specific patterns, leading to reduced predictive accuracy [68]. *Feature Skew* is caused by the use of different types of sensors or varying measurement resolutions across infrastructures [208, 209]. *Quality Skew* emerges from inconsistencies in data collection, such as missing values or noise from faulty sensors [69, 210]. Finally, *Quantity Skew* happens when different amounts of data are contributed by clients, which can lead to imbalanced model updates [71, 72]. To tackle these challenges, various strategies have been proposed in FL frameworks. Clustering-based approaches, such as FedCluster [74] and ClusterFL [211], group clients with similar data patterns to enable more focused aggregation of local models. However, while clustering can be effective, it introduces additional computational complexity and faces scalability challenges as the number of clients increases [212]. Another strategy is parameter decoupling, which separates models into shared components and client-specific components. Methods like FedPer [85] and FedRep [86] allow clients to individually train their personalized parts of the model while collaborating on a common base. While these techniques enhance personalization, they require careful tuning to strike a balance between global and local adaptations, especially for clients with limited data. Knowledge distillation presents an alternative by enabling local models to share knowledge through aggregated representations instead of exchanging raw model parameters [213, 214]. This technique improves model generalization while reducing communication overhead [165, 215]. Similarly, prototype-based FL methods, like those discussed in [73], generate low-dimensional representations of local data. These prototypes are shared and aggregated to refine global models. However, ensuring that prototypes remain representative across diverse data profiles continues to be a challenge [89]. Addressing the challenges posed by data heterogeneity is essential for improving the robustness of FL-based models. Future research should

focus on adaptive aggregation techniques and dynamic personalization strategies to improve model performance in real-world, diverse conditions.

2.8.1 Gap Analysis

Existing approaches to addressing heterogeneity in FL have made significant strides in mitigating challenges such as distribution skew, communication overhead, and personalization. Techniques like client clustering, parameter decoupling, knowledge distillation, and prototype-based learning demonstrate promise in improving model generalization and adaptability. However, critical gaps remain that limit their practical applicability and effectiveness in real-world scenarios. First, clustering methods often rely on static assumptions about client data distributions, which may not hold in dynamic environments where client data evolves over time [216]. This raises scalability concerns and limits adaptability to shifting client participation patterns. Additionally, it is difficult to decide how many clusters should be made, as this decision can significantly impact the performance and efficiency of the clustering approach [216]. Second, parameter decoupling and personalization strategies require manual tuning of shared versus client-specific components, posing challenges for clients with limited computational resources or sparse local data [217]. This can lead to suboptimal performance and increased complexity in the training process. Third, while knowledge distillation reduces communication costs, it introduces trade-offs between prototype quality, aggregation efficiency, and privacy preservation, especially as the number of clients grows. Prototype-based methods, meanwhile, struggle to maintain representative global prototypes under extreme feature or label skew, often sacrificing granularity for scalability [218].

2.8.2 Future Directions and Proposed Solutions

A key unresolved challenge lies in unifying these approaches into a holistic framework that dynamically balances generalization, personalization, and scalability while preserving privacy. For instance, few works address the interplay between multiple heterogeneity dimensions (e.g., feature skew compounded by device heterogeneity) or the long-term effects of evolving client data distributions. Furthermore, reliance on idealized assumptions, such as uniform client participation or stationary data, limits applicability in practical deployments where client availability and data quality fluctuate. Finally, there is a need for lightweight, adaptive mechanisms to automate the trade-off between shared and personalized model components without extensive hyperparameter tuning. These gaps motivate my work, which introduces an adaptive single layer aggregation framework to address scalability, dynamic adaptation of data heterogeneity while minimizing computational and communication overhead.

2.9 Performance metrics

In this section, different metrics are discussed that were used to evaluate the various parameters of the system. These metrics include loss functions, which assess the model's performance by measuring the difference between predicted and actual outcomes, thereby indicating how well the model is learning from the training data. Energy consumption measures the efficiency of the training process, highlighting the resources used by client devices during model updates. Communication cost examines the bandwidth and time required for data transmission between the central server and client devices, emphasizing the importance of optimizing these exchanges for better performance. Finally, Levene's test for data heterogeneity analyzes the variability in data distributions among clients, which can significantly impact the effectiveness and robustness of the FL approach. Together, these metrics provide a comprehensive framework for understanding and improving the system's overall performance.

2.9.1 Loss functions

The forecasting accuracy in this study is evaluated using two commonly applied error metrics: the Mean Absolute Error (MAE) and the Mean Absolute Percentage Error (MAPE). These metrics were chosen due to their interpretability and robustness to outliers in practical forecasting scenarios.

The *Mean Absolute Error* quantifies the average magnitude of errors without considering direction:

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \tag{2.6}$$

The *Mean Absolute Percentage Error* measures relative error magnitude as a percentage:

$$\text{MAPE} = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \tag{2.7}$$

MAE provides a straightforward error magnitude while remaining robust to outliers through its use of absolute differences. MAPE complements this by expressing error as a percentage of actual values, facilitating comparisons across datasets with different scales. Note that MAPE should be used carefully when actual values approach zero to avoid division by small numbers.

2.9.2 Energy Consumption:

Energy consumption (E_{com}) for transmitting local models depends on factors like energy per kilobyte, transfer time, and device type. Using [219]'s model, E_{com} is calculated as:

$$E_{com} = R[(\alpha \times t) + (\beta \times D)] \tag{2.8}$$

where R denotes the number of communication rounds, $\alpha = 0.0001$ kWh/sec represents energy per second, $t = 1$ ms is the transmission time, $\beta = 0.015$ kWh/GB is the energy per kilobyte, and D is the data size.

2.9.3 Communication Cost

Communication cost in FL refers to the total resources, primarily bandwidth and time, required to transmit model updates between the central server and client devices, including the size of the data sent and the frequency of transmissions. Minimizing these costs is crucial for optimizing system performance, effectively managing network resources, and assessing scalability as the number of clients increases. Additionally, understanding communication costs informs the development of algorithms aimed at reducing data exchange, such as model compression techniques. Ultimately, addressing communication costs is key to enhancing the efficiency and practicality of FL implementations in real-world applications.

2.9.4 Levene's Test

Levene's test [220] was employed to investigate the data heterogeneity representing variability in energy consumption patterns among clients in the smart grid network. This statistical method is robust for evaluating whether different groups have equal variances. It enables the comparison of energy consumption variability across clients and helps identify significant differences in consumption patterns.

The test statistic W in Levene's test is derived from the absolute deviations of data points from their group means. The formula for W is:

$$W = \frac{(N-k)}{(k-1)} \times \frac{\sum_{i=1}^{k} N_i(\bar{Z}_{i\cdot} - \bar{Z}_{\cdot\cdot})^2}{\sum_{i=1}^{k}\sum_{j=1}^{N_i}(Z_{ij} - \bar{Z}_{i\cdot})^2} \tag{2.9}$$

In this equation, W is the test statistic, N is the total number of observations, k is the number of groups, N_i is the number of observations in the ith group, $\bar{Z}_{i\cdot}$ is the mean of the ith group, $\bar{Z}_{\cdot\cdot}$ is the overall mean, and Z_{ij} is the jth observation in the ith group.

Under the assumption of equal variances, W follows an F-distribution with $(k-1)$ and $(N-k)$ degrees of freedom. The p-value reflects the probability of obtaining a test statistic like W if the groups have equal variances. A small p-value suggests that W is unlikely under the assumption of equal variances, indicating significant differences. A large p-value implies that W is more common, suggesting similar variances.

A low p-value, usually below a significance level (e.g., 0.05), indicates that the differences in variance between groups are statistically significant. This leads to the rejection of the null hypothesis of homogeneity, implying data heterogeneity. A high p-value above the significance level indicates insufficient evidence to reject the null hypothesis, suggesting data homogeneity.

2.10 Summary of Literature Review, Research Gap and Link with Challenges

This section synthesizes the key insights from the literature review, identifies critical research gaps, and elucidates their connection to the challenges outlined in Chapter 1. By addressing these gaps, the research aims to enhance the practicality and effectiveness of FL systems, particularly in energy networks.

Summary of Literature Review The literature review in this Chapter has provided an extensive overview of FL, detailing its training process, categories, applications, and inherent challenges. It has delineated the five key phases of the FL training pipeline: global model initialization, client selection and model distribution, local model training, layer-wise aggregation of local updates, and global model updates with convergence validation. The review has also explored various categories of FL based on data partitioning (horizontal, vertical, transfer), system architecture (centralized, decentralized), and operational strategies (cross-device, cross-silo). Furthermore, the applications of FL across different domains, particularly in energy systems for load forecasting, have been emphasized. The review has identified several challenges in FL, including data heterogeneity, communication overhead, resource constraints, privacy-security trade-offs, model convergence, and client dynamics.

Research Gaps and Their Significance: Despite the extensive research on FL, several gaps remain that hinder its practical adoption, particularly in energy networks. These gaps are not merely theoretical; they have profound implications for the performance, security, and efficiency of FL systems. Addressing these gaps is essential for advancing the field and ensuring the successful deployment of FL in real-world applications.

1. **Stealth Attacks**: First and foremost, there is a pressing need to create stealth attacks that current defence frameworks cannot detect, this gap is tied to challenge C1. Most attacks can be detected by the frameworks can be detected defence frameworks. Existing research does not adequately address stealth attacks that subtly degrade model performance over time without triggering conventional detection mechanisms. These attacks are particularly insidious because they can remain undetected for extended periods, gradually eroding the accuracy and reliability of the global model. In energy networks, where precise load forecasting is critical for grid stability and operational planning, the lack of effective defences against stealth attacks can lead to significant economic losses and increased operational risks. For instance, a stealth attack like the Federated Communication Round Attack (FedCRA), which increases communication rounds without affecting model accuracy, can result in higher energy consumption and communication costs, undermining the efficiency and sustainability of the FL system.

2. **Defense Mechanisms**: There is a lack of lightweight defence frameworks that are suitable for resource-constrained edge devices common in smart grids or IoT-based forecasting systems. Current defence mechanisms often introduce additional computational overhead, making them less suitable for edge devices with limited processing power and memory. This gap is critical because it directly impacts the challenge of robust aggregation (C2). Without lightweight and efficient defense mechanisms, FL systems remain vulnerable to adversarial attacks, which can compromise the integrity of the global model and lead to unreliable predictions. For example, in a smart grid scenario, a compromised global model could result in inaccurate load forecasts, potentially causing grid instability and increased operational costs.

3. **Communication Efficiency**: Current techniques for communication and computational efficiency, such as model pruning, quantization, and topology optimization, often rely on static heuristics or manual tuning, which fail to adapt to dynamic network conditions or evolving data distributions. This gap is directly tied to the challenge of communication and computational efficiency (C3). In energy networks, where bandwidth and energy resources are often limited, inefficient communication can lead to increased latency, higher energy consumption, and reduced scalability. For instance, if an FL system cannot dynamically adjust its communication strategy based on real-time network conditions, it may result in excessive data transmission, straining the network and reducing the overall efficiency of the system.

4. **Data Heterogeneity**: Most clustering methods for addressing data heterogeneity rely on static assumptions about client data distributions, which may not hold in dynamic environments. Additionally, there is a need for lightweight, adaptive mechanisms that can dynamically adjust to data drift and varying client participation patterns. This gap is closely linked to the challenge of data heterogeneity (C4). In energy networks, where clients may have diverse and non-IID data distributions due to varying consumption patterns and environmental conditions, static clustering methods can lead to biased global models and reduced forecasting accuracy. For example, if an FL system cannot adapt to dynamic changes in client data distributions, it may fail to provide accurate load forecasts for different regions or consumer groups, resulting in inefficient energy allocation and increased operational costs.

By addressing these research gaps, the research presented in this thesis aims to enhance the security, efficiency, and adaptability of FL systems, particularly in energy networks. The proposed solutions seek to bridge the gap between theoretical promises and practical deployment of FL by focusing on robustness, communication efficiency, and adaptability to data heterogeneity. This will ensure that FL systems can deliver reliable and accurate load forecasts while preserving privacy and operating efficiently in real-world conditions.

Chapter 3

Attack Strategies in Distributed Systems

In the field of FL, the distributed nature of model training introduces unique vulnerabilities that can be exploited by malicious actors. These vulnerabilities are particularly concerning as they can compromise the integrity and performance of the global model. Understanding these attack strategies is crucial for developing robust defense mechanisms that can safeguard FL systems against such threats. This chapter delves into various attack strategies that adversaries can employ to undermine FL systems, with a focus on model poisoning attacks. These attacks involve malicious clients manipulating their local updates to corrupt the global model, thereby degrading its performance and reliability.

Attacks in FL can be broadly classified into two categories: those that directly degrade model performance and those that inflate resource consumption without necessarily affecting performance. Furthermore, attacks can be categorized based on their detectability as abrupt (easily detectable) or stealthy (designed to evade detection mechanisms).

Building on the challenges outlined in Chapter 1, this chapter specifically addresses Challenge C1 (Model Attacks) by exploring how adversarial clients can manipulate the training process. As highlighted in Chapter 1, FL systems are susceptible to various types of attacks that can gradually degrade model performance without immediate detection. This is particularly relevant to the research gap identified in Section 2.5.6 of Chapter 2, where the need for effective defence mechanisms against stealth attacks was emphasized. Existing defence frameworks often fail to detect stealth attacks that subtly manipulate model parameters over time, as discussed in Section 2.6.1. To mitigate these vulnerabilities, robust defence mechanisms are essential to safeguard the integrity and reliability of FL systems.

Stealth attacks, such as the Fed-CRA introduced in Section 3.2, are particularly challenging because they evade traditional detection mechanisms. Fed-CRA, which increases communication rounds without affecting model accuracy, exemplifies how adversaries can exploit FL systems to inflate resource consumption while remaining undetected. This type of attack is critical for testing and developing defence frameworks, as it highlights the need for mechanisms that can identify and mitigate subtle threats. As mentioned in Section 2.5.6, current defence frame-

works are often inadequate against such stealthy tactics, making the development of advanced defence strategies a pressing concern. By examining these attack strategies, this chapter underscores the importance of creating defence mechanisms that are specifically designed to detect and counteract stealth attacks, thereby enhancing the security and efficiency of FL systems.

To address these concerns, this chapter explores a range of attack strategies, starting from basic random approaches to more sophisticated stealth-targeted attacks. The following attack types are examined in increasing order of sophistication:

- **Completely Random Attack (CRA)**: Malicious clients submit entirely random model updates, directly degrading global model performance.
- **Partially Random Attack (PRA)**: A hybrid approach where only portions of model updates are randomized, balancing disruption with plausibility.
- **Model Flipping Attack (MFA)**: Clients invert model parameters to introduce systematic errors in the global aggregation.
- **Perturbed Attack (PA)**: Local updates are systematically perturbed rather than fully randomized, creating targeted model degradation.
- **Federated Communication Round Attack (Fed-CRA)**: A stealthy attack designed to increase training rounds and communication overhead without directly affecting model accuracy, representing a resource-inflating strategy.

The chapter evaluates these attack strategies through experiments with real-world datasets. Results demonstrate significant performance degradation from CRA, PRA, MFA, and PA. Additionally, Fed-CRA's unique impact on system resources is highlighted, showing increased training time, communication resources, and energy consumption despite maintaining model accuracy. By progressing from basic performance-degrading attacks to advanced resource-inflating strategies, this chapter provides a comprehensive analysis of FL vulnerabilities and establishes a foundation for developing effective defense mechanisms. This analysis directly contributes to addressing the research gap identified in Section 2.5.6, where current defense mechanisms were found to be inadequate against stealth attacks that gradually compromise model integrity.

3.1 Performance Degrading Attacks

Based on the discussion of threat modelling described in section 2.5.1, this work assumes an active attacker who possesses the capability to intercept and alter model weights, enabling them to systematically inject errors or biases into the model to achieve specific objectives. This active manipulation poses a significant challenge for maintaining the integrity and reliability of FL systems.

Objective: The objective of these attacks is to create a local update that differs from the original update (local model) but not so much that it can be easily detected.

It is possible to plan a model poisoning assault by carefully adjusting the weights of client-submitted local updates. Each client uses its own data to train a local model in a FL environment, and then it sends back modifications to the server. In order to provide a global model update for each layer, these local changes are usually combined layer by layer at the server.

At the t-th communication round, let $W^i_{t,L}$ represent the weight of the L-th layer from the i-th client. An attacker can send malicious weight changes intended to impair the global model's performance if they manage to breach one or more clients. Poor generalization on unseen data might result from this modification, which can have a substantial impact on the aggregated model.

The aggregation of each layer across K clients can be mathematically expressed as follows:

$$\begin{matrix} G^1_t \\ G^2_t \\ \vdots \\ G^L_t \end{matrix} = \begin{cases} \frac{\sum_{i=1}^{K} W^i_{t,1}}{K} \\ \frac{\sum_{i=1}^{K} W^i_{t,2}}{K} \\ \vdots \\ \frac{\sum_{i=1}^{K} W^i_{t,L}}{K} \end{cases} \tag{3.1}$$

In this equation, G^L_t represents the aggregated model weights for the L-th layer at the t-th communication round. Each layer's weights are averaged over all K clients, thus reflecting contributions from both benign and potentially malicious updates. This means that the integrity of the global model relies heavily on the assumption that the majority of clients are honest, which is a key vulnerability in FL systems. The following attack scenarios can be used to access FL network's robustness:

1. **Completely Random Attack (CRA):**In this attack, a client updates the server with random weights. The attacker selects values from the range of actual update values to generate a fresh random update for each communication round. The randomness of the update is determined using the Mersenne Twister algorithm, a widely used pseudorandom number generator known for its high-quality randomness and long period [221]. The Mersenne Twister algorithm produces a sequence of numbers that approximates the properties of random numbers. It is particularly effective because it can generate a large number of random values quickly, making it suitable for applications where high efficiency and uniform distribution are crucial. By leveraging this algorithm, the attacker ensures that the updates appear legitimate, making it difficult for the server to detect the manipulation. Each round of communication introduces new random updates, further complicating efforts to identify the source of the bias while allowing the attacker to systematically degrade the model's performance. If one client provides random updates while the others send legitimate up-

dates, Eq. 3.1 becomes:

$$\begin{matrix} G_t^1 \\ G_t^2 \\ \vdots \\ G_t^L \end{matrix} = \begin{cases} \frac{\text{random}(W_{t,1}^1)+\sum_{i=2}^{K} W_{t,1}^i}{K} \\ \frac{\text{random}(W_{t,2}^1)+\sum_{i=2}^{K} W_{t,2}^i}{K} \\ \vdots \\ \frac{\text{random}(W_{t,L}^1)+\sum_{i=2}^{K} W_{t,L}^i}{K} \end{cases} \tag{3.2}$$

2. **Partially Random Attack (PRA):** In PRA, an adversary scales the local update by a random seed before transmission, making the updates look similar but adversarially modified. With one client sending partially random updates and others legitimate updates, Eq. 3.1 becomes:

$$\begin{matrix} G_t^1 \\ G_t^2 \\ \vdots \\ G_t^L \end{matrix} = \begin{cases} \frac{(\mathbf{r}\times W_{t,1}^1)+\sum_{i=2}^{K} W_{t,1}^i}{K} \\ \frac{(\mathbf{r}\times W_{t,2}^1)+\sum_{i=2}^{K} W_{t,2}^i}{K} \\ \vdots \\ \frac{(\mathbf{r}\times W_{t,L}^1)+\sum_{i=2}^{K} W_{t,L}^i}{K} \end{cases} \tag{3.3}$$

where $\mathbf{r}$ is a random seed, updated per communication round using the Mersenne Twister algorithm [221]. By adjusting $\mathbf{r}$, the adversary can control the degree of deviation in the update, enabling a finer control over the attack's impact.

3. **Model Flipping Attack (MFA):** In MFA, the adversary flips the original update by multiplying it by -1 before sending it to the server [94]. If only one client sends a flipped update, Eq. 3.1 is updated as follows:

$$\begin{matrix} G_t^1 \\ G_t^2 \\ \vdots \\ G_t^L \end{matrix} = \begin{cases} \frac{1}{K}\left((-1)\times W_{t,1}^1 + \sum_{i=2}^{K} W_{t,1}^i\right) \\ \frac{1}{K}\left((-1)\times W_{t,2}^1 + \sum_{i=2}^{K} W_{t,2}^i\right) \\ \vdots \\ \frac{1}{K}\left((-1)\times W_{t,L}^1 + \sum_{i=2}^{K} W_{t,L}^i\right) \end{cases} \tag{3.4}$$

This approach, often referred to as "model flipping," can significantly degrade model accuracy by reversing the learning progress contributed by legitimate clients.

4. **Perturbed Attack (PA):** In PA, an adversarial client introduces controlled random perturbations to the updates, generated using a Gaussian distribution. Let P_1, P_2, and P_L represent the perturbation matrices for respective layers, where the perturbation for the first layer is defined as:

$$P_1 \sim \mathcal{N}(0, \sigma^2 I) \tag{3.5}$$

where $\mathcal{N}(0, \sigma^2 I)$ is a multivariate normal distribution with mean 0, covariance matrix $\sigma^2 I$, and I as the identity matrix with dimensions matching $W^1_{t,1}$. Eq. 3.1 then updates as:

$$\begin{matrix} G_t^1 \\ G_t^2 \\ \vdots \\ G_t^L \end{matrix} = \begin{cases} \frac{(P_1 + W^1_{t,1}) + \sum_{i=2}^{K} W^i_{t,1}}{K} \\ \frac{(P_2 + W^1_{t,2}) + \sum_{i=2}^{K} W^i_{t,2}}{K} \\ \vdots \\ \frac{(P_L + W^1_{t,L}) + \sum_{i=2}^{K} W^i_{t,L}}{K} \end{cases} \tag{3.6}$$

Through such perturbations, an adversary can introduce subtle but effective alterations to the model, potentially reducing its accuracy or shifting the model's decision boundary.

3.1.1 Experiments and Results

The HUE dataset (Dataset 1), obtained from the Harvard Dataverse and released by Stephen Makonin [222], was used to assess the effects of various model poisoning techniques. Comprehensive energy usage data from residential customers of BCHydro, a regional electricity provider in British Columbia, Canada, is included in this dataset. Time-series data records trends of energy use for different homes over a given time period and is part of the HUE dataset. Each household exhibits a unique energy consumption profile, influenced by several factors such as household size, the number and types of electrical appliances used, and individual daily routines. Typically, most energy consumption values fall within the range of 0 to 5 kWh, reflecting the variations in usage across different times of the day and week. An example of the data distribution is illustrated in Fig. 3.1, showcasing the typical consumption trends observed in the dataset.

Preprocessing: I used a rolling mean with a window of five to stabilize the energy use data in order to minimize swings and improve trend clarity. Because raw data might be noisy and contain transient anomalies that could mask underlying trends, this preprocessing phase is essential. We can examine consumption trends more precisely and produce more trustworthy forecasts by smoothing the data. Five features—the value from the previous hour, the previous 24 hours, the previous week, and the averages for the preceding hour and week—were extracted from the processed data [24]. A thorough picture of energy use across time is provided by these aspects, which aid in capturing both short-term and long-term consumption trends.

Deep Learning Model: For time-series forecasting, a three-layer neural network was created, comprising a 32-neuron Long Short-Term Memory (LSTM) layer, two dense layers of 28 neurons each, and a single neuron. To maximize model efficiency, a 12-hour look-back

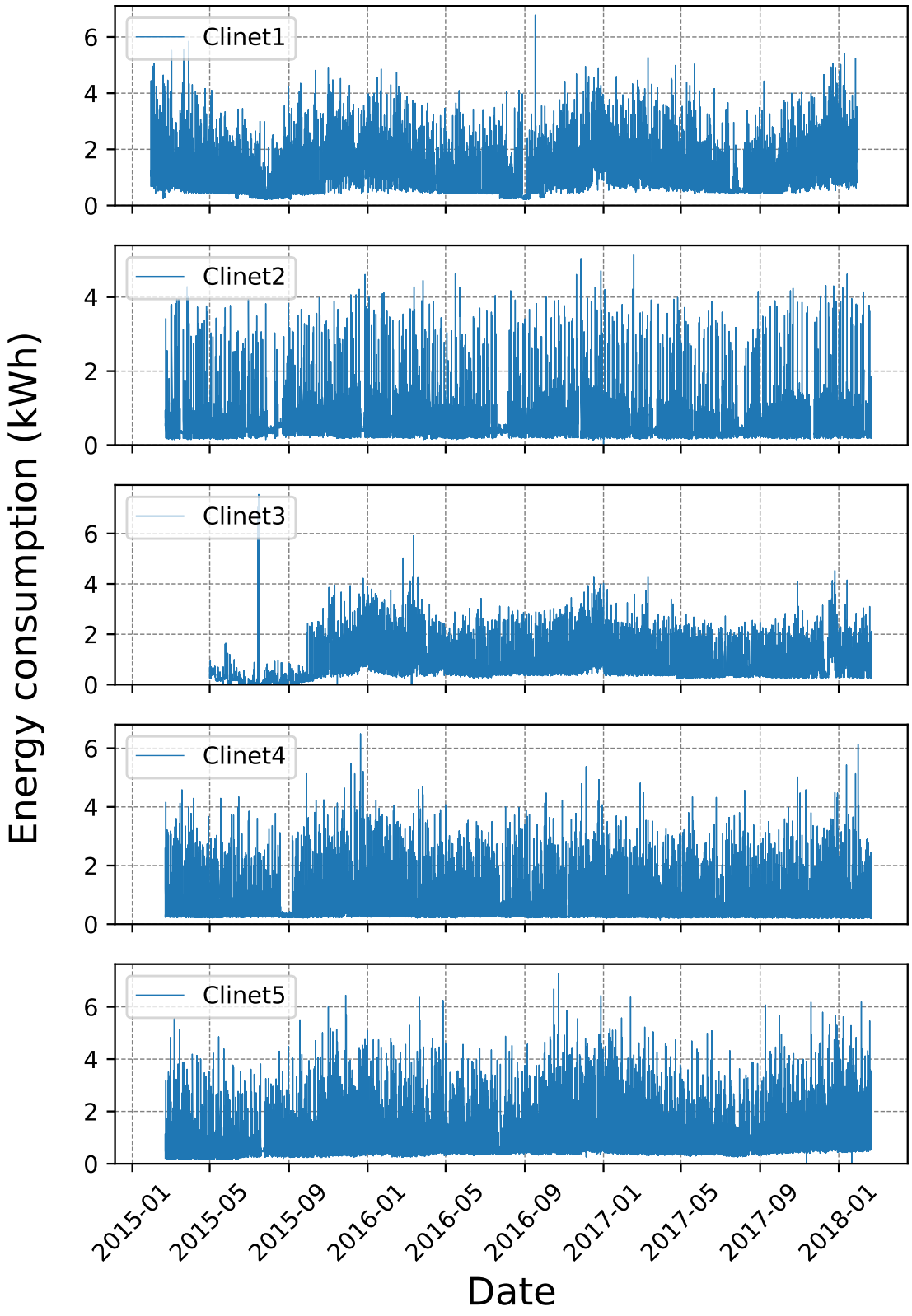


Figure 3.1: Overview of dataset 1.

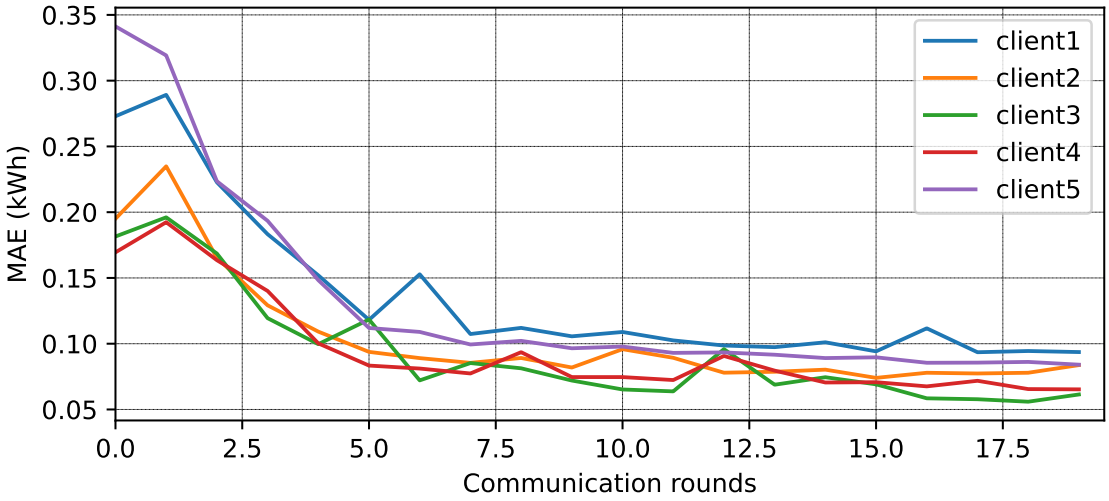


Figure 3.2: MAE during FL training process of global model on Dataset 1.

window was chosen after experimenting with different configurations. Using a grid search technique, hyperparameters such as neuron count, learning rate, batch size, and dropout rate were methodically tuned. To ensure that the model is well-tuned to capture the subtleties of energy consumption patterns, it uses the Adam optimizer for efficient convergence with adaptive learning rates, the Rectified Linear Unit (ReLU) activation function to introduce non-linearity, and mean squared error (MSE) loss to measure prediction accuracy.

Baseline results: The MAE was plotted during the 20 communication rounds of the FL training process, as illustrated in Fig. 3.2. FedAvg serves as the aggregate framework in this case. By the end of the training procedure, it was evident that all of the clients' MAEs had converged. 0.076 kWh was the average client MAE.

As observed, the MAE of most clients exhibits a downward trend with an increasing number of communication rounds, reflecting the gradual improvement in the global model's performance. However, fluctuations in the MAE values of different clients can be attributed to several factors. Firstly, data heterogeneity is a significant concern in practical FL scenarios. Clients often possess non-IID data, and variations in data distributions can lead to differing model performances. Clients with more representative data may achieve lower MAE values more rapidly, while those with significantly deviating data may experience slower convergence and higher MAE values.

Effect of model poisoning updates:

The designed attacks called completely random attack (CRA), partially random attack (PRA), model flipping attack (MFA) and perturbed attack (PA) are applied on the weights of local models. This effect is presented in Fig. 3.3. The density plots associated with each attack reveal distinct patterns in how they influence the model's output distributions. In the MFA, the density plot shows a noticeable shift, resulting in a bimodal distribution where certain predictions are significantly altered. This highlights the model's vulnerability to targeted manipulations. Con-

versely, the CRA presents a nearly uniform distribution, indicating that the model struggles to discern any meaningful patterns, thereby diluting its predictive power.

The PRA offers a more complex landscape, as the density plot demonstrates concentrated distributions while others remain dispersed. This indicates that the attack selectively impacts certain predictions, exploiting the model's weaknesses without overwhelming it with noise. Finally, the PA shows subtle shifts in the output distribution. This suggests that the weight modifications have a targeted effect, altering predictions in a manner that may not be immediately apparent but can lead to significant misclassifications.

The effect these attacks are Fig. 3.4. The graph shows the average MAE for all the attacks. It can be observed that the highest impact is caused by PA, resulting in an MAE of 0.270 kWh, while the lowest impact is caused by PRA, with an MAE of 0.088 kWh. This value is very close to the baseline (no-attack) condition of 0.076 kWh. The results reveal that the PA induces the highest MAPE, while the PRA has the least impact. This disparity can be explained by the distinct mechanisms of these attack strategies. The PA introduces controlled random perturbations to the model updates. Although these perturbations are subtle, they can accumulate over time and mislead the model's learning direction toward incorrect patterns. By directly targeting model parameters and introducing targeted disturbances, PA significantly affects model accuracy, leading to biased predictions for certain data features and a higher MAPE. In contrast, the PRA only randomizes a subset of model parameters, introducing noise without entirely disrupting the model's overall structure. Consequently, the impact on model accuracy is relatively limited, as the unaffected parameters can still guide the model toward the correct optimization direction to some extent, resulting in a lower MAPE. The comparison of MAPE values across different attack types underscores the varying threats posed by different attack strategies to FL systems. It emphasizes the need for attack-specific defense mechanisms. For instance, advanced anomaly detection techniques and robust aggregation algorithms are required to address subtle perturbations introduced by PA, while data validation and model parameter filtering can mitigate the effects of PRA. Overall, this figure provides a quantitative comparison of the effectiveness of different model poisoning attacks, offering valuable insights for the design of defense frameworks to enhance the security and robustness of FL systems.

3.2 Stealth Communication Round Attack (Fed-CRA)

The attacks presented in previous sections are primarily concerned with performance degradation in the system; however, this attack focuses on the resources of the system, which include time, energy, and communication resources. Since it does not affect model performance, it is categorized as a stealth attack, addressing the gap presented in Section 2.5.3 and Table 2.1.

Objective: Fed-CRA's main goal is to carry out a model poisoning attack that preserves the forecasting error while increasing the number of communication cycles between the server and

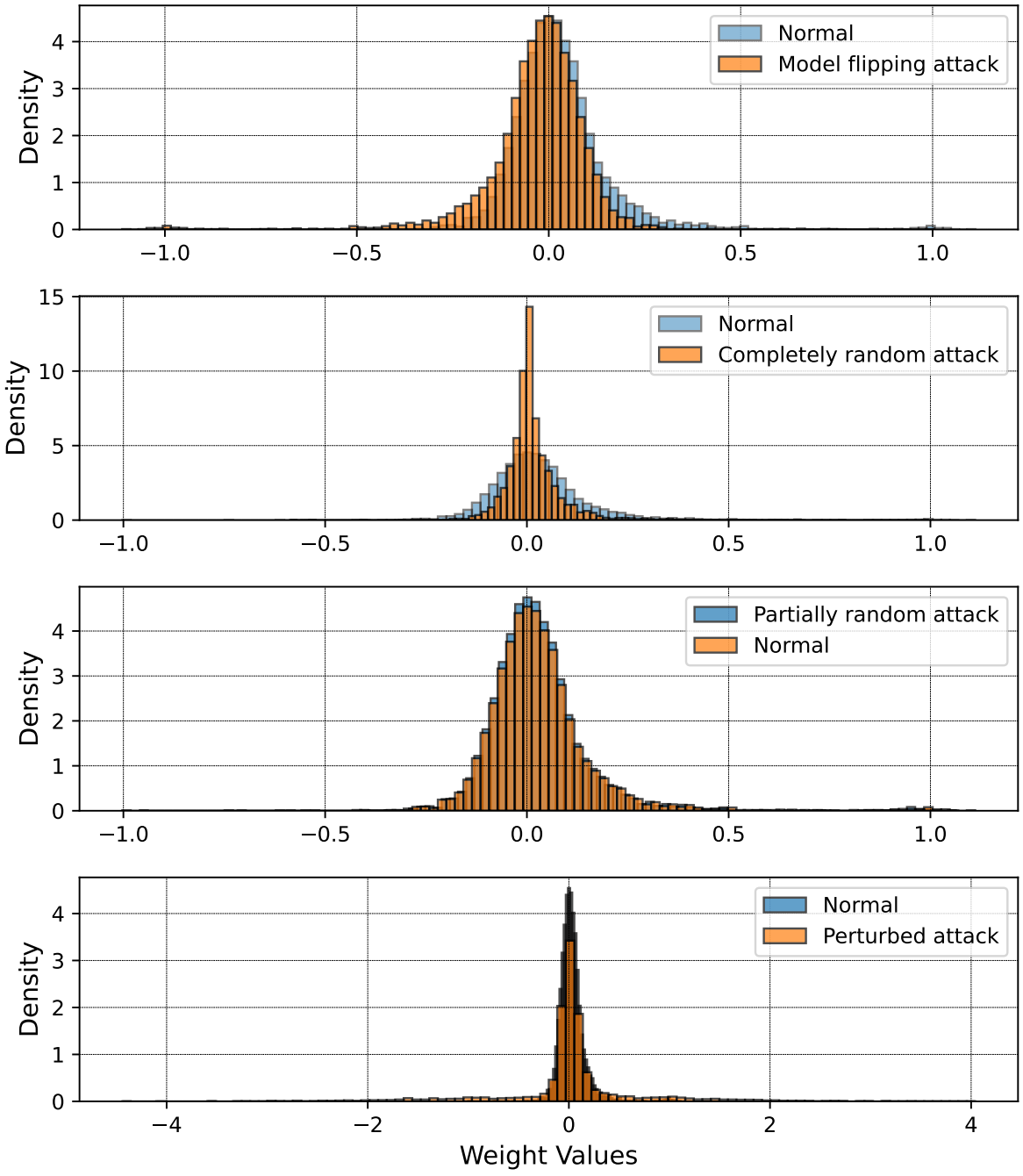


Figure 3.3: Examples of all attacks include: actual and attacked weights.

