Federated Learning for Next Generation Intelligent Applications

Ahsan Raza Khan

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

James Watt School of Engineering
College of Science and Engineering
University of Glasgow

March 2025

Abstract

The rapid proliferation of smart devices and Internet of Things (IoT) technologies has revolutionised data collection for artificial intelligence (AI)-driven applications, enabling rapid training and near real-time inference. However, the traditional centralised learning approaches require transferring vast amounts of raw data from end devices to a central server. This process leads to substantial network overhead, increased latency, and significant privacy concerns, hindering the scalability and responsiveness of intelligent applications. This thesis exploits federated learning (FL) as a distributed, on-device learning framework that enables collaborative model training without raw data sharing. The distributed architecture of FL offers privacy by design and reduces communication costs by exchanging the model parameters that align with principles of data sovereignty and regulatory compliance. Despite its advantages, FL faces significant challenges in real-world applications, and this thesis aims to address the following three critical challenges: **C1)** data diversity; **C2)** robust aggregation ensuring privacy and security in the training process; and finally, **C3)** energy efficiency. The first contribution introduces the similarity-driven truncated aggregation (SDTA) framework, designed to tackle challenges **C1** and **C2**. SDTA measures the similarity among the model updates to identify and filter the anomalous updates, mitigating the impact of attacks and overfitting without accessing client data. Additionally, it incorporates differential privacy (DP) to strengthen training privacy. The second contribution introduces the semantic-aware federated blockage prediction (SFBP) framework, addressing challenges **C1** and **C3**. Using multi-modal fusion and a lightweight computer vision model for edge-based semantic extraction, the proposed framework reduces communication costs and inference delays while maintaining high prediction accuracy. Additionally, SFBP incorporates a filter mechanism to minimise the effects of noisy or adversarial updates. The third contribution addresses **C1** and **C3** and develops a hybrid neuromorphic federated learning (HNFL) framework for outdoor human activity recognition (HAR) using wearable sensors. The proposed spiking-long short-term memory (S-LSTM) model combines the energy-efficient spiking neural networks with the sequential data handling strengths of LSTM networks. This approach improves the accuracy while ensuring data privacy and reducing computational costs, making it suitable for deployment on resource-constrained edge devices. Finally, to address challenges **C2** and **C3**, the federated fusion quantisation (FFQ) framework is proposed to improve HAR models in indoor settings. FFQ combines FL with edge-based preprocessing, feature

engineering, and model compression to achieve a low false positive rate, essential for applications like fall detection. A customised FedDist algorithm is used for global model aggregation, effectively reducing overfitting in diverse data. Additionally, FFQ applies model compression and quantisation-aware training to lower communication overhead without compromising accuracy. These contributions advance FL by enhancing scalability, robustness, and efficiency, paving the way for next-generation intelligent systems.

Keywords

Federated Learning, Energy Efficiency, Robust Aggregation, Semantic Information Processing, Vision-Aided Wireless Communication, Neuromorphic Computing.

Acknowledgements

I extend my deepest gratitude to all those who have supported and guided me throughout my doctoral journey. First and foremost, I am profoundly grateful to my supervisor, **Dr. Ahmed Zoha**, whose exceptional mentorship and unwavering support have been the cornerstone of my academic and personal development. His insightful advice and encouragement have continually inspired me to strive for excellence. His dedication to fostering a collaborative and intellectually stimulating environment has enhanced my research skills and broadened my perspective. His patience, wisdom, and willingness to guide me through challenges have been invaluable.

I am also deeply indebted to my co-supervisor, **Prof. Muhammad Imran**, for his persistent support and invaluable insights. His visionary leadership has been instrumental in shaping the direction of my research. His ability to anticipate future trends and encourage innovative thinking has significantly enriched my academic experience. Under his guidance, I have learnt to approach problems with a forward-thinking mindset, which has not only enhanced the quality of my work but also prepared me for future challenges in the field. I am incredibly grateful to **Dr. Lina Mohjazi** and **Prof. David Flynn** for their persistent support and invaluable insights. Their expertise and guidance have significantly enriched my research experience.

A special thanks go to **Prof. Sajjad Hussain**, **Habib Ullah Manzoor**, **Dr. Iftikhar Ahmed** and **Fahad Ayaz**, whose collaborative spirit and scholarly discussions have contributed significantly to the progression of my work. I want to thank the University of Glasgow, especially the Communication Sensing and Imagining (CSI) group, for providing a nurturing and stimulating academic environment during challenging times. The resources and support offered have been instrumental in facilitating my research endeavours.

Above all, I am eternally grateful to my mother and father, whose unwavering love and support have been the foundation of all my achievements. Their endless encouragement and belief in my abilities have given me the strength and determination to pursue my dreams. To my sisters, thank you for your constant support and for always being there when I needed you. To my loving wife, whose patience, understanding, and encouragement have been invaluable throughout this journey, thank you for your endless support and for being my pillar of strength. Lastly, to my daughters Zimal and Imsaal, who have been a source of joy and inspiration, your smiles have brightened even the most challenging days.

Declaration

University of Glasgow

College of Science & Engineering

Statement of Originality

Name: Ahsan Raza Khan
Registration Number: XXXXXXXX

I certify that the thesis presented here for examination for a PhD degree in the University of Glasgow is solely my own work other than where I have indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it) and that the thesis has not been edited by a third party beyond what is permitted by the University's PGR Code of Practice.

I declare that the thesis does not include work forming part of a thesis presented successfully for another degree.

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 if any issues are raised regarding good research practice-based on review of the thesis, the examination may be postponed pending the outcome of any investigation of the issues.

Signature:
Date: 12/03/2025

Dedication

To my beloved father, ***Nadeem Khalil Khan****, whose unwavering love, wisdom, and sacrifices continue to inspire me every day. Though you are no longer with us, your guidance and values have shaped the person I am today. This work is a tribute to your memory.*

Statement of Copyright

Contents

List of Tables

List of Figures

List of Publications

Journal

1. **Ahsan Raza Khan**, H. U. Manzoor, R.N.B Rais, S. Hussain, M. A. Imran, and A. Zoha: "Semantic-Aware Federated Blockage Prediction (SFBP) in Vision-Aided Next-Generation Wireless Network" (2025) IEEE Transaction on Network Service Management https://ieeexplore.ieee.org/abstract/document/10820115

2. **Ahsan Raza Khan**, M. Al-Quraan, L. Mohjazi, D. Flynn, M. A. Imran, and A. Zoha, "Similarity Driven Truncated Aggregation (SDTA) for Privacy-Preserving Short-Term Load Forecasting" (2025), Elsevier Internet of Things, Vol no. 31, ISSN 2542-6605 https://doi.org/10.1016/j.iot.2025.101530

3. **Ahsan Raza Khan**, H. U. Manzoor, F. Ayaz, M. A. Imran, and A. Zoha: "A Privacy and Energy-Aware Federated Framework for Human Activity Recognition" (2023) Sensors, Vol no. 23: 9339. https://doi.org/10.3390/s23239339

4. **Ahsan Raza Khan**, R.N.B. Rais, S. Sohaib, M. A. Imran, and A. Zoha: "FedFusionQuant (FFQ): Federated Learning with Feature Fusion and Model Quantisation for Human Activity Recognition using CSI" (2024) IEEE Transaction in Sustainable Computing (Major Revision)

5. **Ahsan Raza Khan**, I. Ahmad, L. Mohjazi, S. Hussain, R.N.B Rais, M. A. Imran, and A. Zoha: "Latency-Aware Blockage Prediction in Vision-Aided Federated Wireless Networks" (2023) Frontiers in Communications and Networks, Vol. no. 4, p.1130844. https://doi.org/10.3389/frcmn.2023.1130844

6. Sana Hafeez, **Ahsan Raza Khan**, M. Al-Quraan, L. Mohjazi, A. Zoha, M. A. Imran, and Yao Sun: "Blockchain-Assisted UAV Communication Systems: A Comprehensive Survey" (2023) IEEE Open Journal of Vehicular Technology, Vol. no. 4, pp.558-58. https://ieeexplore.ieee.org/document/10182294

7. M. Al-Quraan, **Ahsan Raza Khan**, L. Mohjazi, Antony Centeno, A. Zoha, and M. A. Imran: "Intelligent Beam Blockage Prediction for Seamless Connectivity in Vision-Aided

Next-Generation Wireless Networks" (2022), IEEE Transactions on Network and Service Management, Vol. no. 20 (2), pp.1937-1948. https://ieeexplore.ieee.org/document/9926150

8. M. Al-Quraan, **Ahsan Raza Khan**, A. Centeno, A. Zoha, M. A. Imran, and L. Mohjazi. "FedraTrees: A Novel Computation-Communication Efficient Federated Learning Framework Investigated in Smart Grids" (2023) Engineering Applications of Artificial Intelligence, Vol. no. 124: 106654. https://doi.org/10.1016/j.engappai.2023.106654

9. H. U. Manzoor, **Ahsan Raza Khan**, D. Flynn, Muhammad Mahtab Alam, Muhammad Akram, M. A. Imran, and A. Zoha: "FedBranched: Leveraging Federated Learning for Anomaly-Aware Load Forecasting in Energy Networks" (2023) Sensors, Vol. no. 7: 3570. https://doi.org/10.3390/s23073570

10. **Ahsan Raza Khan**, U. Ahmed, A. Mahmood, P. Yadav, M. A. Imran, and A. Zoha: "SpikeNet: A Hybrid Model for Short-Term Load Forecasting using Spike Neural Network" (2024) IEEE Transaction on Artificial Intelligence (Under Review)

11. I. Ahmad, **Ahsan Raza Khan**, A. Jabbar, M. Alquraan, L. Mohjazi, Masood Ur Rehman, M. A. Imran, A. Zoha, S. Hussain: "Proactive Blockage Prediction for UAV-assisted Handover in Future Wireless Network" (2024) Elsevier Physical Communication (Under review)

12. U. Ahmed, **Ahsan Raza Khan**, A. Mahmood, Iqra Rafiq, R. Ghannam, and A. Zoha: "Short-Term Global Horizontal Irradiance Forecasting Using Weather Classified Categorical Boosting" (2023) Applied Soft Computing Vol. no. 155 111441. https://doi.org/10.1016/j.asoc.2024.111441

13. S. Bhatti, **Ahsan Raza Khan**, A. Zoha, S. Hussain, and R. Ghannam: "A Machine Learning Frontier for Predicting LCOE of Photovoltaic System Economic" (2023) Advanced Energy and Sustainability Research, Vol. no. 5(8). https://doi.org/10.1002/aesr.202300178

Conference Proceedings

1. H. U. Manzoor, **Ahsan Raza 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, In 2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), (2023) Sept. 5-8 (pp. 1-7). IEEE.

2. Iftikhar Ahmed, **Ahsan Raza Khan**, R.N.B Rais, A. Zoha, M. A. Imran, and S. Hussain: "Vision-Assisted Beam Prediction for Real World 6G Drone Communication" In 2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), (2023) Sept. 5-8 (pp. 1-7). IEEE.

3. **Ahsan Raza Khan**, S. M. Bokhari, S. Khosravi, S. Hussain, R. Ghannam, M. A. Imran, and A. Zoha: "Feature Selection Mechanism for Attention Classification Using Gaze Tracking Data" In 2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS), (2022) Oct 24-26 (pp. 1-4). IEEE.

4. H. U. Manzoor, M. S. Khan, **Ahsan Raza 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), (2022) Oct 24-26 (pp. 1-4). IEEE.

5. H. U. Manzoor, **Ahsan Raza 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) (2022) June 14-17 (pp. 644-649). IEEE.

6. **Ahsan Raza Khan**, S. Khosravi, S. Hussain, R. Ghannam, A. Zoha, and M. A. Imran: "Execute: Exploring eye tracking to support e-learning." In 2022 IEEE Global Engineering Education Conference (EDUCON) (2022), March 28-31 (pp. 670-676). IEEE.

7. M. Al-Quraan, **Ahsan Raza Khan**, L. Mohjazi, A. Centeno, A. Zoha, and M. A. Imran: "A hybrid data manipulation approach for energy and latency-efficient vision-aided UDNs" In Eighth International Conference on Software Defined Systems (SDS) (2021), Dec 6-9 pp. 1-7. IEEE.

8. **Ahsan Raza Khan**, A. Zoha, L. Mohjazi, H. Sajid, Q. H. Abbasi, M. A. Imran: "When federated learning meets vision: An outlook on opportunities and lenges." In EAI International Conference on Body Area Networks (2021) Dec 25 (pp. 308-319). Springer.