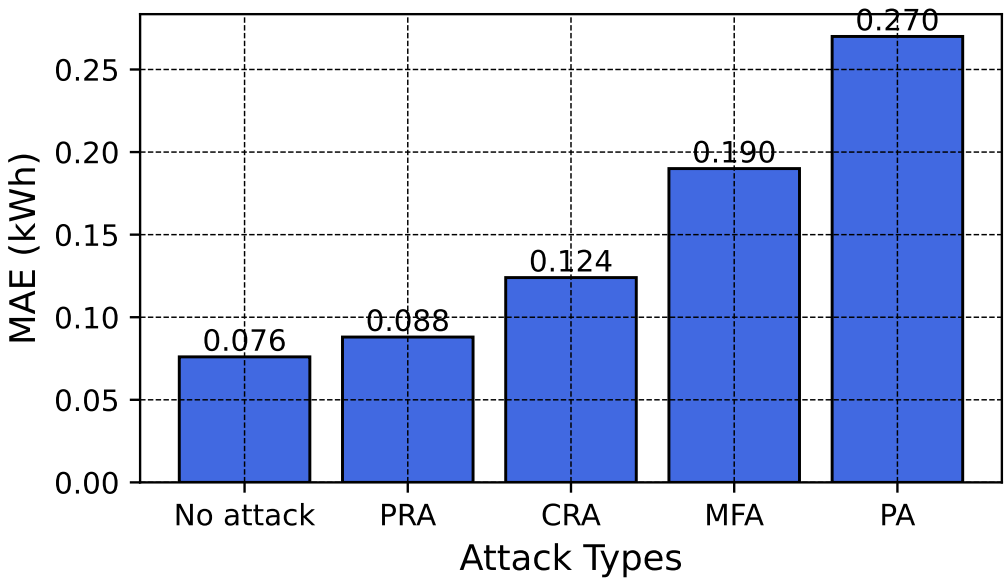


Figure 3.4: Effect of different model posing attacks when client 1 was subjected to attack.

clients. This method requires more energy to be used when training the global model because it increases these communication rounds. Since there is no discernible drop in model accuracy, this feature enables the attack to be categorized as a stealth attack, which makes it challenging to identify.

It is presumed in this scenario that the attacker does not have direct access to the clients' data. Therefore, the attacker's only realistic option is to change the weights of the local models that the clients have submitted. This technique is similar to those used in Man-in-the-Middle (MITM) attacks, in which an interceptor obtains unapproved access to the channel of communication. With this access, the attacker can intercept client model updates and substitute them with malicious updates intended to undermine the integrity of the model as a whole. Wang et al. [223] provide a comprehensive discussion on this form of attack, emphasizing that attackers can adopt two primary approaches: they can either alter the updates in real networks or construct fictitious networks under their control to facilitate their schemes. By leveraging these tactics, the attacker can effectively manipulate the learning process without raising immediate suspicion, thereby undermining the efficacy of the FL system.

A common strategy employed in MITM attacks involves the removal of encryption from the compromised communications, thereby enabling the attacker to observe, modify, or redirect the model updates being exchanged between clients and the server [224]. The difficulty in identifying such attacks arises from the fact that the attacker might not only monitor the updates but may also re-encrypt them before forwarding to the intended recipient, thereby concealing their activities. Additionally, this kind of interference can be categorized as a backdoor attack, wherein the client becomes compromised, and the attacker gains control over the training process [113,115].

In my investigation, i deliberately chose not to include attacks that could potentially af-

fect model accuracy [97, 225]. The rationale behind this decision is that state-of-the-art defense frameworks are adept at identifying and mitigating such accuracy-impacting attacks [94,97,134]. Instead, our focus was directed toward designing a targeted attack [124], specifically aimed at increasing the communication rounds rather than jeopardizing the accuracy of the global model. This strategy not only preserves the model's integrity but also results in increased energy consumption during the training phase, as the server and clients engage in more frequent exchanges of information.

To implement this attack strategy, we constructed malicious updates using a random seed, denoted as $\mathbf{r}$. This seed is generated within a specified range, between 0.5 and 1. The approach involves modifying only the biased portion of the first layer in the compromised local model, while ensuring that the weights in other layers remain unchanged. This method is critical because incorporating the seed into more than one layer could inadvertently compromise the stealthiness of the attack. We aimed to craft updates that closely mirror actual model updates to avoid detection, as completely random updates can be quickly flagged by state-of-the-art detection systems, as noted in [94, 111].

To ensure the stealth of the attack, we deliberately avoided using entirely random updates, which are easily detectable by modern defense mechanisms. Instead, we concentrated on formulating an update that is similar to the legitimate model updates, thereby ensuring that the malicious alteration does not affect the forecasting error. The random seed is generated using the Mersenne Twister (MT) algorithm [221], which is a widely recognized pseudorandom number generator utilized across various computational and scientific applications. The MT algorithm is celebrated for its robust statistical properties and its capacity to generate an extensive sequence of numbers before repeating, which is essential for maintaining the stealthiness of Fed-CRA [226–228].

The seed value, which is the initial input used by the Mersenne Twister algorithm [221], is where the number creation process starts. This algorithm employs a highly intricate mathematical formula that transforms the seed value into a seemingly random sequence of numbers, thus providing the randomness needed for our updates. Consequently, we can modify Equation 1 to reflect this attack mechanism as follows:

$$G_t^k = \begin{Bmatrix} G_t^1 \\ G_t^2 \\ \vdots \\ G_t^L \end{Bmatrix} = \begin{Bmatrix} \frac{\sum_{i=1}^{K} W_{t,1}^i}{K} \\ \frac{\sum_{i=1}^{K} W_{t,2}^i}{K} \\ \vdots \\ \frac{\sum_{i=1}^{K} W_{t,L}^i}{K} \end{Bmatrix}, \begin{Bmatrix} \frac{\sum_{i=1}^{K} B_{t,1}^i \times \mathbf{r}}{K} \\ \frac{\sum_{i=1}^{K} B_{t,2}^i}{K} \\ \vdots \\ \frac{\sum_{i=1}^{K} B_{t,L}^i}{K} \end{Bmatrix} \tag{3.7}$$

In this equation, G_t^k represents the global model parameters at time t, with the first part indicating the aggregation of model weights from all clients, while the second part incorporates the biased updates influenced by the random seed $\mathbf{r}$. This carefully crafted update mechanism is

central to the effectiveness and stealth of the Fed-CRA attack.

3.2.1 Experimental Results

The dataset utilized in this section was sourced from PJM Interconnection LLC [229], which provided a collection of ten datasets corresponding to ten distinct substations across their grid. For this study, Dataset 2, referred to as APE, was selected, as illustrated in Fig. 3.5. The data was organized into ten clients, with each client allocated 2,267 data samples, allowing for a decentralized approach to analysis and modeling. This dataset is specifically employed for STLF, a crucial task for energy management and planning.

To enhance the accuracy of future energy demand predictions, five relevant features were generated, as detailed in previous studies [24,93,94]. These features are designed to capture various temporal aspects of energy consumption and include: (1) the previous hour's consumption value, which provides immediate context; (2) the previous day's consumption value, offering insights into daily trends; (3) the previous week's consumption value, allowing for the identification of weekly patterns; (4) the 24-hour average value, which smooths out short-term fluctuations; and (5) the weekly average value, which helps in understanding longer-term consumption behaviors. These features collectively enhance the model's ability to predict short-term energy needs effectively.

Deep Learning Model: A three-layered artificial neural network (ANN), which is intended to identify intricate patterns in the data, was used to anticipate the energy demand for both datasets. With 100 neurons in the first layer, the model can learn a wide variety of features from the input data and successfully capture complex correlations in the patterns of energy usage. The model's capacity to generalize was improved by the second layer, which had 50 neurons and functioned as a hidden layer that improved the features that the first layer had retrieved and helped to reduce dimensionality. One neuron, which produces the anticipated energy demand value, makes up the last layer. Each layer made use of the Rectified Linear Unit (ReLU) activation function, which is well-known for its ability to add non-linearity to the model and facilitate quicker training convergence. Because of its flexible learning rate capabilities, which aid in effectively navigating the loss terrain during training, the Adam optimizer was used. The model was adjusted to reduce mistakes in energy demand projections by using mean squared error (MSE), which provides a measure of the average squared differences between anticipated and actual values. This architecture is appropriate for short-term load forecasting activities because it was created to improve predicted accuracy and robustness.

Global Dataset: o provide a comprehensive overview of the entire data landscape, the global dataset for this simulation is constructed by aggregating 10% of the data from each client. This method enhances the assessment of the global model's predictive accuracy by incorporating variances from multiple sources, ensuring an objective evaluation of the model's performance across different client scenarios. The global dataset serves as a robust benchmark, enabling cen-

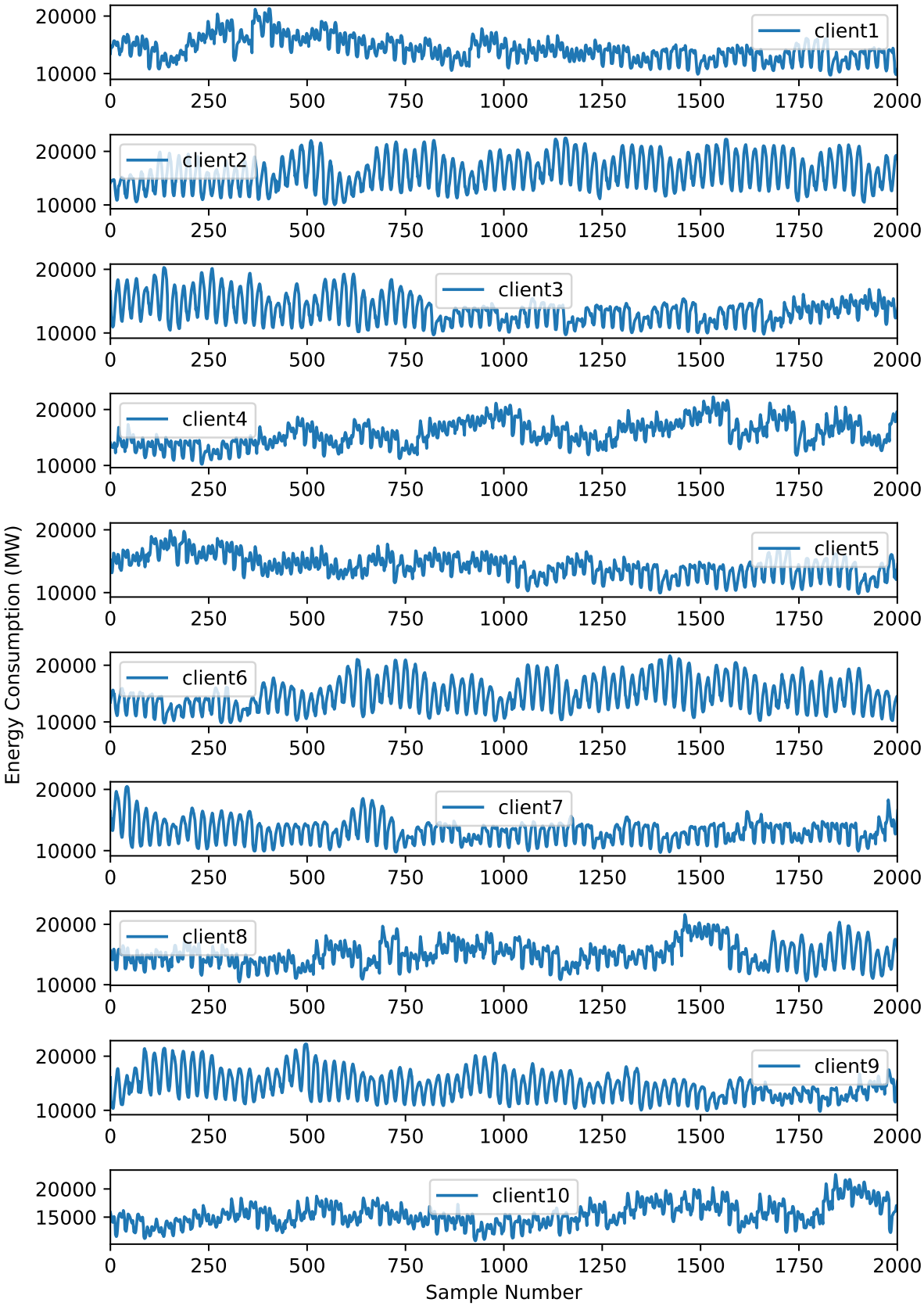


Figure 3.5: Sample of the dataset 2.

tralized access for both training and evaluation stages of the model. By utilizing a portion of each client's data, this approach mitigates potential biases that might arise from relying solely on a single client's dataset, thereby improving the reliability and generalizability of the forecasting results. Additionally, this strategy facilitates a more nuanced comparison of the model's performance against a collective dataset, offering valuable insights into the model's ability to adapt to various operational contexts. The global dataset is securely stored on the server upon construction, ensuring that it remains a trusted reference point throughout the model's development and assessment phases.

Stopping Criterion:A thorough stopping criterion was painstakingly created for this collection of simulations in order to optimally cease the global model's training process. This criterion is essential for figuring out how many communication rounds are best for reaching convergence during the training stage. Prematurely ending the training phase can result in low forecasting accuracy since the model might not have fully mastered the underlying patterns in the data. Conversely, if training continues beyond the point of convergence, it not only wastes computational resources but also risks overfitting, where the model starts to learn noise instead of the true signal. To establish this stopping point, the criterion was defined by closely monitoring the loss function of the global model throughout the training iterations. Specifically, if there is no significant improvement in the loss function over 20 consecutive communication rounds, the training process will automatically cease. This strategy ensures that the model is trained efficiently, striking a balance between achieving high accuracy and conserving valuable resources in the computational environment.

Baseline Results: The mean absolute percentage error (MAPE), which provides a quantitative indicator of predicting accuracy, was computed for both the global model and each individual client following the FL procedure. Fig. 3.6 provides a visual comparison of the global model and client performance indicators, illustrating these findings. Interestingly, after 72 communication rounds, the global model was able to converge, suggesting that the iterative training procedure successfully adjusted the model parameters to reduce predicting errors. The average MAPE for all clients was 2.79%, however the global model showed a great predictive performance with a MAPE of 2.75%. The efficacy of the FL strategy in improving forecasting dependability is demonstrated by this little accuracy difference, which implies that the global model was able to generalize better than individual client models thanks to the combined insights of several clients.

Effect of Fed-CRA: A sample of actual and attacked weight are plotted in Fig. 3.7 showcasing the similarity between actual and attacked weights. This proves the stealthiness of the attack. The stealthiness of the generated attack is evaluated using cosine similarity, which measures the difference between the actual weights and biases of client 1 and client 2, as well as the attacked weights and biases of client 1 compared to client 2's actual weights and biases. The analysis shows that, under normal conditions (with neither client 1 nor client 2 being attacked), the co-

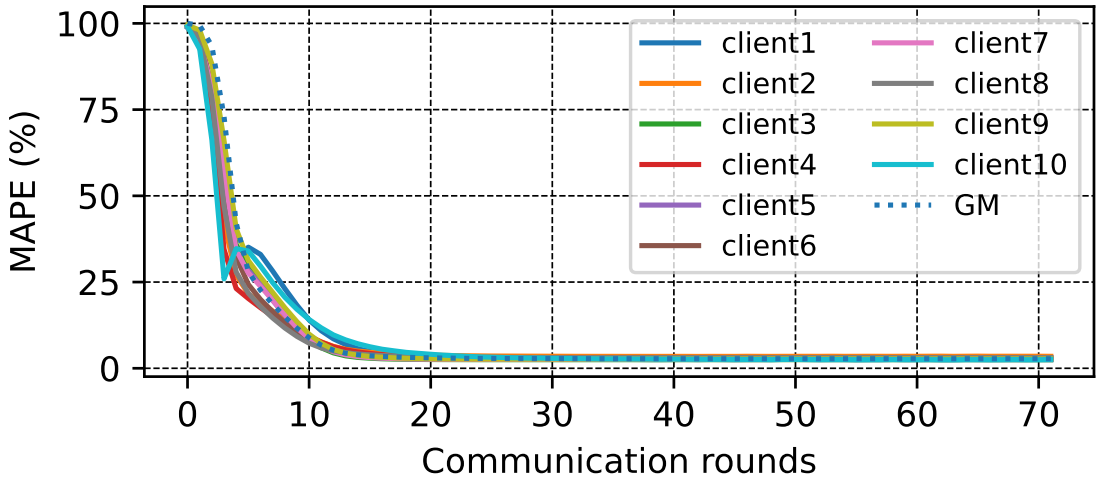


Figure 3.6: Baseline results for dataset 2.

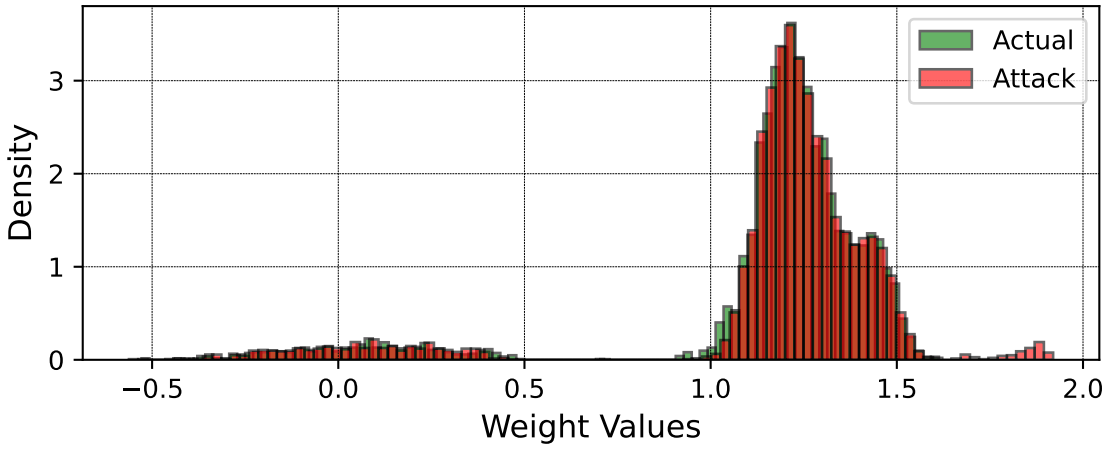


Figure 3.7: A sample of the local model of client one's weight distribution, both as it is and as it was attacked.

sine similarity of their actual weights is 0.99996781. Interestingly, when client 1 is targeted by an attack, the cosine similarity slightly increases to 0.99999588, demonstrating the attack's subtlety and effectiveness.

The percentage of attacked clients was varied from zero to one hundred, and the MAPE of the global model, along with the corresponding communication rounds, was measured. The results are plotted in Fig.3.8. It can be observed that as the percentage of attacked clients increased, the corresponding communication rounds also increased, while the MAPE remained unchanged. This further demonstrates the stealthiness of the attack. Fed-CRA successfully increased the communication rounds from 72 to 485.

Communication Cost: Communication cost, defined as the total data exchanged between clients and the server [165], plays a crucial role in influencing a system's energy efficiency and overall performance [24, 219]. This cost is intricately linked to both the number of commu-

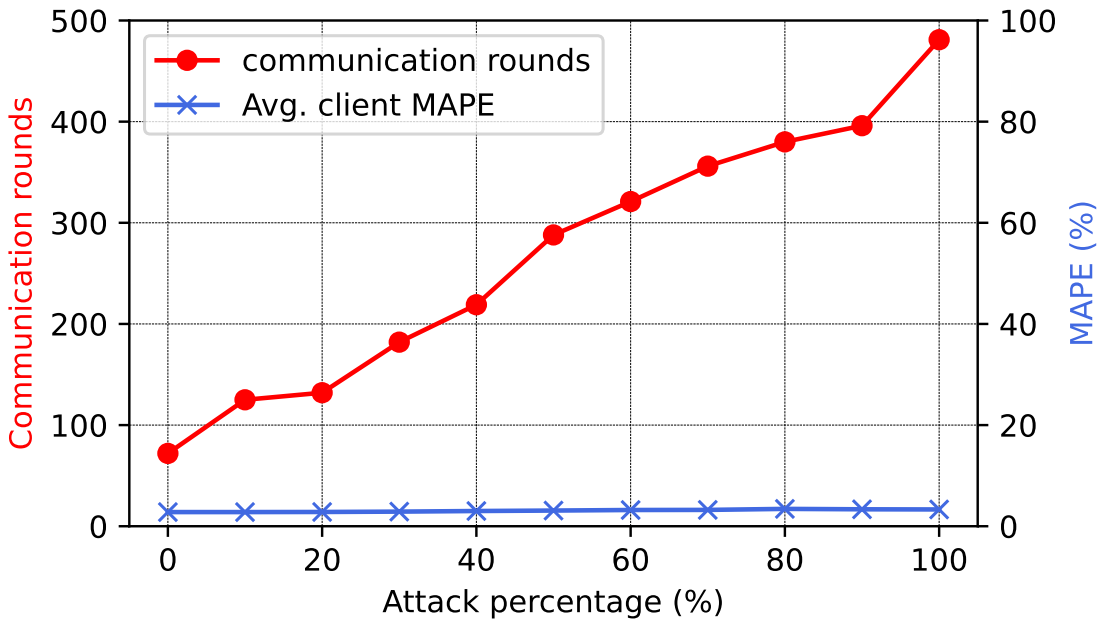


Figure 3.8: Communication rounds (red) and MAPE (blue) resulting from Fed-CRA global model training.

nication rounds and the size of the data exchanged during each round. Notably, the machine learning model is transmitted twice per round: first, from the client device to the server in the form of the local model, and subsequently, the global model is sent back to the clients. In the baseline scenario, which consists of 72 communication rounds, the total communication overhead amounts to 5.4 MB for a model size of 38 KB. However, under the Fed-CRA framework, when all clients are compromised, the number of required communication rounds escalates to 485, which significantly increases the communication overhead to 36.86 MB. This represents a staggering 582.59% increase in communication costs, highlighting the substantial impact that the choice of communication strategy can have on resource consumption and efficiency in FL environments.

Energy consumption: Energy consumption (E_{com}) for transmitting local models depends on factors like energy per kilobyte, transfer time, and device type. Using [219]'s model, E_{com} is calculated as:

$$E_{com} = R[(\alpha \times t) + (\beta \times D)] \tag{3.8}$$

where R denotes the number of communication rounds, $\alpha = 0.0001$ kWh/sec represents energy per second, $t = 1$ ms is the transmission time, $\beta = 0.015$ kWh/GB is the energy per kilobyte, and D is the data size (38 kB in this case). For baseline FL, E_{com} was 41.04 kWh per device. Under Fed-CRA, when all clients are compromised, E_{com} increased to 276.35 kWh. This elevated demand raises concerns about cost, resource strain, and sustainability.

3.3 Discussion

The varying impacts of these attacks on the FL system also reflect differences in the ability of defense mechanisms to address them. Model poisoning attacks, such as the PRA and CRA, directly target the integrity of the model updates. Traditional defense mechanisms can often identify and mitigate these attacks to some extent, especially when the attacks are not highly sophisticated. However, for more advanced model poisoning attacks like the PA, which introduce subtle but targeted perturbations, defense mechanisms need to be more sophisticated. Advanced anomaly detection techniques and robust aggregation methods are better suited to handle such attacks. These frameworks can detect anomalies in the model updates and either exclude or down-weight the malicious updates, thereby maintaining the integrity of the global model.

The MFA presents a different challenge. It reverses the learning direction of the model, which can be difficult for some defense mechanisms to detect as it may not necessarily increase the model's prediction error but rather misdirects the model's learning. Defense mechanisms that monitor the direction of model updates and the consistency of the model's behavior over time are more effective against MFA.

The Fed-CRA is unique in that it does not directly affect model accuracy but instead increases communication overhead. Defense mechanisms targeting Fed-CRA need to focus on identifying abnormal communication patterns rather than model update content. Mechanisms that monitor communication frequency and data transmission volumes can detect such attacks. For example, by setting thresholds for acceptable communication rounds and data sizes, any deviation beyond these thresholds can trigger investigative or corrective actions to mitigate the attack's impact.

The ability of defense mechanisms to cope with different types of attacks varies. Basic defense mechanisms may suffice for simple model poisoning attacks, but advanced and targeted defense strategies are required to effectively address sophisticated attacks like Perturbed Attacks and Model Flipping Attacks. For stealthy attacks like Fed-CRA, defense mechanisms need to shift their focus from model accuracy to communication patterns and resource usage. This discussion highlights the importance of selecting and implementing appropriate defense mechanisms based on the specific types of attacks that an FL system may encounter.

3.4 Concluding Remarks

In this chapter, various attack strategies that can compromise FL systems have been explored, with a particular focus on model poisoning attacks. These attacks aim to degrade the performance of the global model by introducing malicious updates from compromised clients. The key findings and contributions can be summarized as follows:

- **Performance Degrading Attacks:** Several types of model poisoning attacks, including

Completely Random Attack (CRA), Partially Random Attack (PRA), Model Flipping Attack (MFA), and Perturbed Attack (PA), were introduced and evaluated. These attacks simulate real-world scenarios where malicious clients can manipulate their local updates to corrupt the global model. The experiments demonstrated the significant impact these attacks can have on model performance, thereby underscoring the necessity for robust defence mechanisms to address these vulnerabilities. This addresses the challenge of model attacks (C1).

- **Stealth Communication Round Attack (Fed-CRA):** A novel attack, Fed-CRA, was proposed. This attack aims to increase the number of communication rounds between the server and clients while keeping the accuracy unchanged. Designed to create inefficiencies in the learning process without being easily detected, the results indicated that Fed-CRA can significantly increase communication overhead, thus addressing the challenge of communication and computational efficiency (C3).

- **Impact on Model Performance:** The experimental results revealed that the highest impact on model performance was caused by the Perturbed Attack (PA), resulting in an MAE of 0.270 kWh, while the lowest impact was caused by the Partially Random Attack (PRA), with an MAE of 0.088 kWh. This highlighted the varying degrees of vulnerability of FL systems to different types of attacks and emphasized the importance of developing defence mechanisms that can effectively mitigate these threats.

Additional Insights: Some attacks, such as the Perturbed Attack (PA), resulted in high MAE but were relatively easier to detect due to their overt impact on model performance. In contrast, Fed-CRA caused significant operational overhead without triggering detection mechanisms, thereby emphasizing the need for defence strategies specifically tailored to stealthy versus blatant attacks. In STLF applications, where communication latency and forecasting precision are critical, attacks like Fed-CRA could substantially inflate operational costs while escaping anomaly detectors, thus posing risks to grid reliability and efficiency.

The findings from this chapter underscore the critical need for robust defence frameworks to protect FL systems against adversarial attacks. The proposed attack strategies provide valuable insights into the vulnerabilities of FL systems and highlight the importance of developing advanced anomaly detection and mitigation techniques. Future research directions could focus on designing defence mechanisms that can effectively detect and neutralize both overt and stealthy attacks, ensuring the integrity, efficiency, and reliability of FL systems in real-world applications, particularly in critical domains such as energy management.

Chapter 4

Novel Attack Resolution Frameworks

FL systems offer significant advantages in terms of privacy preservation and decentralized training. However, they remain susceptible to various types of attacks, particularly model poisoning attacks. As highlighted in Chapter 1, these attacks can compromise the integrity of the global model by subtly manipulating local model updates, leading to degraded performance and inaccurate predictions. The challenges posed by such adversarial activities are further compounded by the need to balance communication efficiency and computational constraints, as outlined in Section 1.2 (Challenges C2 and C3).

This chapter introduces three novel defense frameworks designed to address these challenges and enhance the robustness of FL systems in real-world applications:

- **Federated Random Layer Aggregation (FedRLA)**: FedRLA directly targets Challenge C3 by reducing communication overhead while maintaining model accuracy. This framework enhances global model training efficiency and defends against adversarial attacks by aggregating only a single, randomly chosen neural network layer during each communication round. This method reduces data exchange between devices and the server, streamlining communication and enhancing privacy. Experimental results demonstrate that FedRLA achieves comparable model accuracy to traditional methods while reducing communication costs by a factor of 3.56.

- **Layer-Based Anomaly Aware Federated Averaging (LBAA-FedAvg)**: LBAA-FedAvg addresses Challenge C2 by introducing a robust aggregation mechanism that detects and mitigates adversarial updates. It leverages anomaly detection to identify deviations in model updates caused by adversarial clients. By clustering the weights of each layer and selectively excluding compromised layers from the aggregation process, LBAA-FedAvg ensures the integrity of the global model. Experiments show that LBAA-FedAvg maintains a stable average client MAPE even under varying attack scenarios.

- **Federated Incentivized Averaging (Fed-InA)**: Fed-InA focuses on Challenges C2 and C3 by developing a novel scoring mechanism to identify and neutralize stealth attacks,

thereby reducing communication costs. This framework introduces a scoring mechanism that evaluates clients based on their contribution to the model's accuracy and reliability. By rewarding good clients and penalizing malicious ones, Fed-InA encourages honest participation and contributes to the overall integrity and performance of the FL system. Experimental results indicate that Fed-InA significantly reduces communication rounds while maintaining model accuracy, even in the presence of adversarial attacks.

The need for such defense frameworks is further underscored by the research gap identified in Section 2.2.5, where existing defense mechanisms were found to be inadequate against sophisticated stealth attacks. As discussed in Section 2.10, traditional defense frameworks often struggle with detecting and mitigating stealth attacks that subtly manipulate model parameters over time. The proposed frameworks in this chapter aim to fill this gap by providing advanced defense strategies specifically designed to counteract such threats. Each framework is designed to complement the others, providing a comprehensive defense strategy against the diverse threats facing FL systems. By explicitly addressing the challenges and building on the insights from the literature survey in, these frameworks contribute to the development of more secure, efficient, and reliable FL systems, particularly in sensitive applications such as short-term load forecasting (STLF) where model integrity and user privacy are paramount.

4.1 Federated Random Layer Aggregation

FedRLA is introduced as a novel approach to enhance global model training efficiency and defend against adversarial attacks. Unlike traditional FedAvg, FedRLA aggregates a single, randomly chosen neural network layer during each communication round, leaving other layers unchanged. This reduces data exchange between devices and the server, streamlines communication, and enhances privacy by limiting shared model information. The server dynamically selects the layer to aggregate, ensuring all layers contribute over successive rounds, with randomness provided by the robust Mersenne Twister (MT) algorithm [221]. This method improves security, resource efficiency, and privacy, making it ideal for privacy-sensitive applications such as household energy forecasting. Details of the framework are provided in Algorithm **??** and illustrated in Fig. 4.1.The training process for energy forecasting using FedRLA follows five key steps:

1. **Sending Generic Models to Clients:** The server initiates the process by distributing a generic ML model comprising L layers to all K edge devices. This model is meticulously designed based on historical data and expert insights, ensuring that it incorporates foundational knowledge relevant to the specific task. Additionally, the server includes information specifying which layer, denoted as L_r, should be returned after local training, allowing for targeted updates.

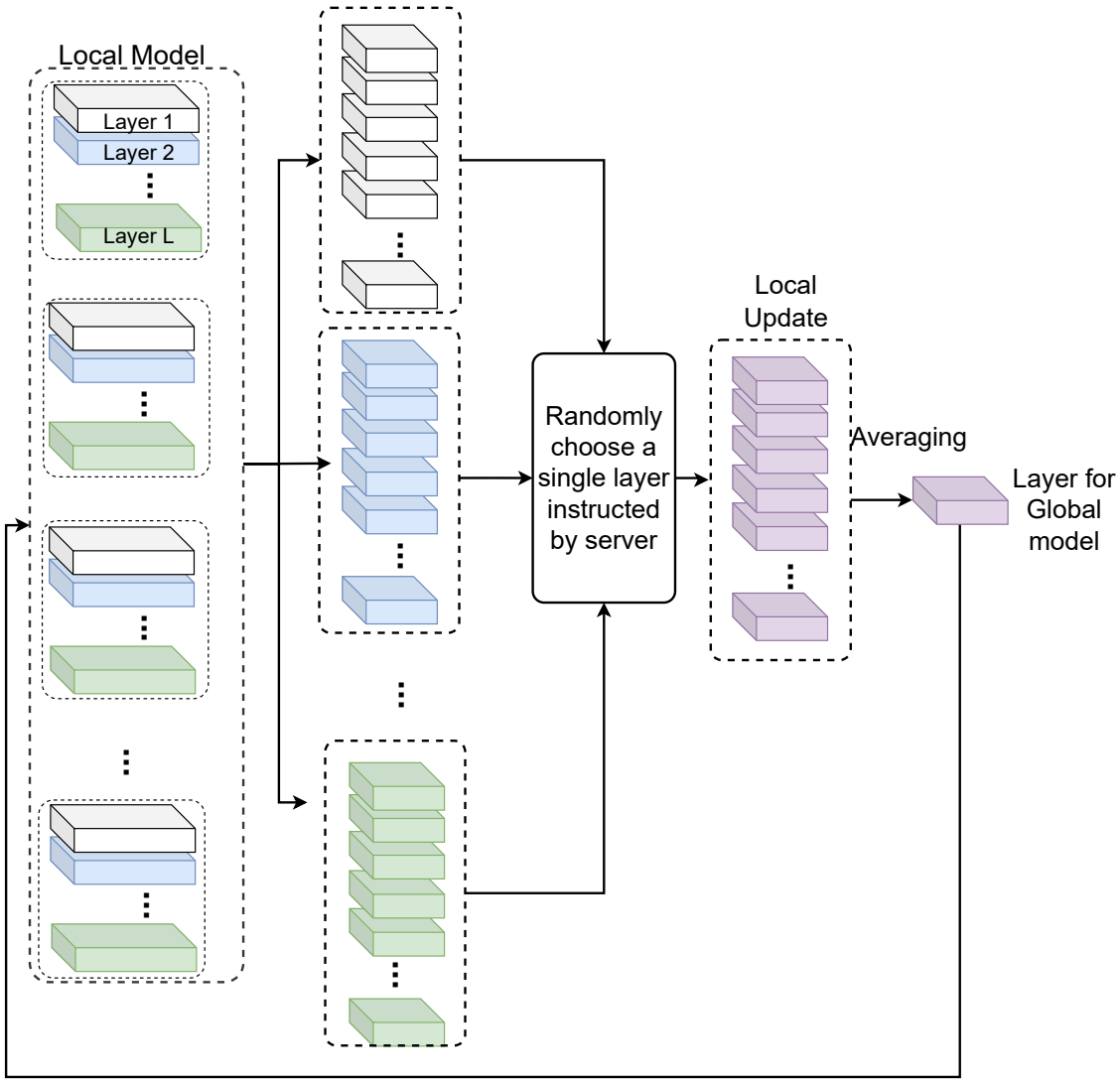


Figure 4.1: Schematic diagram of FedRLA process.

2. **Local Model Training:** Each edge device independently trains the received model using its unique energy consumption data. This step is crucial as it ensures the development of personalized local models that accurately reflect the individual behaviors and consumption patterns of different households, thereby enhancing the relevance and effectiveness of the predictions.

3. **Model Update and Loss Calculation:** Upon completing the training phase, the edge devices transmit their local model's weights and biases, represented as $UP = \{W_{t,r}^1, W_{t,r}^2, \ldots, W_{t,r}^K\}$, back to the server. Additionally, they send the corresponding loss functions, denoted as σ_k, which quantify the performance of each local model. Here, $W_{t,r}^K$ specifically refers to the weights of layer L_r for the Kth client, providing insight into the model's learning progress.

4. **Model Aggregation:** The server aggregates the weights received from the local models

to formulate the global model, denoted as G_t. This aggregation employs the Federated Averaging (FedAvg) algorithm, but it is applied solely to the selected random layer L_r:

$$G_t = \frac{\sum_{i=1}^{K} W_{t,l_{\text{rand}}}^{i}}{K}$$

In this equation, $W_{t,l_{\text{rand}}}^{i}$ represents the weights of the randomly chosen layer l_{rand} from the ith client, ensuring that the global model benefits from diverse local insights while maintaining focus on specific layers.

5. **Global Model Sharing:** In the end, the server completes one communication round by sharing the updated global model G_t with every edge device. By repeating this iterative process, the accuracy of the global model can be continuously improved over subsequent rounds, ultimately improving the predictive power of the system.

This framework is outlined in Algorithm 1 for client side and 2 for server side and visually represented in Fig. 4.1.

Algorithm 1 Federated Random Layer Aggregation (FedRLA) - Client Side

Initialization:
Initialize local model $W_{0,l}^{k}$ for all layers l.
Receive global model and l_{rand} from the server.
for $t = 1$ to T **do**
    Local Training and Update:
    Update local model using local data.
    Create an update for the randomly selected layer: $W_{t,l_{\text{rand}}}^{k} \leftarrow \text{LocalUpdate}(W_{t,l_{\text{rand}}}^{k})$
    Send $W_{t,l_{\text{rand}}}^{k}$ back to the server.
end for

Privacy Enhancement: The privacy of the model is significantly enhanced in FedRLA through the selective aggregation of a single random layer. Traditional FL methods require clients to transmit their entire model update to the server, which can expose a significant amount of information about the client's local data and model. In contrast, FedRLA limits the exposure by only transmitting a single randomly selected layer. This means that during each communication round, only a fraction of the model's information is sent over the network. By reducing the amount of information transmitted, the risk of data leakage is minimized. Additionally, since the layer to be transmitted is randomly chosen in each round, it becomes more difficult for potential attackers to reconstruct the full model or infer sensitive information from the transmitted data. This randomization adds an element of unpredictability, making it harder for attackers to target specific layers or extract meaningful patterns from the transmitted information.

Furthermore, the selective aggregation approach in FedRLA ensures that the global model is built from diverse contributions across different layers from various clients. This diversity makes

Algorithm 2 Federated Random Layer Aggregation (FedRLA) - Server Side

Initialization:
Initialize global model $W_{0,l}$ for all layers l.
Set communication rounds T, number of clients K, and number of layers L.
Send global model and initial l_{rand} to all clients.
for $t = 1$ to T **do**
  Aggregation:
  for each layer l **do**
    if $l = l_{\text{rand}}$ **then**
      Aggregate updates for the selected layer: $W_{t,l} \leftarrow \frac{1}{K} \sum_{i=1}^{K} W_{t,l_{\text{rand}}}^{i}$
    end if
  end for
  New Instructions:
  Choose new l_{rand} for next communication round.
  Send updated global model and new l_{rand} to all clients.
end for
return Trained global model.

it difficult for any single client's update to have a disproportionate influence on the global model, thereby reducing the risk of model inversion attacks or other privacy-threatening activities. By limiting the information transmitted and introducing randomness in the selection of layers, FedRLA provides a robust defense against privacy breaches while maintaining the efficiency and effectiveness of the FL process.

4.1.1 Experiments and Results

In this set of studies, Dataset 1 was selected due to its robust foundation for examining trends in energy usage. A rolling mean of 5 was applied to the dataset to mitigate fluctuations and stabilize the energy use statistics. This technique effectively smoothed out short-term anomalies, resulting in a more consistent pattern that accurately reflected the underlying trends in consumption over time. Additionally, five characteristics were developed to enhance the dataset beyond the preprocessing step: the past-hour value, past-hour average, past-week average, past-hour value, and past-24-hour value [24]. These features aimed to improve the model's predictive power by capturing both short-term and long-term consumption patterns.

The model architecture consisted of a three-layer neural network, comprising an LSTM (long short-term memory) layer followed by two dense layers. The final output layer featured a single neuron to estimate the energy consumption value, with the first dense layer containing 32 neurons and the second layer containing 28 neurons. The design focused on minimizing model size while maintaining performance for deployment in edge machine learning environments. After evaluating various options, a 12-hour look-back window was chosen, as it effectively captured relevant temporal dependencies in the data. Hyperparameters such as batch size, learning rate, dropout rate, and the number of neurons per layer were fine-tuned using grid search to optimize

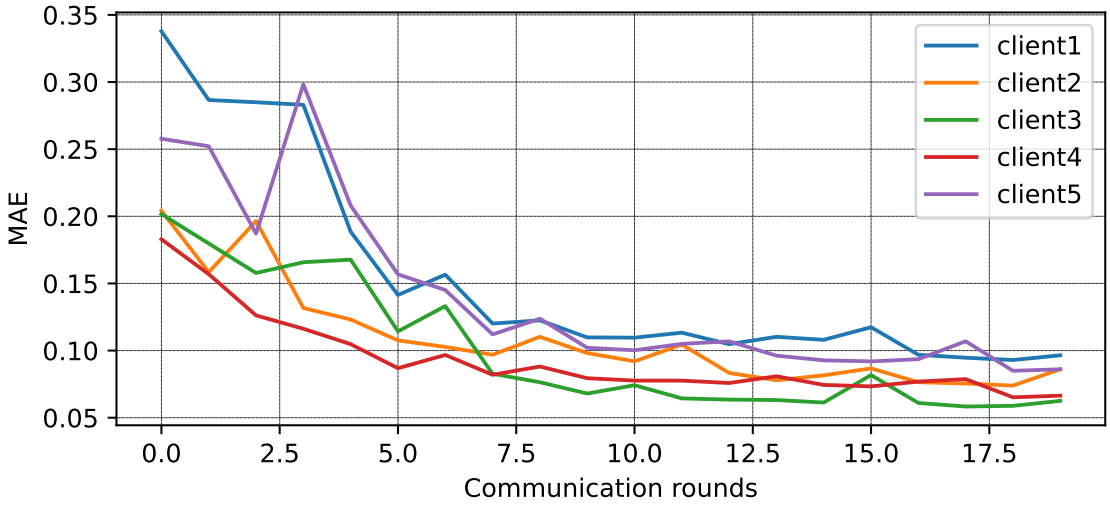


Figure 4.2: MAE of every client following 20 rounds of communication.

model performance. The Adam optimizer was selected for its training effectiveness, and all layers utilized the ReLU activation function. The model's prediction accuracy was assessed using mean squared error as the loss function.

Baseline results Fig. 4.2 displays the FedRLA results, which plot each client's mean absolute error (MAE) against the communication rounds. With an average client MAE of 0.80 kWh after 20 rounds, FedRLA's performance closely resembles the baseline findings in Fig. 3.2. This shows that a single neural network layer, rather than the complete network, can be aggregated to create a global model. In addition to yielding results that are equivalent, this method lowers communication overhead, increasing the scalability and efficiency of network resources.

Communication cost: The data sent between devices and the server was used to calculate the communication cost in this study [165]. The LSTM-based model, sized at 23,220 bytes, comprised three layers: 19,968 bytes for the first, 3,696 bytes for the second, and 28 bytes for the last. Over 20 communication rounds, FedRLA transmitted 0.781 MB of data, reducing communication by 3.56 times compared to FedAvg while maintaining the same forecasting accuracy. This efficiency arises from FedRLA's reduced selection probability for the largest layer, which contained the most parameters. With only three layers, FedRLA achieved notable communication savings, which would further increase in models with more layers.

FedRLA under Adversarial Attacks: During training, FedRLA only communicates with one neural network layer per communication round, which significantly reduces the likelihood that adversarial updates can compromise the entire model. This design choice introduces a probabilistic element to the aggregation process. Since the probability of any given layer being selected for aggregation is $1/L$ (where L is the total number of layers in the model), the impact of malicious updates is naturally confined to a single layer at a time. This probabilistic layer selection inherently dilutes the influence of adversarial updates across the entire model, making it less probable that any single attack can systematically degrade the global model's performance.

In this investigation, four different model poisoning attacks were employed to evaluate FedRLA's robustness: Partially Random Attack (PRA), Completely Random Attack (CRA), Model Flipping Attack (MFA), and Perturbed Attack (PA) (see Section 3.1 for detailed descriptions of these attacks). The results of these experiments are summarized in Figure 4.3, which illustrates the Mean Absolute Error (MAE) under different attack scenarios.

- In the absence of any attacks, both FedRLA and FedAvg achieved a steady MAE of approximately 0.079 kWh, establishing a baseline for model performance.
- Under the Partially Random Attack (PRA), FedRLA recorded an MAE of 0.081 kWh, compared to FedAvg's 0.088 kWh. This indicates that FedRLA's selective layer aggregation strategy effectively limits the spread of malicious updates.
- In the case of the Completely Random Attack (CRA), FedRLA showed greater resilience with an MAE of 0.109 kWh, while FedAvg's MAE increased to 0.124 kWh.
- For the Model Flipping Attack (MFA), FedRLA achieved an MAE of 0.081 kWh, whereas FedAvg's MAE rose to 0.19 kWh. This highlights FedRLA's ability to maintain the coherence of the model's learning direction.
- During the Perturbed Attack (PA), FedRLA maintained a stable MAE of 0.0805 kWh, while FedAvg's MAE sharply increased to 0.27 kWh. This demonstrates FedRLA's effectiveness against sophisticated attack vectors.

These results collectively highlight FedRLA's superior adaptability and robustness against a variety of adversarial attacks. By aggregating only a single randomly selected layer per communication round, FedRLA inherently reduces the attack surface for potential adversaries, thereby enhancing the security and reliability of the federated learning process.

Although FedRLA has built-in protection against model poisoning, the addition of other defense mechanisms like ZeKoC [111], FedClamp [93], LBAA-FedAvg [94], and ShieldFL [147] could increase its efficacy and computational efficiency and guarantee a more dependable and safe FL environment.

4.1.2 Computational Efficiency

Computational efficiency is assessed by analyzing CPU and memory usage over time, which provides insights into the operational demands of a system. High CPU usage may indicate excessive computations, inefficient algorithms, or suboptimal implementation strategies, while low CPU usage suggests potential underutilization of available computational resources. Similarly, excessive memory consumption can lead to performance degradation due to cache misses or frequent swapping with disk storage, which increases latency. Thus, monitoring these metrics over

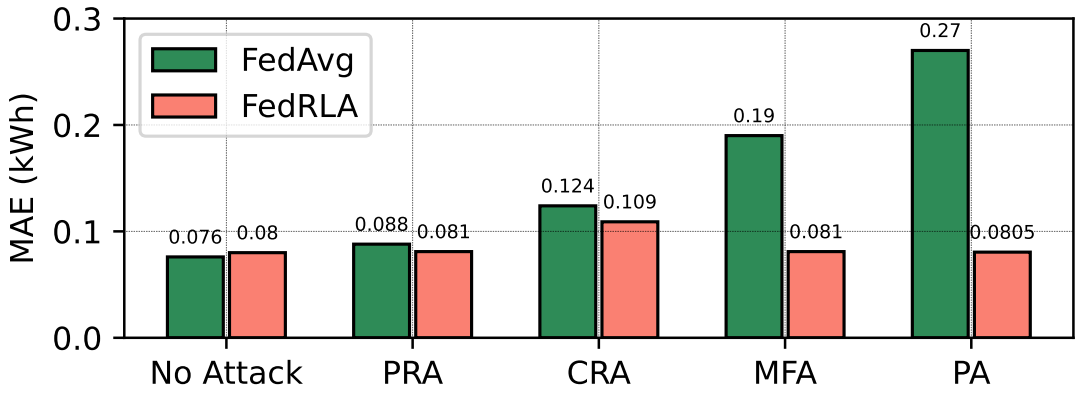


Figure 4.3: FedRLA and FedAvg are compared under various hostile attacks.

time aids in identifying bottlenecks, such as redundant computations, inefficient data structures, or inappropriate resource allocation.

Some algorithms optimize processing speed by utilizing more memory to store precomputed values, thereby reducing the need for repeated calculations. In contrast, other algorithms conserve memory by recomputing values as needed, which can lead to increased CPU usage. Achieving an optimal balance between CPU and memory efficiency is crucial for enhancing overall system performance while minimizing resource consumption, particularly in resource-constrained environments.

The average CPU usage during the training phase is calculated as the mean of the CPU utilization at the beginning and end of the training process. This measurement offers a reliable estimate of the overall CPU load experienced during training. The peak memory usage is determined by identifying the maximum memory consumption recorded during the training phase, ensuring that the system had sufficient resources to accommodate the demands placed upon it. These metrics serve as essential indicators for evaluating the computational efficiency and scalability of the FL model.

The following section offers a comprehensive comparison of the computational efficiency of FedRLA against other frameworks, including FedProx [230], ZeKoC [111], FRA [129], and FedAvg. Fig. 4.4 presents a detailed comparative evaluation of these five frameworks across three critical performance indicators: average CPU usage, peak memory usage, and time taken to execute tasks. For the purpose of effective visualization, normalized values are employed, which allow for equitable comparison across different scales. Each axis on the chart corresponds to one of these indicators, plotted in a circular format to facilitate intuitive and immediate comparison. The working principles of the chosen frameworks are as follows:

- **FedProx:** FedProx is designed to handle heterogeneous client participation and varying computation capabilities. It introduces a proximal term to the optimization problem, allowing clients to perform a variable amount of work and enabling more flexible and ef-

ficient updates. This makes FedProx particularly suitable for environments with diverse computational resources.

- **ZeKoC:** ZeKoC focuses on enhancing the security and robustness of FL systems through zero-knowledge clustering. It ensures that the global model is not compromised by malicious clients by clustering clients based on their data characteristics and only aggregating updates from trusted clusters. This approach is effective in maintaining model integrity but may require additional computational overhead for clustering.

- **RFA:** RFA, or Robust Federated Aggregation, is a novel approach to federated learning that enhances the aggregation process's robustness against potential poisoning of local data or model parameters from participating devices. It utilizes the geometric median for aggregating updates, computed efficiently using a Weiszfeld-type algorithm. RFA is agnostic to the level of corruption and can aggregate model updates without revealing individual contributions from devices. The convergence of the robust federated learning algorithm is established for stochastic learning of additive models with least squares.

- **FedAvg:** FedAvg is the traditional federated averaging algorithm where all clients' model updates are averaged to form the global model. While simple and effective in homogeneous environments, it can struggle with communication efficiency and may be less robust to adversarial attacks in heterogeneous settings.

The blue line on the graph indicates average CPU usage, revealing that FedProx exhibits the highest usage at approximately 1 (normalized value), while FedAvg shows the lowest usage at around 0.73 (normalized value). The orange line represents peak memory usage, with ZeKoC consuming the most memory at around 1 (normalized value) and RFA being the most memory-efficient at about 0.85 (normalized value). The green line illustrates the time taken by each method, with RFA demonstrating the highest efficiency in terms of time, completing tasks in approximately 0.43 (normalized value), whereas FedProx is the least efficient, requiring around 1 (normalized value) to complete the same tasks.

Notably, FedRLA utilizes computational resources similarly to FedAvg, making it particularly suitable for deployment on resource-constrained devices. It achieves a balanced performance profile in terms of average CPU usage (0.78, normalized value), peak memory usage (0.87, normalized value), and time taken (0.42, normalized value). This balanced profile establishes FedRLA as a robust choice for environments where effective resource management is critical.

4.1.3 Discussion

FedRLA is specifically designed for resource-constrained devices, effectively balancing efficiency, security, and privacy in the context of FL. By aggregating only a single randomly se-

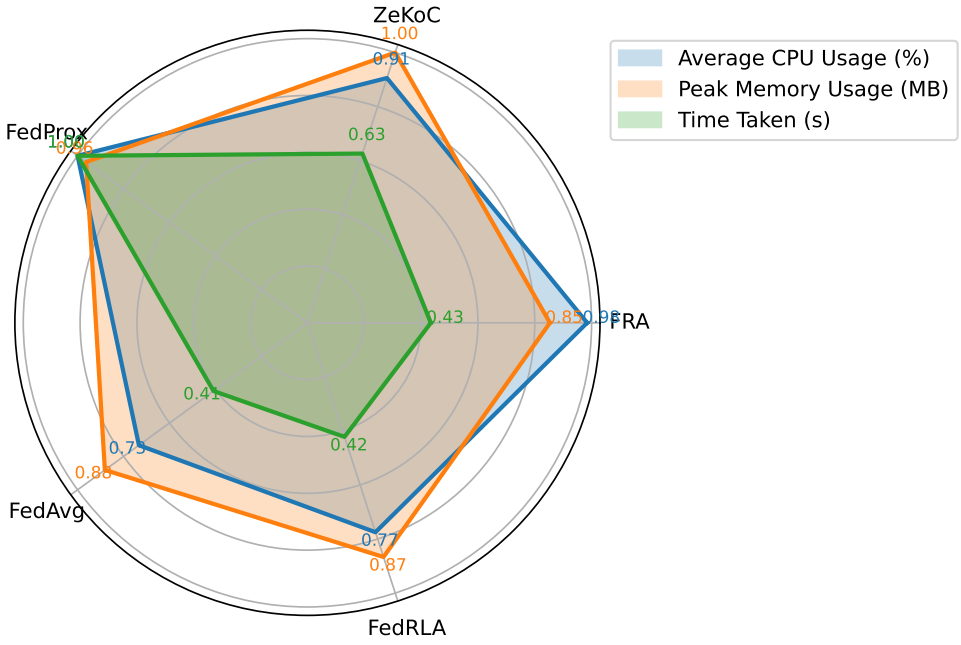


Figure 4.4: Comparison of Different Robust FL Frameworks in Terms of Resource Utilization

lected layer per round, it achieves a significant reduction in communication overhead—up to 92.97% for household-level data and 93.66% for grid-level data with 8-bit quantization—while maintaining predictive accuracy. This reduction is particularly crucial for smart meters and edge devices that operate under limited bandwidth and computational capabilities.

In contrast to traditional FL methods such as FedAvg and FedProx, which transmit full model updates, FedRLA minimizes both communication and computational costs without compromising model performance. The selective layer aggregation approach not only enhances efficiency but also improves resilience against adversarial attacks by limiting the impact of potentially malicious updates. As a result, forecasting errors are lower compared to those observed with FedAvg, providing an added layer of reliability.

Privacy is inherently reinforced within the framework, as transmitting only a single layer significantly reduces the potential for data exposure. This privacy-preserving effect is further enhanced through the implementation of differential privacy (DP), ensuring stable performance even under stringent privacy constraints. Furthermore, FedRLA is compatible with various quantization techniques, allowing it to maintain accuracy while significantly reducing both storage and processing requirements.

Overall, FedRLA demonstrates strong computational efficiency, with an average CPU usage of 7.4%, peak memory usage of 12,588 MB, and an execution time of 41.79 seconds. These performance metrics are comparable to those of FedAvg but exhibit higher efficiency than those of FedProx. Although there is a slight increase in computational load compared to FedAvg, FedRLA offers a more favorable trade-off by enhancing security, communication efficiency, and privacy. These characteristics render it a practical and effective choice for FL applications

in energy-constrained environments where resource management is paramount.

4.2 LBAA-FedAvg

Many existing defense frameworks, as outlined in Chapter 2, address the challenge of anomalous clients by excluding their entire local model from the aggregation process. While this method is effective for fully compromised models, it proves inefficient in scenarios where only specific layers of a local model are targeted by attacks. The exclusion of the entire model results in the loss of valuable information contained within unaffected layers, consequently diminishing the robustness and overall performance of the aggregation process.

To tackle this issue, I propose Layer-Based Anomaly-Aware Federated Averaging (LBAA-FedAvg). This innovative approach diverges from traditional methods by isolating adversarially compromised layers while retaining the unaffected layers during the aggregation phase. By concentrating on layer-level granularity, LBAA-FedAvg excludes only the maliciously modified portions of a model, thereby preserving the integrity of the global model and maximizing the utility of the available data. This strategy is particularly pertinent for systems that have experienced partial attacks, where targeted defenses are essential for maintaining both security and efficiency within FL environments.

The aggregation framework of LBAA-FedAvg, illustrated in Fig. 4.5, divides each neural network layer into two clusters: one representing the larger, unaffected cluster and the other representing the smaller, potentially compromised cluster. The underlying assumption is that the number of compromised clients will be fewer than that of normal clients. This allows for the effective isolation and exclusion of the compromised layer while retaining the remainder of the model for aggregation. Consequently, this approach ensures that each layer of the global model benefits from insights gathered from multiple clients, thereby enhancing the overall learning process and resilience against adversarial threats.

The inclusion or exclusion of clients during the aggregation process is determined by specific clustering criteria, as depicted in Fig. 4.6. This criterion mandates that there must be a larger gap between the centroids of each cluster than the maximum Euclidean distances between any two clients within those clusters. Specifically, for each layer, the weights from different clients are clustered into two groups. The validity of the clustering is assessed by comparing the distance between the centroids of the two clusters (D_3) with the maximum Euclidean distances within each cluster (D_1 and D_2). If D_3 exceeds both D_1 and D_2, it indicates that the clusters are sufficiently distinct and the larger cluster is selected for aggregation. This ensures that only the weights from the more representative and potentially non-adversarial cluster are used to update the global model, thereby enhancing the robustness and integrity of the aggregation process.

The steps for implementing Layer-Based Anomaly-Aware Federated Averaging (LBAA-FedAvg) are as follows:

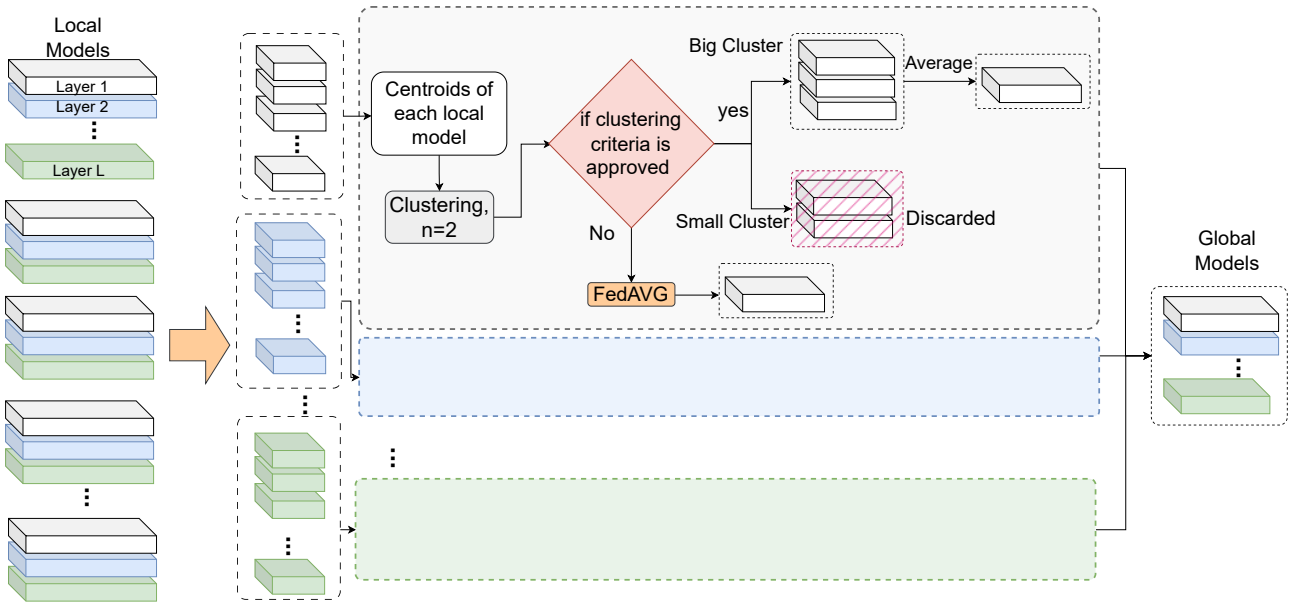


Figure 4.5: Block diagram of LBAA-FedAVG).

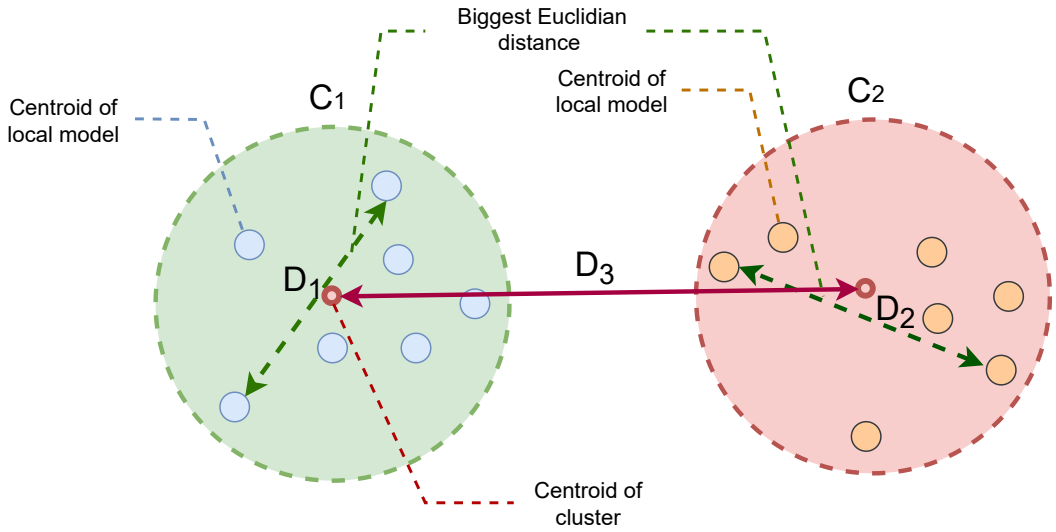


Figure 4.6: Illustration of the clustering criterion for LBAA-FedAvg, where C_1 and C_2 denote two clusters, D_1 and D_2 represent the maximum Euclidean distances within each cluster, and D_3 is the distance between the centroids of C_1 and C_2.

1. Receive updates from local clients: $w_1^t, w_2^t, \ldots, w_k^t$.

2. Split the layers of local models into separate weight groups for granular analysis.

3. Compute the centroids for each layer's weights:

$$w_{c1} = \text{avg}(w_1^t), \quad w_{c2} = \text{avg}(w_2^t), \quad \ldots, \quad w_{ck} = \text{avg}(w_k^t).$$

4. Cluster the weights for each layer into two distinct groups, denoted as C_1 and C_2.

5. Calculate the maximum Euclidean distance, D_1, for clients in cluster C_1.

6. Calculate the maximum Euclidean distance, D_2, for clients in cluster C_2.

7. Compute the distance D_3 between the centroids of clusters C_1 and C_2.

8. If $D_3 > D_1$ and $D_3 > D_2$, select the larger cluster and aggregate the weights from that cluster using:

$$\sum_{k=0}^{n_{\max}} \frac{1}{n_{\max}} w_k^t.$$

   Repeat this process for all layers.

9. If the clustering condition is not met, proceed with the standard Federated Averaging (FedAvg) method as outlined in [231].

This method ensures that only compromised layers are excluded from the aggregation process, thereby preserving both the efficiency and integrity of the overall model aggregation.

Design Insights: A clustering-based aggregation technique called LBAA-FedAvg is based on the idea that there will be fewer compromised clients than healthy ones. It should be noted, though, that the error rate of the aggregated model may be higher than that of the conventional FedAvg technique if the percentage of compromised clients is greater than 50%. The flexibility of LBAA-FedAvg to selectively include or exclude specific layers according to the type of attack detected is a noteworthy feature. Because of this flexibility, compromised clients can be included in the aggregate process while the negatively impacted layers are excluded. A compromised client's model will therefore probably continue to resemble that of the other healthy clients even if it is included in the aggregation even though it does not satisfy the clustering condition. This resemblance increases the resilience of the aggregation process by reducing the overall effect of the compromised client's existence on the model's performance.

4.2.1 Experiments and Results:

In order to assess LBAA-FedAvg's efficacy against standard FedAvg, we performed model flipping attacks using compromised clients that ranged from 10% to 50%. The local models were explicitly targeted by these attacks, which sequentially attacked the first, second, and third layers of each of their three layers. The findings, as shown in Fig. 4.7, show a distinct pattern: the MAPE in FedAvg increased in tandem with the rise in the proportion of compromised clients. Attacks on the first layer, in particular, had the least effect on the MAPE, indicating that it might have more resilient features or be less vulnerable to hostile manipulation.

In contrast, when the attack percentage was between 10% and 40%, LBAA-FedAvg was able to sustain an average MAPE of 2.8%. This steady performance demonstrates how well LBAA-FedAvg reduces the effects of hostile attacks. However, the MAPE in LBAA-FedAvg

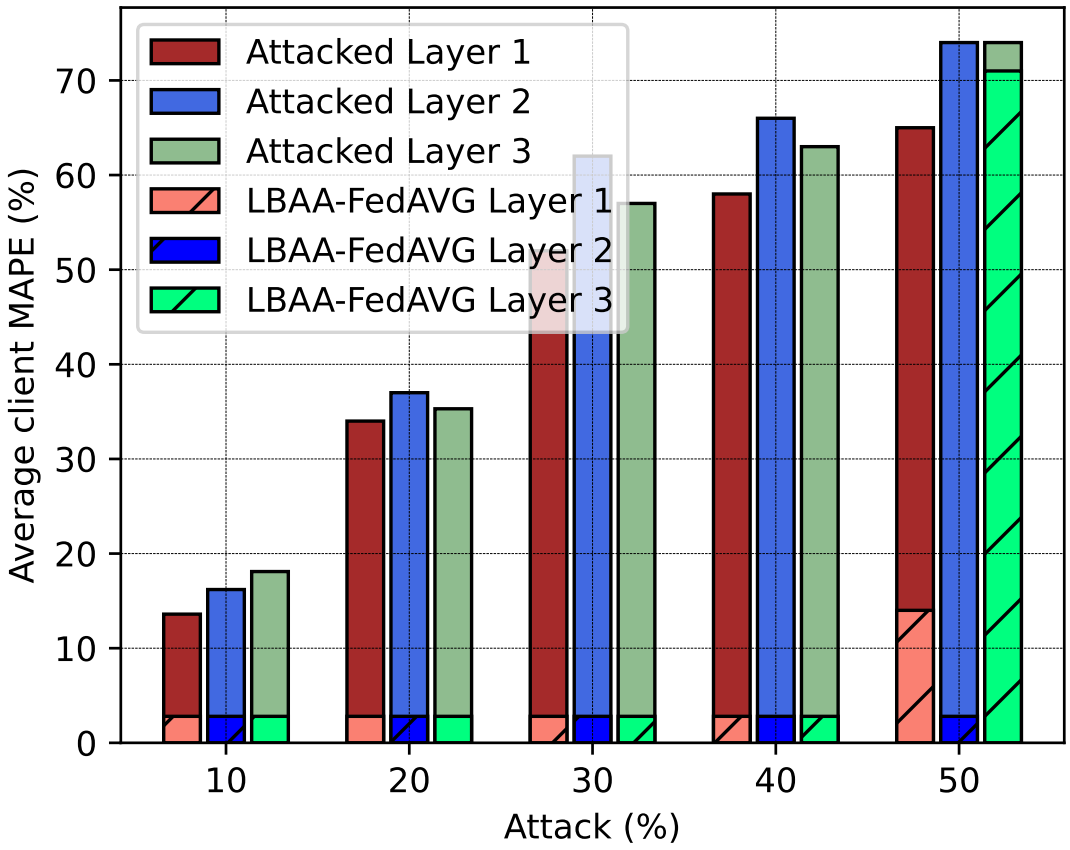


Figure 4.7: Effect of LBAA-FedAvg on average client MAPE .

significantly increased when the attack percentage increased to 50%. This decrease can be ascribed to difficulties precisely locating the larger cluster in the midst of a greater number of compromised customers. It is interesting that LBAA-FedAvg outperformed FedAvg in spite of this rise at the greatest attack percentage, proving its efficacy and robustness in situations even when a sizable fraction of clients were compromised. The benefits of using layer-level granularity to counter adversarial threats in FL environments are demonstrated by this performance.

Resource Utilization: Examining the resource consumption of LBAA-FedAvg across a range of resources allows for an assessment of its overall resource use. This analysis considers five important factors: disk space, communication rounds, global model training time, CPU consumption, and memory utilization. In the absence of adversarial attacks, Fig. 4.8 illustrates the resource usage of both FedAvg and LBAA-FedAvg. Both strategies demonstrate similar levels of CPU, memory, and disk space usage, suggesting that their fundamental resource requirements are essentially the same.There is a minor discrepancy in the number of communication rounds needed; FedAvg required 65 rounds, while LBAA-FedAvg required 67 rounds. However, the training time shows a more noticeable difference. FedAvg took 266.27 seconds to complete training, while LBAA-FedAvg took 317 seconds, which is approximately

19% longer. The clustering procedure used by LBAA-FedAvg in each communication round introduces computational overhead, which is the main cause of this increase in training time.

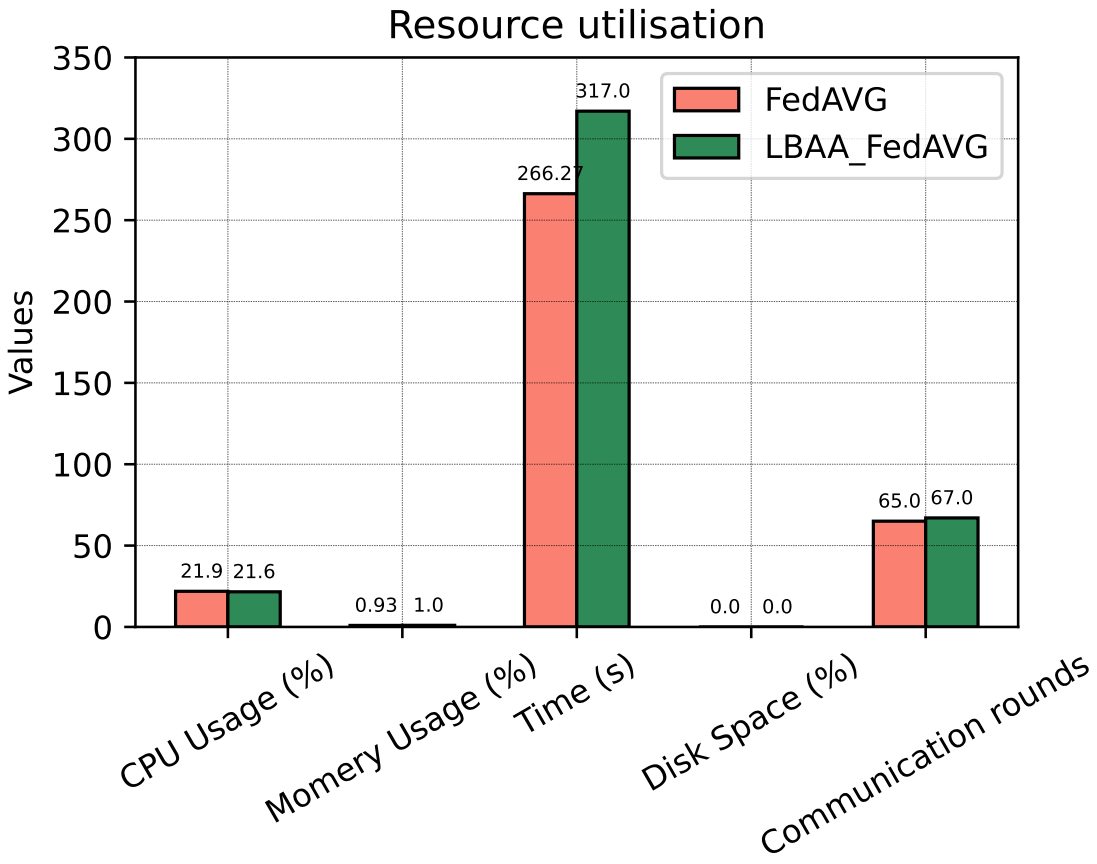


Figure 4.8: Resource utilisation of FedAvg and LBAA-FedAVG.

4.3 Federated Incentivized Averaging (Fed-InA)

Fed-CRA is a subtle and difficult-to-detect poisoning attack due to its random nature, where it may appear in one round and disappear in the next. This unpredictability makes it challenging to identify compromised clients without prolonged observation [232]. To address this, an incentive-based aggregation approach is proposed that rewards clients providing beneficial updates, inspired by real-world incentive systems [233]. Various incentive mechanisms, including contract theory [234], game theory [235], and deep reinforcement learning [236], have been explored in FL.

To counter the Fed-CRA attack, Fed-InA is introduced, a scoring-based aggregation method that rewards valuable updates and penalizes attempts to manipulate the global model. Fed-InA incorporates clustering, scoring, and scored averaging components, allowing it to assess each layer's weights and biases independently during aggregation. The method processes updates by breaking them down into layers, weights, and biases, assigning rewards based on the detection

of adversarial components in both weights and biases [233]. Fed-InA is represented graphically in Fig. 4.9 and in the form of algorithm in algorithm 3.