Book Chapters

1. **Ahsan Raza Khan**, H. U. Manzoor, F. Ayaz, M. A. Imran, and A. Zoha: "Hybrid Neuromorphic Federated Learning (HNFL): A Spiking-LSTM for Human Activity Recognition using Wearable Sensors" (2024). Accepted in Wiley `https://doi.org/10.1002/9781394257744.ch6`.

2. Iftikhar Ahmed, **Ahsan Raza Khan**, R.N.B Rais, A. Zoha, M. A. Imran, and S. Hussain: “Multimodal Beam Prediction for Enhanced Beam Management in Drone Communication" (2024), Accepted in Wiley `https://doi.org/10.1002/9781394257744.ch7`.

List of Abbreviations

5G	Fifth Generation
6G	Sixth Generation
AI	Artificial Intelligence
AMI	Advanced Metering Infrastructure
Amp	Amplitude
AR	Augmented Reality
ARIMA	Auto-Regressive Integrated Moving Average
ARMA	Auto-Regressive Moving Average
B5G	Beyond Fifth Generation
BPTT	Backpropagation Through Time
CDFL	Cross-Domain Federated Learning
CFO	Carrier Frequency Offset
CSI	Channel State Information
CTF	Combined Temporal Features
CV	Computer Vision
DFL	Decentralised Federated Learning
DL	Deep Learning
DP	Differential Privacy
DNN	Deep Neural Network
DWT	Discrete Wavelet Transform
EMD	Empirical Mode Decomposition
EN	Edge Node
FedAKD	Federated Augmented Knowledge Distillation
FedAVG	Federated Averaging
FedBranched	Federated Branched
FedDist	Federated Distance
FFQ	Federated Fusion Quantisation
FedMA	Federated Matched Averaging
FedMeta	Federated Meta-Learning
FedProx	Federated Proximal

FedSplit	Federated Splitting
FN	False Negative
FP	False Positive
FS	Federated Server
FTL	Federated Transfer Learning
GDPR	General Data Protection Regulation
HAR	Human Activity Recognition
HE	Homomorphic Encryption
HFL	Horizontal Federated Learning
HNFL	Hybrid Neuromorphic Federated Learning
IID	Independent and Identically Distributed
IMF	Intrinsic Mode Function
IoT	Internet of Things
kWh/GB	Kilowatt-hours per Gigabyte
LDA	Linear Discriminant Analysis
LIF	Leaky Integrate-and-Fire
LOS	Line-of-Sight
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MEC	Mobile Edge Computing
ML	Machine Learning
mmWave	Millimetre Wave
NA	No Activity
NLOS	Non-Line-of-Sight
OFDM	Orthogonal Frequency Division Multiplexing
PHO	Proactive Handover
QAT	Quantisation-Aware Training
RSSI	Received Signal Strength Indicator
S-CNN	Spiking Convolutional Neural Network
SD-FedAVG	Similarity-Driven Federated Averaging
SDTA	Similarity-Driven Truncated Aggregation
SFBP	Semantic-Aware Federated Blockage Prediction
SGD	Stochastic Gradient Descent
SMPC	Secure Multi-Party Computation
S-LSTM	Spiking-Long Short-Term Memory
SNN	Spiking Neural Network
STDP	Spike-Timing-Dependent Plasticity

STLF	Short-Term Load Forecasting
TEE	Trusted Execution Environment
THz	Terahertz
TP	True Positives
TN	True Negatives
UCI	University of California, Irvine
USRP	Universal Software Radio Peripheral
VFL	Vertical Federated Learning
YOLOv5	You Only Look Once Version 5

List of Symbols

N	Total number of edge nodes (ENs) or clients
$\mathscr{D}_i$	Local dataset at the i-th edge node
$\mathbf{X}_{ij}$	Feature vector of the j-th sample in the i-th local dataset
$\mathbf{y}_{ij}$	Target class or value for the j-th sample in the i-th local dataset
$\ell_i(\omega, \mathscr{D}_i)$	Local loss function at the i-th edge node
ω	Global model parameters
ω_i^t	Model parameters of the i-th client at communication round t
η	Learning rate for local training
$\nabla \ell_i(\omega)$	Gradient of the local loss function at the i-th edge node
$\lvert\mathscr{D}_i\rvert$	Size of the local dataset $\mathscr{D}_i$
E	Number of local epochs per communication round
C	Fraction of clients participating in each communication round
L	Total number of layers in the model
$\Delta\omega_i^{t,l}$	Local model update for the i-th client at layer l and round t
$S_{(i,j)}^{t,l}$	Cosine similarity between updates of clients i and j at layer l in round t
$\mathbf{S}^{t,l}$	Similarity matrix for layer l in communication round t
$\bar{S}_i^{t,l}$	Average similarity score for client i at layer l
Z	Truncation percentage for misaligned updates
N_{Tr}^l	Number of truncated ENs for layer l
N_{R}^l	Number of remaining ENs for aggregation at layer l
$\Delta\omega^{t,l}$	Aggregated update for layer l in round t
x_o	Scaled value of the input feature after Min-Max normalization
$Min(x)$	Minimum value of the feature x
$Max(x)$	Maximum value of the feature x
ε	Privacy budget in differential privacy
δ	Probability of exceeding the privacy loss budget ε
σ	Noise scale in the Gaussian mechanism
Δs	Sensitivity of the function in differential privacy
τ	Threshold for filtering noisy updates
β	Leakage factor in spiking neural networks

v_{th}	Membrane potential threshold for spiking
$V_i^{(l)}(t)$	Membrane potential of neuron i in layer l at time t
$o_i^{(l)}(t)$	Spike output of neuron i in layer l at time t
γ	Scaling factor for surrogate gradient
t_{cap}	Time required for image capture
t_{tx}	Time required for data transmission
t_{inf}	Time required for model inference
E_{est}	Energy efficiency estimate
α, β	Computation and communication constants
t_c	Computation time for local training
P_{trn}	Size of transmitted parameters
$\hat{X}_i(t)$	Denoised signal for client i at time t
$IMF_{ik}(t)$	Intrinsic mode function for k at client i
$r_i(t)$	Residue of decomposed signal at client i
$\Phi_{j,i}$	Wavelet coefficients at level j and location i
$\tilde{\Phi}_{j,i}$	Denoised wavelet coefficients at level j and location i
τ_j	Threshold for wavelet coefficients at level j
$Diff_i^W$	Differential feature for client i in window W
H_i	Combined temporal features for client i
$Q(\cdot, b)$	Quantisation function with bit precision b
$\hat{\omega}_i^{t+1}$	Quantised local model parameters for client i at round $t+1$
$dist(\hat{N}_1, \hat{N}_2)$	Euclidean distance between neurons $\hat{N}_1$ and $\hat{N}_2$
C_b	Between-class scatter matrix in LDA
C_w	Within-class scatter matrix in LDA
C_t	Total scatter matrix in LDA

Chapter 1

Introduction

1.1 Background

The rapid progress in big data and artificial intelligence (AI) has brought significant changes to the modern technological landscape. The proliferation of Internet of Things (IoT) devices and smart mobile gadgets, alongside advancements in communication technologies, has fueled an unprecedented surge in data generation. According to the data protection corporation, 80 billion IoT devices are expected to be connected in future systems, collectively producing an impressive 163 zettabytes of data worldwide [1, 2]. This massive data influx, coupled with the growing capabilities of edge processing, is driving the development of innovative, data-driven applications across sectors. These AI-driven applications include smart healthcare, wireless communication systems, e-banking, live surveillance, augmented reality, and smart cities [3].

Despite these opportunities, traditional machine learning (ML) model training, which heavily depends on centralised data processing, faces significant obstacles. In a centralised approach, data from various distributed sources is sent to a central server for training and analysis [4]. While this method is effective for handling large datasets, it presents several challenges that make it less practical for future applications [5, 6]:

- **Privacy Concerns:** Centralised data collection often includes handling sensitive information and posing risks to user privacy. Data protection laws like the General Data Protection Regulation (GDPR) in Europe and the Consumer Privacy Bill of Rights in the United States impose strict rules on data collection, storage, and processing [7], [8]. These regulations aim to protect user data, requiring explicit consent from data owners, which limits data accessibility and impacts model performance.

- **Latency and Time Sensitivity:** Transferring large volumes of data to a central server results in high latency due to transmission delays [9]. This can be particularly problematic in applications that require quick responses, such as smart healthcare, autonomous vehicles, drone surveillance, and augmented reality, where timely decisions are crucial.

- **Communication Costs and Bandwidth Utilisation:** The transfer of unstructured raw data to a centralised location burdens the backbone network, increasing communication costs and resulting in substantial network footprints.
- **Storage Costs:** Centralised data collection incurs additional storage and maintenance costs. Additionally, data collection is a very time-consuming and expensive process.

In response to these challenges, a new approach has emerged that brings computation closer to the data source. Mobile edge computing (MEC) leverages the processing power of edge devices and servers, reducing the need for data transfers to central servers [10]. While MEC lowers the latency and saves bandwidth, it still involves data transfer to edge servers, raising ongoing privacy concerns that discourage collaborative model training. This has led to the development of federated learning (FL) [11], a paradigm that allows computation to occur on edge devices, thus maintaining user privacy. Google, a pioneer in this field, utilises FL algorithms extensively to improve services like Gboard and predictive text [12]. Although FL was initially focused on smartphone and edge device applications, its integration with IoT sensors and advanced AI tools has opened up numerous industrial applications in Industry 4.0, digital healthcare, smart cities, smart buildings, drug discovery, video surveillance, digital imaging, and AR/VR, as well as in autonomous vehicles [13].

For example, vision processing has become a crucial technology, especially in healthcare and smart city applications. Vision sensors produce massive data that current wireless networks struggle to handle efficiently, particularly for time-sensitive use cases. The primary challenge lies in the high communication costs and delays caused by network congestion [1]. However, 5G connectivity combined with FL enables a range of vision-based applications in areas like smart healthcare, wireless communications, live traffic monitoring, and incident management [14]. Many of these applications require strong privacy protections and low latency, making the 5G and FL combination a promising solution for next-generation intelligent systems.

1.2 Scope and Motivation

FL has addressed several critical limitations inherent in centralised learning frameworks by enabling collaborative model training across distributed data sources without transferring raw data to a central server. FL inherently enhances privacy, reduces communication costs, and decreases latency. These advantages make FL particularly appealing for intelligent applications in domains where data sensitivity and the need for real-time response are paramount, such as smart grids, wireless communication, and healthcare [15]. The decentralised nature of FL aligns with the principles of data sovereignty, offering a scalable approach that can leverage the computational capabilities of edge devices while preserving user trust and regulatory compliance [9].

Despite its notable advantages, current FL frameworks face several challenges that hinder

their widespread adoption and effectiveness in complex, real-world environments. One key challenge is data diversity, as client devices often generate non-independent and identically distributed (non-IID) data due to variations in user behavior and device specifications. This heterogeneity can significantly impact the convergence and generalisation capabilities of the global model, leading to suboptimal performance [16, 17]. Variations in data distributions among clients, stemming from differences in user behaviour, device specifications, and environmental contexts, complicate the training process, making it challenging to achieve robust, consistent model performance. Additionally, many applications using multi-modal data fusion to train a robust model make FL more complicated.

Another significant challenge is energy efficiency, which combines computation cost and communication overhead. The computation cost refers to the amount of energy required to train a model on the edge device, whereas the communication cost is the amount of energy needed to model parameter sharing during the training. FL requires iterative communication between client devices and a central server to update and synchronise the global model. This process results in substantial communication costs, which strain network resources and limit scalability [18]. Furthermore, frequent model updates increase the energy consumption of participating devices, posing a problem for battery-operated edge nodes, particularly in IoT-based applications where resource constraints are critical [19]. Adversarial attacks and security threats further complicate FL implementations. While preserving data privacy, the decentralised nature of FL exposes the system to various vulnerabilities, such as model poisoning, where malicious participants deliberately corrupt their local models to degrade the overall model performance [16, 17]. Additionally, inference attacks, where adversaries aim to extract sensitive information from shared model updates, pose significant risks, undermining the privacy benefits that FL seeks to uphold [20]. Addressing these security concerns is essential to ensuring the trustworthiness and reliability of FL systems, especially in critical applications like healthcare and smart infrastructure.

The choice of aggregation mechanisms in FL is another area that presents challenges. The commonly used federated averaging (FedAVG) algorithm, while effective for many scenarios, is often inadequate for handling data diversity and robustness to adversarial contributions [21]. Aggregation strategies need to account for the quality and trustworthiness of client updates, ensuring that outliers or malicious data do not disproportionately affect the global model. Furthermore, current aggregation mechanisms often overlook the nuances of layer-wise contributions from client models, which could be pivotal in managing model convergence and addressing data variability.