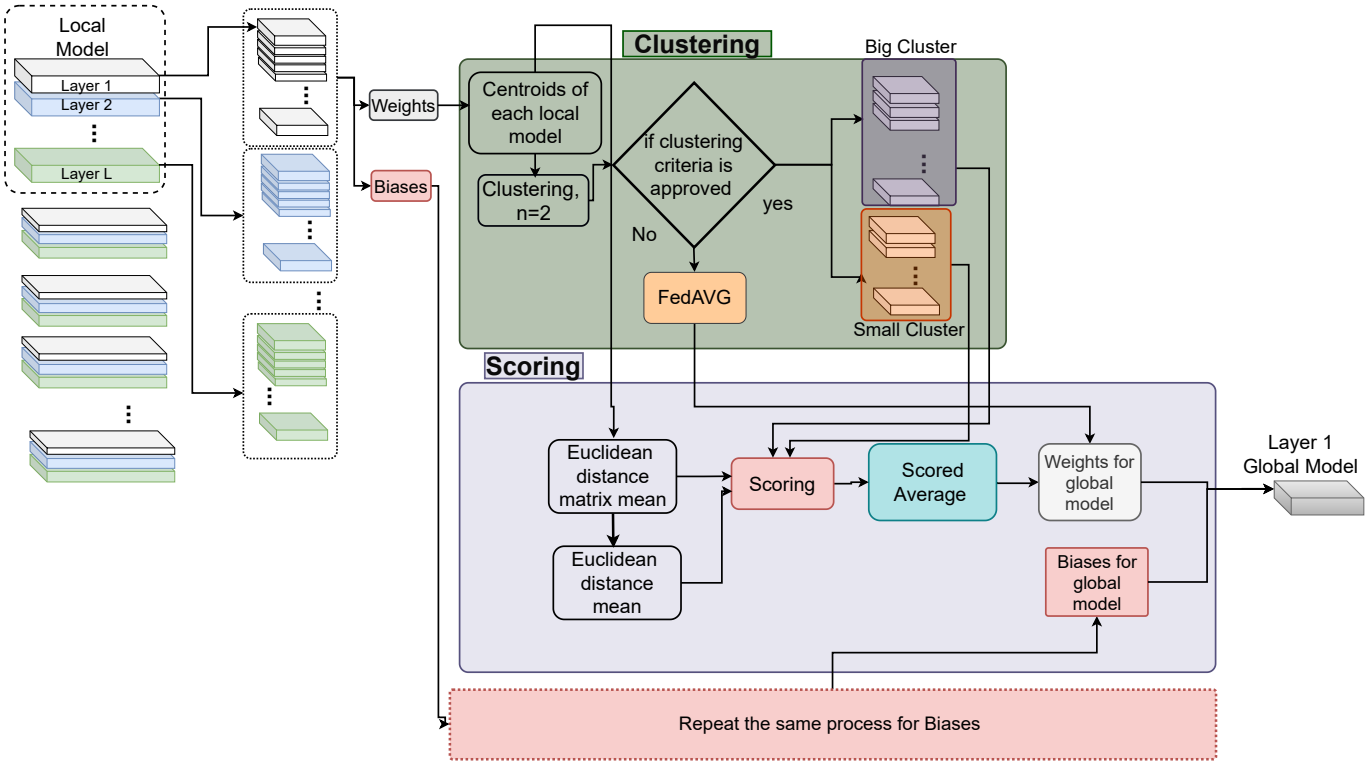


Figure 4.9: Block diagram of Fed-InA.

The weights for the new global model are computed as follows:

1. **Initialization:** After clients K send their updated weights W_t^k to the server, initialize the client score S_k^t. Identify all layers L and store them separately in $W_{t,l}^k$, where $k \in K$ and $l \in L$ represent the clients and layers, respectively.

2. **Clustering:** To facilitate clustering, find the centroids $C_{t,l}$ for each layer and perform clustering with $n = 2$, generating two clusters: $C_{t,l}^1$ and $C_{t,l}^2$.

3. **Clustering criteria:** Calculate the Euclidean distances within each cluster and store them in $d_{t,l}^1$ and $d_{t,l}^2$. Identify the largest Euclidean distances, $D_{t,l}^1$ and $D_{t,l}^2$, and the distance $D_{t,l}^3$ between the centroids of $C_{t,l}^1$ and $C_{t,l}^2$. If $D_{t,l}^3$ is greater than $D_{t,l}^1$ and $D_{t,l}^2$, the clustering is accepted; otherwise, skip clustering and use FedAvg as described in [231]. If clustering is not approved, it indicates that the clients are too close to each other, suggesting no attack detection. A graphical representation of the clustering criteria is shown in Fig. 4.6.

4. **Scoring:** The larger cluster from $C_{t,l}^1$ and $C_{t,l}^2$ receives a positive score, while the smaller cluster is penalized with a negative score. To calculate the score, first determine the pairwise Euclidean distances of all clients, $z_{k,l}$, from $C_{t,l}$. Then, compute the average pairwise

distance $z_{k,l}$ and the client-wise average $D_{avg,l}$. For clients in the larger cluster, update their score as:

$$S_t^k = S_{t-1}^k + \left| D_{avg,l} - z_{k,l} \right|$$

For clients in the smaller cluster, update their score as:

$$S_t^k = S_{t-1}^k - \left| D_{avg,l} - z_{k,l} \right|$$

5. **Scored average:** Normalize the client scores S_n^t so that their sum equals 1. The global model is then obtained by computing the scored average:

$$G_t^k = (W_t^1 \times S_n^1) + (W_t^2 \times S_n^2) + (W_t^3 \times S_n^3) + \cdots + (W_t^k \times S_n^k)$$

Design insights: The scoring method in this algorithm rewards clients who contribute positively to the model and penalizes those attempting to manipulate it. Based on their Euclidean distance from the mean, clients are assigned scores that encourage competition, motivating them to enhance their models and align with the global model. While the algorithm does not explicitly reference game theory concepts such as Nash equilibria, it incorporates mechanisms that promote desired behaviors and deter harmful actions, reflecting the core principles of game theory. Fed-InA, a clustering-based aggregation method, assumes that compromised clients are fewer than healthy ones, a common assumption in prior work, which typically considers attack percentages up to 50%. If compromised clients exceed 50%, the aggregation process may select compromised clients during clustering and incentivize them accordingly.

4.3.1 Experiments and Results

Fed-InA was applied to Fed-CRA to evaluate its performance. The effectiveness of Fed-InA depends on the proportion of compromised clients, which should ideally be less than 50%, as the method relies on clustering-based techniques. Therefore, Fed-CRA was evaluated with attack percentages ranging from 10% to 50%. The results, including the average client MAPE and the number of communication rounds, are presented in Fig. 4.10. These results demonstrate that implementing Fed-InA can significantly reduce communication rounds. The observed reduction in communication rounds is attributed to a greater focus on clients providing superior updates during the aggregation of the global model. Importantly, Fed-InA managed to keep the communication rounds below 120 even as the attack percentage in Fed-CRA increased from 10% to 50%. Fig. 4.10 provides a comparison of communication rounds when the system was subjected to Fed-CRA. The shaded bars in the figure illustrate the effect of Fed-InA on the performance of Fed-CRA. It is evident that the impact of Fed-InA increases as the attack percentage rises.

Computational Analysis To evaluate the computational efficiency of Fed-InA, a detailed computational analysis was conducted [94], considering the following key parameters:

Algorithm 3 Federated Incentivized Averaging (Fed-InA) for weights of global model

Initialize global model parameters W_0
for $t = 1$ to T **do**
  for each client $k \in K$ **do**
    Client k sends local model parameters W_t^k
  end for
  for $l = 1$ to L **do**
    for each client $k \in K$ **do**
      Separate layer l parameters from W_t^k and store them in $W_{t,l}^k$
    end for
    $C_{t,l} = \frac{1}{K}\sum_{k=1}^{K} W_{t,l}^k \leftarrow$ Compute centroid
    Perform k-means clustering with $n = 2$ on $W_{t,l}^k$ to form clusters $C_{t,l}^1$ and $C_{t,l}^2$
    $d_{t,l}^1$ and $d_{t,l}^2 \leftarrow$ Euclidean distance within Cluster
    $D_{t,l}^1$ and $D_{t,l}^2 \leftarrow$ max distance for $C_{t,l}^1$ and $C_{t,l}^2$
    $D_{t,l}^3 \leftarrow$ distance between centroids of $C_{t,l}^1$ and $C_{t,l}^2$
    if $D_{t,l}^3 > D_{t,l}^1$ and $D_{t,l}^3 > D_{t,l}^2$ **then**
      Clustering approved
      for each client $k \in K$ **do**
        $z_{k,l} \leftarrow$ pairwise distance of client k from centroid $C_{t,l}$
        $D_{avg,l} = \frac{1}{K}\sum_{k=1}^{K} d_{k,l} \leftarrow$ averaged pairwise distance
        if client k is in the larger cluster **then**
          Update the previous score, S_{t-1} of client k
          $S_t^k = S_{t-1}^k + \left|D_{avg,l} - z_{k,l}\right|$
        else
          Update the previous score, S_{t-1} of client k
          $S_t^k = S_{t-1}^k - \left|D_{avg,l} - z_{k,l}\right|$
        end if
      end for
      $Sn_t \leftarrow$ Normalised score of all k clients
      $Sn_t = (S_t^1, S_t^2, S_t^3, ..., S_t^k)/\sum S_t^k$
      $G_t^k \leftarrow$ Global model
      $G_t^k = (W_t^1 \times Sn_t^1) + (W_t^2 \times Sn_t^2) + (W_t^3 \times Sn_t^3) + ... + (W_t^k \times Sn_t^k)$
    else
      Ignore scoring and compute FedAVG on all clients K as mentioned in [231]
      $G_t^k \leftarrow \sum_{k=0}^{K} \frac{1}{K} W_t^k$
    end if
  end for
end for

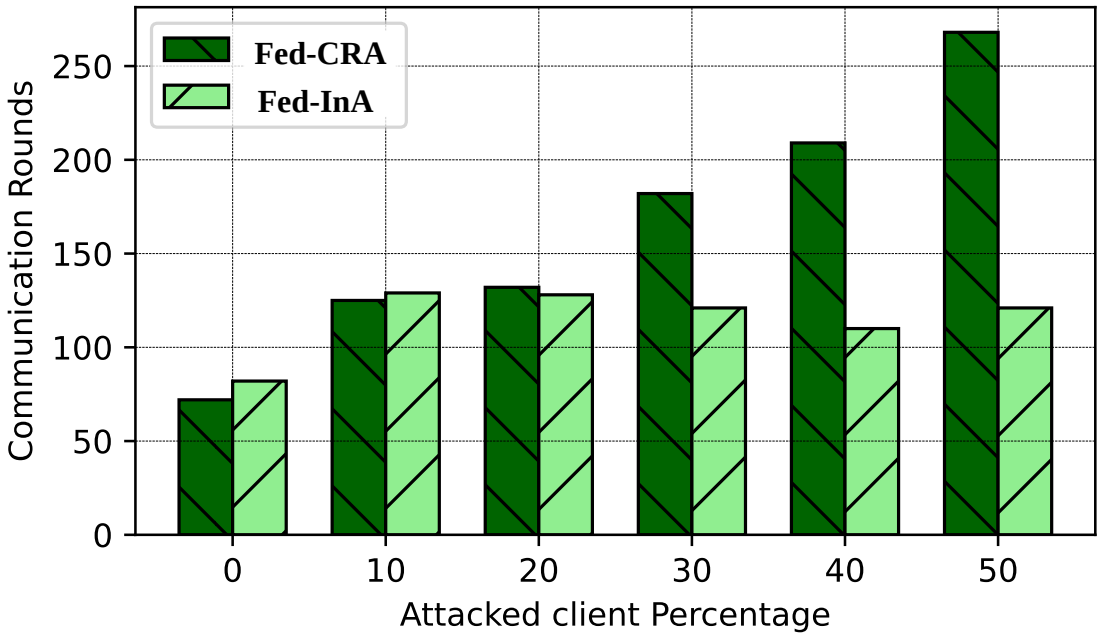


Figure 4.10: Effect of Fed-InA on Fed-CRA.

1. CPU Usage: This metric quantifies the computational requirements of the algorithm, providing insight into its temporal complexity. Higher CPU usage indicates a greater computational burden, while lower usage suggests enhanced efficiency and reduced computational demands.

2. Memory Usage: This parameter reflects the algorithm's scalability and memory efficiency. A significant increase in memory usage may indicate inefficiencies or higher complexity, suggesting a greater requirement for resources.

3. Time: This measure reflects the total computational time required by the algorithm, offering insights into its performance and computational demands. It aids in optimizing and selecting algorithms based on the time required for effective execution.

4. Disk Space: This metric assesses the space complexity of the algorithm, indicating how the storage requirements grow with the size of the input or other related factors. It provides an understanding of the algorithm's efficiency in utilizing storage resources.

The performance of Fed-InA was compared to the baseline results of FL using FedAvg under conditions with no attack. Given that FedAvg is not capable of detecting Fed-CRA, which is designed to increase training time, the comparison was made under no-attack conditions for a fair evaluation. The summarized results, presented in Fig. 4.11, indicate that the primary resource constraint for Fed-AwR, in comparison to FedAvg, is the computational time. FedAvg completed the process in 266 seconds, while Fed-AwR required 293 seconds, reflecting an 11% increase. The additional time required by Fed-AwR is attributed to the clustering and scoring processes performed in each communication round.

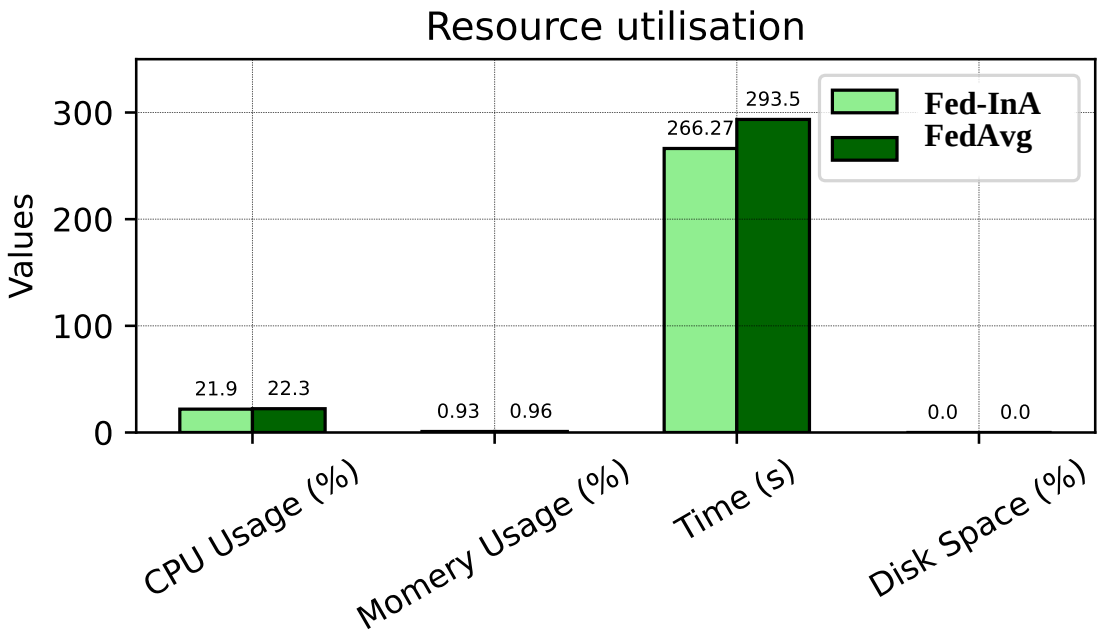


Figure 4.11: Resource utilisation of Fed-InA.

4.3.2 Discussion

This section delves into a comprehensive analysis of Fed-InA, comparing it against several state-of-the-art robust aggregation frameworks: FedClamp [93], LBAA-FedAvg [94], ZeKoC [111], and Cluster FL [237], under the challenging scenario of Fed-CRA. The evaluation was meticulously conducted using the standard FedAvg algorithm on a dataset where 50% of the participating clients were compromised by adversarial behavior. This adversarial setup is designed to simulate real-world scenarios where a significant portion of clients may exhibit malicious or anomalous behavior, thereby testing the robustness of each framework.

The frameworks compared can be broadly categorized based on their operational strategies: post-aggregation and in-aggregation approaches. Post-aggregation frameworks, such as Cluster FL and FedClamp, execute the entire FL training process first and then identify anomalous clients by comparing their behavior against the majority. These frameworks assume that the majority of clients are benign and use this assumption to detect outliers. Once identified, these anomalous clients are subsequently clustered separately to mitigate their impact on the global model. This approach is effective in scenarios where the majority of clients are indeed benign, but it may fail when the adversarial clients are able to mimic the behavior of the majority.

On the other hand, in-aggregation frameworks like LBAA-FedAvg [94] and ZeKoC [111] dynamically adjust client contributions during the FL training process. ZeKoC employs a clustering-based methodology, where clients are grouped into clusters using a comprehensive three-step process. This process involves initial clustering, refinement of clusters based on client behavior, and final adjustment of cluster assignments. Separate global models are then generated

for each cluster, allowing for more granular control over the aggregation process. In comparison, LBAA-FedAvg uses a performance-based clustering mechanism that restricts the number of clusters to two at any given time. This simplifies the aggregation process while maintaining robustness against adversarial attacks by ensuring that only the top-performing clients influence the global model.

Despite these varied strategies, none of these frameworks were capable of detecting the additional communication rounds introduced by Fed-CRA. As depicted in Fig. 4.12, the primary limitation of these frameworks is their reliance on clustering methodologies that focus on identifying poorly performing clients. In Fed-CRA, all clients exhibit uniformly consistent behavior, effectively bypassing detection mechanisms that depend on performance-based differentiation. This uniformity makes it difficult for traditional clustering-based approaches to identify malicious clients, as they do not stand out in terms of performance metrics.

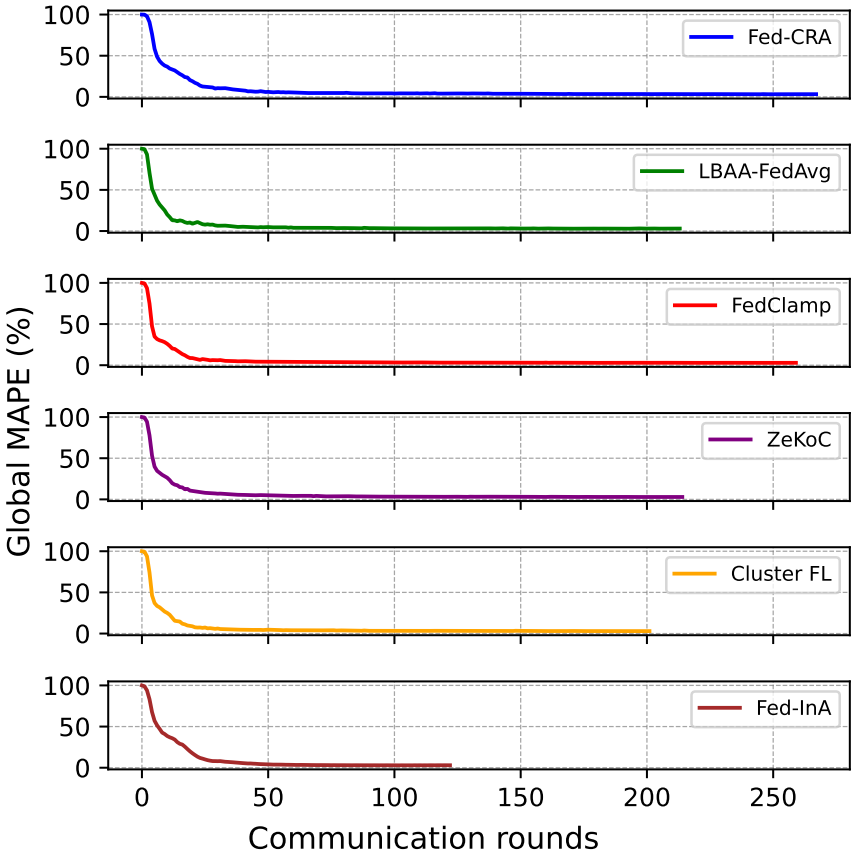


Figure 4.12: Comparison of different state-of-the-art robust aggregation frameworks with Fed-CRA and Fed-InA. The figure illustrates the number of communication rounds required by each framework under adversarial conditions. Fed-InA significantly reduces the number of rounds compared to Fed-CRA, demonstrating its robustness and efficiency.

Even Fed-InA, despite its advanced incentive-based aggregation mechanism, was not able to completely eliminate the effects of Fed-CRA. However, it demonstrated a significant improve-

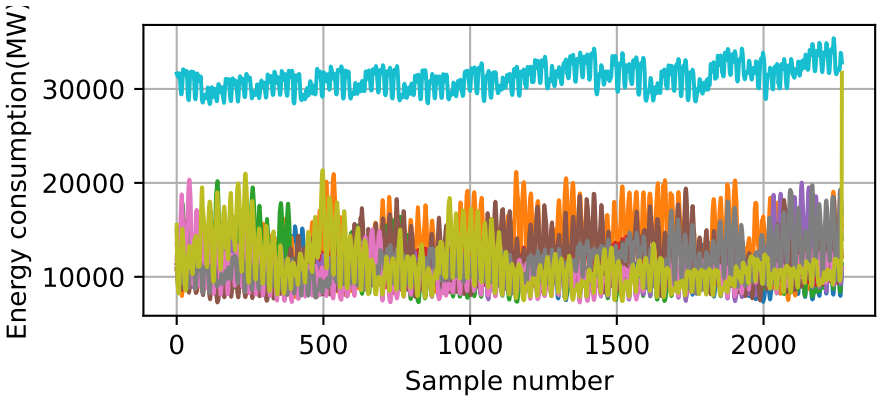


Figure 4.13: Visual representation of the heterogeneous dataset. The figure shows the data distribution across different clients, highlighting the significant deviation in client 10's data pattern.

ment in mitigating its impact. The baseline FL process required 72 communication rounds, while Fed-CRA inflated this to 263 communication rounds by introducing deceptive patterns in the client updates. These deceptive patterns are designed to mislead the aggregation process, causing unnecessary communication overhead. Fed-InA effectively reduced this to 120 communication rounds, all while maintaining comparable model performance. This reduction showcases Fed-InA's capability to counteract adversarial strategies while improving communication efficiency. The incentive-based mechanism of Fed-InA allows it to dynamically adjust client contributions based on their behavior, thereby reducing the influence of malicious clients and stabilizing the global model.

To further assess Fed-InA's robustness in heterogeneous data scenarios, a behavioral shift was induced in client 10 (Fig. 4.13), causing it to exhibit a significantly different data pattern compared to other clients, simulating non-IID (non-identically and independently distributed) conditions. Non-IID data is a common challenge in FL, where clients may have data that is not representative of the overall distribution. This deviation affected the aggregation process during the first 15 communication rounds, as shown in Fig. 4.14. During this period, the global model's performance was unstable due to the conflicting updates from client 10. In response, Fed-InA dynamically adjusted its incentive mechanism, progressively reducing the incentives for client 10. This adjustment is based on the observed behavior of client 10, which was identified as potentially misleading. By reducing the incentives, Fed-InA effectively discouraged client 10's updates, thereby reducing its influence on the global model. As the incentives approached zero, client 10's influence diminished, allowing the remaining clients to align their updates and stabilize the global model's performance. This dynamic adjustment mechanism is a key feature of Fed-InA, enabling it to adapt to changing client behaviors and maintain overall model accuracy even in challenging non-IID scenarios. In summary, Fed-InA demonstrates robustness against adversarial attacks and non-IID data conditions through its advanced incentive-based aggregation mechanism. While it cannot completely eliminate the effects of adversarial strategies like Fed-CRA, it significantly mitigates their impact, improving communication efficiency and

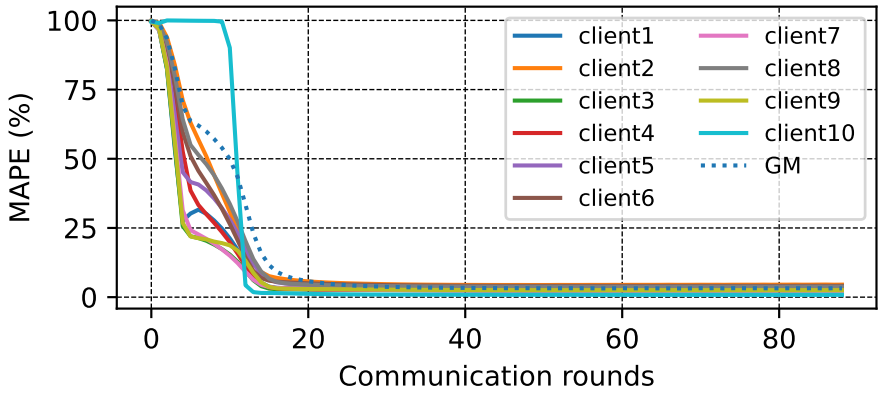


Figure 4.14: Effect of Fed-InA on the heterogeneous dataset. The figure illustrates the stabilization of the global model's performance as Fed-InA adjusts its incentive mechanism to mitigate the influence of client 10.

maintaining model performance. This makes Fed-InA a promising approach for FL in real-world scenarios where data heterogeneity and adversarial behavior are common challenges.

4.4 Concluding Remarks

In this chapter, i explored the critical need for robust defense mechanisms in FL systems, particularly in the face of adversarial attacks that can compromise the integrity and performance of the global model. The proposed defense frameworks aim to enhance the security and resilience of FL systems by effectively detecting and mitigating the impact of various types of attacks. The key contributions and findings of this chapter can be summarized as follows:

- **Federated Random Layer Aggregation (FedRLA)**: FedRLA introduces a novel approach to enhance global model training efficiency and defend against adversarial attacks. By aggregating only a single, randomly chosen neural network layer during each communication round, FedRLA reduces data exchange between devices and the server, thereby streamlining communication and enhancing privacy. This method not only improves security but also significantly reduces communication overhead, making it ideal for privacy-sensitive applications such as household energy forecasting. Experimental results demonstrate that FedRLA achieves comparable model accuracy to traditional methods while reducing communication costs by a factor of 3.56. This addresses the challenge of communication efficiency (C3) by minimizing the amount of data transmitted during each round.

- **Layer-Based Anomaly Aware Federated Averaging (LBAAFedAvg)**: LBAAFedAvg leverages anomaly detection to safeguard against deviations in model updates caused by adversarial clients. By clustering the weights of each layer and selectively excluding com-

promised layers from the aggregation process, LBAAFedAvg ensures the integrity of the global model. This approach effectively mitigates the impact of partial attacks, preserving the accuracy and robustness of the FL system. Experiments show that LBAAFedAvg maintains a stable average client Mean Absolute Error (MAPE) even under varying attack scenarios, demonstrating its effectiveness in enhancing model security. This framework addresses the challenge of robust aggregation (C2) by detecting and isolating attacked layers during the aggregation process.

- **Federated Incentivized Averaging (Fed-InA)**: Fed-InA introduces a scoring mechanism that evaluates clients based on their contribution to the model's accuracy and reliability. By rewarding good clients and penalizing malicious ones, Fed-InA encourages honest participation and contributes to the overall integrity and performance of the FL system. This incentivization strategy effectively identifies and mitigates stealth attacks, which are particularly challenging to detect due to their subtle nature. Experimental results indicate that Fed-InA significantly reduces communication rounds while maintaining model accuracy, even in the presence of adversarial attacks. This framework addresses the challenge of stealth attacks (C2) by introducing a novel scoring mechanism that rewards honest clients and penalizes malicious ones.

The findings from this chapter highlight the importance of developing tailored defense mechanisms to address specific vulnerabilities in FL systems. The proposed frameworks—FedRLA, LBAAFedAvg, and Fed-InA, demonstrate significant improvements in security and model performance by employing advanced anomaly detection techniques and selective model aggregation. These advancements contribute to the ongoing research in secure FL, providing practical solutions to enhance the robustness and reliability of FL systems in real-world applications. Future work will focus on further refining these frameworks and exploring additional techniques to enhance the security and efficiency of FL in diverse applications.

Chapter 5

Novel Framework for Data Heterogeneity in FL

Data heterogeneity presents a significant challenge in FL systems, leading to biased global updates, slower convergence, and reduced model accuracy. This issue is particularly pronounced in real-world deployments where client heterogeneity and resource constraints are common. As outlined in Section 1.2.3 (Challenge C4), FL systems often struggle with data heterogeneity, which can severely impact model performance and fairness. This challenge is further elaborated in Section 2.6.1, where the gap analysis highlights the need for robust solutions to handle non-IID data distributions. In the realm of smart energy networks, device-level variance and non-IID data distributions exacerbate these problems, affecting the efficiency of smart grid management and posing risks to grid stability. STLF is a critical application in this domain, where accurate predictions are essential for efficient energy allocation and operational planning.

Building upon the challenges identified in Chapter 1 and the gap analysis from Chapter 2, this chapter evaluates two complementary solutions: FedBranched and ASLA. FedBranched directly addresses Challenge C4 by enhancing personalization and fairness through its innovative clustering approach. ASLA, on the other hand, not only contributes to resolving data heterogeneity (C4) but also specifically targets Challenge C3 by optimizing communication and computational efficiency. Both frameworks are designed to ensure the equitable contribution of clients and improve the robustness of global models under non-IID settings.

By addressing data heterogeneity and optimizing resource usage, we aim to pave the way for more resilient, efficient, and reliable energy systems. This chapter thus directly responds to Challenges C3 and C4, contributing to the mitigation of the heterogeneity and efficiency gaps previously identified. The key contributions of this chapter are:

- **FedBranched Framework**: This section presents FedBranched, a novel framework that employs Hidden Markov Model (HMM) clustering to address data heterogeneity in FL. Through comprehensive analysis, the framework demonstrates its effectiveness in handling diverse data distributions and improving model performance and convergence in

energy networks.

- **Adaptive Single Layer Aggregation (ASLA) Framework**: This section introduces ASLA, which simplifies the aggregation process by focusing on a single layer of neural networks. ASLA is designed to tackle both communication and computational efficiency (C3) and data heterogeneity (C4). It achieves this by reducing communication overhead and maintaining model accuracy while enhancing computational efficiency.

Key Difference: FedBranched and ASLA both aim to enhance FL systems but differ in their primary focus and methodology. FedBranched employs HMM clustering to tackle data heterogeneity (C4), improving model performance and convergence in energy networks. In contrast, ASLA addresses both communication/computational efficiency (C3) and data heterogeneity (C4) by simplifying the aggregation process to a single neural network layer, reducing overhead while maintaining accuracy.

5.1 FedBranched

This section presents FedBranched, a novel framework designed to address data heterogeneity in FL systems. The following subsections provide a detailed exploration of the framework's motivation, driven by the need to handle non-IID data distributions in smart energy networks. The methodology is then delved into, outlining how FedBranched employs HMM clustering to enhance model convergence. Subsequent sections detail the simulation setup, experiments conducted, and the results obtained, which demonstrate the framework's effectiveness in improving forecasting accuracy and its ability to adapt to diverse data patterns. Additionally, valuable design insights gained from implementing FedBranched are shared, offering a comprehensive understanding of its functionality and potential applications.

5.1.1 Motivation

FedBranched is a zero-knowledge FL framework developed to enhance model convergence through the innovative use of Hidden Markov Model (HMM) clustering. The primary motivation behind FedBranched stems from the challenge of data heterogeneity in FL systems, which can lead to biased global updates, slower convergence, and reduced accuracy in load forecasting. As outlined in Section 1.2.3 (Challenge C4), data heterogeneity is a significant issue in real-world FL deployments, particularly in smart energy networks where device-level variance and non-IID data distributions are common. Traditional FL approaches often struggle to handle these diverse data patterns effectively, resulting in inefficient energy allocation, increased operational costs, and heightened risks of grid instability. FedBranched addresses this challenge by employing HMM clustering to group clients with similar performance characteristics, ensuring more accurate and personalized model updates for each cluster.

5.1.2 Methodology

The FedBranched framework effectively groups clients into distinct branches based on the Euclidean distances of their MAPE, assigning a unique global model to each branch. The Hidden Markov Model is a generative probabilistic model that describes a system where a sequence of hidden states $\mathbf{Z}$ generates observable variables $\mathbf{X}$. In this context, the hidden states are not directly observable and transition according to a first-order Markov chain, which captures the temporal dependencies between states. The integration of HMM in the clustering process ensures robust grouping by taking into account the probabilistic nature of the data distribution, as highlighted in previous studies [238]. This allows the framework to adaptively account for variations in client performance and data characteristics.

The framework limits clustering to two branches at each step of the aggregation process to maximize resource utilization and improve computational efficiency. By reducing the maximum number of branches to $n/2$, where n is the total number of clients participating in the training, this restriction effectively reduces the overall number of global models needed. In addition to simplifying model administration, this design decision lowers the communication overhead involved in updating multiple global models.

A generic machine learning model is distributed to each participating client at the start of the procedure. Each client uses this model to train on its own local dataset, resulting in the production of local models. These local models are then sent back to the central server along with their corresponding loss values. The server aggregates the local models into a single global model, represented by M. The loss function employed in this framework is the MAPE.

After completing a predefined number of communication rounds, the server evaluates the MAPE for all clients. If convergence is not achieved, clients are divided into two branches, labeled Branch 1 and Branch 2, using HMM clustering applied to the Euclidean distances of their MAPEs. A new generic ML model is then provided to each branch, and the local models from each branch are aggregated to create branch-specific global models, M_1 and M_2.

5.1.3 Simulation Setup

To evaluate the performance of FedBranch in comparison to vanilla FL, we established an FL environment comprising nine clients, utilizing a real-world energy dataset sourced from PJM Interconnection LLC [229]. This dataset provides a comprehensive view of energy consumption patterns across different substations and is specifically designed for short-term load forecasting applications. Each column in the dataset corresponds to energy usage recorded at a specific substation, allowing for the analysis of energy consumption trends over time.

The dataset encompasses a wide range of energy usage scenarios, reflecting variations in consumption due to factors such as time of day, day of the week, and seasonal changes. This diversity in the data is essential for training robust models that can generalize well to different

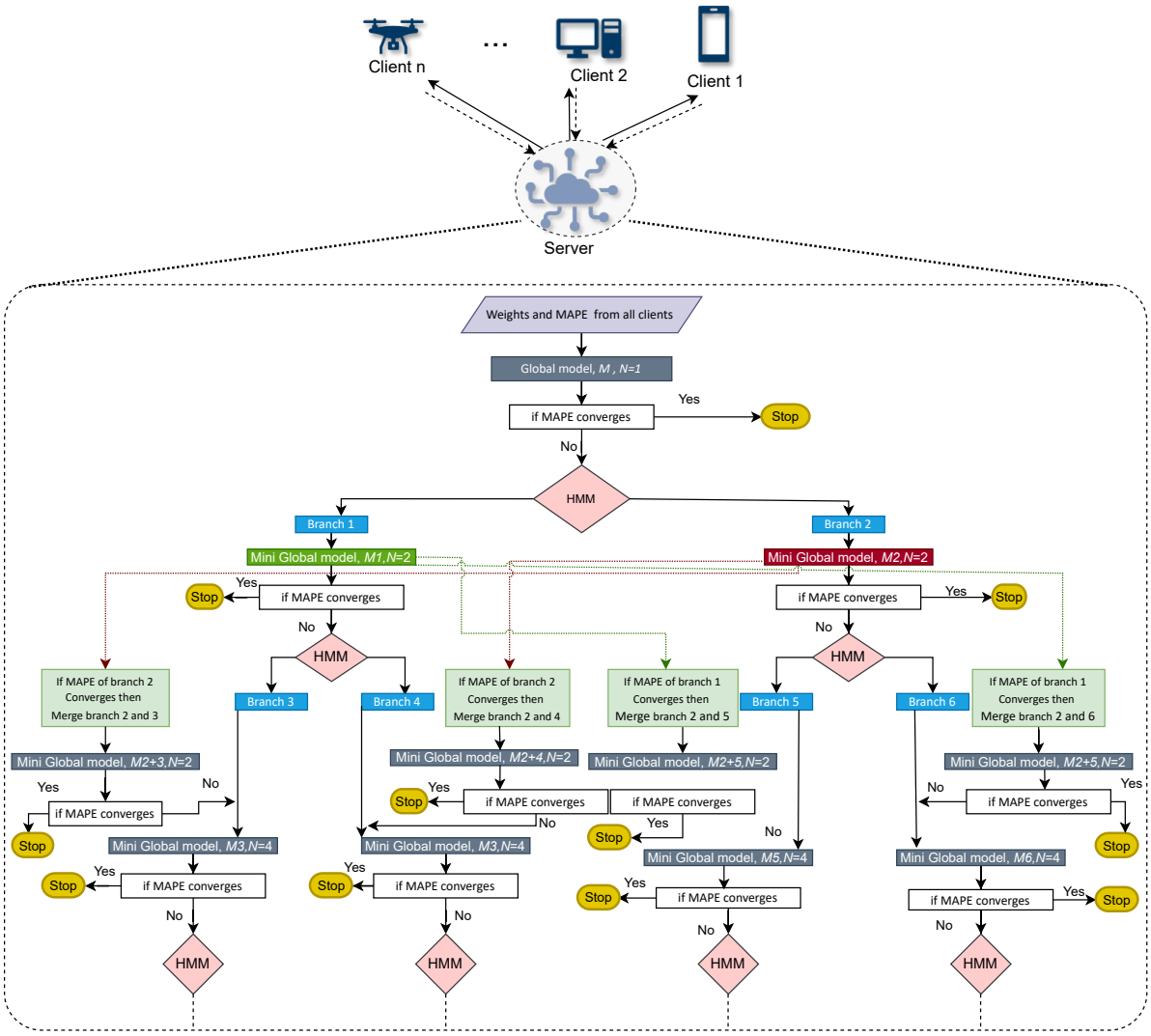


Figure 5.1: Framework of FedBranched.

conditions. Each client in the study was allocated 13,896 samples, which are drawn from the energy consumption records of individual substations. These samples are time-series data points that capture the dynamic nature of energy usage, making them suitable for training models to predict future energy demand.

As illustrated in Fig. 5.2, the energy consumption patterns vary significantly across different substations. The figure shows the energy consumption (in MW) over a range of sample numbers for each of the nine clients. The distinct lines represent the energy usage trends for each client, highlighting the diversity in consumption patterns. For example, client 1 exhibits the highest mean energy usage, with consumption levels frequently exceeding 15,000 MW. This suggests it may represent a substation serving a densely populated or industrial area with consistently high demand. Client 7, on the other hand, demonstrates the lowest mean energy usage, typically below 5,000 MW, potentially indicating a substation in a residential area with lower overall consumption. Client 2 is identified as having the most outliers, indicated by sudden spikes in

energy consumption that deviate from the typical patterns observed in the dataset. These outliers could be due to specific events or anomalies in energy usage, such as sudden increases in demand during particular periods.

The *p*-values from Levene's test are shown in Table 5.1. Consistently low *p*-values across all client pairs highlight significant differences in energy consumption variability, confirming the presence of data heterogeneity within the smart grid network.

Table 5.1: Matrix of p-values of Levene's test for Data 1

	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9
C 1	0.0	1.02×10^{-59}	0.0	0.0	8.39×10^{-1}	0.0	0.0	0.0	0.0
C 2	1.02×10^{-59}	0.0	0.0	0.0	1.17×10^{-56}	0.0	0.0	0.0	0.0
C 3	0.0	0.0	0.0	0.0	0.0×10^{0}	3.64×10^{-226}	2.94×10^{-1}	0.0	0.0
C 4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C 5	8.39×10^{-1}	1.17×10^{-56}	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C 6	0.0	0.0	3.64×10^{-226}	0.0	0.0	0.0	4.02×10^{-183}	0.0	0.00
C 7	0.0	0.0	2.94×10^{-1}	0.0	0.0	4.02×10^{-183}	0.0	0.0	0.0
C 8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.83×10^{-118}
C 9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.83×10^{-118}	0.0

The table above presents the matrix of *p*-values obtained from Levene's test. Each cell in the table represents the *p*-value for the comparison between two clients (e.g., C1 vs. C2, C1 vs. C3, etc.). The diagonal elements are zero because the variance comparison within the same group is not meaningful.

- **Low *p*-values (e.g., 1.02×10^{-59}):** These indicate significant differences in variances between the compared groups, suggesting heterogeneity in energy consumption patterns.
- **High *p*-values (e.g., 0.0 or 8.39×10^{-1}):** These suggest that there is no significant difference in variances between the compared groups, indicating homogeneity in energy consumption patterns.

The dataset was specifically designed for STLF and includes five key features that are predictive of future energy demand: the last-hour value, the last-day value, the last-week value, the 24-hour average, and the weekly average. These features were selected based on their relevance to forecasting energy load and their ability to capture temporal trends in energy usage.

For the modeling aspect, a three-layer ANN was designed for STLF. This architecture consists of an input layer with 100 neurons, a hidden layer with 50 neurons, and an output layer with a single neuron to predict energy load. All layers employ the ReLU (Rectified Linear Unit) activation function, which is effective in mitigating the vanishing gradient problem often encountered in deep learning. The Adam optimizer, recognized for its adaptive learning rate capabilities, was utilized, and mean squared error served as the loss function to quantify prediction accuracy.

The dataset was split into training and testing subsets with a ratio of 70/30, ensuring effective validation against unseen data. The FL process was conducted over 30 communication

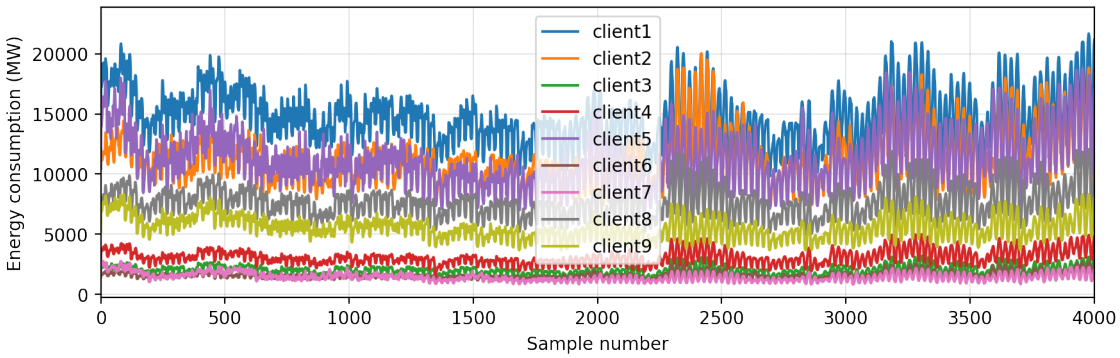


Figure 5.2: Dataset with nine substations.

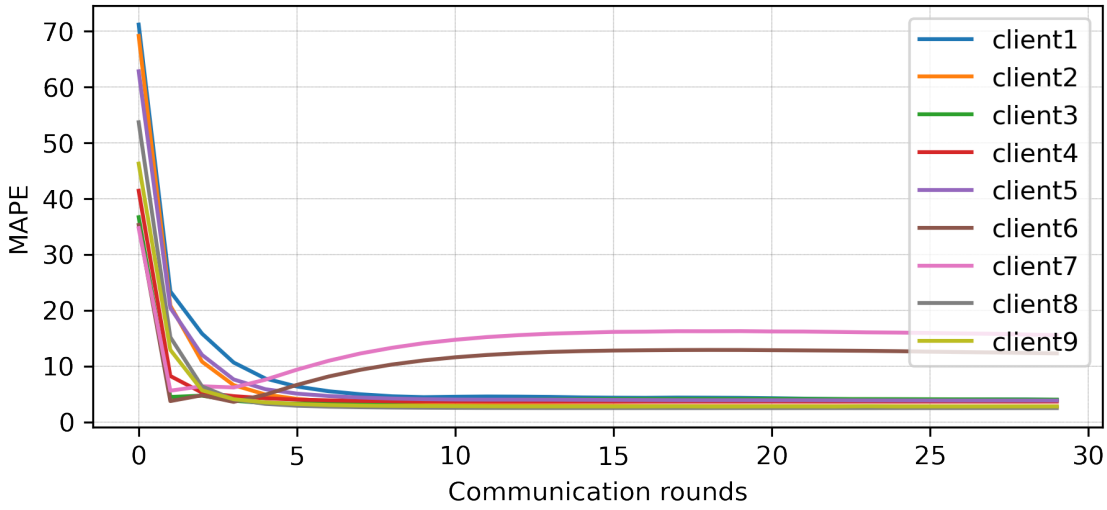


Figure 5.3: Baseline results representing MAPE of all clients during training process of global model.

rounds, with each client performing 15 local epochs to refine models. A batch size of 300 was used during local training, allowing for efficient processing of respective datasets. FedAvg was employed as the server aggregation method, which combines the locally trained models to produce a consolidated global model, thereby enabling collaborative learning while preserving data privacy.

5.1.4 Experiments and Results

Baseline results: Fig. 5.3 shows the MAPE for each client's local model across communication rounds in traditional FL. By the eighth round, most clients converge, except for clients 6 and 7. After 30 rounds, client 2 achieves the best MAPE of 2.51%, while client 7 has the worst at 15.71%. These results indicate that a single global model is inadequate, and clients 6 and 7 may require a separate global model.

Fedbranched results: Fedbrached results in two mini global models after two clustering

Table 5.2: Comparison of MAPE between Traditional FL and FedBranch.

Client	Vanilla FL (%)	FedBranch (%)	Percentage Improvement (%)
1	4.13	2.78	1.35
2	2.80	2.66	0.14
3	4.05	2.98	1.07
4	3.63	3.95	-0.32
5	3.88	3.65	0.23
6	12.43	2.61	9.82
7	15.71	4.35	11.36
8	2.51	2.50	-0.01
9	2.85	2.82	0.03

rounds. After running FedBranch for two clustering rounds, two mini global models, $M_{2,4}$ and M_3 , were finalized. Table 5.2 summarizes the results, showing that FedBranched improved performance compared to traditional FL. The highest improvement, 11.36%, was observed for client 7, while client 4 experienced a slight decline of -0.32%. The clustering mechanism used by FedBranch is illustrated in Fig. 5.4, where clients 3, 4, and 7 formed one cluster, and clients 1, 2, 5, 8, and 9 were grouped into another. Each cluster was assigned its own global model. These results demonstrate that FedBranch effectively enhances forecasting accuracy for highly diverse datasets. The average MAPE across all clients decreased from 5.172% in traditional FL to 2.83% with FedBranch, highlighting its ability to address data heterogeneity. The results showing the convergence of all the MAPE of all clients is presented in Figs. 5.5(a) and 5.5(b). Fig. 5.5(a) represents MAPE form client number 3, 6 and 7 while Fig. 5.5(b) shows the MAPE of cleint number 1, 2, 4, 5, 8 amd 9

5.1.5 Design Insights:

FedBranched is an innovative framework that employs a probabilistic clustering approach tailored for FL with heterogeneous data. One of its key strengths is that it requires no prior knowledge of the dataset, making it adaptable to various applications. The framework ensures convergence of the loss function by clustering clients based on the sum of Euclidean distances of their respective loss values, utilizing HMM to enhance the robustness of the clustering process.

To optimize model aggregation, FedBranched integrates a multi-stage clustering mechanism that minimizes the total number of clusters and global models. By restricting each stage to a maximum of two clusters, the framework provides better control over the training process and facilitates more efficient model updates. Furthermore, by leveraging the loss functions rather than the model weights, FedBranched maintains compatibility with techniques such as differential privacy [18] and homomorphic encryption [18], thereby ensuring that data security and privacy are prioritized throughout the training process.

Despite these advantages, FedBranched does have its limitations. For instance, it is less

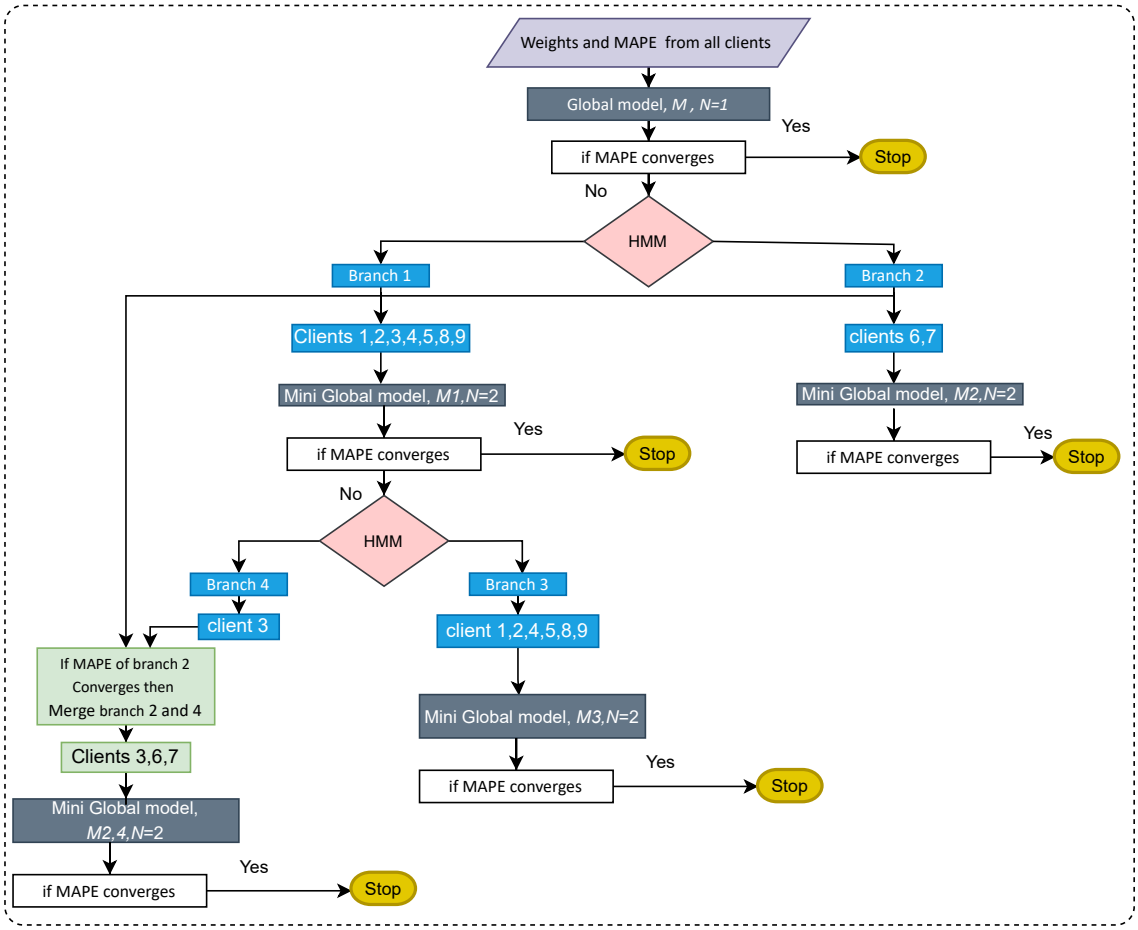


Figure 5.4: Graphical elaboration of FedBranch on considered example.

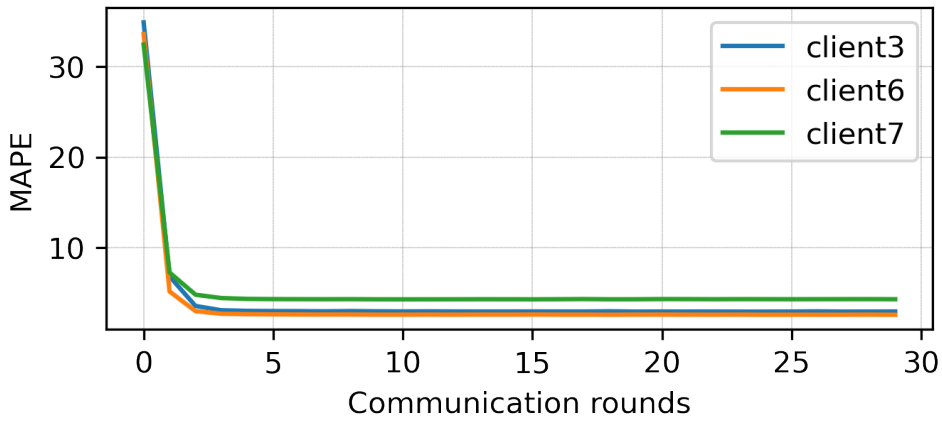
effective when the number of clients is fewer than five. In such cases, the lack of sufficient data diversity can hinder accurate clustering, which is crucial for optimal model performance. Additionally, the multi-stage clustering approach necessitates increased computational resources and energy consumption.

In terms of operational efficiency, while Vanilla FL required only 30 communication rounds, FedBranched necessitated 150 rounds (30 rounds per stage for five stages). This substantial increase in communication rounds significantly escalates the overall computational effort required for training.

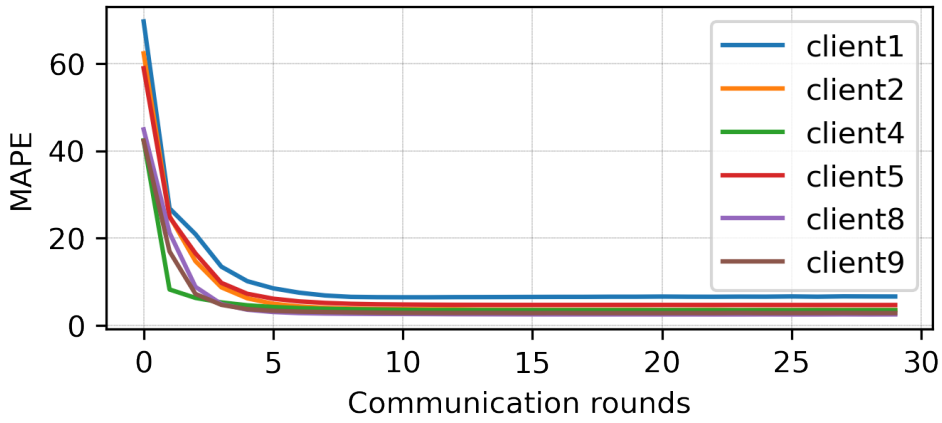
Energy consumption during the training process, denoted as E_{com}, is influenced by various factors such as data transfer size, the efficiency of the communication channel, and computation time. Following the formulation from prior research [219], E_{com} is calculated using the equation:

$$E_{com} = R[(\alpha \cdot t) + (\beta \cdot D)]$$

In this equation, R represents the number of communication rounds, α signifies the energy consumed per second (in kWh), t denotes the computation time (in seconds), β indicates the energy consumed per kilobyte (KB) of data transferred, and D refers to the data transfer size (in



(a) FL Results when branch 2 + 4.



(b) FL Result of branch 3.

Figure 5.5: FL results of 2nd round of clustering after 30 communication rounds.

KB).

In these experiments, i found that each global or local model had a size of 38 KB. The energy consumption per KB, denoted as β, was set at 0.015 kWh/GB [239], while the energy consumed per second, α, was determined to be 0.0001 kWh/sec. For the Vanilla FL setup, with computation time $t = 3.39$ minutes (equivalent to 203.4 seconds), the calculated energy consumption E_{com} was approximately 27 W. In contrast, the FedBranched framework saw an increase in energy consumption to 136 W, primarily due to the additional communication rounds required for effective clustering and model aggregation.

5.2 Adaptive Single Layer Aggregation (ASLA)

The following subsections delve deeper into the architectural design of ASLA, illustrating how its adaptive framework dynamically adjusts to client resources. Furthermore, an in-depth quan-

tization analysis is provided to demonstrate how ASLA reduces communication overhead and enhances computational efficiency through precision optimization. At the end ASLA is applied to STLF in heterogeneous STLF application.

5.2.1 Motivation

The Adaptive Single-Layer Aggregation (ASLA) framework is an advanced solution designed to tackle several critical challenges in FL , particularly in domains such as energy networks and load forecasting. These challenges include data heterogeneity, which arises when clients possess diverse datasets that may vary significantly in distribution and characteristics; resource constraints, where computational power and energy availability are limited; and communication overhead, which refers to the bandwidth and energy costs associated with transmitting model updates between clients and the central server. Traditional FL aggregates all layers of the neural network models from participating clients, which can lead to excessive communication costs and inefficiencies, particularly for resource-constrained devices. ASLA addresses these issues by aggregating only selected layers based on device capability and experimental results, thereby reducing communication costs and enhancing computational efficiency while maintaining model accuracy.

5.2.2 Architecture

ASLA simplifies the aggregation process by focusing on a single layer of neural networks. Resource-constrained devices aggregate only the last layer, while more capable devices can aggregate an optimal layer determined through experimentation. This approach significantly reduces the amount of data transmitted during model updates, leading to more efficient use of resources and faster convergence times. The framework dynamically adjusts based on client resources to balance global and local learning needs, as illustrated in Fig. 5.6.

Stopping Criteria: Efficient training in FL necessitates mechanisms that prevent resource wastage during redundant iterations, which can be particularly detrimental in environments with limited computational and communication resources. ASLA incorporates dual stopping criteria implemented at both the client and server levels to enhance training efficiency:

1. **Client Level:** Training on a client is halted when its loss function exhibits no improvement for a specified number of consecutive communication rounds. This criterion eliminates unnecessary computational effort and reduces network transmission for models that are not converging.

2. **Server Level:** At the server level, training globally terminates when a predefined percentage (x) of clients cease sending updates. This criterion ensures consistency in model performance and enhances the overall efficiency of the FL process.

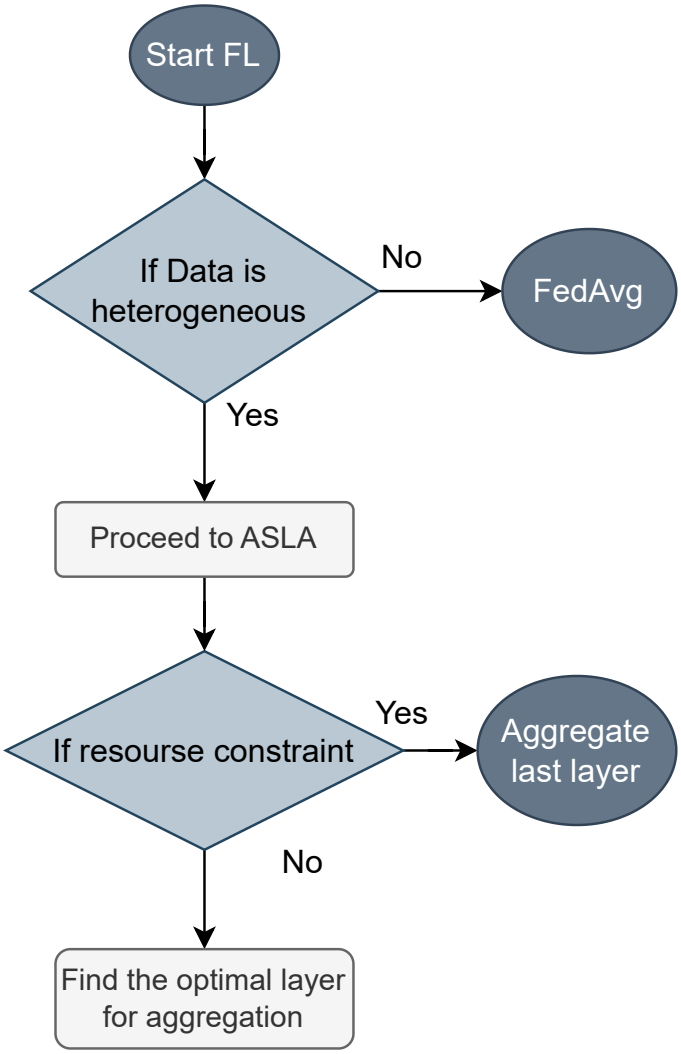


Figure 5.6: Adaptive framework of ASLA.

The early stopping mechanism employed by ASLA is visually summarized in Fig. 5.7, which highlights its critical role in optimizing both communication and computational resources.

5.2.3 Quantization Analysis

To address the high communication and memory demands in FL, ASLA employs weight quantization. This technique reduces neural network weights from the standard 32-bit floating-point representation to a more compact 8-bit fixed-point precision. This reduction in weight size leads to a drastic decrease in the amount of data transmitted during model updates, resulting in several key benefits:

1. **Lower Bandwidth Usage:** The smaller model sizes facilitate faster data transmission, particularly in environments where bandwidth is limited.

2. **Reduced Memory Requirements:** Quantized weights occupy significantly less memory, making ASLA suitable for deployment on edge devices with constrained storage capacity.

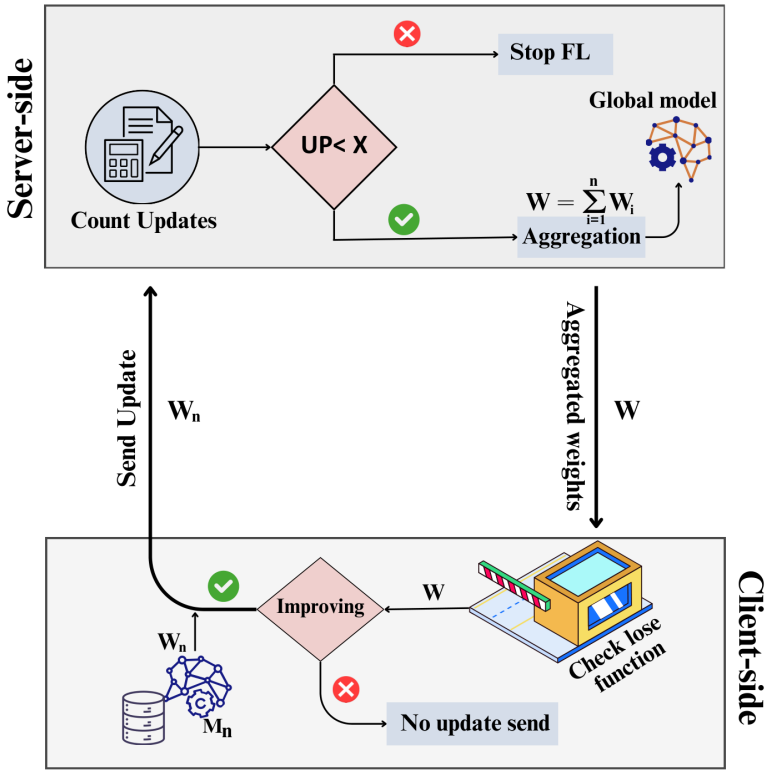


Figure 5.7: Block diagram of early stopping criteria.

3. **Improved Communication Efficiency:** Faster model updates enhance the overall efficiency of the FL process, enabling more timely forecasting and decision-making.

By default, Python employs a 32-bit floating-point representation, which adheres to the IEEE 754 standard [240]. This representation offers a wide range of numbers with variable precision but comes at the cost of increased computational complexity and higher hardware resource consumption. In contrast, fixed-point arithmetic provides a simpler and more efficient alternative, particularly for applications with limited computational power and hardware resources. The $Q_{n.m}$ format represents a number using n bits for the integer part and m bits for the fractional part. This method reduces memory usage and computational overhead, enabling faster processing and simpler hardware implementations [195].

5.2.4 Experiments and Results

To evaluate the Adaptive Single-Layer Aggregation (ASLA) framework and compare its performance with that of vanilla FL , a FL environment consisting of ten clients was established. This setup utilized real-world energy data sourced from PJM Interconnection LLC [241]. Each client was assigned a total of 13,896 samples, showcasing significant diversity in data across the clients. This data heterogeneity is visually represented in Fig. 5.8, which illustrates the variations in the load profiles among the participating clients.

The collected data was specifically employed for STLF, utilizing five key features: the previous hour's load value, the previous day's load value, the previous week's load value, the average load over the last 24 hours, and the average load over the last week [24]. These features were carefully selected to capture the temporal dynamics of energy consumption, which are crucial for making accurate predictions.

For the modeling process, a three-layer artificial neural network (ANN) was constructed to effectively learn the underlying patterns in the data. The architecture of the ANN consisted of 100 neurons in the first layer, 50 neurons in the second layer, and a single neuron in the final layer, which outputs the forecasted load value. All layers employed the ReLU activation function to introduce non-linearity into the model, enhancing its ability to capture complex relationships in the data. The Adam optimizer was utilized to minimize the mean square error, serving as the loss function, which is a common choice for regression tasks.

To ensure robust evaluation, the dataset was divided into a training set and a testing set in a 70/30 ratio. The FL process was executed for 100 communication rounds, with each client performing 1 local epoch per round and utilizing a batch size of 300 samples. For server-side model aggregation, the FedAvg algorithm [231] was employed, which averages the model updates from all participating clients to produce a global model that benefits from the collective knowledge of the decentralized data.

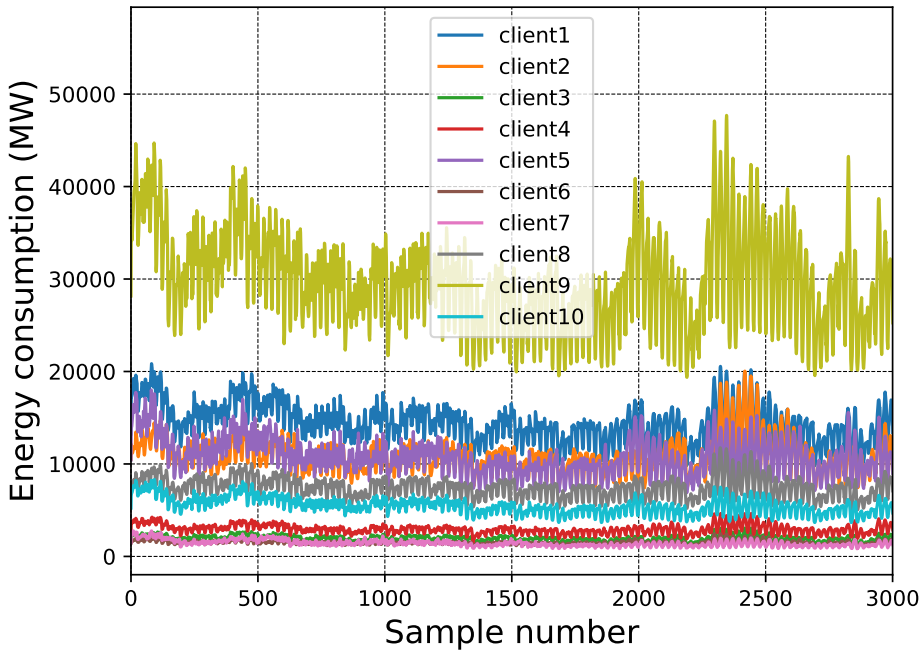


Figure 5.8: A sample of used dateset containing ten different clients

Baseline results: After 100 communication rounds of FL, the MAPE of the local models did not converge effectively, as shown in Fig. 5.9. Only Clients 2, 8, and 10 exhibited signs of

convergence, while Clients 6 and 7 faced significant challenges. The average MAPE was a concerning 79.1 percent, indicating that the models struggled to stabilize around a consistent error rate. This lack of convergence suggests that clients may be overly influenced by updates from others, which can obscure the unique characteristics of their local datasets. As a result, high model diversity may become detrimental, preventing effective learning. In contrast, centralized learning, where each client trains independently on its own data, achieved a MAPE of approximately 3.1 percent. This stark difference underscores the advantages of centralized approaches, allowing for focused learning that captures local data nuances effectively. The elevated MAPE in the FL setup likely stems from clients learning excessively from each other, leading to aggregated models that do not generalize well, ultimately compromising overall system performance.

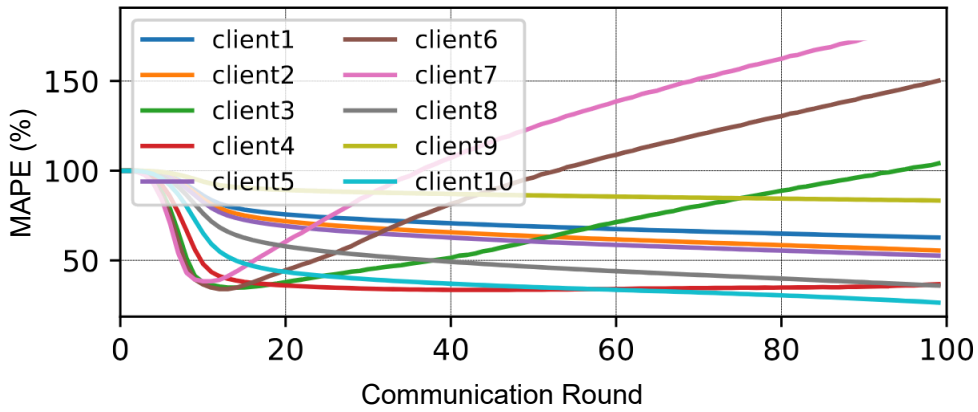


Figure 5.9: MAPE of all local clients in the baseline simulation. The x-axis represents the MAPE of all clients, while the y-axis represents the communication rounds.

Effect of single layer aggregation: Simulations were conducted to improve local learning by aggregating layers in different ways. Initially, only the first layer was aggregated while the others remained unchanged. Gradually, more layers were aggregated. The results, shown in Table 5.3, indicated that not aggregating all layers significantly improved the MAPE for clients. However, when all layers were aggregated, the MAPE increased sharply to 79.1% for the two datasets. This demonstrates that partial aggregation of layers enhances local learning and reduces the effect of other clients' models, leading to better overall performance.

Further testing was done by aggregating each layer individually while leaving others unchanged, as shown in Table 5.4. Aggregating any single layer resulted in an average MAPE of about 3.3%. This suggests that aggregating just one layer is sufficient, reducing communication time and improving model performance. The local MAPE for all clients in both datasets is shown in Fig. 5.10, where the MAPE for all clients converges when only the first layer is aggregated.

Effect of Quantization: Communication overhead can be significantly reduced by sharing only one layer of the model, with additional improvements achieved through the quantization of weights in both local and global models. Experiments conducted in Python evaluated various

Table 5.3: Average client MAPE of step by step aggregation

	Layer 1	Layer 1-2	Layer 1-3
MAPE(%)	3.26	3.29	79.1

Table 5.4: Only a single layer aggregation

	Layer 1	Layer 2	Layer 3
MAPE(%)	3.29	3.3	3.32

quantization strategies, including float16, fixed-point formats of 32, 16, and 8 bits. A three-layer neural network was utilized, and only the first layers of local models were aggregated while tracking the average MAPE across clients.

In the 32-bit fixed-point system, 8 bits are allocated for the integer part and 24 bits for the fractional part, offering a wide dynamic range and high precision. The 16-bit fixed-point system designates 5 bits for the integer part and 11 bits for the fractional part, striking a balance between range and memory efficiency. The 8-bit system utilizes 2 bits for the integer part and 6 bits for the fractional part, minimizing resource consumption even further. These configurations are particularly advantageous for embedded systems that operate under strict resource constraints.

The results for Data 1, illustrated in Fig. 5.11, show only minor variations in the Mean Absolute Percentage Error (MAPE) when reducing the bit depth from 32 to 16 bits, with both configurations yielding an MAPE of approximately 3.3%. Even with the 8-bit fixed-point format, the MAPE remains at 3.4%, which is deemed acceptable for the intended application. This demonstrates that quantization not only reduces communication overhead but also maintains an adequate level of accuracy suitable for practical use in resource-limited environments.

Effect of Layer-wise Aggregation and Stopping Criteria:

To enhance energy and communication efficiency in the FL framework, an early stopping rule was implemented. If a client fails to show improvement in Mean Absolute Percentage Error (MAPE) over five consecutive rounds, it ceases sending updates. The server then relies on the last update received from that client. Additionally, if updates are not received from a specified number of clients, denoted as x (in this case, $x = 3$), the aggregation process is terminated.

In these experiments, the number of communication rounds was not fixed, unlike previous setups that used a total of 100 rounds. With 8-bit quantization and a focus on aggregating only the first layer, the average MAPE results for clients are shown in Fig. 5.12. Aggregating the first layer resulted in a MAPE of 3.26% over 114 rounds, while the second layer achieved a MAPE of 3.25% with 102 rounds. The third layer produced a MAPE of 3.50% across 110 rounds, indicating that the second layer performed the best.

These results underscore the benefits of optimizing each layer individually, demonstrating that effective resource and communication management can improve performance. Balancing accuracy and the number of communication rounds is crucial when designing distributed learning systems, as it significantly impacts the overall efficiency and effectiveness of the learning

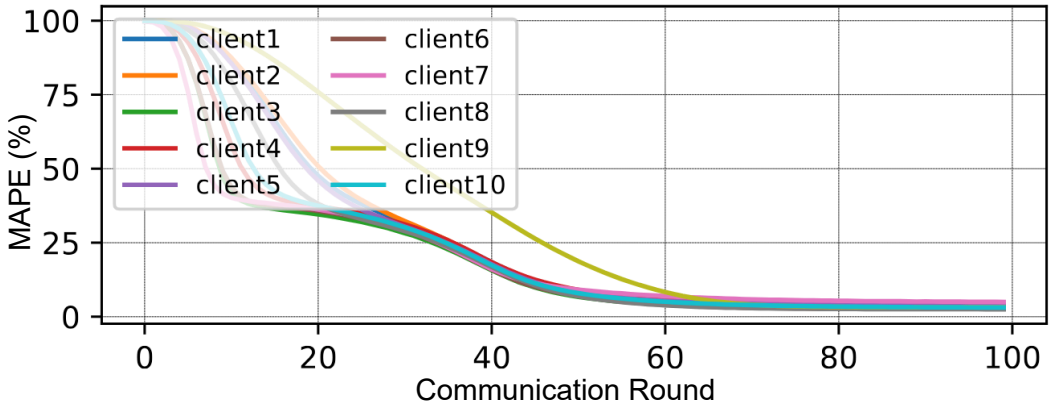


Figure 5.10: MAPE of all clients during training of global model when only first layers were aggregated. The x-axis represents the MAPE of all clients, while the y-axis represents the communication rounds.

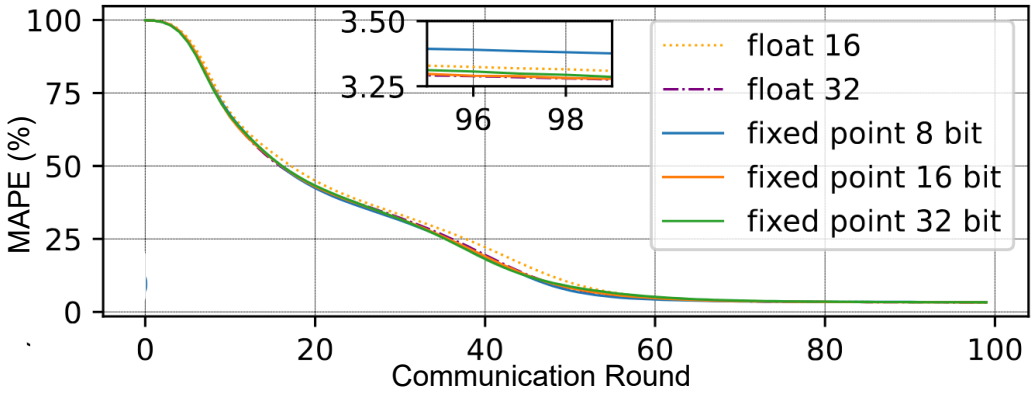


Figure 5.11: The effect of quantization on communication rounds and on average client MAPE. The x-axis represents the MAPE of all clients, while the y-axis represents the communication rounds.

process.