1.3 Problem Statement and Objectives

Despite the advancements in FL, significant challenges persist in achieving optimal efficiency, security, and robustness across various intelligent applications. The scope of this thesis addresses three critical challenges that limit the deployment of FL systems in real-world scenarios, which include:

C1 Data Diversity: Data generated by edge devices in FL systems is inherently diverse and non-IID due to variations in user behaviour, device specifications, and environmental contexts. The data diversity adversely affects the convergence and generalisation capabilities of the global model, leading to suboptimal performance [16, 17]. Traditional FL techniques often employ clustering approaches to manage heterogeneous data. While clustering works well for highly variable and unpredictable data, such as household energy forecasting, it adds unnecessary complexity for more uniform data, like substation-level energy consumption. Furthermore, clustering methods often use predefined static clusters, which struggle to capture transient data variations, limiting the model's adaptability to evolving patterns. Additionally, training multiple models for distinct clusters increases computational overhead, hindering scalability. Moreover, modern intelligent applications often require multi-modal data fusion, where data from different modalities (e.g., visual, sensory, textual) are integrated to train robust models. Current FL frameworks struggle to manage these challenges effectively, underscoring the need for solutions that handle diverse and multi-modal data seamlessly without the drawbacks of clustering.

C2 Robust Aggregation: While FL ensures privacy by keeping raw data on local devices, it remains vulnerable to adversarial attacks. Malicious participants deliberately corrupt their local models (model poisoning) to degrade overall performance. Additionally, the inference attacks pose additional risks, where adversaries aim to extract sensitive information from shared model updates, undermining the privacy benefits of FL. Furthermore, the commonly used FedAVG with clustering is inadequate for handling malicious contributions. Moreover, integrating differential privacy (DP) within clustering frameworks adds another layer of difficulty. Noise injection required for DP can disproportionately impact smaller clusters, leading to degraded accuracy. Moreover, clustering assumes synchronous communication and consistent client participation, which is challenging to maintain in real-world FL systems with intermittent connectivity. These issues highlight the need for robust aggregation mechanisms that enhance security, protect against adversarial attacks, and maintain model adaptability without compromising performance or scalability.

C3 Energy Efficiency Energy efficiency is a major challenge in FL, especially for resource-constraint edge devices like IoT sensors and wearables. In FL, energy consumption comes from two sources: computation cost, the energy needed for local model training, and

communication cost, the energy used to transmit model updates during training. Frequent communication between client devices and the central server creates substantial overhead, increasing bandwidth usage and draining device batteries, which impacts scalability and real-time performance. Addressing these challenges requires innovative solutions like model compression, semantic information extraction, and efficient aggregation techniques. These methods can reduce the communication and computational load while maintaining seamless integration with existing FL frameworks and ensuring high model performance.

1.3.1 Aims and Objectives

In response to three challenges identified in the problem statement, the objectives of this thesis are outlined as follows:

1. Design and implement FL frameworks capable of effectively managing diverse and heterogeneous data distributions, including multi-modal datasets, to ensure robust and scalable model training. This involves developing energy-efficient solutions that integrate advanced data processing techniques, seamless multi-modal fusion, and adaptive aggregation methods while avoiding the limitations of clustering and specific hardware dependencies

2. Investigate the vulnerabilities of FL systems, particularly the impact of adversarial and poisoning attacks on model performance. The main aim is to develop and implement robust aggregation strategies to identify and mitigate malicious updates, ensuring the integrity and reliability of the training process while addressing the privacy challenges.

3. Design and implement a scalable and energy-efficient hybrid FL framework optimised for resource-constrained edge devices. The aim is to develop lightweight neuromorphic architectures that integrate model compression and feature engineering to reduce computational demands and communication overhead while ensuring high performance and real-time responsiveness, even in applications with stringent privacy and latency requirements.

1.4 Research Contributions

The main contributions of this thesis are itemised as follows:

- The first contribution of this thesis introduces the similarity-driven truncated aggregation (SDTA) framework to address the challenges **C1** and **C2**, leveraging the similarity measures, filtering process, and DP mechanism. The similarity measure aligns the model

updates and detects the anomalies, whereas the filter process excludes the extreme values to mitigate the adversarial updates. DP is incorporated to safeguard model updates during training to further enhance security. Unlike traditional methods, SDTA employs weighted averaging to address adversarial attacks, such as model sign inversion, without completely excluding client contributions. Extensive simulations on real-world substation data for short-term load forecasting (STLF) demonstrate that SDTA outperforms existing methods like FedAVG and federated distance (FedDist), especially in scenarios involving privacy constraints, adversarial challenges, and client dropouts.

- The second contribution of this thesis introduces the semantic-aware federated blockage prediction (SFBP) framework, addressing challenges **C1** and **C3**. The proposed SFBP employs the multi-modal fusion, lightweight computer vision (CV) model for edge-based semantic extraction, anomaly detection, and filtering for robust aggregation. This framework leverages bimodal visual and wireless sensing data to improve blockage prediction accuracy, enhancing proactive handover (PHO) performance in communication networks. A lightweight CV model is used for edge-based semantic extraction, significantly reducing communication costs and inference latency. To ensure robust model training, SFBP incorporates similarity-driven FedAVG (SD-FedAVG), which uses anomaly detection and filtering to mitigate the impact of noisy or adversarial updates. Detailed analyses explore the effects of noise and data variability on blockage prediction accuracy, including the impact of false positives (FP) and false negatives (FN) on PHO success rates. Additionally, the framework compares the energy efficiency (measured in kilowatt-hours per gigabyte (kWh/GB)) for raw data transfer in centralised training with model parameter sharing in FL, highlighting substantial energy savings through semantic information sharing. Benchmarking against state-of-the-art methods demonstrates SFBP's superior trade-offs between computational efficiency and prediction accuracy.

- The third contribution of this thesis proposes a hybrid neuromorphic federated learning (HNFL) framework for outdoor HAR using wearable sensing time series data, addressing challenges **C1** and **C3**. The HNFL framework integrates multi-modal data fusion with a spiking long short-term memory (S-LSTM) model, which combines the event-driven processing capabilities of spiking neural networks (SNNs) with the sequential data handling strengths of LSTM networks. This innovative approach enhances activity recognition accuracy while ensuring data privacy and reducing computational costs. This study also thoroughly evaluates the performance of the S-LSTM model using two publicly available datasets, demonstrating its superiority over traditional models like LSTM, spiking convolutional neural networks (S-CNN), and standard convolutional neural networks (CNN). Additionally, the research explores the impact of random client selection on the HAR model's performance, providing insights into balancing computational and communica-

tion efficiency with model accuracy.

- The final contribution of this thesis introduces the federated fusion quantisation (FFQ) framework, addressing challenges **C2** and **C3**. This framework enhances the performance of indoor HAR models by combining the FL with edge-based preprocessing and feature engineering. FFQ extracts and combines the statistical and differential features from raw signals, achieving a low false positive rate, particularly for critical tasks like fall detection. This framework incorporates a tailored FedDist algorithm for global model aggregation to address overfitting in diverse data. The FedDist algorithm uses a modified divergence metric to adjust model parameters based on dissimilarity measures of model updates. Additionally, the FFQ framework employs model compression with quantisation-aware training (QAT), significantly reducing communication overhead while maintaining model accuracy.

1.5 Thesis Organisation

This thesis is structured into seven chapters, each addressing specific aspects of the research challenges, proposed solutions, and their applications. The detailed organisation is as follows:

Chapter 1 provides an overview of the research context, motivation, and objectives of the thesis. It highlights the contributions and outlines the challenges addressed, laying the foundation for the subsequent chapters.

Chapter 2 comprehensively reviews the existing literature on FL and its applications. It begins by exploring state-of-the-art FL techniques, limitations, and their applications in energy networks, communication systems, and healthcare domains. The chapter also analyses critical challenges associated with data diversity, adversarial robustness, privacy, and energy efficiency. This review identifies the research gaps, particularly in model aggregation, multi-modal data fusion, adversarial attack resilience, and computational efficiency. The chapter concludes with a discussion of potential application areas for FL, emphasising the need for innovative frameworks to address the identified three challenges discussed in **Section 1.3**.

Chapter 3 introduces the SDTA framework, which addresses challenges **C1** and **C2**, leveraging the similarity measures, filtering process, and DP mechanism. First, it draws on the motivation of this study and discusses the proposed framework, simulation setup, dataset used, results, and outcomes.

Chapter 4 introduces SFBP framework, addressing challenges **C1** and **C3**. This chapter discusses the process of multi-modal fusion, edge-based semantic information extraction, and robust aggregation mechanisms to mitigate adversarial updates. Finally, the proposed framework is tested on a wireless communication application, and the results and outcomes are discussed.

Chapter 5 focuses on the HNFL framework, which addresses challenges **C1** and **C3**. This chapter discusses the multi-modal data fusion with a novel S-LSTM model, combining the

energy-efficient event-driven processing of SNNs with the sequential data handling capabilities of LSTM networks. The chapter evaluated the effectiveness of the proposed framework in outdoor HAR applications using wearable sensing data. It benchmarked against traditional models such as LSTM, S-CNN, and CNN, demonstrating the robustness and efficiency of the HNFL framework.

Chapter 6 introduces the FFQ framework, which addresses challenges **C2** and **C3**. This chapter discusses the integration of FL with edge-based preprocessing and feature engineering for contactless sensing in indoor HAR applications. Additionally, this chapter investigated the impact of feature fusion in achieving a low false positive rate, which is critical for tasks like fall detection. This chapter also discusses QAT for model compression, significantly reducing communication overhead while maintaining model accuracy.

Chapter 7 summarises the contributions of this thesis, highlighting the advancements made in FL frameworks to address the three challenges identified in **Section 1.3**. It also discusses the broader implications of the proposed solutions and their applicability across various domains. The chapter concludes with potential directions for future research, such as exploring adaptive aggregation methods, integrating FL with novel sensing modalities, and extending the frameworks to new application areas.

Chapter 2

Literature Review

In recent years, federated learning (FL) has emerged as a transformative paradigm in distributed machine learning (ML), enabling collaborative model training across decentralised data sources without requiring data sharing. This chapter reviews the fundamental aspects of FL, focusing on its application in next-generation intelligent systems that demand strict privacy, high energy efficiency, and resilience against data and system heterogeneity. The chapter begins by outlining the foundational concepts and architectural frameworks of FL and introducing key protocols and mechanisms that make FL a viable solution for distributed learning across diverse environments. It also covers the critical aspects of privacy and security, exploring established methods such as DP and secure multi-party computation to protect data confidentiality. Furthermore, this chapter specifically aligns the literature review with three critical challenges identified in Section 1.3, and how these factors impact model performance and convergence in FL settings. For instance, Section 2.3, 2.4 summarise the literature for challenge **C1**, whereas Section 2.5, 2.6 covers for challenge **C2**. A literature review for challenge **C3** is presented in 2.7, and a review of FL applications in smart grids, wireless communications systems, and healthcare, identifying current limitations and research gaps, is presented in Section 2.8. Finally, Section 2.9 summarises the research gap and links it with three core challenges identified in the thesis.

2.1 Federated Learning: An Overview

FL is a decentralised ML paradigm that enables multiple clients or devices to collaboratively train a shared global model while keeping their local data private. Unlike traditional centralised approaches, where data is aggregated on a central server, FL allows model training to occur directly on edge nodes (EN) [11]. This approach addresses critical concerns related to data privacy, security, and communication overhead, making it particularly suitable for applications involving sensitive information or bandwidth constraints [13, 22]. The motivation for FL arises from the increasing need to harness distributed data generated by edge devices, such as smartphones, internet of things (IoT) devices, and organisational silos, without compromising user privacy

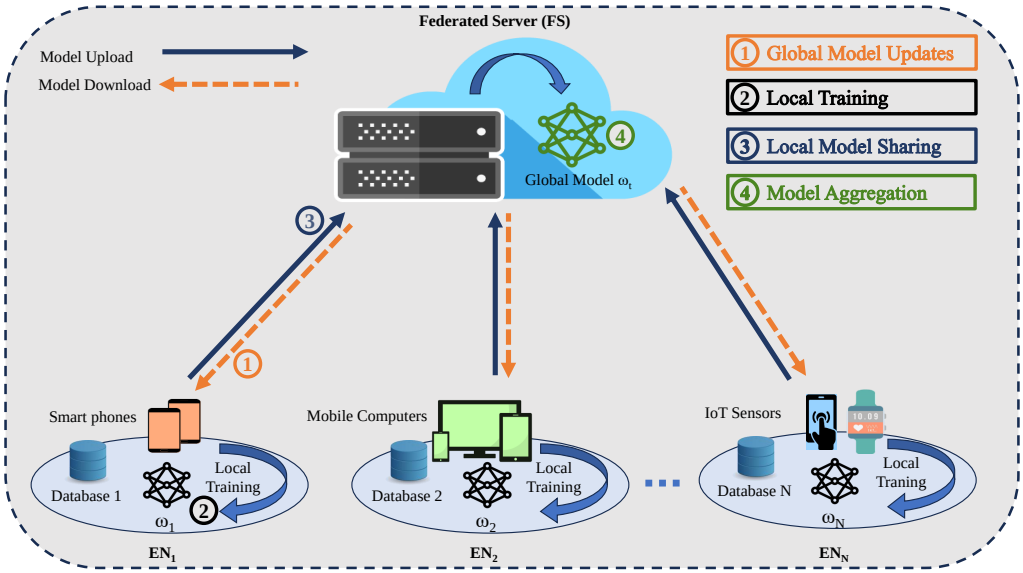


Figure 2.1: The generic framework and training process of FL.

or violating data protection regulations like general data protection regulation (GDPR) [23]. FL mitigates the risks associated with data breaches and unauthorised access by ensuring that raw data remains on local devices, fostering trust among participants in collaborative learning environments.