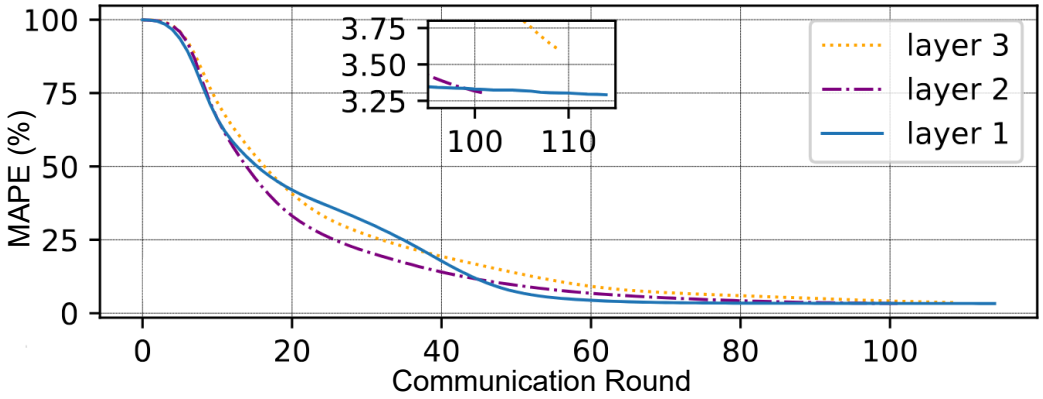


Figure 5.12: The effect of stopping criteria on communication rounds and on average client MAPE. The x-axis represents the MAPE of all clients, while the y-axis represents the communication rounds.

5.2.5 Findings and Insights

The ASLA framework effectively addresses issues of data heterogeneity in distributed load forecasting while improving communication, computational efficiency, and memory usage by sharing only a single layer of the neural network and implementing stopping criteria and quantization.

Key Advantages of the ASLA Framework

1. Enhanced privacy
2. Reduced computation
3. Reduced memory requirements
4. Lower communication costs

Enhanced Privacy: Privacy protection is crucial in distributed learning systems. ASLA enhances privacy by sharing only a single layer of a model's weights. This makes it harder for attackers to reverse-engineer users' private data, minimizing the risk of information leakage. Since shared models typically perform best on the data they are trained on, adversaries can exploit prediction accuracy gaps to deduce sensitive information. By limiting the shared data to a single layer, the ASLA framework reduces the likelihood of such reverse-engineering attacks, offering better privacy protection.

Reduced Computation: Fixed-point arithmetic, particularly with 8-bit quantization, provides several computational benefits over floating-point arithmetic. Floating-point systems,

while more complex, require more silicon area, leading to slower speeds and higher power consumption. In contrast, 8-bit fixed-point systems are simpler, faster, and more energy-efficient. This makes them ideal for applications with power and speed requirements, such as IoT devices. Transitioning from 32-bit floating-point to 8-bit fixed-point systems offers substantial savings in hardware design, speed, and power, leading to improved energy efficiency and reduced operational costs.

Reduction in Memory Usage: Efficient memory usage is crucial in resource-constrained environments like IoT devices [205]. To calculate the size of a neural network, we focus on the weights, excluding biases for simplicity. The total weight size is determined by counting the connections between layers, multiplying by the weight size in bytes (e.g., 32-bit floating-point), and converting to kilobytes.

For a 3-layer network:

- Input to the first hidden layer: $100 \times 100 = 10000$.
- First to second hidden layer: $100 \times 50 = 5000$.
- Second hidden layer to output: $50 \times 1 = 50$.

Total weights: 15050.

With 32-bit weights (4 bytes), the total size is $15050 \times 4 = 60200$ bytes, or 58.72 KB. The breakdown is: first layer (39.06 KB), second layer (19.53 KB), and last layer (0.19 KB). Biases are minimal (0.59 KB) and can be ignored. Similar calculations apply for 16-bit and 8-bit systems, and the LSTM model used with Data 2, summarized in Table 5.5.

Table 5.5: Sizes of Different Layers of Local Model

	32 bit (KB)	16 bit (KB)	8 bit (KB)
Entire NN	58.72	29.36	14.68
Layer 1	39.06	19.53	9.765
Layer 2	19.53	9.765	4.8825
Layer 3	0.19	0.095	0.0475

Communication Cost: Communication cost is intrinsically linked to the volume of data transferred between clients and the server within a distributed learning framework. The ASLA approach mitigates this communication burden by limiting the data shared to a single layer of the neural network. This strategy not only reduces the total amount of data transmitted but also facilitates more efficient utilization of network resources.

The communication cost is quantified by multiplying the volume of data transferred by the number of communication rounds. As illustrated in Fig. 5.13, the choice of which layer to aggregate significantly influences communication costs. Selecting the last layer for aggregation results in substantial reductions in communication costs , with reductions of up to 829.2 times

when transitioning from a 32-bit system to an 8-bit system. This notable decrease in communication overhead enhances scalability and energy efficiency, rendering the system more suitable for large-scale distributed learning applications.

Moreover, the analysis of communication costs revealed significant savings achieved by aggregating later layers and employing fixed-point arithmetic. Specifically, the communication expense for the final layer using an 8-bit fixed-point system was considerably lower, resulting in communication cost reductions of 829.2 times compared to the first layer utilizing a 32-bit floating-point system. This highlights the potential for enhancing FL systems through strategic selection of layers and quantization methods, thus improving efficiency while maintaining model accuracy. For optimal results, it is recommended to utilize the last layer, which provides satisfactory outcomes alongside the lowest communication costs.

The ASLA framework distinctly outperforms several state-of-the-art frameworks. For instance, FedKD [165], a communication-efficient framework based on knowledge distillation, achieved only a 19-fold improvement in communication costs. Another framework, FedProto [242], which employed prototype learning to address heterogeneity in FL, accomplished a 161.25-fold improvement in communication costs compared to FedAvg. A framework introduced in [21] aimed to minimize communication costs but achieved only a 34% enhancement. Similarly, the authors in [243] reported a 95% reduction in communication costs. SmartIdx [244] achieved a 69.2-fold improvement in communication expenses. Lastly, FedPSO [245] used partial swarm optimization to reduce communication costs, resulting in a 55% enhancement.

Furthermore, the framework is considerably more straightforward and can be scaled to larger applications with relatively less complexity. This simplicity not only mitigates implementation challenges but also ensures that the approach can be easily adopted and deployed in practical scenarios, providing substantial advantages in terms of both communication efficiency and operational viability.

5.3 Comparison and Discussion of FedBranched and ASLA

While FedBranched and ASLA both aim to enhance FL systems in energy networks, they employ different strategies and address distinct aspects of the challenges posed by real-world deployments. This section provides a comprehensive comparison of these frameworks, guiding the selection of the most suitable approach based on specific requirements and constraints.

5.3.1 FedBranched vs. ASLA

FedBranched primarily targets data heterogeneity (C4) by using HMM clustering to group clients with similar data patterns, allowing for personalized model training within each cluster. This approach is particularly effective in scenarios where data distributions vary significantly

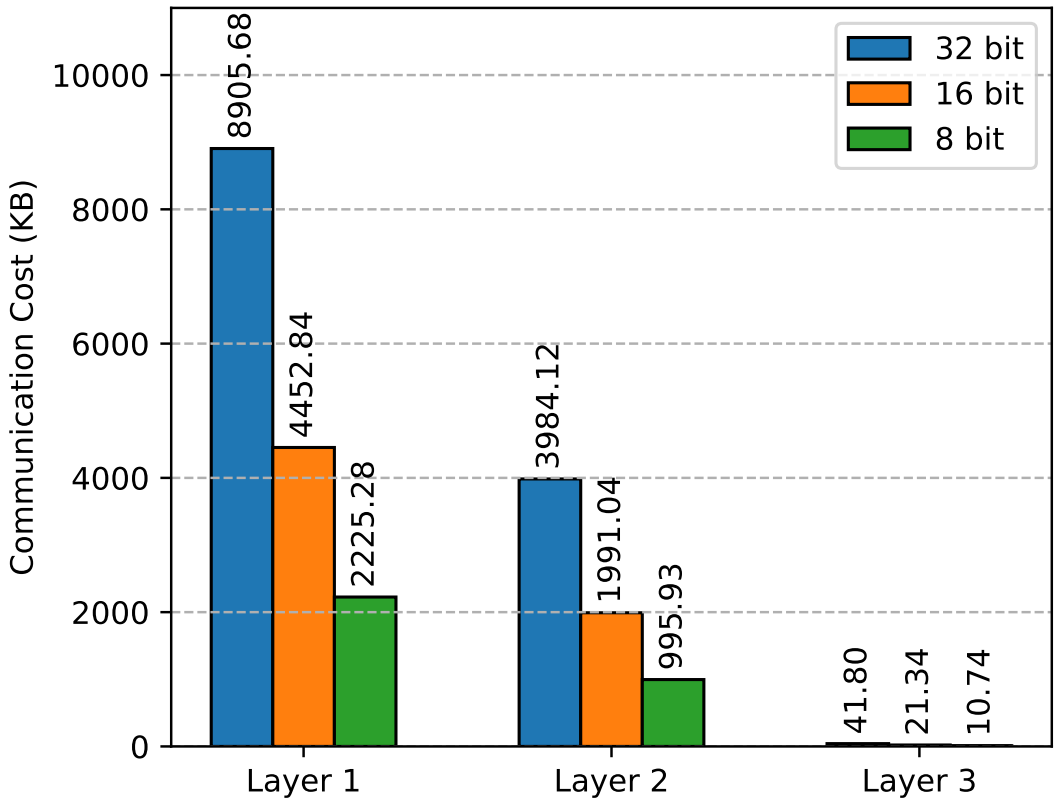


Figure 5.13: Communication costs for different layers of a three-layered neural network, when only single layer was used for aggregation

across clients, such as in smart energy networks with diverse consumer behaviors and grid infrastructures.

ASLA, on the other hand, focuses on optimizing communication and computational efficiency (C3) by aggregating only a single neural network layer. This reduces the communication overhead and makes the framework suitable for resource-constrained environments. Additionally, ASLA also addresses data heterogeneity (C4) by allowing selective layer aggregation based on client capabilities, which helps in maintaining model accuracy while reducing the amount of data transmitted.

5.3.2 When to Use Each Framework

The choice between FedBranched and ASLA depends on the specific requirements and constraints of the application:

- **FedBranched** is more suitable when the primary challenge is data heterogeneity and personalized models are needed for different client clusters. It is particularly beneficial in scenarios where the number of clients is sufficiently large to leverage the clustering advantages. However, FedBranched may require more computational resources due to the clustering process and the management of multiple global models.

- **ASLA** is preferable when communication efficiency and computational resource constraints are critical concerns. Its single-layer aggregation strategy significantly reduces the data transmitted between clients and the server, making it ideal for large-scale deployments with limited bandwidth or high communication costs.

In scenarios where both data heterogeneity and resource constraints are significant, a combination of FedBranched and ASLA could be explored. FedBranched can first cluster clients based on data characteristics, and within each cluster, ASLA can be applied to further optimize communication and computation. This hybrid approach may offer a balanced solution that addresses both challenges effectively.

5.3.3 Summary Comparison

Table 5.6: Summary Comparison of FedBranched and ASLA

Aspect	**FedBranched**	**ASLA**
Primary Challenge Addressed	Data heterogeneity (C4)	Communication and computational efficiency (C3) and data heterogeneity (C4)
Aggregation Strategy	Clusters clients using HMM based on MAPE and assigns unique models to each cluster	Aggregates only a single neural network layer, with options for different layers based on client capability
Resource Utilization	May require more computational resources due to clustering and multiple models	Reduces communication overhead and computational load by limiting aggregation to one layer
Suitability	Ideal for scenarios with significant data distribution variations	Best for resource-constrained environments where communication efficiency is critical
Scalability	Suitable for deployments with a moderate to large number of clients	Highly scalable due to reduced communication and computation requirements
Model Personalization	High, as each cluster has a personalized global model	Moderate, as personalization is limited to layer-specific updates

Understanding the strengths and limitations of FedBranched and ASLA enables practitioners to select the most appropriate framework for their specific use case. FedBranched offers superior handling of data heterogeneity through its innovative clustering approach, while ASLA provides significant advantages in resource-constrained settings with its efficient aggregation strategy. In complex scenarios requiring both heterogeneity handling and efficiency, a hybrid approach combining these frameworks may yield optimal results.

This comparison section can be placed at the end of Chapter 5, after the individual framework discussions, to provide a comprehensive synthesis of FedBranched and ASLA.

5.4 Concluding Remarks

In this chapter, we have explored the significant impact of data heterogeneity on the performance of FL systems, particularly in energy networks. Data heterogeneity, characterized by variations in energy consumption patterns and data distributions across different clients, poses a major challenge to the convergence and accuracy of FL models. Addressing this challenge is crucial for enhancing the robustness and efficiency of FL systems in managing and optimizing energy resources. The key findings and contributions of this chapter can be summarized as follows:

- **FedBranched Framework**: FedBranched addresses the challenge of data heterogeneity (C4) by employing a probabilistic clustering approach using Hidden Markov Models (HMM) to categorize clients based on their data characteristics. This method effectively handles diverse data distributions by creating distinct branches for clients with similar data profiles, allowing for tailored model training within each branch. The results demonstrate a significant improvement in model performance and convergence, with a notable reduction in the average Mean Absolute Percentage Error (MAPE) from 5.172% in traditional FL to 2.83% with FedBranched. This framework effectively mitigates the impact of data heterogeneity, ensuring more accurate and reliable predictions in energy networks.

- **Adaptive Single Layer Aggregation (ASLA) Framework**: ASLA tackles the challenge of communication and computational efficiency (C3) by simplifying the aggregation process. Instead of aggregating all layers of the neural network, ASLA focuses on a single layer, reducing the amount of data transmitted and processed during each communication round. This approach not only minimizes communication overhead but also enhances the scalability of FL systems. The incorporation of quantization techniques further optimizes data transmission, while stopping criteria ensure that training halts when performance plateaus. The results show that ASLA achieves a substantial reduction in communication costs, with an 829.2 times decrease compared to traditional methods, without compromising model accuracy. This makes ASLA highly suitable for resource-constrained environments and underscores its effectiveness in improving the efficiency of FL systems.

This chapter has provided valuable insights into addressing the challenges of data heterogeneity in FL systems. The proposed FedBranched and ASLA frameworks offer effective solutions to enhance the robustness and efficiency of FL models in energy networks. By explicitly addressing the challenges of data heterogeneity (C4) and communication efficiency (C3), these frameworks pave the way for more reliable and efficient FL applications in diverse environments.

Chapter 6

DRLA: A Decentralised Defence Framework for FL

In this chapter, DFL is compared with CFL under adversarial attacks for load forecasting, and performance is evaluated. Furthermore, DRLA is proposed to further reduce the effect of adversarial attacks. Building on the FedRLA framework introduced in Chapter 4, this chapter extends the defence strategy to decentralized settings through the DRLA framework. The DRLA framework leverages the principles of FedRLA while adapting them to the decentralized environment to enhance robustness against adversarial threats. As with the approaches discussed in Chapter 4, DRLA continues the focus on mitigating the impact of adversarial attacks while preserving model accuracy and integrity. This extension to DFL not only addresses the vulnerabilities highlighted in the threat models of Chapter 3 but also aligns with the defence strategies developed in Chapter 4. Specifically, DRLA addresses Challenge C2 (robust aggregation) and Challenge C3 (communication and computational efficiency) by introducing a novel aggregation method that selectively aggregates model layers in a decentralized manner. This approach directly responds to the need for enhanced security and efficiency in FL systems as outlined in Chapter 1.

DFL offers a transformative approach to address the inherent challenges of traditional FL by eliminating the dependency on a central server. Unlike CFL, which relies on a central server for aggregating updates and coordinating the global model, DFL employs a peer-to-peer (P2P) communication model. In this model, nodes can dynamically switch between acting as servers and clients, fostering a more collaborative and resilient environment [246, 247]. This decentralized framework is crucial for enhancing security, fairness, and resilience in FL systems. By removing the central server, DFL mitigates the Single Point of Failure (SPF) vulnerability and reduces the risk of biased or malicious interventions by a central authority [248, 249]. This is particularly important in critical applications like energy networks, where the integrity and reliability of the model are paramount.

The need for DFL is further emphasized by the vulnerabilities associated with CFL. As discussed in Chapter 3, adversarial attacks can significantly impact the performance and security

of FL systems. DFL, by its nature, inherently reduces some of these risks. For instance, the decentralized architecture makes it more difficult for attackers to compromise the entire system, as they would need to target multiple nodes instead of a single central server. This aligns with the defence strategies presented in Chapter 4, where frameworks like FedRLA were introduced to enhance robustness against adversarial attacks. DRLA, as an extension of FedRLA, brings these robust aggregation techniques to decentralized settings, ensuring that the benefits of selective layer aggregation are preserved even in the absence of a central coordination point [155]. By doing so, DRLA not only addresses the specific challenges outlined in Chapter 1 but also advances the field of FL by providing a more secure and efficient alternative to traditional centralized approaches.

The key contributions of this chapter are:

- **DFL Framework:**
  - Presented a novel DFL framework to mitigate adversarial attacks in FL systems.
  - Conducted a comparative analysis of DFL with traditional Centralized FL (CFL) under adversarial attacks for load forecasting.
  - Evaluated the performance of DFL and CFL across various communication topologies (line, ring, and bus), demonstrating the robustness of DFL in limiting the impact of compromised clients.
  - Measured and analyzed the communication costs associated with each topology, highlighting the efficiency benefits of DFL.
- **Decentralized Random Layer Aggregation (DRLA):**
  - Introduced the Decentralized Random Layer Aggregation (DRLA) framework, an adaptation of FedRLA for decentralized settings.
    - Building on the FedRLA framework from Chapter 4, DRLA extends its robust aggregation strategy to DFL environments.
  - Applied DRLA to the line, ring, and bus topologies and tested its performance under attack conditions.
  - Demonstrated how DRLA reduces the impact of adversarial attacks by aggregating only a single, randomly selected layer per round in a decentralized manner.
  - Analyzed the improvements in communication efficiency and model robustness provided by DRLA in decentralized environments.

6.1 Decentralized FL (DFL)

In distributed machine learning, communication rounds between clients and servers are crucial for ensuring effective model training and achieving accurate predictions. In centralized FL (CFL), clients participate concurrently, meaning that multiple clients can send their updates to the server simultaneously. In this context, the order in which the server aggregates these updates has minimal impact on convergence, as the server can effectively average the contributions from all clients without being significantly affected by the sequence of incoming data, which allows for a more streamlined and efficient learning process [250]. However, in DFL, the iteration order of clients plays a significant role in determining the performance of individual models. The lack of a centralized server in DFL means that the order in which clients communicate their updates can lead to variations in model convergence and accuracy, making it a critical factor in the learning process [251]. To optimize performance in DFL, various strategies for managing client iteration order can be implemented, including sequential patterns, where clients update the server in a fixed order; cyclic patterns, which ensure that all clients are included in the update process over time; random patterns, allowing for varied participation that can enhance robustness; and parallel patterns, where multiple clients send updates concurrently. Each of these strategies influences not only the convergence speed but also the overall efficacy of the model, highlighting the importance of effective communication round management in decentralized environments [252].

In STLF, efficient communication round management becomes paramount. STLF relies on the collaborative training of models across diverse datasets distributed among various clients, such as smart meters and energy sensors, to predict future energy consumption accurately. The application of STLF in smart grids necessitates a robust and efficient FL framework due to the dynamic nature of energy consumption patterns and the need for real-time predictions. The communication strategies employed in DFL directly impact the model's ability to adapt to these dynamic conditions while preserving data privacy and ensuring secure model updates. By optimizing communication rounds through strategies such as sequential, cyclic, random, and parallel patterns, DFL can enhance the convergence and accuracy of STLF models, making it a suitable choice for this critical application in smart energy systems

6.1.1 Communication Methods

- **Pointing:** A simple, one-to-one communication model resembling direct interaction, where a node sends a specific message to another. This method is efficient for targeted updates but may not scale well in larger networks due to increased communication overhead.
- **Gossip Protocol:** A decentralized approach involving random, peer-to-peer exchanges that facilitate fluid information sharing among nodes. Each node periodically selects a peer to exchange information, effectively spreading updates throughout the network.

This protocol enhances resilience and convergence speed, particularly in dynamic environments [253].

- **Broadcast Protocol:** A one-to-all communication model where updates are disseminated simultaneously across the network. This method ensures that all nodes receive the same information at the same time, promoting synchronization. However, it may lead to network congestion if the volume of data is high, necessitating careful management of bandwidth [254].

DFL networks leverage diverse topologies like grids, rings, and fully connected structures, each with unique convergence characteristics [255]. Unlike CFL, DFL involves multiple model versions and lacks a centralized server, complicating knowledge consolidation and access for clients.

6.1.2 Design Consideration

In this work, a pointing and broadcast iterative method is adopted with a sequential communication protocol across three topologies: line, ring, and bus. The choice of these topologies facilitates efficient communication while aligning with decentralization principles, thereby avoiding star or mesh structures that can introduce unnecessary complexity and single points of failure.

The sequential communication protocol ensures methodical information flow by organizing the order of message exchanges, which enhances clarity and reduces the risk of data collisions. This systematic approach allows nodes to process updates in an orderly fashion, promoting synchronization within the network.

In contrast, the broadcast mechanism enables simultaneous transmission of updates to all nodes within the network, significantly enhancing robustness against connectivity disruptions. By allowing every node to receive critical information at the same time, the system can maintain operational integrity even if some connections fail. This dual approach of pointing for targeted communication and broadcasting for widespread dissemination effectively balances efficiency and reliability in decentralized settings.

6.1.3 DFL with Line Network Communication Topology

To transition from centralized FL (CFL) to decentralized FL (DFL) while maintaining consistency, a line communication topology is adopted. In this setup, Client 1 initiates the process by sending its locally trained model to Client 2. Client 2 then aggregates this model with its own, resulting in an updated global model. This refined model is then passed sequentially through Clients 3, 4, and 5, with each client contributing to further enhancements based on their local data. Upon reaching Client 5, the model follows a reverse path back to Client 1, completing one full communication round. This round-trip communication ensures that all clients can benefit

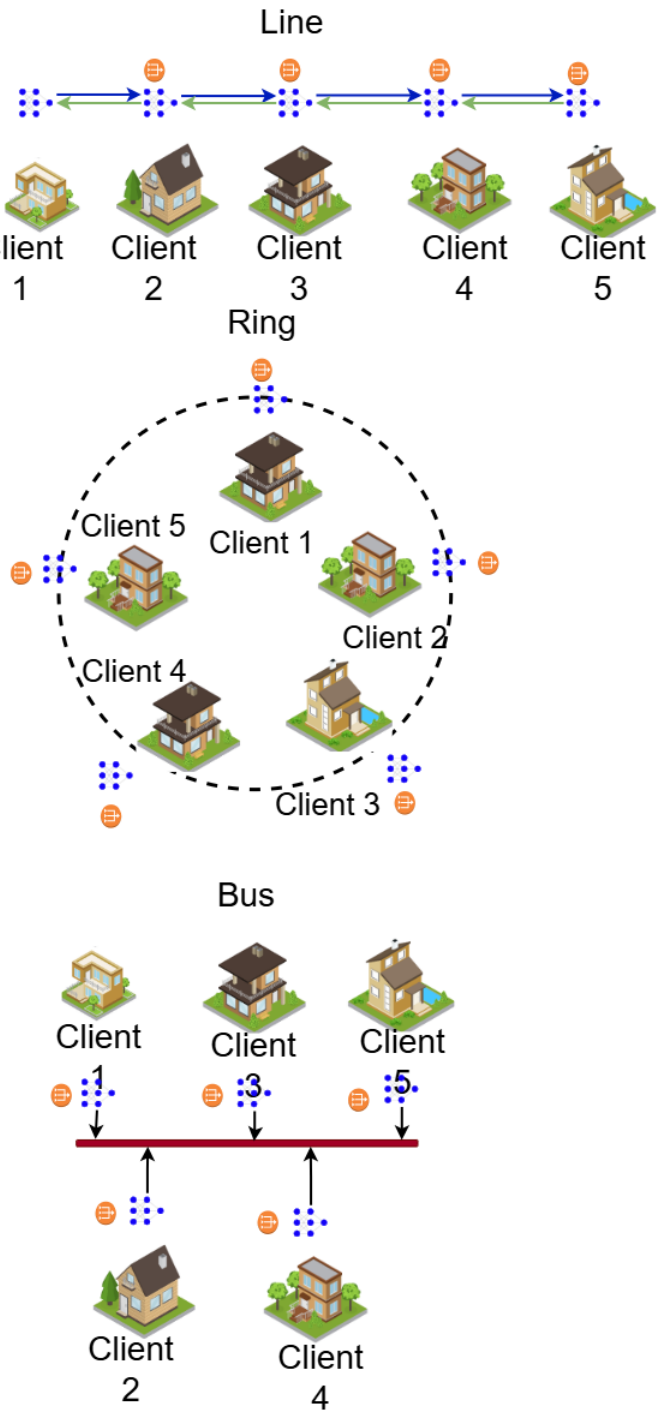


Figure 6.1: Line, Ring and Bus ring topologies used in DFL for load forecasting

from the updates made at each stage. At every aggregation point, the process is executed using the aggregation equation defined in Equation 2.3. The structure and flow of the Line-DFL topology are illustrated in Fig. 6.1, highlighting the sequential nature of communication and collaboration among the clients.

6.1.4 DFL with Ring Network Communication Topology

In the ring topology, Client 1 initiates the communication process by sharing its locally trained model with Client 2. Client 2 then aggregates this model with its own, creating an updated version, which is subsequently forwarded to Client 3. This process continues as the model cycles through Clients 4 and 5, effectively forming a closed loop. Upon reaching Client 5, the global model is passed back to Client 1, marking the completion of one iteration. Communication rounds are considered complete when the model returns to its origin, ensuring that all clients participate in the iterative refinement of the model. At each stage of this process, aggregation is performed using Equation 2.3, facilitating consistent updates across the network. The structure and flow of the Ring-DFL topology are illustrated in Fig. 6.1, emphasizing the interconnected nature of client communication and model aggregation.

6.1.5 DFL with Bus Network Communication Topology

In the bus topology, Client 1 initiates the process by broadcasting its locally trained model to all other clients in the network. Each subsequent client—Clients 2, 3, 4, and 5—then broadcasts its own updated model to the rest of the clients. This collective broadcasting continues until every client has contributed, thus completing one communication round. This topology is particularly advantageous for ensuring fault tolerance, as the process can still proceed even if one or more clients fail to respond, allowing for robust model updates despite potential communication disruptions. The structure and flow of the DFL Bus topology are illustrated in Fig. 6.1, highlighting the collaborative nature of client interactions and the inclusive model aggregation process.

6.2 Experiments and Results

In this experiment, we used Dataset 1, as described in Section 3.1.1. The dataset underwent the same preprocessing steps and featured the same attributes and deep learning model.

6.2.1 Baseline Results

In the experiments, baseline results for CFL were obtained by conducting 20 communication rounds, with each client performing 1 local epoch of training on their individual datasets. Fig. 6.2 displays the MAE for all clients throughout the global model's training process, revealing

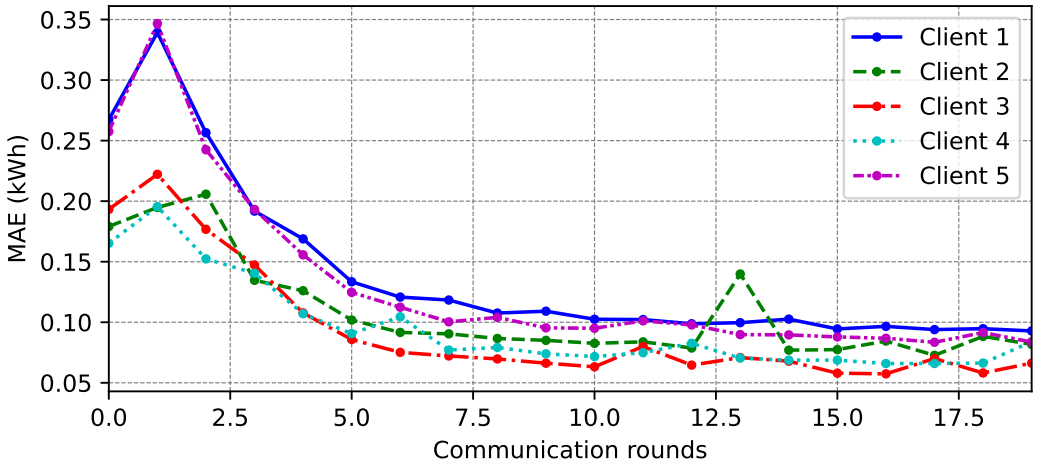


Figure 6.2: Client-wise MAE for all clients during the global model's training phase.

an average MAE of 0.076 kWh, which indicates a strong predictive capability of the model. This value suggests that the predictions made by the model were consistently close to the actual energy consumption values.

Additionally, baseline results for DFL are presented in Fig. 6.3, featuring the Mean MAE for a prediction task involving five clients using three distinct distributed FL topologies: line, ring, and bus. Each topology represents a different configuration for client communication, affecting how updates and model parameters are shared among clients. Remarkably, the average MAE remained consistent at 0.076 kWh across all topologies, as shown in Fig. 6.2. This consistency underscores the robustness of the model across various network configurations, suggesting that it can maintain high performance regardless of the underlying communication structure.

The learning behaviours of these three communication topologies were strikingly similar, closely resembling the performance observed in CFL, as indicated in Fig. 6.2. This finding suggests that the choice of topology did not significantly impact the model's learning efficacy. Such results are encouraging for practical applications, as they imply that system designers can select from multiple topologies without fearing a detrimental effect on model accuracy. Moreover, this flexibility can facilitate easier implementation in diverse environments, allowing for optimization based on specific operational constraints or preferences. Overall, the experiments highlight the model's adaptability and effectiveness in various decentralized settings.

6.2.2 Communication Cost

Communication cost is calculated as the amount of data transfer between clients and the server. This metric is crucial in evaluating the overall performance of distributed systems, as high communication costs can lead to increased latency and resource consumption. The energy efficiency of any system is directly related to the amount of transmitted data. To calculate communication, it is necessary to determine the size of the local and global models. The primary factor in de-

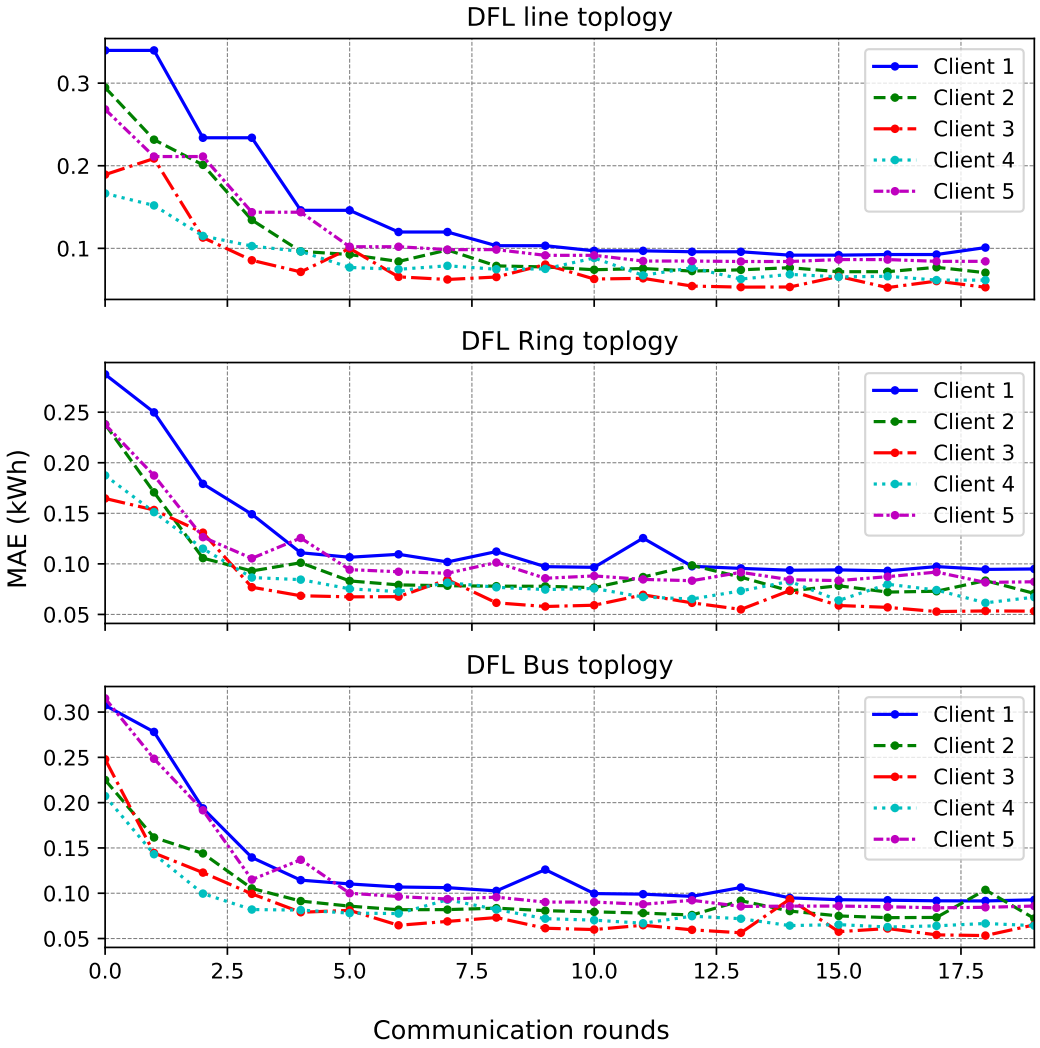


Figure 6.3: Client-wise MAE observed during the global model's training in the context of DFL.

termining the size of a neural network model is the total number of parameters, which includes both weights and biases. Each layer's parameters are calculated by multiplying the number of input units by the number of output units, with an additional parameter for each output unit to account for biases.

The designed deep learning model had three dense layers, the first layer's parameters are the product of the input size plus one (for the bias) and the 64 output units. The second layer's parameters result from multiplying the 64 units from the previous layer by the 32 output units, again adding one for bias. The final layer reduces the output to a single unit, contributing an additional set of parameters. The total number of parameters in this case is 2561, which translates to approximately 10 KB of memory, as each 32-bit floating-point parameter occupies 4 bytes.

In CFL, the communication cost is calculated by measuring the data transmitted by a single device, multiplying it by the number of devices, and then multiplying that result by the num-

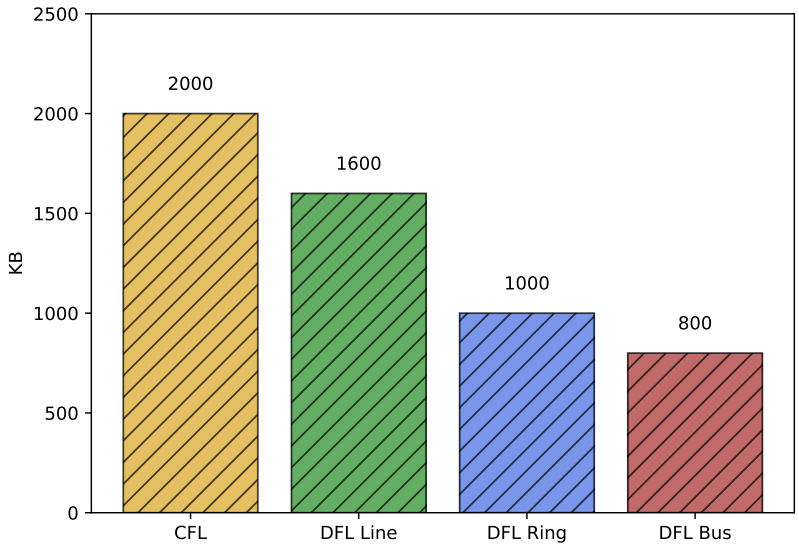


Figure 6.4: Communication cost of CFL and DFL frameworks.

ber of communication rounds. The same amount of data is also transmitted by the server. In DFL Line, Client 1 sends the model to Client 2, and this process continues until the last client is reached. The process then reverses direction. In each communication round, the model is transmitted by $2 \times (\text{number of devices}) - 2$. Multiply this amount by the number of communication rounds to get the total communication cost. In Ring DFL, Client 1 sends data to Client 2, and the process continues until the last client is reached. Then, the last client sends the model back to Client 1. Thus, the number of model transmissions is equal to the number of clients, which is then multiplied by the total number of communication rounds. In Bus DFL, during each communication round, only one client sends the model to all other clients. Therefore, the total number of model transmissions is equal to the number of devices minus one, and this is multiplied by the number of communication rounds to calculate the communication cost.

The exact communication cost is plotted in Figure 6.4.t is evident that CFL exhibits the highest communication cost, followed by DFL Line, DFL Ring, and finally DFL Bus. This variation is attributed to the differences in topologies, which influence the efficiency of data transmission between clients and the server. CFL requires each device to communicate extensively, leading to increased data transfer and higher costs. In contrast, DFL Line minimizes communication by allowing sequential data transfer, thereby reducing the overall cost. DFL Ring further optimizes communication by creating a circular path for data exchange, which enhances efficiency. Finally, DFL Bus allows a single client to communicate with all others, but still incurs lower costs compared to CFL due to reduced redundancy in data transmission. Thus, the choice of topology plays a critical role in determining communication costs in distributed systems.

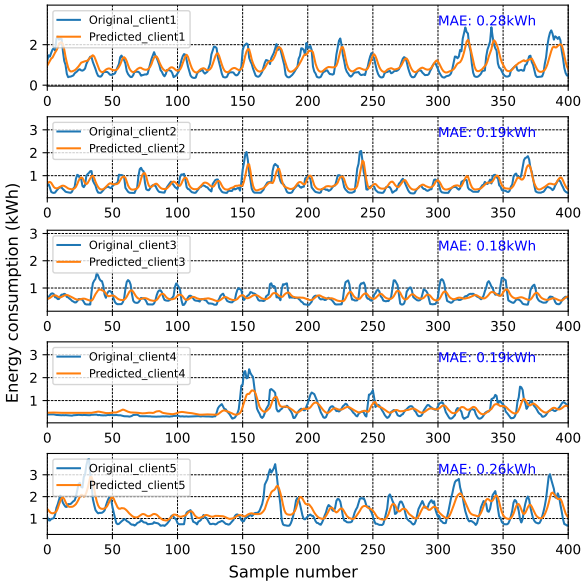


Figure 6.5: Comparison of actual and predicted curves in the CFL bus topology during a model flipping attack targeting Client 1.

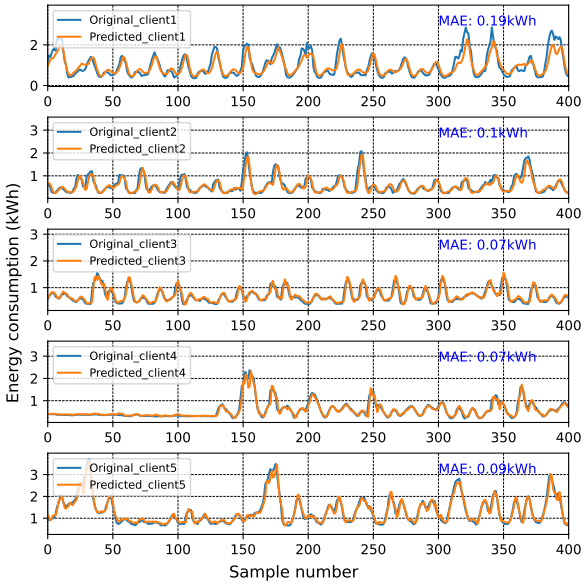


Figure 6.6: Comparison of actual and predicted curves in the DFL line topology during a model flipping attack targeting Client 1.

6.2.3 Effect of Adversarial Attack

In this work, a model flipping attack is considered. The details of the attack are presented in Section 3.1. In this attack, the weights of the local model are multiplied by -1 to create adversarial effects, effectively reversing the model's predictions. This manipulation can lead to significant misclassification, thereby compromising the model's accuracy and reliability. The implications of such adversarial attacks will be explored in depth, emphasizing their potential impact on the overall security and performance of FL systems.

The actual and predicted curves are illustrated in Fig. 6.5, providing a clear depiction of the repercussions of a model inversion attack on Client 1 within the centralized FL (CFL) framework. This approach, while capable of achieving high accuracy through centralized training, inadvertently exposes all participating clients to significant security vulnerabilities. As a result, a consistent Mean Absolute Error (MAE) of approximately 0.2 kWh is observed across all clients. Among them, Clients 1 and 5 are particularly vulnerable, experiencing the most pronounced effects of the attack, which compromises their predictive accuracy and reliability.

In stark contrast, DFL methods employing bus, line, and ring topologies demonstrate a remarkable resilience to model inversion attacks. This robustness is evidenced by significantly lower MAE values, illustrating the effectiveness of decentralized approaches in safeguarding client data. Figs. 6.6, 6.7, and 6.8 visually represent the prediction outcomes derived from these diverse topologies.

Clients 3 and 4 stand out with notably low MAEs, achieving values as low as 0.07 kWh in both line and ring topologies. This indicates a strong performance and suggests that these topologies are particularly effective in mitigating the impacts of attacks. However, not all clients

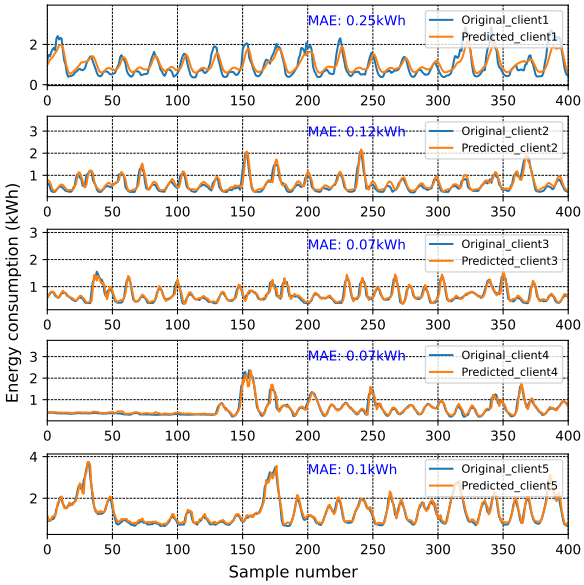


Figure 6.7: Comparison of actual and predicted curves in the DFL ring topology during a model flipping attack on Client 1.

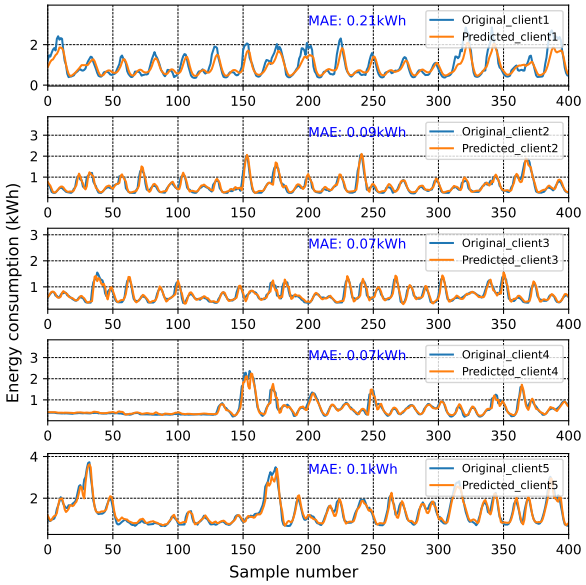


Figure 6.8: Comparison of actual and predicted curves in the DFL bus topology during a model flipping attack targeting Client 1.

are equally insulated; for instance, in the bus topology, Client 5 registers an MAE of 0.1 kWh, which, while still relatively low, is slightly elevated compared to the 0.09 kWh observed for other clients. This highlights that certain clients remain susceptible to minor disruptions. Similarly, in the ring topology, Client 1 displays a higher MAE of 0.25 kWh, underlining the variability in client performance under different topological configurations.

Overall, the DFL framework significantly limits the adverse impacts of model inversion attacks, confining them to specific clients rather than affecting all participants uniformly, as seen in the CFL approach. This distinction underscores the importance of topology selection in designing more secure FL systems, enhancing both individual client resilience and overall network integrity. By leveraging decentralized strategies, the DFL approach not only maintains predictive accuracy but also fortifies the system against potential security threats, ensuring a more robust and reliable learning environment.

6.3 Decentralised Random Layer Aggregation (DRLA)

In conventional FL, the central server aggregates all neural network layers to create a comprehensive global model during each communication round. This holistic approach, while effective in enhancing model accuracy, often results in increased communication overhead, which can strain network resources and lead to delays. Furthermore, the aggregation process may introduce potential vulnerabilities, as it becomes a target for adversarial attacks aimed at compromising the integrity of the model.

To address these significant challenges and to enhance both energy efficiency and resilience

Algorithm 4 Decentralized Federated Random Layer Aggregation (DRLA)

Device Initialization:
Initialize global model, $W_{0,l}^k$ for all layers, where l represents the layer number and k represents the client number.
Each client initializes its local model $W_{t,l}^k \leftarrow W_{0,l}^k$ for all layers l.
Set communication rounds, T, number of clients, K, and number of layers in local models, L.
Each client selects a random layer l_{rand}.
for $t = 1$ to T **do**
  Device Side (Local Training and Update):
  for each client $i = 1$ to K in parallel **do**
    Train the local model on local data.
    Create an update for the randomly selected layer: $W_{t,l_{\text{rand}}}^k \leftarrow \text{LocalUpdate}(W_{t,l_{\text{rand}}}^k)$.
  end for
  Device Side (Peer-to-Peer Aggregation):
  for each client $i = 1$ to K in parallel **do**
    Exchange $W_{t,l_{\text{rand}}}^k$ with a random subset of peers (P2P communication).
    Aggregate updates received from peers for the selected layer: $W_{t,l_{\text{rand}}}^k \leftarrow \frac{1}{|P|}\sum_{j\in P} W_{t,l_{\text{rand}}}^j$, where P is the set of peers.
  end for
  Device Side (Model Update):
  for each client $i = 1$ to K in parallel **do**
    Update the local model $W_{t+1,l}^k$ for all layers l:
    $W_{t+1,l}^k \leftarrow W_{t,l}^k$ for $l \neq l_{\text{rand}}$.
    $W_{t+1,l_{\text{rand}}}^k \leftarrow \frac{1}{|P|}\sum_{j\in P} W_{t,l_{\text{rand}}}^j$.
    Select new l_{rand} for the next communication round.
  end for
end for
return Trained local models $W_{T,l_{\text{rand}}}^k$ for the randomly selected layers l_{rand} on clients K.

to adversarial threats, the Decentralised Random Layer Aggregation (DRLA) framework has been adopted [107]. The detailed algorithm for DRLA is presented in Algorithm 4.

DRLA introduces an innovative strategy by randomly selecting a single layer for aggregation in each communication round while leaving all other layers unchanged. This selective aggregation not only significantly reduces the amount of data transmitted between clients and the server, thereby minimizing communication costs, but also enhances the robustness of the model against adversarial attacks. By making the aggregation process less predictable, DRLA complicates the task for attackers attempting to exploit vulnerabilities in the data exchange.

When applied to Decentralized FL (DFL), DRLA has demonstrated substantial improvements in both model performance and security. For instance, the average MAE results, as depicted in Fig. 6.9, were evaluated under a model-flipping attack originating from Client 1. In the DFL bus and ring topologies, the MAE was successfully reduced to 0.09 kWh, showcasing the strong defensive capabilities of DRLA during adversarial conditions.

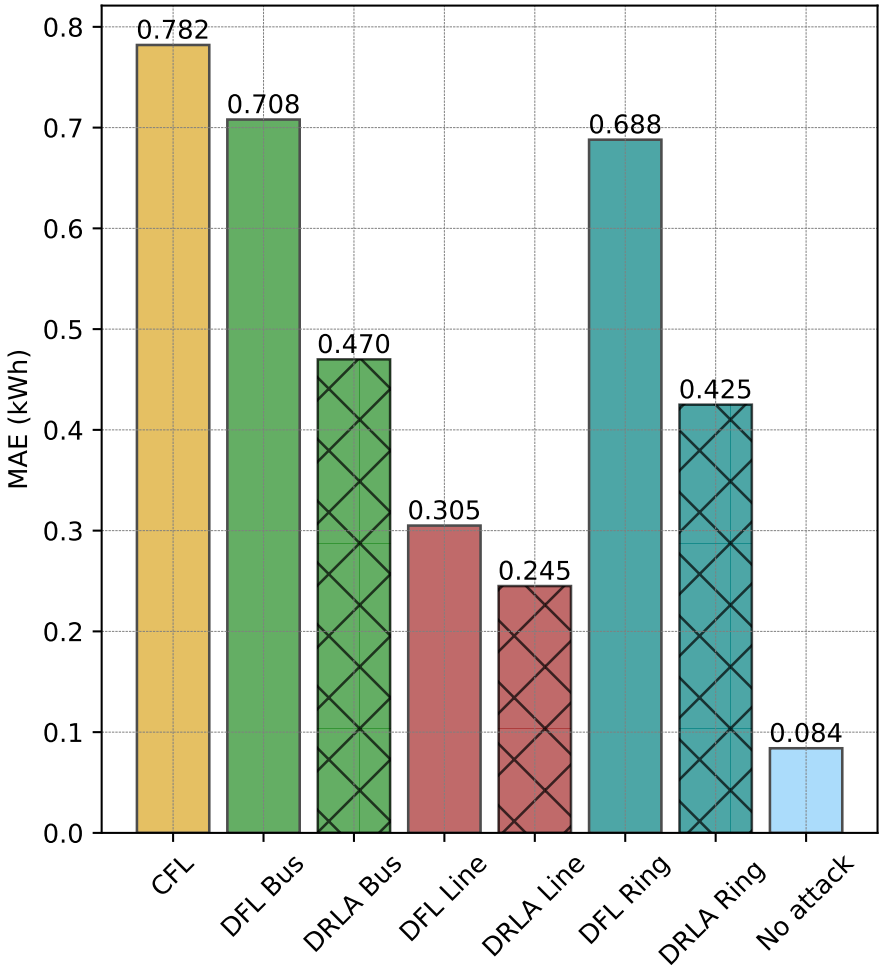


Figure 6.9: Analysis of average client MAE under model flipping attack targeting Client 1: A comparison across CFL, DFL, and DRLA frameworks.

Furthermore, in the ring topology, the MAE registered at 0.102 kWh during the attack, which is notably higher than the significantly lower MAE of 0.076 kWh observed when no attacks were present. This comparison underscores the effectiveness of DRLA across various communication topologies, illustrating its potential to maintain model integrity and accuracy even in the face of adversarial challenges. Overall, the implementation of DRLA in DFL not only optimizes communication efficiency but also fortifies the system against security threats, establishing it as a valuable approach within the evolving landscape of FL.

6.4 Concluding Remarks

In this chapter, i explored the application of Decentralized FL (DFL) for mitigating adversarial attacks and enhancing communication efficiency in FL systems, particularly in load forecasting. The key contributions and findings of this chapter can be summarized as follows:

- **Robust Aggregation (C2)**: The experiments conducted demonstrate that DFL topolo-

gies, including line, ring, and bus, significantly reduce the impact of adversarial attacks on model performance. This is in contrast to Centralized FL (CFL), where a single compromised client can degrade the global model's accuracy. The results show that DFL effectively confines the adverse effects of attacks to specific clients, thereby preserving the integrity and accuracy of the overall system. Additionally, the Decentralized Random Layer Aggregation (DRLA) framework further enhances the robustness of the system by aggregating only a single, randomly selected layer per round. This selective aggregation strategy not only reduces the amount of data transmitted but also minimizes the impact of potential adversarial updates. By focusing on a single layer, DRLA ensures that the global model remains robust even in the presence of malicious clients. This addresses the challenge of robust aggregation (C2) by effectively detecting and mitigating the impact of anomalous updates.

- **Communication Efficiency (C3)**: DFL leverages peer-to-peer communication, eliminating the need for a central server and thereby reducing communication overhead. This decentralized approach not only enhances privacy but also improves the system's resilience against single points of failure. The DRLA framework further optimizes communication by aggregating only a single, randomly selected layer per round. This selective aggregation strategy significantly reduces the amount of data transmitted, making DFL highly suitable for bandwidth-constrained environments. This addresses the challenge of communication and computational efficiency (C3) by minimizing the communication overhead and enhancing the scalability of the system.

This chapter has provided valuable insights into the potential of DFL for enhancing the security and efficiency of FL systems. The proposed frameworks and methods demonstrate significant improvements in mitigating adversarial attacks and optimizing communication, paving the way for more robust and scalable FL solutions. The findings presented here contribute to the ongoing research in secure and efficient FL, highlighting the importance of decentralized approaches in addressing the challenges of modern distributed machine learning environments.

Chapter 7

Conclusion and Future Work

This final chapter synthesizes the research presented in this thesis, highlighting the key contributions and their impact on addressing the critical challenges in FL, as discussed in Section 1.2. These challenges include managing data diversity, ensuring robustness against adversarial threats, and optimizing resource efficiency, all of which require innovative real-world application solutions. This thesis has advanced FL by proposing novel frameworks that address these gaps, making it more effective in energy forecasting. The chapter consolidates these technical contributions, situates them within the broader context of secure and efficient FL, and outlines future research directions that build on the foundational insights established throughout this work. The goal is to provide a comprehensive summary of the progress made and to offer a clear roadmap for future advancements in the field.

7.1 Summary of Contributions

This thesis has made significant contributions to the field of FL by addressing key challenges related to adversarial robustness, communication efficiency, and data heterogeneity. The following is a synthesis of the contributions made in each chapter and their relevance to the overall goals of the research. In Chapter 3, a thorough analysis of various adversarial attack strategies was conducted to evaluate the robustness of FL systems. These attacks, including the novel Federated Communication Round Attack (Fed-CRA), highlighted vulnerabilities in FL systems and emphasized the need for robust defence mechanisms. This work specifically targeted the challenge of ensuring robustness against adversarial attacks (C1) and laid the foundation for developing subsequent defence frameworks. Chapter 4 introduced three novel defence frameworks designed to enhance robustness (C2) and communication efficiency (C3):

- **Federated Random Layer Aggregation (FedRLA)**: This framework improves communication efficiency by aggregating only a single randomly selected neural network layer during each communication round, thereby reducing data exchange and enhancing pri-

vacy. FedRLA effectively mitigated adversarial attacks while maintaining model accuracy.

- **Layer-Based Anomaly Aware Federated Averaging (LBAA-FedAvg)**: By incorporating anomaly detection and selective layer exclusion, this framework ensures the integrity of the global model against adversarial updates, demonstrating strong robustness and stability under various attack scenarios.
- **Federated Incentivized Averaging (Fed-InA)**: Introducing a scoring mechanism to reward honest clients and penalize malicious ones, Fed-InA promotes honest participation and effectively identifies and mitigates stealth attacks, further enhancing the system's robustness and efficiency.

Chapter 5 proposed two frameworks to address data heterogeneity (C4) and further optimize communication and computational efficiency (C3):

- **FedBranched**: Utilizing probabilistic clustering based on Hidden Markov Models (HMM), this framework categorizes clients to handle diverse data distributions, significantly improving model performance and convergence in energy networks.
- **Adaptive Single Layer Aggregation (ASLA)**: By simplifying the aggregation process to a single neural network layer and incorporating quantization techniques, ASLA drastically reduces communication overhead while maintaining model accuracy, making it highly suitable for resource-constrained environments.

Chapter 6 presented the Decentralized Federated Learning (DFL) framework and the Decentralized Random Layer Aggregation (DRLA) mechanism. DFL leverages peer-to-peer communication topologies to eliminate the single point of failure associated with centralized systems. DRLA, a combination of FedRLA and DFL, extends the principles of FedRLA to decentralized settings by aggregating a single, randomly selected layer in each communication round within a decentralized environment. This approach further enhances robustness and communication efficiency.

Cross-Cutting Synthesis

The frameworks developed in this thesis offer distinct advantages depending on the scenario:

- **FedRLA** is ideal for centralized FL environments where communication efficiency and robustness against adversarial attacks are critical.
- **ASLA**, with its significant reduction in communication costs and adaptability to resource constraints, is best suited for large-scale deployments with limited bandwidth.

- **DRLA**, as an extension of FedRLA to decentralized settings, provides enhanced security and resilience in environments where a central server is either unavailable or undesirable.
- **LBAA-FedAvg** and **Fed-InA** provide additional robustness through anomaly detection and client incentivization, respectively, making them valuable for scenarios with high adversarial risk.
- **FedBranched** addresses data heterogeneity through clustering, making it suitable for applications with diverse data distributions.

Table 7.1: Comparative Strengths of Frameworks

Framework	**Key Strengths**	**Ideal For**	**Challenge Addressed**
FedRLA	Minimizes communication costs by aggregating a single neural network layer per round, offering high resilience against adversarial attacks while maintaining model accuracy.	Centralized FL systems with bandwidth limitations where efficient communication and robust security are paramount.	C2, C3
LBAA-FedAvg	Combines anomaly detection with selective layer aggregation to maintain model integrity, effectively countering partial adversarial updates and ensuring stable model performance.	FL environments facing partial adversarial attacks, requiring strong model integrity and robustness.	C2
Fed-InA	Utilizes a scoring system to distinguish between honest and malicious clients, reducing communication rounds and fostering a trustworthy FL environment.	Scenarios with potential stealth attacks and limited resources, needing efficient communication and honest client participation.	C2
FedBranched	Employs HMM-based clustering to categorize clients, addressing data heterogeneity and enhancing model convergence in applications with varied data distributions.	Energy networks and other applications with diverse and non-IID data distributions, requiring personalized model training.	C4
ASLA	Drastically cuts communication overhead using single-layer aggregation and quantization, making it highly efficient in resource usage while preserving model accuracy.	Large-scale FL deployments with stringent bandwidth constraints and resource-limited clients.	C3, C4
DRLA	Extends FedRLA principles to decentralized settings, enhancing security through P2P communication and maintaining robust aggregation of model updates.	Decentralized FL systems requiring high resilience against adversarial attacks and efficient communication without a central server.	C2, C3

Reflecting on the objectives outlined in Section 1.3 and the research gaps identified in Chapter 2, this thesis has made substantial progress in advancing the practical deployment of FL

systems. The proposed frameworks not only address the technical challenges of robustness, efficiency, and heterogeneity but also contribute to the broader goal of enabling more secure and efficient intelligent systems in real-world applications. By synthesizing insights across Chapters 3 to 6, this work provides a comprehensive toolkit for practitioners and researchers to select the most appropriate FL framework based on their specific requirements and operational contexts. The summary of purposed frameworks, according to their applications, is presented in table 7.1. The thesis has thus achieved its aim of developing robust, efficient, and heterogeneous data-aware FL frameworks for secure and privacy-conscious applications, particularly in energy forecasting. The contributions made provide a solid foundation for future research and practical implementations in the field of FL.

7.2 Limitations of the Research

- **Data Assumption Limitations**: The research assumes that the data distribution across clients, despite being non-IID, follows certain identifiable patterns that can be leveraged by the proposed frameworks like FedBranched and ASLA. However, in real-world scenarios, data distributions can be extremely complex and dynamic, potentially deviating from the assumed patterns and thus affecting the performance of these frameworks.

- **Attack Model Limitations**: The study focuses on specific types of adversarial attacks, such as model poisoning and stealth attacks like Fed-CRA. While these attacks are representative, there may be other sophisticated attack vectors not considered in this research that could potentially compromise the proposed defense frameworks.

- **Client Resource Heterogeneity**: Although the research acknowledges the heterogeneity of client resources, it simplifies this aspect to a certain extent. In practice, the diversity in computational power, memory, and network conditions among clients can be more pronounced, which might influence the effectiveness and applicability of the proposed frameworks.

- **Scalability Limitations**: The proposed frameworks have been tested and validated within a specific range of client numbers and data scales. When applying these frameworks to larger-scale FL systems with thousands or even millions of clients, new challenges may emerge, such as increased communication overhead and heightened computational complexity, which are not fully addressed in this research.

- **Privacy Protection Limitations**: While the research emphasizes privacy preservation through techniques like model quantization and differential privacy, achieving a high level of privacy protection often comes at the cost of reduced model accuracy and increased

computational overhead. The proposed methods may not fully satisfy the stringent privacy requirements of certain highly sensitive applications.

7.2.1 Tradeoffs Made in the Research

- **Model Accuracy vs. Communication Efficiency**: The proposed frameworks, such as FedRLA and ASLA, reduce communication costs by aggregating only a single layer of the neural network. However, this approach may lead to a certain degree of decline in model accuracy compared to traditional FL methods that aggregate all layers. The research attempts to strike a balance between communication efficiency and model accuracy but may not achieve optimal results in all scenarios.

- **Robustness vs. Computational Overhead**: To enhance the robustness of FL systems against adversarial attacks, the research introduces defense mechanisms like Fed-InA and DRLA. These mechanisms increase computational overhead by incorporating clustering, scoring, and other operations. In resource-constrained environments, this additional computational burden may affect the system's overall efficiency. The research makes a tradeoff between robustness and computational overhead, aiming to provide a reasonable level of protection without significantly impacting system performance.

- **Personalization vs. Global Model Performance**: FedBranched addresses data heterogeneity by clustering clients and training personalized models for each cluster. While this approach improves the personalization of models for individual clients, it may result in suboptimal global model performance compared to a single global model trained on homogeneous data. The research balances personalization and global model performance but may not achieve the best of both worlds in certain cases.

- **Privacy Protection vs. Model Utility**: In order to protect data privacy, techniques like model quantization and differential privacy are employed, which may limit the utility of the model to some extent. The research explores the tradeoff between privacy protection and model utility, striving to maximize model utility while ensuring a certain level of privacy but potentially falling short of satisfying both aspects fully.

- **Complexity vs. Practicality**: The proposed frameworks introduce additional complexity in terms of algorithms and system design. While these complex frameworks offer improved performance in specific aspects, they may reduce the practicality and ease of implementation of FL systems. The research seeks to balance complexity and practicality but may not fully achieve simplicity and ease of use in real-world deployments.

7.3 Future Work

The research presented in this thesis has significantly advanced FL by developing novel defense frameworks, addressing data heterogeneity, and optimizing communication efficiency. However, there are several promising directions for future work that can further enhance the robustness, efficiency, and applicability of FL systems.

7.3.1 Combining Presented Techniques

In future work, I plan to integrate robust aggregation, communication efficiency, computational efficiency, privacy awareness, and stopping criteria into a single, cohesive framework. This integrated approach aims to enhance the overall performance and security of FL systems by addressing multiple challenges simultaneously.

7.3.2 Evaluation on Other Domains

While the techniques presented in this thesis have been evaluated on energy networks, their applicability and effectiveness need to be tested across other domains to ensure their versatility and robustness. For example, in healthcare, the robust aggregation and privacy-preserving techniques can be evaluated in medical imaging to ensure they maintain data confidentiality and improve model performance. In autonomous vehicles, the communication efficiency and computational efficiency techniques can be tested to ensure they enable timely and accurate decision-making while minimizing resource usage. In smart cities, the stopping criteria and privacy awareness techniques can be assessed in applications such as traffic management and environmental monitoring to ensure they provide efficient and secure solutions. In supply chain and logistics, the robust aggregation and communication efficiency techniques can be evaluated in inventory management, demand forecasting, and delivery route optimization to ensure they provide accurate predictions and efficient communication.

7.3.3 Additional Areas for Future Work

Future work will also explore methods to improve interoperability between different FL frameworks and systems, ensuring seamless integration and operation. Enhancing the scalability of FL systems to handle large numbers of clients and diverse datasets efficiently will be another focus. Additionally, developing user-friendly interfaces and tools to facilitate the adoption of FL by non-expert users will promote wider deployment and use.

Bibliography

[1] T. R. Wanasinghe, R. G. Gosine, L. A. James, G. K. Mann, O. De Silva, and P. J. Warrian, "The internet of things in the oil and gas industry: a systematic review," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8654–8673, 2020.

[2] C. Lee and G. Ahmed, "Improving iot privacy, data protection and security concerns," *International Journal of Technology, Innovation and Management (IJTIM)*, vol. 1, no. 1, pp. 18–33, 2021.

[3] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A survey on federated learning for resource-constrained iot devices," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 1–24, 2021.

[4] S. S. Albouq, A. A. Abi Sen, N. Almashf, M. Yamin, A. Alshanqiti, and N. M. Bahbouh, "A survey of interoperability challenges and solutions for dealing with them in iot environment," *IEEE Access*, vol. 10, pp. 36 416–36 428, 2022.

[5] A. Taïk and S. Cherkaoui, "Electrical load forecasting using edge computing and federated learning," in *ICC 2020-2020 IEEE international conference on communications (ICC)*. IEEE, 2020, pp. 1–6.

[6] S. M. Kerner. Colonial pipeline hack explained: Everything you need to know. [Online]. Available: https://www.techtarget.com/whatis/feature/Colonial-Pipeline-hack-explained-Everything-you-need-to-know

[7] J. Greig. Suspected china-backed hackers target 7 indian electricity grid centers. [Online]. Available: https://therecord.media/suspected-china-backed-hackers-target-7-indian-electricity-grid-centers

[8] A. Meehan. Lithuanian energy firm disrupted by ddos attack. [Online]. Available: https://www.infosecurity-magazine.com/news/lithuanian-energy-ddos-attack/

[9] J. Greig. Ohio's largest oil producer says 'no impact' seen after cyberattack. [Online]. Available: https://therecord.media/encino-energy-cyberattack-alleged-data-leak-alphv

[10] M. Worley. Ibm security x-force threat intelligence index 2023. [Online]. Available: https://www.ibm.com/downloads/cas/DB4GL8YM

[11] H. U. Manzoor, A. R. Khan, M. Al-Quraan, L. Mohjazi, A. Taha, H. Abbas, S. Hussain, M. A. Imran, and A. Zoha, "Energy management in an agile workspace using ai-driven forecasting and anomaly detection," in *2022 4th Global Power, Energy and Communication Conference (GPECOM)*. IEEE, 2022, pp. 644–649.

[12] X. Luo, L. O. Oyedele, A. O. Ajayi, O. O. Akinade, J. M. D. Delgado, H. A. Owolabi, and A. Ahmed, "Genetic algorithm-determined deep feedforward neural network architecture for predicting electricity consumption in real buildings," *Energy and AI*, vol. 2, p. 100015, 2020.

[13] Ausgrid, "Ausgrid distribution zone substation data fy23, https://www.ausgrid.com.au/industry/our-research/data-to-share/distribution-zone-substation-data, accessed on 22/07/2024." [Online]. Available: "https://www.ausgrid.com.au/Industry/Our-Research/Data-to-share/Distribution-zone-substation-data"

[14] (2017) European environment agency. energy and climate change. [Online]. Available: https://www.eea.europa.eu/signals/signals-2017/articles/energy-and-climate-change

[15] W. Hurst, C. A. C. Montañez, and N. Shone, "Time-pattern profiling from smart meter data to detect outliers in energy consumption," *IoT*, vol. 1, no. 1, p. 6, 2020.

[16] E. Skomski, J.-Y. Lee, W. Kim, V. Chandan, S. Katipamula, and B. Hutchinson, "Sequence-to-sequence neural networks for short-term electrical load forecasting in commercial office buildings," *Energy and Buildings*, vol. 226, p. 110350, 2020.

[17] S. Chen, Y. Ren, D. Friedrich, Z. Yu, and J. Yu, "Prediction of office building electricity demand using artificial neural network by splitting the time horizon for different occupancy rates," *Energy and AI*, vol. 5, p. 100093, 2021.

[18] H. U. Manzoor, A. Shabbir, A. Chen, D. Flynn, and A. Zoha, "A survey of security strategies in federated learning: Defending models, data, and privacy," *Future Internet*, vol. 16, no. 10, p. 374, 2024.

[19] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, and W. Zhang, "A survey on federated learning: challenges and applications," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 2, pp. 513–535, 2023.

[20] 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 & Tutorials*, vol. 23, no. 3, pp. 1759–1799, 2021.

[21] G. Paragliola and A. Coronato, "Definition of a novel federated learning approach to reduce communication costs," *Expert Systems with Applications*, vol. 189, p. 116109, 2022.

[22] J. Liu, J. Huang, Y. Zhou, X. Li, S. Ji, H. Xiong, and D. Dou, "From distributed machine learning to federated learning: A survey," *Knowledge and Information Systems*, vol. 64, no. 4, pp. 885–917, 2022.

[23] S. Vahidian, M. Morafah, C. Chen, M. Shah, and B. Lin, "Rethinking data heterogeneity in federated learning: Introducing a new notion and standard benchmarks," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 3, pp. 1386–1397, 2023.

[24] H. U. Manzoor, A. R. Khan, D. Flynn, M. M. Alam, M. Akram, M. A. Imran, and A. Zoha, "Fedbranched: Leveraging federated learning for anomaly-aware load forecasting in energy networks," *Sensors*, vol. 23, no. 7, p. 3570, 2023.

[25] H. U. Manzoor, A. Jafri, and A. Zoha, "Lightweight single-layer aggregation framework for energy-efficient and privacy-preserving load forecasting in heterogeneous smart grids," 2024.

[26] P. Qi, D. Chiaro, A. Guzzo, M. Ianni, G. Fortino, and F. Piccialli, "Model aggregation techniques in federated learning: A comprehensive survey," *Future Generation Computer Systems*, 2023.

[27] Y. Liu, A. Huang, Y. Luo, H. Huang, Y. Liu, Y. Chen, L. Feng, T. Chen, H. Yu, and Q. Yang, "Fedvision: An online visual object detection platform powered by federated learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 08, 2020, pp. 13 172–13 179.

[28] A. R. Khan, H. U. Manzoor, R. N. B. Rais, S. Hussain, L. Mohjazi, M. A. Imran, and A. Zoha, "Semantic-aware federated blockage prediction (sfbp) in vision-aided next-generation wireless network," *Authorea Preprints*, 2024.

[29] Y. Liu, Y. Kang, T. Zou, Y. Pu, Y. He, X. Ye, Y. Ouyang, Y.-Q. Zhang, and Q. Yang, "Vertical federated learning: Concepts, advances, and challenges," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[30] S. Saha and T. Ahmad, "Federated transfer learning: concept and applications," *Intelligenza Artificiale*, vol. 15, no. 1, pp. 35–44, 2021.

[31] H. U. Manzoor, S. Hussain, D. Flynn, and A. Zoha, "Centralised vs. decentralised federated load forecasting: Who holds the key to adversarial attack robustness?" *Authorea Preprints*, 2024.

[32] L. Yuan, Z. Wang, L. Sun, S. Y. Philip, and C. G. Brinton, "Decentralized federated learning: A survey and perspective," *IEEE Internet of Things Journal*, 2024.

[33] C. Huang, J. Huang, and X. Liu, "Cross-silo federated learning: Challenges and opportunities," *arXiv preprint arXiv:2206.12949*, 2022.

[34] K. Müller and F. Bodendorf, "Cross-silo federated learning in enterprise networks with cooperative and competing actors," in *The Human Side of Service Engineering, AHFE 2023 International Conference, San Francisco, USA*, 2023.

[35] S. Dai and F. Meng, "Addressing modern and practical challenges in machine learning: A survey of online federated and transfer learning," *Applied Intelligence*, vol. 53, no. 9, pp. 11 045–11 072, 2023.

[36] A. Naeem, T. Anees, R. A. Naqvi, and W.-K. Loh, "A comprehensive analysis of recent deep and federated-learning-based methodologies for brain tumor diagnosis," *Journal of Personalized Medicine*, vol. 12, no. 2, p. 275, 2022.

[37] J. Lo, T. Y. Timothy, D. Ma, P. Zang, J. P. Owen, Q. Zhang, R. K. Wang, M. F. Beg, A. Y. Lee, Y. Jia *et al.*, "Federated learning for microvasculature segmentation and diabetic retinopathy classification of oct data," *Ophthalmology Science*, vol. 1, no. 4, p. 100069, 2021.

[38] S. Naz, K. T. Phan, and Y.-P. P. Chen, "A comprehensive review of federated learning for covid-19 detection," *International Journal of Intelligent Systems*, vol. 37, no. 3, pp. 2371–2392, 2022.

[39] M. J. Baucas, P. Spachos, and K. N. Plataniotis, "Federated learning and blockchain-enabled fog-iot platform for wearables in predictive healthcare," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 4, pp. 1732–1741, 2023.

[40] W. Yang, Y. Zhang, K. Ye, L. Li, and C.-Z. Xu, "Ffd: A federated learning based method for credit card fraud detection," in *Big data–bigData 2019: 8th international congress, held as part of the services conference federation, SCF 2019, san diego, CA, USA, June 25–30, 2019, proceedings 8*. Springer, 2019, pp. 18–32.

[41] A. Oualid, Y. Maleh, and L. Moumoun, "Federated learning techniques applied to credit risk management: A systematic literature review," *EDPACS*, vol. 68, no. 1, pp. 42–56, 2023.

[42] B. Guembe, A. Azeta, V. Osamor, and R. Ekpo, "A federated machine learning approaches for anti-money laundering detection," *Available at SSRN 4669561*, 2023.

[43] S. Guntuka, "Ai-driven algorithmic trading: Advanced techniques reshaping financial markets," *INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING AND TECHNOLOGY (IJCET)*, vol. 15, no. 5, pp. 564–571, 2024.

[44] T. Zeng, O. Semiari, M. Chen, W. Saad, and M. Bennis, "Federated learning on the road autonomous controller design for connected and autonomous vehicles," *IEEE Transactions on Wireless Communications*, vol. 21, no. 12, pp. 10 407–10 423, 2022.