2.1.1 FL Training Process

FL training process is an iterative and collaborative procedure involving a federated server (FS) and multiple clients (also referred to as EN or participants). The objective is to minimise a global loss function by aggregating locally computed updates from the EN. The generic FL training process can be described in four key steps as shown in Fig 2.1, which are repeated across multiple communication rounds until the model converges. Consider a federated system with a central server FS, and N ENs, index i where $i \in \{1,2,\ldots,N\}$, holds a distinct local dataset $\mathscr{D}_i$. These datasets are defined as $\mathscr{D}_i = \{(\mathbf{X}_{ij}, \mathbf{y}_{ij})\}_{j=1}^{\mathscr{D}_i}$, with $\mathscr{D}_i$ indicating the number of samples at the i-th EN. Each sample comprises a feature vector $\mathbf{X}_{ij}$ and the targeted class $\mathbf{y}_{ij}$.

The primary objective of the FL framework is to train a global model ω collaboratively by minimising the overall loss across all ENs, without sharing the raw data from each EN. This can be formulated as [4, 22]:

$$\min_{\omega \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^{N} \ell_i(\omega, \mathscr{D}_i), \tag{2.1}$$

where $\ell_i(\omega, \mathscr{D}_i)$ denotes the local loss function at EN i, and ω represents the global model parameters. For classification, it is a cross-entropy loss, while mean absolute error (MAE) is

employed for regression problems. The local loss function at EN i is defined as:

$$\ell_i(\omega, \mathscr{D}_i) = \frac{1}{\mathscr{D}_i} \sum_{j=1}^{\mathscr{D}_i} \left| \mathbf{y}_{ij} - \hat{\mathbf{y}}_{ij}(\omega) \right|, \tag{2.2}$$

where $\mathbf{y}_{ij}$ and $\hat{\mathbf{y}}_{ij}(\omega)$ are the actual and targeted variables, respectively.

FL Training Steps:

FL model training is an iterative process that involves several communication rounds between the FS and ENs. The entire process is divided into four steps.
Step 1: The FS initialises the global model parameters ω^0 and distributes them to all clients. Mathematically, this process is represented as:

$$\omega^0 \rightarrow \text{Clients } \{1, 2, \ldots, N\}. \tag{2.3}$$

Step 2: Each EN i receives a copy of the global model ω^t at round t and performs local training using its dataset $\mathscr{D}_i$. The client updates the model parameters by minimising the local loss function ℓ_i, typically using stochastic gradient descent (SGD) for E local epochs:

$$\omega_i^{t+1} = \omega_i^t - \eta \nabla \ell_i(\omega_i^t), \tag{2.4}$$

where η is the learning rate, and $\nabla \ell_i(\omega_i^t)$ is the gradient of the local loss function at client i.
Step 3: After local training, each EN sends its updated model parameters ω_i^{t+1} to the central server mathematically represented as:

$$\{\omega_1^{t+1}, \omega_2^{t+1}, \ldots, \omega_N^{t+1}\} \rightarrow \text{Server} \tag{2.5}$$

Step: 4 The FS aggregates the received local updates to form a new global model ω^{t+1}. A common aggregation method is federated averaging (FedAVG), which computes a weighted average of the client updates based on their dataset sizes [11]:

$$\omega^{t+1} = \sum_{i=1}^{N} \frac{|\mathscr{D}_i|}{\sum_{j=1}^{N} |\mathscr{D}_j|} \omega_i^{t+1}, \tag{2.6}$$

The server updates the global model with ω^{t+1} and distributes it to the clients for the next round:

$$\omega^{t+1} \rightarrow \text{Clients } \{1, 2, \ldots, N\}. \tag{2.7}$$

This process repeats for multiple rounds until the model converges or reaches a predefined performance threshold.

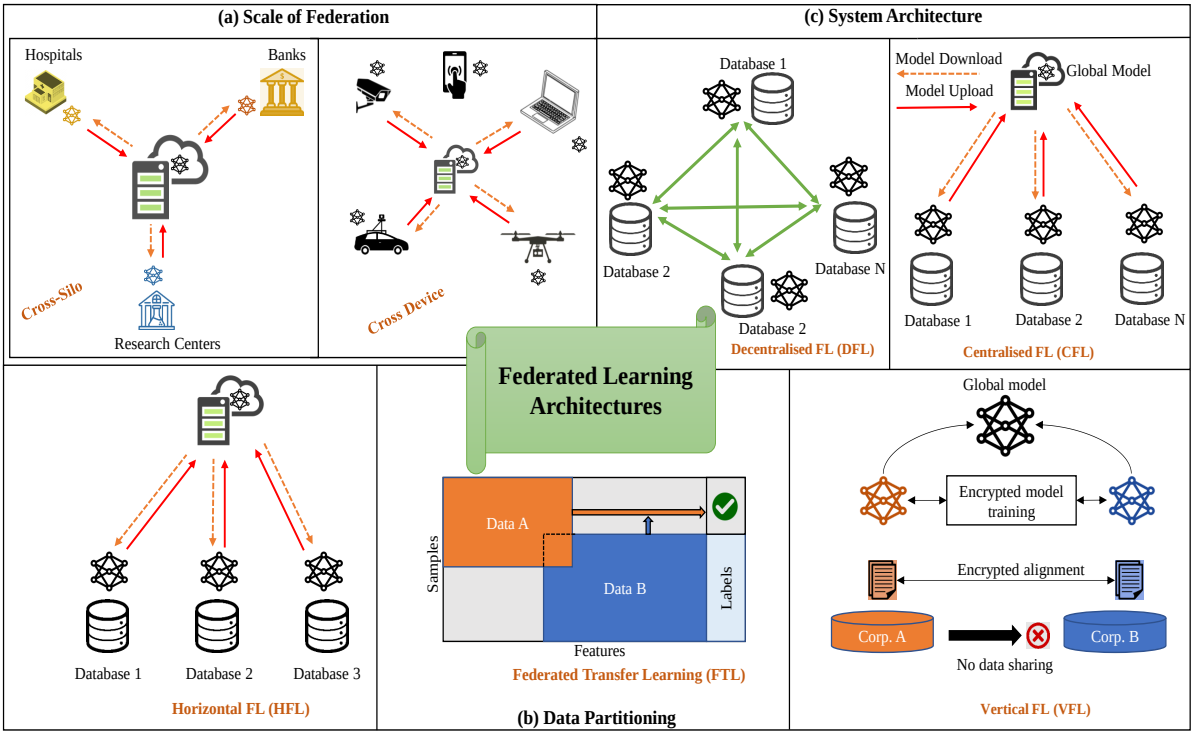


Figure 2.2: Classification of FL where (a) Scale of federation, (b) data partitioning, (c) system architecture.

2.1.2 Classification of FL

The recent advancement in FL can be categorised into three classes, which include (a) scale of federation or operational strategies, (b) data partitioning, and (c) system architecture [19].

Scale of Federation

- The scale of federation is highly dependent on the number of ENs participating in the training process. This involves a limited number of reliable and often organisational clients (e.g., hospitals, banks). Clients are stable, have significant computational resources, and are continuously available [19]. The cross-silo architecture is shown in Fig. 2.2 (a).

- Conversely, cross-device involves a large number of unreliable and heterogeneous devices (e.g., smartphones, IoT devices). Clients may have intermittent availability and limited computational power [15].

The typical examples of cross-device are Google Gboard [11], and cross-silo is NVIDA Clara [24] for brain tumour segmentation. In FL applications, the scale of federation is highly dependent on the nature of data and the user's intent. For instance, in brain tumour segmentation, the data is usually stored in silos placed in different geographical locations, and as a result, the cross-silo mechanism is used for model training.

Data Partitioning

Data partitioning plays a key role in FL, where it is broadly divided into horizontal, vertical and transfer learning, as shown in Fig. 2.2 (b) [15].

- **Horizontal Federated Learning (HFL)**: Clients share the same feature space but have different samples. This approach is suitable when datasets across clients have the same structure (e.g., mobile devices with similar data types) [15, 22]. The classic example of horizontal FL is Google Gboard, with the assumption of honest consumers and a secure, centralised server for global model training [11].
- **Vertical Federated Learning (VFL)**: Clients have different feature spaces but share the same sample IDs. This approach is applicable when institutions hold different information about the same set of individuals (e.g., different companies holding complementary customer data) [5, 25]. The real-world use case for vertical FL may be a scenario where the credit card sales team of a bank trains its ML model by using information from online shopping. In this case, only everyday bank and e-commerce website users will participate in the training process. With this liaising of secure information exchange, banks can improve their credit services and provide incentives to active customers [25].
- **Federated Transfer Learning (FTL)**: The transfer FL approach uses a pre-trained model on a similar dataset to solve a completely new problem set. The real-time example of transfer FL could be similar to vertical FL with small modifications. In this approach, the condition of similar users with matching data for model training can be relaxed to create a diverse system to serve individual customers [26, 27]. It is a personalised model training for individual users to exploit the better generalisation properties of the global model, which can be achieved by either data interpolation, model interpolation, and user clustering [3].

Classification based on System Architecture

Based on system architecture, FL is divided into centralised and decentralised learning.

- **Centralised FL (CFL)**: Features a central server that orchestrates the training process, aggregates local model updates, and redistributes the global model. This architecture simplifies coordination but introduces a single point of failure and potential privacy risks if the server is compromised [19].
- **Decentralised FL (DFL)**: Eliminates the need for a central server by allowing clients to communicate directly with each other in a peer-to-peer network [28, 29]. This architecture enhances robustness and reduces centralisation risks but may face challenges in synchronisation and increased communication overhead.

2.2 Performance and Energy Metrics in FL

To rigorously evaluate the performance of both regression and classification models within this research, we employ widely adopted metrics to ensure a comprehensive assessment of predictive capabilities across different problem settings.

2.2.1 Regression Models

For regression-based tasks, such as short-term load forecasting (STLF), mean absolute error (MAE) and mean absolute percentage error (MAPE) are used extensively. MAE captures absolute deviations without regard to scale, while MAPE normalizes the errors relative to actual values, thus providing interpretability in percentage terms. Mathematically, these metrics are expressed as follows [30]:

$$\text{MAE} = \frac{1}{T}\sum_{t=1}^{T}|A_t - F_t|, \tag{2.8}$$

$$\text{MAPE} = \frac{1}{T}\sum_{t=1}^{T}\left|\frac{A_t - F_t}{A_t}\right| \times 100 \tag{2.9}$$

where A_t denotes the actual value, F_t represents the predicted value, and T is the total number of samples.

2.2.2 Classification Metrics

In classification scenarios such as blockage prediction (binary classification) and human activity recognition (HAR; multi-class classification), accuracy alone might not be sufficient, particularly in datasets exhibiting class imbalance. Thus, additional metrics such as Precision, Recall, and F1-score are employed.

- **Accuracy:** It indicates the proportion of correct predictions among the total predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.10}$$

  where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

- **Precision:** This metric evaluates the accuracy of positive predictions, given by the formula:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{2.11}$$

where TP represents true positives and FP denotes false positives.

- **Recall (sensitivity):** This measure assesses the model's ability to identify all relevant instances, defined as:
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{2.12}$$
with FN indicating false negatives.

- **F1-score:** Offering a balance between precision and recall, this metric is particularly valuable in the presence of class imbalance. It is calculated as:
$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{2.13}$$

Collectively, these metrics facilitate a balanced and insightful performance evaluation, ensuring robust assessments across diverse problem contexts.

2.2.3 Computational Efficiency

In addition to evaluating performance using error and accuracy metrics, assessing computational efficiency is crucial to determine the scalability and practicality of the proposed frameworks, especially for deployment in resource-constrained environments. Computational complexity is commonly quantified using Floating Point Operations (FLOPs), which measure the number of arithmetic operations required by an algorithm or model during each computation or communication round. The total computational complexity per communication round can be generically expressed as:

$$\text{FLOPs}_{\text{Total}} = N \cdot F_{\text{Local}} + F_{\text{Agg}} + F_{\text{Algo}}, \tag{2.14}$$

where N denotes the number of participating ENs, F_{Local} is the computational cost associated with local training at each EN, F_{Agg} represents the aggregation computational overhead at the central server, and F_{Algo} corresponds to any additional computational overhead specific to the aggregation algorithm employed, such as similarity computation or truncation mechanisms.

Local training cost, F_{Local}, involves both forward and backward passes through the neural network model, with complexity depending on model architecture, including layer type, neuron count, and training sample size. For instance, in the case of recurrent architectures such as LSTM, the local computational complexity is proportional to the number of hidden units, layers, and the size of training samples processed at each EN. Formally, this can be represented as:

$$F_{\text{Local}} \propto S \cdot H \cdot L, \tag{2.15}$$

where S represents the training sample size per EN, H denotes the number of hidden units per layer, and L indicates the total number of layers in the LSTM model.

Aggregation cost, F_{Agg}, involves the computational complexity of combining local updates at the server and is generally dependent on the number of parameters in the neural network model and the number of participating edge nodes. Mathematically, it can be expressed as:

$$F_{\text{Agg}} \propto K \cdot M, \tag{2.16}$$

where M denotes the total number of model parameters.

Algorithm-specific computational overhead, F_{Algo}, varies significantly depending on the aggregation approach utilised. For example, conventional aggregation methods, such as FedAVG, primarily involve simple averaging operations, resulting in relatively low computational overhead. Conversely, advanced aggregation algorithms, such as those involving similarity-driven mechanisms or truncation strategies, entail additional operations like cosine similarity calculations, sorting, or thresholding operations, thus introducing an extra computational cost. Specifically, for L layers and N participating clients, the complexity introduced by similarity-driven mechanisms can be approximated as:

$$F_{\text{Algo}} \propto L \cdot N \cdot M. \tag{2.17}$$

Thus, comprehensive evaluation of FLOPs provides a practical measure for understanding model scalability, energy consumption, and suitability for real-world, resource-limited deployments.

In addition, we have expanded our evaluation criteria to include energy efficiency, which is a crucial factor in FL where computational and communication resources are limited. The energy efficiency metric is based on the computational requirements of local training and the amount of data transmitted during each communication round, which is represented by the following formula: [31, 32]

$$E_{est} = R[(\alpha * t_{com}) + N(\beta * P_{trn})], \tag{2.18}$$

where α is the computation constant having dimensions of energy per second and β is the communication constant with dimensions of energy per kilobyte. R represents the number of communication rounds; N is the number of participants; t_{com} is the computation time, which is dependent on the device type; and P_{trn} is the data payload size per communication round.

2.3 Heterogeneity and Data Diversity in FL

Heterogeneity is a significant challenge in FL, arising from diverse data distributions, varying computational capabilities, and network conditions across the ENs. Unlike traditional centralised learning, FL involves training on decentralised, non-independent and identically distributed (non-IID) data from multiple sources, where data from each EN represents unique local patterns or biases due to different environments, user behaviours, and data collection meth-

ods [33]. The data heterogeneity leads to suboptimal performance, where the global model needs help to generalise effectively across all clients. Additionally, system heterogeneity, caused by disparities in hardware, network stability, and computational power among ENs introduces further complexities, as slower or less capable devices hinder the training process [34]. Addressing heterogeneity in FL requires specialised aggregation techniques, robust optimisation algorithms, and strategies for managing resource-constrained clients to ensure both model accuracy and fairness across all participants. This section focuses on data diversity, related challenges, and methods to overcome these challenges. The data diversity is typically classified into statistical heterogeneity and domain shift [35].

2.3.1 Statistical Heterogeneity and Client Drift

A primary challenge in FL is statistical heterogeneity, where clients collect data with unique distributions and characteristics, resulting in non-IID datasets [36, 37]. For instance, devices in distinct environments capture data that varies regarding feature distributions, label proportions, and data imbalance. This diversity leads to client drift, where local models diverge from each other during training, creating a challenge for the global model to converge effectively [38]. This client drift, where local model updates deviate significantly from the global model, leads to unstable convergence and often deteriorates model performance. Furthermore, the inconsistency in data distribution results in suboptimal global models that struggle to adapt to new data distributions, limiting their effectiveness and accuracy across diverse client populations [39].

2.3.2 Domain Shift

Domain shift is another critical aspect of data heterogeneity, where clients capture data from different domains or under varying conditions. This issue is common in applications like healthcare, where medical institutions collect patient data with vastly different characteristics due to regional and demographic factors [38]. Similarly, in industrial applications, two robots operating in separate geographical locations may encounter different environmental conditions, leading to domain-specific feature distributions. Such domain shifts can hinder model performance as the global model struggles to accommodate diverse contexts, reducing adaptability and accuracy across different clients [40].

2.3.3 Challenges Arising from Data Diversity in FL

- **Slow convergence and model instability:** Non-IID data often cause each model to diverge during local training, leading to difficulties in achieving stable convergence of the global model. The reliance on stochastic gradient descent (SGD) exacerbates this issue, as gradients from clients with dissimilar data conflicts slow down convergence and cause

oscillations [41].

- **Client-specific performance variability:** The global model exhibits inconsistent performance across ENs due to data diversity. Clients with limited or unrepresentative data negatively influence the global model, reducing overall model accuracy [42]. Furthermore, clients with highly skewed data distributions cause the global model to overfit specific patterns, compromising generalisation.

- **Increased computational complexity:** Handling data diversity often demands additional computational resources, especially for clients with constrained capabilities. Strategies to mitigate heterogeneity involve added operations for client clustering, data normalisation, or model adaptation, further straining devices with limited processing power [43].

2.3.4 Techniques to Address Data Diversity in FL

Several techniques have been proposed to overcome data diversity challenges, focusing on adapting the FL process to handle heterogeneous data effectively. Clustering-based methods group clients with similar data characteristics, allowing for more targeted model aggregation within clusters. This approach minimises the impact of non-IID data by aligning the model updates within each group, enhancing training performance and reducing convergence issues [44, 45].

Another approach is personalised FL, which customises global models for individual clients through transfer learning and meta-learning [46, 47]. By tailoring models to specific data distributions, personalised FL improves local model performance and mitigates the adverse effects of data heterogeneity on convergence. Additionally, normalisation techniques have been introduced to adjust local model updates before aggregation, addressing disparities in model contributions and reducing gradient scattering [48].

Domain adaptation techniques have also been explored to handle domain shifts by aligning feature distributions across domains. These methods allow the global model to generalise better by minimising the divergence between domain-specific and global feature representations [49]. Finally, some studies propose sharing a small public dataset across clients to create a common data representation, thus bridging the distribution gap between clients [50]. This shared dataset aids in stabilising the global model and improving its robustness in diverse settings. In conclusion, addressing data diversity in FL is essential to enhance model convergence, performance, and adaptability. Clustering, personalisation, domain adaptation, and normalisation techniques provide viable solutions, each tailored to specific aspects of heterogeneity, allowing for more robust and effective FL models.

2.4 Multi-Modal Fusion in FL

Multi-modality in FL aims to leverage diverse data sources (e.g., text, image, audio) in a decentralised setting to improve model performance while preserving data privacy [51]. Multi-modality is valuable, especially for applications in healthcare, wireless communication systems, autonomous systems, and smart cities, where integrating multi-modal data sources can improve the robustness of predictions [52]. However, multi-modal fusion introduces several complex challenges, primarily due to the inherent diversity in data distribution, different formats, and feature representation across different modalities [53].

Data heterogeneity, modality alignment, and communication efficacy are among the primary challenges in multi-modal fusion [51–53]. Unlike conventional unimodal FL, which typically shares a similar structure with client data, multi-modal fusion needs to resolve inconsistencies that vary in data characteristics and processing requirements. This heterogeneity can impede model convergence and diminish performance if not effectively managed [54]. This complexity is frequently observed in the healthcare sector, where the structure and relevance of various modalities, such as clinical text, imaging, and lab results, necessitate meticulous integration to generate meaningful insights [55].

Key challenges in multi-modal fusion for FL include:

- **Data Heterogeneity:** Different data types (e.g., images vs. text) have varying feature representations, making it challenging to align and aggregate them in a unified model. This heterogeneity can lead to client drift and impede model convergence if fusion strategies do not account for these differences [54].

- **Modality Alignment:** Another intricate issue is consistency maintenance across multi-modalities, particularly when specific clients possess only partial data. Sophisticated alignment techniques are necessary to align modalities while preserving their individual contributions to the global model [55].

- **Communication Efficiency:** The computational requirements and data size are typically increased by multi-modal fusion, which results in significant communication overhead.

Several approaches have emerged to address these challenges faced by multi-modal data fusion in FL. Modality-specific training is one widely used technique where client devices train models on individual modalities and then aggregate latent representations in a shared feature space [54]. For instance, attention mechanisms are commonly applied to capture complementary information across modalities, allowing models to selectively integrate relevant features from each modality [56]. Another approach involves hierarchical fusion architectures, where early fusion integrates raw modality data before local training. In contrast, late fusion combines predictions from each modality, enhancing alignment and preserving the individual strengths of each data type [57].

To further address heterogeneity, dynamic modality fusion and similarity-based aggregation techniques have been proposed [58]. These methods enable the model to prioritise modalities with the most informative representations, reducing noise and improving convergence. Privacy-preserving techniques, such as homomorphic encryption (HE) and differential privacy (DP), are integrated to protect sensitive multi-modal data during transmission. Recent studies have also explored autoencoder and graph neural network-based methods for modality alignment, which help create a common latent space for data integration, thus improving the overall performance and robustness of the global model [59].

2.5 Aggregation Mechanism in FL

In FL, aggregation mechanisms are central to the collaborative training process, enabling the integration of model updates from multiple clients to produce an optimal global model. The choice of aggregation method can significantly affect model performance, especially when dealing with challenges like non-IID data, system heterogeneity, and limited computational resources on ENs [60]. This subsection examines the most commonly used aggregation techniques, including gradient descent (GD), SGD, FedAVG, federated matched averaging (FedMA), federated proximal (FedProx), and federated splitting (FedSplit) [34, 60, 61].

GD is one of the foundational methods used in centralised ML. It involves iteratively adjusting model parameters by descending along the gradient of the loss function, gradually minimising the difference between predicted and actual values [62]. While effective, GD can be computationally intensive when applied directly in FL, requiring each client to compute the gradient on its entire dataset. SGD is a variant of GD that reduces computational load by using a randomly selected subset of data, known as a mini-batch, to approximate the gradient. In FL, SGD is commonly used in local training on each client, as it strikes a balance between computational efficiency and accuracy, making it suitable for large-scale, resource-constrained settings [33]. In the FL setting, Federated (FedSGD) is exploited to quantify how frequently the global FL model needs to be updated [11]. Despite its advantages, the FedSGD requires large communication rounds proportionate to the data volume, which can strain communication links and consume unnecessary bandwidth.

To address the challenge of frequent updates, McMahan *et al.* [12], introduced FedAVG, which computes a weighted average of model updates from all participating ENs where each EN is weighted based on the size of its dataset. Mathematically, this is expressed as:

$$\omega^{t+1} = \frac{1}{\sum_{i=1}^{N} |\mathscr{D}_i|} \sum_{i=1}^{N} |\mathscr{D}_i| \cdot \omega_i^{t+1}, \tag{2.19}$$

where ω^{t+1} represents the global model at round $t+1$, N is the number of clients, $\mathscr{D}_i$ is the dataset of client i, and ω_i^{t+1} is the locally updated model from client i. FedAVG is simple, com-

putationally efficient, and well-suited to scenarios with IID data across multiple ENs. However, it faces challenges with non-IID data distributions, as averaging non-uniform updates can lead to suboptimal model convergence and performance [63]. Moreover, FedAVG can be sensitive to client dropout, which may distort the global model if certain clients disproportionately influence the aggregation process.