[45] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, "Federated learning for industrial internet of things in future industries," *IEEE Wireless Communications*, vol. 28, no. 6, pp. 192–199, 2021.

[46] Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Z. Li, L. Lyu, and Y. Liu, "Privacy-preserving blockchain-based federated learning for iot devices," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1817–1829, 2020.

[47] S. Banabilah, M. Aloqaily, E. Alsayed, N. Malik, and Y. Jararweh, "Federated learning review: Fundamentals, enabling technologies, and future applications," *Information processing & management*, vol. 59, no. 6, p. 103061, 2022.

[48] F. Alfayez and S. B. Khan, "User-centric secured smart virtual assistants framework for disables," *Alexandria Engineering Journal*, vol. 95, pp. 59–71, 2024.

[49] D. Javeed, M. S. Saeed, P. Kumar, A. Jolfaei, S. Islam, and A. N. Islam, "Federated learning-based personalized recommendation systems: An overview on security and privacy challenges," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 2618–2627, 2023.

[50] S. P. Ramu, P. Boopalan, Q.-V. Pham, P. K. R. Maddikunta, T. Huynh-The, M. Alazab, T. T. Nguyen, and T. R. Gadekallu, "Federated learning enabled digital twins for smart cities: Concepts, recent advances, and future directions," *Sustainable Cities and Society*, vol. 79, p. 103663, 2022.

[51] S. Zhang, J. Li, L. Shi, M. Ding, D. C. Nguyen, W. Tan, J. Weng, and Z. Han, "Federated learning in intelligent transportation systems: Recent applications and open problems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 5, pp. 3259–3285, 2023.

[52] Y. Pang, Z. Ni, and X. Zhong, "Federated learning for crowd counting in smart surveillance systems," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 5200–5209, 2023.

[53] P. Arora, V. Khullar, I. Kansal, R. Kumar, and R. Popli, "Privacy-preserving federated learning system (f-ppls) for military focused area classification," *Multimedia Tools and Applications*, pp. 1–27, 2024.

[54] T. Ding, L. Liu, Z. Yan, X. Lu, and J. Hu, "Federated reinforcement learning for intelligent route planning in aerial-terrestrial network," *IEEE Internet of Things Journal*, 2024.

[55] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, M. D. Mueck, and S. Srikanteswara, "Energy demand prediction with federated learning for electric vehicle networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[56] M. N. Fekri, K. Grolinger, and S. Mir, "Distributed load forecasting using smart meter data: Federated learning with recurrent neural networks," *International Journal of Electrical Power & Energy Systems*, vol. 137, p. 107669, 2022.

[57] J. Jithish, B. Alangot, N. Mahalingam, and K. S. Yeo, "Distributed anomaly detection in smart grids: a federated learning-based approach," *IEEE Access*, vol. 11, pp. 7157–7179, 2023.

[58] C. Si, H. Wang, L. Chen, J. Zhao, Y. Min, and F. Xu, "Robust co-modeling for privacy-preserving short-term load forecasting with incongruent load data distributions," *IEEE Transactions on Smart Grid*, 2024.

[59] S. Zhang, Y. Li, S. Zhang, F. Shahabi, S. Xia, Y. Deng, and N. Alshurafa, "Deep learning in human activity recognition with wearable sensors: A review on advances," *Sensors*, vol. 22, no. 4, p. 1476, 2022.

[60] J. Wang, X. Chen, F. Zhang, F. Chen, and Y. Xin, "Building load forecasting using deep neural network with efficient feature fusion," *Journal of Modern Power Systems and Clean Energy*, vol. 9, no. 1, pp. 160–169, 2021.

[61] L. Wen, K. Zhou, and S. Yang, "Load demand forecasting of residential buildings using a deep learning model," *Electric Power Systems Research*, vol. 179, p. 106073, 2020.

[62] J. W. Taylor, "Short-term electricity demand forecasting using double seasonal exponential smoothing," *Journal of the Operational Research Society*, vol. 54, no. 8, pp. 799–805, 2003.

[63] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, "Short-term load forecasting with deep residual networks," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3943–3952, 2018.

[64] C. Vigurs, C. Maidment, M. Fell, and D. Shipworth, "Customer privacy concerns as a barrier to sharing data about energy use in smart local energy systems: A rapid realist review," *Energies*, vol. 14, no. 5, p. 1285, 2021.

[65] G. Singh and J. Bedi, "A federated and transfer learning based approach for households load forecasting," *Knowledge-Based Systems*, vol. 299, p. 111967, 2024.

[66] M. Savi and F. Olivadese, "Short-term energy consumption forecasting at the edge: A federated learning approach," *IEEE Access*, vol. 9, pp. 95 949–95 969, 2021.

[67] Y. L. Tun, K. Thar, C. M. Thwal, and C. S. Hong, "Federated learning based energy demand prediction with clustered aggregation," in *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 2021, pp. 164–167.

[68] T. Zhang, L. Gao, C. He, M. Zhang, B. Krishnamachari, and A. S. Avestimehr, "Federated learning for the internet of things: Applications, challenges, and opportunities," *IEEE Internet of Things Magazine*, vol. 5, no. 1, pp. 24–29, 2022.

[69] S. Yang, H. Park, J. Byun, and C. Kim, "Robust federated learning with noisy labels," *IEEE Intelligent Systems*, vol. 37, no. 2, pp. 35–43, 2022.

[70] G. Gad and Z. Fadlullah, "Federated learning via augmented knowledge distillation for heterogenous deep human activity recognition systems," *Sensors*, vol. 23, no. 1, p. 6, 2022.

[71] Z. Chai, A. Ali, S. Zawad, S. Truex, A. Anwar, N. Baracaldo, Y. Zhou, H. Ludwig, F. Yan, and Y. Cheng, "Tifl: A tier-based federated learning system," in *Proceedings of the 29th international symposium on high-performance parallel and distributed computing*, 2020, pp. 125–136.

[72] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 2022, pp. 965–978.

[73] S. Wu, J. Chen, X. Nie, Y. Wang, X. Zhou, L. Lu, W. Peng, Y. Nie, and W. Menhaj, "Global prototype distillation for heterogeneous federated learning," *Scientific Reports*, vol. 14, no. 1, p. 12057, 2024.

[74] D. Lin, Y. Guo, H. Sun, and Y. Chen, "Fedcluster: A federated learning framework for cross-device private ecg classification," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2022, pp. 1–6.

[75] A. Shabbir, H. U. Manzoor, K. Arshad, K. Assaleh, Z. Halim, and A. Zoha, "Sustainable and lightweight defense framework for resource constraint federated learning assisted smart grids against adversarial attacks," *Authorea Preprints*, 2024.

[76] S. Malaviya, M. Shukla, and S. Lodha, "Reducing communication overhead in federated learning for pre-trained language models using parameter-efficient finetuning," in *Conference on Lifelong Learning Agents*. PMLR, 2023, pp. 456–469.

[77] D. Roschewitz, M.-A. Hartley, L. Corinzia, and M. Jaggi, "Ifedavg: Interpretable data-interoperability for federated learning," *arXiv preprint arXiv:2107.06580*, 2021.

[78] G. S. Nariman and H. K. Hamarashid, "Communication overhead reduction in federated learning: a review," *International Journal of Data Science and Analytics*, pp. 1–32, 2024.

[79] H. U. Manzoor, A. Jafri, and A. Zoha, "Adaptive single-layer aggregation framework for energy-efficient and privacy-preserving load forecasting in heterogeneous federated smart grids," *Internet of Things*, p. 101376, 2024.

[80] X. Li, Z. Qu, B. Tang, and Z. Lu, "Fedlga: Toward system-heterogeneity of federated learning via local gradient approximation," *IEEE Transactions on Cybernetics*, vol. 54, no. 1, pp. 401–414, 2023.

[81] H. Wang and J. Xu, "Friends to help: Saving federated learning from client dropout," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 8896–8900.

[82] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li, "A survey of federated learning for edge computing: Research problems and solutions," *High-Confidence Computing*, vol. 1, no. 1, p. 100008, 2021.

[83] J. S.-P. Díaz and Á. L. García, "Study of the performance and scalability of federated learning for medical imaging with intermittent clients," *Neurocomputing*, vol. 518, pp. 142–154, 2023.

[84] Z. Yang, S. Zhang, C. Li, M. Wang, H. Wang, and M. Zhang, "Efficient knowledge management for heterogeneous federated continual learning on resource-constrained edge devices," *Future Generation Computer Systems*, vol. 156, pp. 16–29, 2024.

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

[86] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *International conference on machine learning*. PMLR, 2021, pp. 2089–2099.

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

[88] S. Mohammadi, A. Balador, S. Sinaei, and F. Flammini, "Balancing privacy and performance in federated learning: A systematic literature review on methods and metrics," *Journal of Parallel and Distributed Computing*, p. 104918, 2024.

[89] B. Yan, H. Zhang, M. Xu, D. Yu, and X. Cheng, "Fedrfq: Prototype-based federated learning with reduced redundancy, minimal failure, and enhanced quality," *IEEE Transactions on Computers*, 2024.

[90] W. Xiong and R. Lagerström, "Threat modeling–a systematic literature review," *Computers & security*, vol. 84, pp. 53–69, 2019.

[91] B. Li, L. Fan, H. Gu, J. Li, and Q. Yang, "Fedipr: Ownership verification for federated deep neural network models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4521–4536, 2022.

[92] W.-T. Lin, G. Chen, and X. Zhou, "Privacy-preserving federated learning for detecting false data injection attacks on power system," *Electric Power Systems Research*, vol. 229, p. 110150, 2024.

[93] H. U. Manzoor, M. S. Khan, A. R. Khan, F. Ayaz, D. Flynn, M. A. Imran, and A. Zoha, "Fedclamp: An algorithm for identification of anomalous client in federated learning," in *2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, 2022, pp. 1–4.

[94] H. U. Manzoor, A. R. Khan, T. Sher, W. Ahmad, and A. Zoha, "Defending federated learning from backdoor attacks: Anomaly-aware fedavg with layer-based aggregation," in *2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2023, pp. 1–6.

[95] L. Lyu, H. Yu, X. Ma, C. Chen, L. Sun, J. Zhao, Q. Yang, and S. Y. Philip, "Privacy and robustness in federated learning: Attacks and defenses," *IEEE transactions on neural networks and learning systems*, 2022.

[96] A. Shabbir, H. U. Manzoor, R. A. Ahmed, and Z. Halim, "Resilience of federated learning against false data injection attacks in energy forecasting," in *2024 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*. IEEE, 2024, pp. 245–249.

[97] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *NDSS*, 2021.

[98] T. Kim, S. Singh, N. Madaan, and C. Joe-Wong, "pfeddef: Defending grey-box attacks for personalized federated learning," *arXiv preprint arXiv:2209.08412*, 2022.

[99] K. N. Kumar, C. Vishnu, R. Mitra, and C. K. Mohan, "Black-box adversarial attacks in autonomous vehicle technology," in *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. IEEE, 2020, pp. 1–7.

[100] O. Zari, C. Xu, and G. Neglia, "Efficient passive membership inference attack in federated learning," *arXiv preprint arXiv:2111.00430*, 2021.

[101] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[102] P. Liu, X. Xu, and W. Wang, "Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives," *Cybersecurity*, vol. 5, no. 1, p. 4, 2022.

[103] 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, no. 3, pp. 2496–2501, 2021.

[104] D. Li, W. E. Wong, W. Wang, Y. Yao, and M. Chau, "Detection and mitigation of label-flipping attacks in federated learning systems with kpca and k-means," in *2021 8th International Conference on Dependable Systems and Their Applications (DSA)*. IEEE, 2021, pp. 551–559.

[105] S. Andreina, G. A. Marson, H. Möllering, and G. Karame, "Baffle: Backdoor detection via feedback-based federated learning," in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2021, pp. 852–863.

[106] X. Lin, Z. Liu, D. Fu, R. Qiu, and H. Tong, "Backtime: Backdoor attacks on multivariate time series forecasting," *Advances in Neural Information Processing Systems*, vol. 37, pp. 131 344–131 368, 2024.

[107] H. U. Manzoor, K. Arshad, K. Assaleh, and A. Zoha, "Enhanced adversarial attack resilience in energy networks through energy and privacy aware federated learning," *Authorea Preprints*, 2024.

[108] X. Zhou, M. Xu, Y. Wu, and N. Zheng, "Deep model poisoning attack on federated learning," *Future Internet*, vol. 13, no. 3, p. 73, 2021.

[109] N. Wang, Y. Xiao, Y. Chen, Y. Hu, W. Lou, and Y. T. Hou, "Flare: defending federated learning against model poisoning attacks via latent space representations," in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, 2022, pp. 946–958.

[110] T. Zou, Y. Liu, Y. Kang, W. Liu, Y. He, Z. Yi, Q. Yang, and Y.-Q. Zhang, "Defending batch-level label inference and replacement attacks in vertical federated learning," *IEEE Transactions on Big Data*, 2022.

[111] Z. Chen, P. Tian, W. Liao, and W. Yu, "Zero knowledge clustering based adversarial mitigation in heterogeneous federated learning," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1070–1083, 2020.

[112] Y. Zhao, J. Chen, J. Zhang, D. Wu, J. Teng, and S. Yu, "Pdgan: A novel poisoning defense method in federated learning using generative adversarial network," in *Algorithms and Architectures for Parallel Processing: 19th International Conference, ICA3PP 2019, Melbourne, VIC, Australia, December 9–11, 2019, Proceedings, Part I 19*. Springer, 2020, pp. 595–609.

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

[114] Z. Zhang, A. Panda, L. Song, Y. Yang, M. Mahoney, P. Mittal, R. Kannan, and J. Gonzalez, "Neurotoxin: Durable backdoors in federated learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 26 429–26 446.

[115] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" *arXiv preprint arXiv:1911.07963*, 2019.

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

[117] K. Yoo and N. Kwak, "Backdoor attacks in federated learning by rare embeddings and gradient ensembling," *arXiv preprint arXiv:2204.14017*, 2022.

[118] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[119] X. Yuan, X. Ma, L. Zhang, Y. Fang, and D. Wu, "Beyond class-level privacy leakage: Breaking record-level privacy in federated learning," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2555–2565, 2021.

[120] H. Yang, M. Ge, D. Xue, K. Xiang, H. Li, and R. Lu, "Gradient leakage attacks in federated learning: Research frontiers, taxonomy and future directions," *IEEE Network*, 2023.

[121] Y. Dong, Y. Wang, M. Gama, M. A. Mustafa, G. Deconinck, and X. Huang, "Privacy-preserving distributed learning for residential short-term load forecasting," *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 16 817–16 828, 2024.

[122] E. Hallaji, R. Razavi-Far, M. Saif, B. Wang, and Q. Yang, "Decentralized federated learning: A survey on security and privacy," *IEEE Transactions on Big Data*, 2024.

[123] A. Wainakh, E. Zimmer, S. Subedi, J. Keim, T. Grube, S. Karuppayah, A. Sanchez Guinea, and M. Mühlhäuser, "Federated learning attacks revisited: A critical discussion of gaps, assumptions, and evaluation setups," *Sensors*, vol. 23, no. 1, p. 31, 2022.

[124] N. M. Jebreel and J. Domingo-Ferrer, "Fl-defender: Combating targeted attacks in federated learning," *Knowledge-Based Systems*, vol. 260, p. 110178, 2023.

[125] G. Bao and P. Guo, "Federated learning in cloud-edge collaborative architecture: key technologies, applications and challenges," *Journal of Cloud Computing*, vol. 11, no. 1, p. 94, 2022.

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

[127] S. Shen, S. Tople, and P. Saxena, "Auror: Defending against poisoning attacks in collaborative deep learning systems," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, 2016, pp. 508–519.

[128] T. D. Nguyen, P. Rieger, R. De Viti, H. Chen, B. B. Brandenburg, H. Yalame, H. Möllering, H. Fereidooni, S. Marchal, M. Miettinen *et al.*, "{FLAME}: Taming backdoors in federated learning," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1415–1432.

[129] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.

[130] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[131] M. S. Ozdayi, M. Kantarcioglu, and Y. R. Gel, "Defending against backdoors in federated learning with robust learning rate," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, 2021, pp. 9268–9276.

[132] C. Wu, S. Zhu, and P. Mitra, "Federated unlearning with knowledge distillation," *arXiv preprint arXiv:2201.09441*, 2022.

[133] C. Wu, X. Yang, S. Zhu, and P. Mitra, "Mitigating backdoor attacks in federated learning," *arXiv preprint arXiv:2011.01767*, 2020.

[134] X. Li, Z. Qu, S. Zhao, B. Tang, Z. Lu, and Y. Liu, "Lomar: A local defense against poisoning attack on federated learning," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 437–450, 2021.

[135] C. Zhao, Y. Wen, S. Li, F. Liu, and D. Meng, "Federatedreverse: A detection and defense method against backdoor attacks in federated learning," in *Proceedings of the 2021 ACM workshop on information hiding and multimedia security*, 2021, pp. 51–62.

[136] 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, 2022.

[137] Z. Zhang, Y. Zhang, D. Guo, L. Yao, and Z. Li, "Secfednids: Robust defense for poisoning attack against federated learning-based network intrusion detection system," *Future Generation Computer Systems*, vol. 134, pp. 154–169, 2022.

[138] S. Lu, R. Li, W. Liu, and X. Chen, "Defense against backdoor attack in federated learning," *Computers & Security*, vol. 121, p. 102819, 2022.

[139] W. Wan, J. Lu, S. Hu, L. Y. Zhang, and X. Pei, "Shielding federated learning: A new attack approach and its defense," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2021, pp. 1–7.

[140] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," *Advances in neural information processing systems*, vol. 31, 2018.

[141] D. Shah, P. Dube, S. Chakraborty, and A. Verma, "Adversarial training in communication constrained federated learning," *arXiv preprint arXiv:2103.01319*, 2021.

[142] E. Hallaji, R. Razavi-Far, M. Saif, and E. Herrera-Viedma, "Label noise analysis meets adversarial training: A defense against label poisoning in federated learning," *Knowledge-Based Systems*, vol. 266, p. 110384, 2023.

[143] M. H. Meng, S. G. Teo, G. Bai, K. Wang, and J. S. Dong, "Enhancing federated learning robustness using data-agnostic model pruning," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2023, pp. 441–453.

[144] X. Jiang and C. Borcea, "Complement sparsification: Low-overhead model pruning for federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 7, 2023, pp. 8087–8095.

[145] D. Yin, Y. Chen, K. Ramchandran, and P. L. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:3708326

[146] E. M. El Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in Byzantium," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 3521–3530. [Online]. Available: https://proceedings.mlr.press/v80/mhamdi18a.html

[147] 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.

[148] L. Muñoz-González, K. T. Co, and E. C. Lupu, "Byzantine-robust federated machine learning through adaptive model averaging," *arXiv preprint arXiv:1909.05125*, 2019.

[149] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "Fltrust: Byzantine-robust federated learning via trust bootstrapping," *arXiv preprint arXiv:2012.13995*, 2020.

[150] X. Cao, J. Jia, and N. Z. Gong, "Provably secure federated learning against malicious clients," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 8, 2021, pp. 6885–6893.

[151] R. Balakrishnan, T. Li, T. Zhou, N. Himayat, V. Smith, and J. Bilmes, "Diverse client selection for federated learning via submodular maximization," in *International Conference on Learning Representations*, 2022.

[152] A. Kumar, V. Khimani, D. Chatzopoulos, and P. Hui, "Fedclean: A defense mechanism against parameter poisoning attacks in federated learning," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4333–4337.

[153] H. Guo, H. Wang, T. Song, Y. Hua, Z. Lv, X. Jin, Z. Xue, R. Ma, and H. Guan, "Siren: Byzantine-robust federated learning via proactive alarming," in *Proceedings of the ACM Symposium on Cloud Computing*, 2021, pp. 47–60.

[154] P. Rieger, T. D. Nguyen, M. Miettinen, and A.-R. Sadeghi, "Deepsight: Mitigating backdoor attacks in federated learning through deep model inspection," *arXiv preprint arXiv:2201.00763*, 2022.

[155] 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.

[156] Q. Mao, L. Wang, Y. Long, L. Han, Z. Wang, and K. Chen, "A blockchain-based framework for federated learning with privacy preservation in power load forecasting," *Knowledge-Based Systems*, vol. 284, p. 111338, 2024.

[157] Z. Batool, K. Zhang, Z. Zhu, S. Aravamuthan, and U. Aivodji, "Block-fest: A blockchain-based federated anomaly detection framework with computation offloading using transformers," in *2022 IEEE 1st Global Emerging Technology Blockchain Forum: Blockchain & Beyond (iGETblockchain)*. IEEE, 2022, pp. 1–6.

[158] W. Zhang, Q. Lu, Q. Yu, Z. Li, Y. Liu, S. K. Lo, S. Chen, X. Xu, and L. Zhu, "Blockchain-based federated learning for device failure detection in industrial iot," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5926–5937, 2020.

[159] H. Guo, Y. Mao, X. He, B. Zhang, T. Pang, and P. Ping, "Improving federated learning through abnormal client detection and incentive." *CMES-Computer Modeling in Engineering & Sciences*, vol. 139, no. 1, 2024.

[160] Y. Bai, G. Xing, H. Wu, Z. Rao, C. Peng, Y. Rao, W. Yang, C. Ma, J. Li, and Y. Zhou, "Isppfl: An incentive scheme based privacy-preserving federated learning for avatar in metaverse," *Computer Networks*, vol. 251, p. 110654, 2024.

[161] X. Jiang, S. Sun, Y. Wang, and M. Liu, "Towards federated learning against noisy labels via local self-regularization," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 862–873.

[162] Z. Chen, Z. Wu, X. Wu, L. Zhang, J. Zhao, Y. Yan, and Y. Zheng, "Contractible regularization for federated learning on non-iid data," in *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2022, pp. 61–70.

[163] Y. Liu, T. Zou, Y. Kang, W. Liu, Y. He, Z. Yi, and Q. Yang, "Batch label inference and replacement attacks in black-boxed vertical federated learning," *arXiv preprint arXiv:2112.05409*, 2021.

[164] X. Liu, H. Li, G. Xu, Z. Chen, X. Huang, and R. Lu, "Privacy-enhanced federated learning against poisoning adversaries," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4574–4588, 2021.

[165] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-efficient federated learning via knowledge distillation," *Nature communications*, vol. 13, no. 1, p. 2032, 2022.

[166] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *International conference on machine learning*. PMLR, 2021, pp. 12 878–12 889.

[167] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan, "Fine-tuning global model via data-free knowledge distillation for non-iid federated learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 174–10 183.

[168] C. Zhang, S. Ekanut, L. Zhen, and Z. Li, "Augmented multi-party computation against gradient leakage in federated learning," *IEEE Transactions on Big Data*, 2022.

[169] V. Mugunthan, A. Polychroniadou, D. Byrd, and T. H. Balch, "Smpai: Secure multi-party computation for federated learning," in *Proceedings of the NeurIPS 2019 Workshop on Robust AI in Financial Services*, vol. 21. MIT Press Cambridge, MA, USA, 2019.

[170] D. Byrd and A. Polychroniadou, "Differentially private secure multi-party computation for federated learning in financial applications," in *Proceedings of the First ACM International Conference on AI in Finance*, 2020, pp. 1–9.

[171] S. Otoum, N. Guizani, and H. Mouftah, "On the feasibility of split learning, transfer learning and federated learning for preserving security in its systems," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[172] V. Turina, Z. Zhang, F. Esposito, and I. Matta, "Federated or split? a performance and privacy analysis of hybrid split and federated learning architectures," in *2021 IEEE 14th International Conference on Cloud Computing (CLOUD)*. IEEE, 2021, pp. 250–260.

[173] C. Thapa, P. C. M. Arachchige, S. Camtepe, and L. Sun, "Splitfed: When federated learning meets split learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8485–8493.

[174] J. Liao, Z. Chen, and E. G. Larsson, "Over-the-air federated learning with privacy protection via correlated additive perturbations," in *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2022, pp. 1–8.

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

[176] J. Sun, A. Li, B. Wang, H. Yang, H. Li, and Y. Chen, "Soteria: Provable defense against privacy leakage in federated learning from representation perspective," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 9311–9319.

[177] H. Lee, J. Kim, R. Hussain, S. Cho, and J. Son, "On defensive neural networks against inference attack in federated learning," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

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

[179] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 656–672.

[180] Y. Chen, F. Luo, T. Li, T. Xiang, Z. Liu, and J. Li, "A training-integrity privacy-preserving federated learning scheme with trusted execution environment," *Information Sciences*, vol. 522, pp. 69–79, 2020.

[181] F. Mo, H. Haddadi, K. Katevas, E. Marin, D. Perino, and N. Kourtellis, "Ppfl: Privacy-preserving federated learning with trusted execution environments," in *Proceedings of the 19th annual international conference on mobile systems, applications, and services*, 2021, pp. 94–108.

[182] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[183] D. Molchanov, A. Ashukha, and D. Vetrov, "Variational dropout sparsifies deep neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2498–2507.

[184] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz, "Importance estimation for neural network pruning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 264–11 272.

[185] Y. He, G. Kang, X. Dong, Y. Fu, and Y. Yang, "Soft filter pruning for accelerating deep convolutional neural networks," *arXiv preprint arXiv:1808.06866*, 2018.

[186] J.-H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5058–5066.

[187] H. Huang, L. Zhang, C. Sun, R. Fang, X. Yuan, and D. Wu, "Distributed pruning towards tiny neural networks in federated learning," in *2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2023, pp. 190–201.

[188] D. Stripelis, U. Gupta, G. V. Steeg, and J. L. Ambite, "Federated progressive sparsification (purge, merge, tune)+," *arXiv preprint arXiv:2204.12430*, 2022.

[189] L. Lei, Y. Yuan, Y. Yang, Y. Luo, L. Pu, and S. Chatzinotas, "Sparsification and optimization for energy-efficient federated learning in wireless edge networks," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 3071–3076.

[190] B. Wang, J. Fang, H. Li, and B. Zeng, "Communication-efficient federated learning: A variance-reduced stochastic approach with adaptive sparsification," *IEEE Transactions on Signal Processing*, 2023.

[191] H. Sun, X. Ma, and R. Q. Hu, "Adaptive federated learning with gradient compression in uplink noma," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16 325–16 329, 2020.

[192] A. M Abdelmoniem, A. Elzanaty, M.-S. Alouini, and M. Canini, "An efficient statistical-based gradient compression technique for distributed training systems," *Proceedings of Machine Learning and Systems*, vol. 3, pp. 297–322, 2021.

[193] B. Guo, Y. Liu, and C. Zhang, "A partition based gradient compression algorithm for distributed training in aiot," *Sensors*, vol. 21, no. 6, p. 1943, 2021.

[194] H. Jin, D. Wu, S. Zhang, X. Zou, S. Jin, D. Tao, Q. Liao, and W. Xia, "Design of a quantization-based dnn delta compression framework for model snapshots and federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 3, pp. 923–937, 2023.

[195] C. Chung-Kuan, "Computer arithmetic algorithms and hardware design," *Lecture notes*, 2006.

[196] D. Zoni and A. Galimberti, "Cost-effective fixed-point hardware support for risc-v embedded systems," *Journal of Systems Architecture*, vol. 126, p. 102476, 2022.

[197] A. Kuzmin, M. Nagel, M. Van Baalen, A. Behboodi, and T. Blankevoort, "Pruning vs quantization: Which is better?" *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[198] M. T. Lê, P. Wolinski, and J. Arbel, "Efficient neural networks for tiny machine learning: A comprehensive review," *arXiv preprint arXiv:2311.11883*, 2023.

[199] S. Branco, A. G. Ferreira, and J. Cabral, "Machine learning in resource-scarce embedded systems, fpgas, and end-devices: A survey," *Electronics*, vol. 8, no. 11, p. 1289, 2019.

[200] J. Cao, R. Wei, Q. Cao, Y. Zheng, Z. Zhu, C. Ji, and X. Zhou, "Fedstar: Efficient federated learning on heterogeneous communication networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023.

[201] J. Wu, F. Dong, H. Leung, Z. Zhu, J. Zhou, and S. Drew, "Topology-aware federated learning in edge computing: A comprehensive survey," *ACM Computing Surveys*, vol. 56, no. 10, pp. 1–41, 2024.

[202] S. Huang, Z. Zhang, S. Wang, R. Wang, and K. Huang, "Accelerating federated edge learning via topology optimization," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2056–2070, 2022.

[203] Z. Niu, H. Dong, A. K. Qin, and T. Gu, "Flrce: Resource-efficient federated learning with early-stopping strategy," 2024.

[204] F. Dehrouyeh, L. Yang, F. B. Ajaei, and A. Shami, "On tinyml and cybersecurity: Electric vehicle charging infrastructure use case," *arXiv preprint arXiv:2404.16894*, 2024.

[205] F. Pereira, R. Correia, P. Pinho, S. I. Lopes, and N. B. Carvalho, "Challenges in resource-constrained iot devices: Energy and communication as critical success factors for future iot deployment," *Sensors*, vol. 20, no. 22, p. 6420, 2020.

[206] O. Sentieys and D. Menard, "Customizing number representation and precision," in *Approximate Computing Techniques: From Component-to Application-Level*. Springer, 2022, pp. 11–41.

[207] P. Labonne, "Asymmetric uncertainty: Nowcasting using skewness in real-time data," *International Journal of Forecasting*, 2024.

[208] Z. Luo, Y. Wang, Z. Wang, Z. Sun, and T. Tan, "Disentangled federated learning for tackling attributes skew via invariant aggregation and diversity transferring," *arXiv preprint arXiv:2206.06818*, 2022.

[209] T. Zhou, J. Zhang, and D. H. Tsang, "Fedfa: Federated learning with feature anchors to align features and classifiers for heterogeneous data," *IEEE Transactions on Mobile Computing*, 2023.

[210] J. Xu, Z. Chen, T. Q. Quek, and K. F. E. Chong, "Fedcorr: Multi-stage federated learning for label noise correction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 184–10 193.

[211] X. Ouyang, Z. Xie, J. Zhou, G. Xing, and J. Huang, "Clusterfl: A clustering-based federated learning system for human activity recognition," *ACM Transactions on Sensor Networks*, vol. 19, no. 1, pp. 1–32, 2022.

[212] M. Yan, L. Wang, X. Wang, L. Li, L. Xu, and A. Fei, "Matching theory aided federated learning method for load forecasting of virtual power plant," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2021, pp. 327–333.

[213] H. Seo, J. Park, S. Oh, M. Bennis, and S.-L. Kim, "16 federated knowledge distillation," *Machine Learning and Wireless Communications*, vol. 457, 2022.

[214] G. Yang and H. Tae, "Federated distillation methodology for label-based group structures," *Applied Sciences*, vol. 14, no. 1, p. 277, 2023.

[215] E. Tanghatari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Federated learning by employing knowledge distillation on edge devices with limited hardware resources," *Neurocomputing*, vol. 531, pp. 87–99, 2023.

[216] M. Mendieta, T. Yang, P. Wang, M. Lee, Z. Ding, and C. Chen, "Local learning matters: Rethinking data heterogeneity in federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8397–8406.

[217] Y. Shen, Y. Zhou, and L. Yu, "Cd2-pfed: Cyclic distillation-guided channel decoupling for model personalization in federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 041–10 050.

[218] M. Asad, S. Shaukat, D. Hu, Z. Wang, E. Javanmardi, J. Nakazato, and M. Tsukada, "Limitations and future aspects of communication costs in federated learning: A survey," *Sensors*, vol. 23, no. 17, p. 7358, 2023.

[219] A. N. Mian, S. W. H. Shah, S. Manzoor, A. Said, K. Heimerl, and J. Crowcroft, "A value-added iot service for cellular networks using federated learning," *Computer Networks*, vol. 213, p. 109094, 2022.

[220] W.-y. Loh, "Some modifications of levene's test of variance homogeneity," *Journal of Statistical Computation and Simulation*, vol. 28, no. 3, pp. 213–226, 1987.

[221] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, no. 1, pp. 3–30, 1998.

[222] S. Makonin, "Hue: The hourly usage of energy dataset for buildings in british columbia," *Data in brief*, vol. 23, 2019.

[223] D. Wang, C. Li, S. Wen, S. Nepal, and Y. Xiang, "Man-in-the-middle attacks against machine learning classifiers via malicious generative models," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2074–2087, 2020.

[224] P. Chatterjee, E. Benoist, and A. Nath, *Applied approach to privacy and security for the Internet of things*. IGI Global, 2020.

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

[226] R. I. Abdelfatah, "A color image authenticated encryption using conic curve and mersenne twister," *Multimedia Tools and Applications*, vol. 79, no. 33, pp. 24 731–24 756, 2020.

[227] F. Masood, W. Boulila, J. Ahmad, Arshad, S. Sankar, S. Rubaiee, and W. J. Buchanan, "A novel privacy approach of digital aerial images based on mersenne twister method with dna genetic encoding and chaos," *Remote Sensing*, vol. 12, no. 11, p. 1893, 2020.

[228] T. Saikawa, K. Tanaka, and K. Tanaka, "Formal verification and code-generation of mersenne-twister algorithm," in *2020 International Symposium on Information Theory and Its Applications (ISITA)*. IEEE, 2020, pp. 607–611.

[229] R. Mulla. Hourly energy consumption. [Online]. Available: https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption

[230] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[231] Y. Zhou, Q. Ye, and J. Lv, "Communication-efficient federated learning with compensated overlap-fedavg," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 1, pp. 192–205, 2021.

[232] T. D. Nguyen, T. Nguyen, P. L. Nguyen, H. H. Pham, K. Doan, and K.-S. Wong, "Backdoor attacks and defenses in federated learning: Survey, challenges and future research directions," *arXiv preprint arXiv:2303.02213*, 2023.

[233] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang, and D. I. Kim, "Incentive design for efficient federated learning in mobile networks: A contract theory approach," in *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*. IEEE, 2019, pp. 1–5.

[234] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, 2019.

[235] S. Feng, D. Niyato, P. Wang, D. I. Kim, and Y.-C. Liang, "Joint service pricing and cooperative relay communication for federated learning," in *2019 International Conference on*

Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). IEEE, 2019, pp. 815–820.

[236] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6360–6368, 2020.

[237] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 8, pp. 3710–3722, 2020.

[238] S. Ghassempour, F. Girosi, and A. Maeder, "Clustering multivariate time series using hidden markov models," *International journal of environmental research and public health*, vol. 11, no. 3, pp. 2741–2763, 2014.

[239] J. Aslan, K. Mayers, J. G. Koomey, and C. France, "Electricity intensity of internet data transmission: Untangling the estimates," *Journal of Industrial Ecology*, vol. 22, no. 4, pp. 785–798, 2018.

[240] P. R. Antony and A. M. Joseph, "Design and implementation of double precision floating point comparator," *Procedia technology*, vol. 25, pp. 528–535, 2016.

[241] R. Mulla, "Hourly energy consumption, https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption, accessed on 08/08/2022." [Online]. Available: "https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption"

[242] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang, "Fedproto: Federated prototype learning across heterogeneous clients," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8432–8440.

[243] G. Paragliola, "Evaluation of the trade-off between performance and communication costs in federated learning scenario," *Future Generation Computer Systems*, vol. 136, pp. 282–293, 2022.

[244] D. Wu, X. Zou, S. Zhang, H. Jin, W. Xia, and B. Fang, "Smartidx: Reducing communication cost in federated learning by exploiting the cnns structures," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 4, 2022, pp. 4254–4262.

[245] S. Park, Y. Suh, and J. Lee, "Fedpso: Federated learning using particle swarm optimization to reduce communication costs," *Sensors*, vol. 21, no. 2, p. 600, 2021.

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

[247] M. Roussopoulos, M. Baker, D. S. Rosenthal, T. J. Giuli, P. Maniatis, and J. Mogul, "2 p2p or not 2 p2p?" in *Peer-to-Peer Systems III: Third International Workshop, IPTPS 2004, La Jolla, CA, USA, February 26-27, 2004, Revised Selected Papers 3*. Springer, 2005, pp. 33–43.

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

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

[250] J. Zhang, H. Zhu, F. Wang, J. Zhao, Q. Xu, and H. Li, "Security and privacy threats to federated learning: Issues, methods, and challenges," *Security and Communication Networks*, vol. 2022, no. 1, p. 2886795, 2022.

[251] X. Sha, W. Sun, X. Liu, Y. Luo, and C. Luo, "Enhancing edge-assisted federated learning with asynchronous aggregation and cluster pairing," *Electronics*, vol. 13, no. 11, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/11/2135

[252] H. Wen, Y. Wu, J. Hu, Z. Wang, H. Duan, and G. Min, "Communication-efficient federated learning on non-iid data using two-step knowledge distillation," *IEEE Internet of Things Journal*, 2023.

[253] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE transactions on information theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[254] A. Nedic, "Asynchronous broadcast-based convex optimization over a network," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1337–1351, 2010.

[255] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.