Multiple variants of FedAVG have been introduced based on the customised requirements to address the limitations. For instance, FedProx was introduced as a modification that includes a proximal term in the local objective function to stabilise updates across clients with varying data distributions and computational capacities [64]. In FedProx, the optimisation problem of each EN is augmented with a penalty term that keeps the local model close to the global model parameters, reducing the drift caused by clients with highly skewed data. This method is particularly beneficial in heterogeneous networks, where client data and resources vary significantly, enabling more robust model convergence. The local objective function for client i in FedProx is given by [64]:

$$\ell_i(\omega) + \frac{\mu}{2}\|\omega - \omega^t\|^2, \tag{2.20}$$

where μ is a regularisation parameter that controls the strength of the proximal term. The proximal term $\|\omega - \omega^t\|^2$ ensures that local updates do not diverge too drastically from the global model, helping to align client updates even under non-IID conditions.

FedMA, another extension of FedAVG, handles heterogeneous model architectures across ENs [65]. FedMA aligns model layers from different EN before averaging, thus enabling aggregation in scenarios where client models have varying architectures. This approach begins by identifying correspondences between layers of each EN's model and then averaging matched parameters. FedMA is particularly useful in applications where each EN uses different model architectures, as it allows for flexibility while still achieving a meaningful global update. However, FedMA can be computationally complex due to the need for matching and alignment before averaging, making it more suitable for federated systems with moderate levels of heterogeneity.

Another promising method for model aggregation is FedSplit, which divides the model parameters into subsets and assigns them to different ENs based on data relevance or computational capabilities [66]. FedSplit enables selective participation, where only relevant ENs update specific model subsets. This approach reduces the computational burden on individual clients, as they are responsible only for particular segments of the model rather than the entire parameter set. By distributing model components across clients, FedSplit facilitates parallel processing, accelerating model convergence [66]. However, it requires careful coordination to ensure that all parts of the model are updated in a synchronised manner, and it may increase communication costs if extensive coordination is needed among clients.

Beyond these primary methods, several other aggregation techniques have been explored to address specific challenges in FL. For example, federated gradient sparsification (FGS) [67], federated distance (FedDist) [68], federated meta-learning (FedMeta) [69], and federated boost-

ing (FedBoost) [70] worked to reduce the computational complexity and communication cost.

In summary, aggregation mechanisms in FL have evolved to meet the diverse needs of distributed systems. While FedAVG remains the most commonly used technique due to its simplicity and efficiency, methods like FedProx and FedMA have been developed to address data heterogeneity and client variability challenges. Advanced methods, such as FedSplit and gradient sparsification, provide further flexibility, catering to applications with unique communication, computational, and privacy constraints. The selection of an aggregation mechanism depends heavily on the specific application requirements and EN characteristics, as well as on factors like data distribution, client reliability, and resource limitations.

2.6 Privacy and Security in FL

As FL grows in prominence as a method for decentralised model training, privacy and security emerge as two of its most crucial considerations. FL preserves data locally, enabling multiple EN to train collaboratively and offers privacy by design [17]. However, the decentralised nature of FL introduces unique privacy and security challenges, as model updates shared during training can inadvertently leak sensitive information by data reconstruction [16, 17]. Furthermore, FL is vulnerable to various adversarial attacks, including poisoning and inference attacks, which can compromise both data confidentiality and model integrity [71, 72].

This section explores the key privacy and security issues in FL, starting with a summary of privacy challenges and common objectives for protecting client data. Next, we explore the essential privacy-preserving techniques used in FL, such as DP, secure multi-party computation (SMPC), and homomorphic encryption (HE), each providing a unique approach to improving data confidentiality. The discussion then shifts to the security landscape of FL, exploring potential adversarial threats such as poisoning and inference attacks, which can compromise model integrity and privacy. Finally, we review existing defence mechanisms designed to counter these threats, highlighting advancements that strengthen the robustness of FL against attacks while maintaining model performance.

2.6.1 Vulnerabilities in FL

While FL offers privacy by design, it remains vulnerable to various adversarial attacks that can compromise the performance, integrity, or even the privacy of individual data points. From a broader perspective, the adversarial attacks in FL can be classified into two main categories, i.e., performance attacks and privacy attacks [17, 28]. Performance attacks aim to degrade model accuracy or robustness, while privacy attacks seek to infer sensitive information from model updates. Understanding these attack types and their operational mechanisms is crucial for implementing effective defences in FL systems.

Performance Attacks

Performance attacks primarily target the predictive capabilities and generalisation of FL models by introducing biases, errors, or disruptions during training. Common types of performance attacks include data poisoning and model poisoning.

1) Data Poisoning Attack: Data poisoning involves manipulating or corrupting the local training data to introduce erroneous patterns into the model. In FL, adversaries can disguise themselves as legitimate clients and insert manipulated data into their training sets to influence the global model. One typical example is label flipping, where specific class labels are deliberately misassigned (e.g., changing all instances of "cat" to "dog") to distort the model's predictions. This type of attack biases the model without altering its structure, making detection challenging in a decentralised setting [73, 74].

Another form of data poisoning is backdoor attacks, which subtly insert specific patterns into the data (e.g., a particular pixel patch in images) that trigger incorrect predictions only when the pattern appears [75]. For example, in a facial recognition model, attackers could add a small patch to an image to make the model misclassify specific individuals. Data poisoning, whether through label flipping or backdoors, can significantly degrade model performance, especially when multiple compromised clients participate in the training process [76].

2) Model Poisoning: Model poisoning targets the integrity of the global model by directly manipulating model updates rather than altering the data itself. In this attack, malicious clients intentionally modify the gradients they send to the server [77]. By sending poisoned gradients, adversaries can skew the global model towards incorrect patterns or objectives. Model poisoning can be targeted where only specific outcomes are affected or untargeted, impacting the model's overall accuracy [5].

In targeted model poisoning, an attacker may use the same data as legitimate clients but alter the objective function to inject specific biases, effectively compromising the model for particular tasks. Untargeted attacks, on the other hand, aim to degrade the global model's performance across all tasks, often by injecting noise or deviating gradients. Model poisoning is especially challenging to detect, as malicious clients can craft gradients that closely resemble legitimate updates, bypassing standard anomaly detection techniques [73, 74].

Privacy Attacks

Privacy attacks in FL aim to extract sensitive information about the local data from the shared model updates. Common types of privacy attacks include model inversion, membership inference, and gradient leakage [19].

1) Model Inversion Attack Model inversion attacks allow adversaries to reconstruct private data based on shared model parameters. In FL, an attacker accessing the global model's gradients or parameters can reverse-engineer data patterns, approximating the original data points [78]. Model inversion can be done with generative adversarial networks (GANs), which

train a generator network to make data samples that look like they came from the targeted client while using the global model as a discriminator [79]. Model inversion threatens privacy, especially in applications handling sensitive information like medical records or financial data.

2) Membership Inference Attacks: Membership inference attacks aim to determine whether a specific data point was part of a client's training dataset by analysing the updates that clients contribute. This type of attack allows adversaries to infer details about individual data points or their characteristics. For instance, by observing clients' gradient updates, an attacker can identify if a particular data sample influenced the model, thus revealing membership information [80]. Membership inference attacks are particularly concerning in applications with sensitive dataset membership, such as healthcare or law enforcement.

3) Gradient Leakage and Gradient Inference Attacks: In gradient leakage attacks, adversaries exploit gradients shared during FL training to infer sensitive details about the data. Gradients can contain information about the direction and magnitude of data changes, which attackers can use to approximate original data points. For example, attackers may leverage gradient information to perform gradient inversion, reconstructing partial or complete data samples from the updates shared by clients [80]. This risk is especially high in FL systems with fewer clients or susceptible data. Implementing privacy-preserving mechanisms such as gradient noise is essential to mitigate leakage risks.

2.6.2 Privacy Challenges and Goals in FL

The impact and likelihood of different types of attacks depend on the resources, knowledge, and access that the attacker has to the system. Attacks like backdoors and model poisoning require a deep understanding and access to the architecture of the FL system. In contrast, membership inference and gradient leakage attacks can be executed more efficiently with minimal resources.

One of the primary privacy risks in FL stems from information leakage through model updates. Despite EN only sharing model parameters, research has shown that an attacker with access to these updates can reconstruct aspects of the local data used to generate them [16, 71, 72]. For instance, gradient inversion techniques enable adversaries to infer data characteristics or approximate data values from the gradient updates, posing a significant privacy threat in sensitive applications like healthcare or financial services [81]. Additionally, differential updates between training rounds can reveal client-specific data patterns over time, enabling attackers to reconstruct details about local data more accurately. This makes FL vulnerable to sophisticated reconstruction attacks, where long-term exposure to model updates can accumulate into a significant privacy breach [20].

Another critical privacy issue in FL involves membership inference attacks, where adversaries attempt to determine whether specific data points were part of a client's dataset. These attacks take advantage of vulnerabilities in model updates to differentiate between data that the model has seen and data that it has not. This allows attackers to verify the presence of spe-

Table 2.1: Summary of Attack Types in FL.

Type of Attack	Source of Attack	Target of Attack	Complexity	Impact	Effectiveness	Detection Challenge
Data Poisoning	Malicious clients	Global model performance	Medium (requires data access)	Biases model predictions, decreases accuracy	High when multiple clients are compromised	Difficult (label flipping or backdoors subtle)
Model Poisoning	Malicious clients	Global model parameters	High (requires gradient manipulation)	Compromises model integrity, affects specific or all outcomes	High (both targeted and untargeted attacks effective)	High (manipulated gradients resemble legitimate updates)
Model Inversion	Adversary accessing model parameters	Reconstruction of private data points	High (requires GANs or advanced reconstruction)	Reveals sensitive data, high privacy risk	High in sensitive applications	High (reconstruction techniques hard to detect)
Membership Inference	Adversary observing gradient updates	Individual data inclusion	Low (easily inferred from updates)	Exposes individual data points, privacy breach	Moderate to High (depending on application)	Moderate (privacy-preserving techniques help)
Gradient Leakage	Adversary analysing shared gradients	Sensitive data patterns	Medium (requires gradient analysis)	Reveals partial data, medium privacy risk	Moderate (more effective with fewer clients)	Moderate to High (privacy-preserving mechanisms needed)

cific records, such as medical information, within client datasets [19]. Membership inference is particularly concerning in applications where the mere presence of data can have sensitive implications, such as records indicating the diagnosis of a disease. While designed to prevent direct data sharing, the FL framework risks exposing sensitive information through such inference attacks due to its iterative update process.

Additionally, the privacy challenges in FL extend beyond individual clients to include risks associated with client-server trust dynamics. In typical FL setups, a central server aggregates model updates from clients, creating a potential single point of vulnerability. The central server could be compromised, either by external attackers or by collusion with malicious clients, leading to unauthorised access to aggregated updates and compromising client privacy [71]. The absence of a direct trust mechanism between clients and the central server increases this risk, as clients must depend on the server to ensure privacy without insight into its data aggregation or handling practices. This dynamic underscores the need for robust privacy guarantees that operate independently of server trustworthiness, particularly for sensitive data applications.

In terms of impact, targeted backdoor attacks are particularly effective as they compromise model predictions without changing the overall structure of the model. On the other hand, untargeted attacks, such as generic model poisoning, reduce the overall accuracy of the model but may be easier to detect. Privacy attacks, including model inversion and membership inference, can be highly effective, especially when data confidentiality is a primary concern. Table 2.1 summarise the targets and complexity of adversarial attacks in FL.

In light of the challenges discussed in above, privacy preservation in FL focuses on three main goals: (a) minimising information leakage, (b) ensuring data confidentiality, and (c) main-

taining the integrity of the learning process.

- The first goal, minimising information leakage, aims to restrict the amount of information inferred from shared model updates. Techniques like DP introduce controlled noise to model updates during training, limiting the potential for data reconstruction while retaining model utility [82].
- The second goal is data confidentiality, which emphasises protecting client data from unauthorised access, particularly during model aggregation on the server. SMPC and HE are commonly employed to enable computations on encrypted data, ensuring that the server only processes encrypted updates without accessing the underlying information [71].
- The final goal is the integrity of the learning process, which addresses the risk of malicious or compromised clients manipulating model updates to inject biased or misleading information into the global model. Robust aggregation techniques, such as anomaly detection and reweighting strategies, are critical to maintaining model reliability in adversarial settings.

2.6.3 Defense Mechanism in FL

Several privacy-preserving methods have recently been proposed to enhance the resilience of FL, each offering a unique balance between privacy and utility. Here, we discuss the primary privacy-preserving techniques, including HE, secure multi-party computation (SMPC), and DP, and advanced model robustness methods, such as anomaly detection, robust aggregation, and pruning.

Defense against Privacy Attacks

Given the variety of attacks in FL, it is imperative to deploy a range of defense frameworks tailored to the specific nature of each attack, considering factors such as device configurations, FL architecture, and available resources.

1) Homomorphic Encryption: HE enables computations on encrypted data without decrypting it, thus maintaining data privacy during model training. In FL, HE allows ENs to send encrypted updates to the central server, which can perform arithmetic operations on the encrypted values. HE has three primary variations based on its computational flexibility and complexity, including full, partial, and somewhat encryption [83, 84]. In full encryption, arbitrary computations are performed on ciphertexts, which offer maximal flexibility but are computationally intensive, often impractical for large-scale applications [85]. Partial encryption supports only one type of operation (e.g., addition or multiplication), while the somewhat HE allows a limited number of both operations [86]. Although HE enhances privacy, the added

computational and memory demands make it challenging to deploy efficiently in FL, particularly when handling non-linear operations required by complex models. Some FL frameworks, such as those using additively HE, aim to balance privacy and computational overhead, but the trade-offs remain a critical consideration in implementing HE [83].

2) Secure Multi-Party Computation: SMPC enables collaborative computations across multiple ENs without sharing individual data, enhancing privacy in distributed settings. In SMPC, each client divides its data into random shares, which are then distributed among other clients for local computation [87]. Once each client computes the function on the shared data, the results are aggregated to produce the desired outcome, ensuring that no single party can access complete information [88]. SMPC has been effectively used in FL to secure private model training. However, it typically incurs significant communication and computational costs due to its dependence on data-sharing among multiple clients [89]. While SMPC provides robust privacy, the high communication overhead makes it less efficient for large-scale FL applications, where a large number of clients participate. However, implementing SMPC on a large scale requires meticulous coordination and can be vulnerable to performance issues, particularly in resource-constrained environments [87].

3) Differential Privacy: DP is widely used in FL to protect against data inference attacks by introducing random noise to the model updates, making it difficult for attackers to derive specific information about individual data points [90]. DP is beneficial for safeguarding against both inference and data poisoning attacks, as it disrupts the underlying gradients that attackers may exploit [82, 90]. In most cases, DP is added to FL using methods such as the exponential noise mechanism or the Gaussian noise addition to gradients, hiding the private data before updates are sent [72]. While DP imposes a lower computational cost compared to HE and SMPC, it can affect model quality due to noise injection. Moreover, DP faces challenges related to cumulative privacy loss across iterative training rounds, as privacy degradation may increase with each successive iteration. Techniques such as privacy amplification by subsampling and controlled noise injection have been proposed to mitigate this issue, providing more stable privacy guarantees over multiple rounds [91].

In FL, DP can be applied at different levels, such as user-level privacy, which protects the presence of an entire dataset, or instance-level privacy, which focuses on securing individual records. User-level DP ensures that the model remains secure even if individual clients or their data are removed from the training process, while instance-level DP provides a more granular protection level, securing each data point within the client's dataset [91]. In summary, HE, SMPC, and DP each bring unique strengths to privacy preservation in FL; they also involve trade-offs between privacy, computational cost, and model utility. The choice of technique often depends on the specific privacy requirements and resource constraints of the FL deployment.

4) Knowledge Distillation: Knowledge distillation (KD) is a method by which knowledge is transmitted from a larger, well-trained model "teacher" to a smaller, more lightweight model, the

"student", by training the student to approximate the teacher's predictions [92]. This method is useful for developing lightweight, efficient models without sacrificing accuracy, making it applicable to a wide range of applications. KD is crucial in the FL as it enables the direct deployment of efficient student models on decentralised devices, thereby enhancing model performance [92].

Instead of sharing entire, huge mentor models, the federated knowledge distillation (FedKD) approach [93], focuses on sharing updates from smaller "mentee" models, which promotes privacy protection. FedKD reduces privacy threats by sharing the most critical updates, minimising the amount of information exchanged. Furthermore, FedKD utilises gradient encryption before transmitting local gradients to the central server, thereby obscuring sensitive patterns included within the gradients [93, 94]. Methods like singular value decomposition (SVD) are employed for gradient compression, reducing sensitive information leakage [19]. In the convergence phase, FedKD progressively enhances compression precision, as gradients possess diminished private information at this level. The integrated methodology of FedKD offers strong privacy protections while preserving superior model performance.

5) Trusted execution environments (TEEs): are secure enclaves on client devices or servers that safeguard sensitive computations and data against unauthorised access and manipulation [17]. In FL, TEEs isolate the model training processes, thus ensuring the confidentiality and integrity of client data against adversaries seeking to intercept or modify updates transferred between clients and the central server [19]. This secure environment fosters trust by guaranteeing that individual contributions remain confidential, thereby encouraging increased client engagement in FL. Additionally, the tamper-resistant features of TEEs in FL enhance training and aggregation by minimising the risk of attacks where adversaries might alter model updates.

For example, IntelSGX proposed in [95] is a privacy-preserving FL framework that guarantees secure local training and global aggregation. By restricting computations to verified TEEs, IntelSGX uses remote attestation to verify integrity. This configuration detects and eliminates manipulated gradients, safeguarding the global model against harmful modifications. Another sophisticated architecture, privacy-preserving federated learning (PPFL) proposed in [96], utilises TEEs on both client and server sides to safeguard against privacy breaches. This framework implements layer-wise training to mitigate memory limitations, with just the final layer retained in the client's TEE to safeguard against membership inference attacks. Encrypted communication across TEEs guarantees data secrecy, significantly reducing risks associated with data reconstruction and property inference attacks [19]. Table. 2.2 provide the summary of defence mechanisms for privacy attacks with their characteristics and weaknesses.

Defense against Model and Data Attack

In addition to privacy-preserving methods, several defence mechanisms aim to enhance model robustness against adversarial manipulations in FL. These techniques focus on detecting and neutralising malicious updates, ensuring the reliability and stability of the global model. This

Table 2.2: Summary of Key Defense Mechanisms in FL for Privacy Attacks.

Defense Mechanism	Definition (Purpose)	Characteristics	Implementation Stages	Weakness	References
HE	Enables computations on encrypted data without decrypting it, maintaining data privacy during model training.	Variations include full, partial, and somewhat HE; supports computation on ciphertext; offers high security.	Server side (aggregation of updates)	Computationally intensive, particularly for non-linear operations, challenging for large-scale FL applications.	[83–86]
SMPC	Supports collaborative computations across clients without data sharing, enhancing privacy in distributed settings.	Uses data-sharing with random shares, ensuring no single party has complete data; robust privacy but high communication overhead.	Client side (local updates sharing)	High communication cost due to data-sharing requirements; performance issues in resource-constrained environments.	[87–89]
DP	Protects against inference attacks by introducing random noise, safeguarding sensitive information.	Different levels (user-level, instance-level); flexible noise injection; efficient in terms of computational cost.	Client and server sides	Potential impact on model utility due to noise; cumulative privacy loss over iterative rounds may reduce effectiveness.	[72, 82, 90, 91]
KD	Transfers knowledge from a larger model to a smaller one to enhance efficiency without sacrificing accuracy.	Creates lightweight student models; FedKD variant with gradient encryption and compression; high privacy protection.	Client side (local model training)	Limited in protecting against sophisticated reconstruction attacks; effectiveness varies with model and task complexity.	[19, 92–94]
TEEs	Provides secure enclaves on devices to protect sensitive computations from unauthorized access.	Secure, isolated environment for computations; supports remote attestation; strong defence against data tampering.	Both client and server sides	Limited by memory constraints; potential overhead with large models or datasets.	[19, 95, 96]

subsection will discuss the defence mechanism for the data and model attacks discussed in Section 2.6.1.

1) Anomaly Detection: Anomaly detection is pivotal in FL training against malicious updates that could degrade model integrity or introduce data poisoning. Typically implemented on the server side, anomaly detection monitors deviations of upcoming updates from normal patterns, identifying and flagging abnormal updates that indicate adversarial behaviour [97, 98]. Several frameworks demonstrate the breadth of anomaly detection strategies in FL. For instance, the local malicious factor (LoMar) employs a two-step strategy to protect against poisoning attacks [99]. Initially, it scores the "maliciousness" of each update by utilising kernel density estimation based on k-nearest neighbours. Subsequently, it applies a threshold to classify updates as either clean or malicious. By successfully removing malicious updates from the aggregation process, this method significantly increases the reliability of the global model. Similarly, FederatedReverse proposed in [100] employs outlier detection and reverse engineering to detect malware triggers in client updates. This approach entails the generation of reverse triggers for each label at the client level, which are subsequently centralised. Median-based filtering is employed to identify anomalies among these triggers. When such anomalies are identified, corrective actions are implemented to mitigate the backdoor effects while maintaining the accuracy of the model.

Another technique presented in [101] is a dynamic defence against Byzantine attacks (DDaBA), which employs the induced ordered weighted averaging (IOWA) operator to provide adaptive protection against Byzantine attacks. This operator assigns dynamic weights to client updates

based on the performance of the validation process. Moreover, the SecFedNIDS framework proposed in [102] incorporates layer-wise relevance propagation and gradient-based anomaly detection to secure intrusion detection systems, filtering both poisoned model updates and data samples. Additionally, frameworks such as BAFFLE [75] employ a client feedback loop to verify the integrity of the model during each training round, thereby ensuring that anomalous updates are identified based on discrepancies in the error rate. Other methods, such as Multi-KRUM [103] and FoolsGold [104], facilitate the detection of coordinated malicious activities without compromising performance by addressing the identification of Sybil groups and gradient variances.

2) Robust Aggregation: The robust aggregation methods usually mitigate the effects of malicious updates, handle data heterogeneity, and optimise model performance. The foundational approach in FL is the FedAVG algorithm [12], where the server aggregates client updates by computing a weighted average, typically based on the number of data samples each client possesses. While simple and effective under ideal conditions, FedAVG is vulnerable to outlier and adversarial updates. Hence, multiple aggregation techniques like FedMA, FedProx, FedSplit, FedMeta, and FedDist [34, 60, 61, 65, 68] are proposed in the literature. These techniques use model updates and modify the aggregation mechanism to deal with vulnerabilities and adversarial updates. The detail of these model aggregation techniques is discussed in Section 2.5.

Apart from these traditional methods, various other techniques have also been proposed. For instance, the author in [103] proposed Kurm, which selects a single client update that is closest to the majority of other updates in terms of Euclidean distance. Krum reduces the influence of outliers by choosing the update with minimal cumulative distance to its nearest neighbours. Additionally, trim mean aggregation and clustering approaches are actively used in literature to mitigate the impact of adversarial updates. Clustering-based methods group client updates based on similarity, which helps to isolate outliers and aggregate only those representative of typical client data. Techniques like similarity-based aggregation calculate cosine similarity scores to form clusters of similar updates, reducing the influence of adversarial noise [35].

Advanced clustering approaches, such as Hierarchical Clustering Aggregation (HCA), extend this concept by iteratively grouping similar updates into hierarchical clusters [105]. The hierarchical structure enables the aggregation mechanism to preserve relationships within client updates while excluding clusters that exhibit high variance or dissimilarity. HCA is beneficial in dynamic FL environments, where clients may have drastically different data distributions or resource constraints, as it helps form a more nuanced and resilient aggregation process. Moreover, Byzantine-resilient methods are specifically designed to counteract malicious clients that aim to disrupt the FL process. Bulyan aggregation, proposed in [18], is one such method that integrates Krum with the coordinate-wise majority voting to tolerate Byzantine faults. This approach identifies and discards updates that deviate significantly from most updates, focusing on updates that maintain model accuracy under adversarial conditions.

Robust client selection frameworks minimise the risk of aggregating malicious or faulty updates by choosing only trustworthy clients in each training round. Techniques like FedClean [106] use reputation-based scoring to select highly trustworthy clients for model aggregation. By continually evaluating client performance, FedClean reduces the impact of unreliable clients on the global model, effectively balancing accuracy and security. Similarly, sampling mechanisms like diverse federated learning (DivFL) select a subset of clients that best represent the diversity of data distributions within the FL environment [107]. This submodular optimisation strategy ensures that the aggregated model reflects a broader range of data, improving its generalisation across heterogeneous clients while reducing reliance on any single client's updates.

3) Model Pruning: Model pruning is a method that is employed to improve the efficiency of neural networks by eliminating redundant or insignificant parameters, such as weights, neurones, or entire layers, without substantially affecting the performance [108]. This process involves the identification and elimination of parameters that have a negligible impact on the output of the neural network, thereby reducing the computational complexity and size of the model [109]. Pruning is classified into two primary categories:

- **Structured Pruning:** This approach eliminates entire network structures, including neurones, filters, and layers [109]. Structured pruning results in more efficient computation and memory usage by simplifying the network architecture. Parallel processing and hardware acceleration can be more readily optimised with the simplified network.

- **Unstructured Pruning:** This method removes the least significant weights by focusing on individual weights and utilising criteria such as magnitude [109]. Although unstructured pruning can generate networks with a high degree of sparsity, it frequently necessitates specialised hardware or libraries to completely capitalise on the sparsity as a result of the irregular remaining network structure.

Pruning is a widely used technique in FL that can considerably enhance the defence of local models by reducing the attack surface and increasing robustness [110]. Pruning mitigates the effects of adversarial manipulations by eradicating redundant or less significant parameters. This decrease in model complexity complicates the process of attackers injecting malicious updates without detection. Furthermore, pruning enhances the generalisation capacity of local models, thereby reducing their susceptibility to minor perturbations introduced by assaults [111]. In addition to reducing communication overhead during parameter sharing, pruned models facilitate quicker and more secure aggregation processes in FL due to their reduced number of parameters. Pruning can be implemented on both the server and client platforms, improving the model's security and efficiency throughout the federated network.

4) Regularisation: Regularisation functions as a strategic defense mechanism in FL, ensuring that the server and ENs are protected from data and model poisoning attacks. On the server side, regularisation techniques incorporate penalty terms in the loss function during model ag-

Table 2.3: Summary of Generic Defense Mechanisms in FL for data and model attack.

Defense Mechanism	Definition	Characteristics	Implementation Stage	Weaknesses	References
Anomaly Detection	Identifies and removes malicious updates by flagging deviations from expected patterns.	Detects abnormal updates, flags suspicious patterns, and often relies on statistical or machine learning-based anomaly detection techniques.	Server-side, often with client feedback to verify integrity.	May struggle with subtle adversarial changes and high computational costs.	[75, 97–103]
Robust Aggregation	Aggregates client updates while mitigating the impact of malicious data and model poisoning.	Uses techniques to average or cluster updates aims to minimise adversarial impact and improve aggregation resilience.	Server-side, during aggregation.	Vulnerable to sophisticated attacks if malicious clients not excluded, possible loss of useful updates.	[12, 34, 60, 61, 65]
Model Pruning	Enhances model efficiency by eliminating redundant or insignificant parameters, such as weights or neurons.	Reduces model complexity, improves computational efficiency, lowers communication overhead, and increases model robustness.	Server- or client-side; during training or before aggregation.	Specialized hardware may be needed for unstructured pruning, risk of accuracy loss.	[108–111]
Regularisation	Adds penalty terms to loss functions to prevent overfitting and enhance model robustness.	Controls model complexity, enhances generalisation and increases model robustness against adversarial manipulations.	Applied on both server and client sides during model training and aggregation.	Potential trade-off in model accuracy may require fine-tuning to avoid underfitting.	[19, 112–114]

gregation. This control of the complexity promotes robust generalisation across diverse client data [19]. This mitigates the risk of overfitting to specific client updates and enhances the resilience of the aggregated global model against adversarial influences.

On the client side, local training incorporates regularisation methods such as weight decay, batch normalisation, and dropout to improve the robustness of individual models further [112]. For example, dropout mitigates the probability of any single neuron becoming excessively dominant, enhancing the model's resilience to adversarial modifications. On the other hand, weight decay penalises heavier weights, thereby mitigating the potential effects of smaller adversarial perturbations to stabilising the model [113].

For instance, the author in [113] proposed local self-regularisation (LSR). This framework is intended to overcome the obstacles presented by noise-labelled data in FL while maintaining data privacy. LSR integrates two types of regularisation: implicit and explicit. Implicit regularisation enhances the model's resilience and confidence in label noise by employing MixUp, which merges data points. In contrast, explicit regularisation employs self-knowledge distillation to align the outputs of augmented and original samples, thereby preventing overfitting on noise labels. Similarly, the authors in [114] presented contractible regularisation (ConTre) as a solution to the non-IID data challenges in FL, emphasising image classification. ConTre improves the adaptability and stability of FL frameworks by progressively decreasing the regularisation effect as the model approaches convergence. Table. 2.3 summarises the characteristics and weaknesses of federated defense mechanisms of data and model attacks.

2.7 Model Complexity, Communication Overhead and Latency

This section provides the summary of model complexity, communication overhead and latency issues in FL. Additionally, it delves into state-of-the-art techniques dealing with the challenges caused by model complexity and communication overhead.

2.7.1 Model Complexity in FL

Model complexity in FL is influenced by factors such as model size, the number of parameters, and computational requirements, which affect the feasibility and effectiveness of deploying models on edge devices with limited resources [33]. It is imperative to manage model complexity in order to guarantee performance, efficiency, and inclusivity across heterogeneous clients as FL scales to larger models and increasingly diverse datasets [28].

High model complexity in FL can impose substantial computational and memory demands on client devices, particularly those with limited processing capacity, such as smartphones or IoT devices [29]. It is challenging for all clients to participate effectively in complex models with many parameters since they demand significant power, processing, and storage resources [60]. Additionally, the training process is slowed by the high complexity, which necessitates large number of communication rounds and can impede real-time applications [12, 29]

Several techniques have been developed to reduce model complexity in FL, enabling efficient training while preserving model accuracy:

- **Model Pruning:** Model pruning removes redundant parameters by selectively eliminating weights or neurons that have minimal impact on model predictions. Techniques such as structured pruning (removing entire neurons or layers) and unstructured pruning (removing individual weights) can help reduce the model's size and computational load, making it more suitable for resource-constrained devices. Structured pruning is advantageous for deployment as it maintains model architecture integrity, facilitating efficient parallelisation on hardware [108].

- **Quantisation:** Quantisation reduces model precision by representing weights and activations with lower-bit representations, such as 8-bit integers instead of 32-bit floating-point values [115]. This reduction can drastically decrease the model's memory requirements and speed up computations. Quantisation-aware training (QAT) allows models to maintain high accuracy despite lower precision, making it an effective strategy for deploying complex models in FL settings where device constraints are significant [116].

- **Knowledge Distillation:** Knowledge distillation transfers knowledge from a complex "teacher" model to a smaller "student" model. The student model learns to replicate the

teacher's performance using a simplified architecture, reducing computation and memory costs while preserving accuracy. In FL, federated knowledge distillation enables each client to train smaller, personalised models using teacher-student frameworks, which also reduces communication overhead [39].

- **Parameter Sharing Strategies:** Techniques such as partial model sharing, layer-based updates, and sparse communication reduce model complexity by transmitting only critical parts of the model. For example, in layer-based sharing, only select neural network layers are updated or shared among clients, reducing the communication cost and allowing simpler models on resource-constrained clients [117].

2.7.2 Communication Overhead in FL

Communication overhead is a major bottleneck in FL, as frequent exchanges of model updates between clients and the server can lead to significant latency and energy consumption. This challenge is compounded by the diverse network conditions across devices, where bandwidth limitations or intermittent connectivity can further delay the aggregation process [35]. As FL typically requires multiple communication rounds to converge, managing communication overhead is essential to make FL viable for large-scale and real-time applications.

Techniques to Address Communication Overhead and Latency

- **Compression and Sparsification:** Compressing model updates reduces the data size transmitted between clients and the server, thereby lowering communication costs. Gradient sparsification and quantisation are two popular techniques for this purpose. Gradient sparsification sends only a subset of significant gradients, often based on a threshold, while setting others to zero [117]. Quantisation, on the other hand, reduces the bit precision of gradients. Combined, these approaches can substantially reduce communication without severely impacting model accuracy [115, 116].

- **FedAVG:** The FedAvg algorithm reduces communication rounds by allowing each client to perform multiple local updates before sending the model to the server. By aggregating model updates after several local epochs, FedAVG reduces the frequency of communication, which is particularly effective in settings with unstable connectivity or limited bandwidth. However, this technique may introduce additional challenges in non-IID data settings, potentially affecting model convergence [12].

- **Client Selection and Scheduling:** Selecting a subset of clients to participate in each training round can lower communication costs. Adaptive client selection methods prioritise clients based on data quality, network conditions, and resource availability, thus balancing communication overhead with model accuracy. Scheduling clients based on

their bandwidth and latency can also optimise communication efficiency, reducing wait times caused by slow devices [82].

- **Asynchronous Federated Learning:** Asynchronous FL techniques allow clients to send updates simultaneously rather than synchronising all clients in each round. This approach reduces latency and enables faster clients to continue training without waiting for slower devices, thereby minimising the impact of stragglers. Asynchronous methods, however, must account for stale updates from slower clients to avoid adverse effects on model performance [33].

- **Hierarchical Federated Learning:** Hierarchical FL reduces communication between clients and the central server by organising clients into clusters or intermediate nodes that aggregate updates locally before sending them to the server. This multi-layered approach reduces the total communication load on the central server and enhances scalability. Hierarchical FL is particularly advantageous in environments with many clients, such as IoT networks, where direct server communication for each client would be infeasible [44].

In summary, balancing complexity and communication efficiency is crucial for effective FL deployment. Complex models generally improve accuracy but lead to higher communication costs due to large model updates. Conversely, reducing model complexity enhances communication efficiency but impacts model performance. Therefore, adopting a hybrid approach, such as a combination of model compression, selective update techniques, and client selection, enables FL systems to dynamically balance these competing demands based on the specific requirements of the deployment environment. The techniques addressing model complexity and communication overhead reflect the ongoing efforts in FL research to accommodate real-world constraints. As FL expands into applications requiring complex models and real-time processing, these strategies enhance scalability, responsiveness, and inclusivity across diverse client populations.

2.8 Application of FL

Recently, a substantial body of research has been conducted, advancing, refining, and identifying the multiple verticals in FL. The versatility of FL lies in its ability to adapt to different environments where data is inherently distributed and privacy is a significant concern. The application areas include smart healthcare, smart grids, wireless communication systems, smart cities, smart industrial control, education, banking, smartphones, IoT, cyber security, and more [33]. While FL does have potential applications in various fields, it is essential to consider the challenges and limitations associated with implementing this technology, such as communication overhead, model aggregation, and data privacy concerns. Additionally, the practical implemen-

tation of FL in real-world scenarios may require significant resources and expertise, making widespread adoption challenging.

In this thesis, we choose the application areas, which include smart grid energy forecasting, wireless communication systems, and healthcare through indoor and outdoor human activity recognition (HAR). These applications are strategically selected based on their alignment with the core challenges and contributions addressed in this thesis. Each domain exemplifies environments where data privacy, efficient communication, computational overhead, and data heterogeneity are pivotal concerns. For instance, smart grid and energy forecasting present unique challenges due to the sensitive nature of energy consumption data and the competitive market structure, making FL an ideal solution for collaborative load forecasting while preserving data privacy.

Wireless communication, especially in the context of 5G and beyond, benefits from FL for managing the dynamic nature of high-frequency networks, where proactive blockage prediction and seamless handovers are necessary for ensuring reliable connectivity. To achieve scalability, future wireless systems will adopt multi-modal data fusion and robust learning mechanisms. Additionally, incorporating FL into wireless systems will allow for efficient resource allocation and network efficiency, enhancing user experiences and overall network reliability. Lastly, healthcare, specifically focusing on HAR using wearable or contactless sensing, underscores the critical need for data privacy, multi-modal fusion, and efficient on-device processing, as personal health data is highly sensitive. The rationale for choosing these applications is their significance to society, their challenges in data management, and their alignment with the innovative solutions proposed in this thesis. These solutions aim to tackle data diversity, energy efficiency, and communication constraints using customised FL mechanisms. This section will provide a thorough literature review of the application areas considered in this thesis and the research gap analysis.

2.8.1 FL Application in Load Forecasting

STLF plays a pivotal role in ensuring the stability and operational efficiency of modern power systems. It enables utility companies to manage the integration of renewable energy sources better, optimise generation scheduling, and enhance demand-side management strategies. The complexity of the current electricity market, marked by deregulation, competition among various stakeholders, and the integration of advanced metering infrastructure (AMI), has made accurate STLF increasingly essential [118, 119].

Various STLF techniques have been developed for both macro-scale (substation/grid level) and micro-scale (household level) forecasting [120–122]. Traditional statistical approaches, such as linear regression, auto-regressive moving average (ARMA), and auto-regressive integrated moving average (ARIMA), have been widely utilised [123, 124]. However, the advent of big data and artificial intelligence (AI) has propelled the use of deep learning (DL) mod-

els that excel at capturing complex, non-linear patterns in load data [30]. While these models have shown significant promise, they often require vast amounts of historical data for training. This requirement typically leads to centralised data aggregation, which poses challenges related to data privacy, high communication costs, and the restricted accessibility of secure data silos [125]. For instance, residential energy data collected at the micro-scale level is highly privacy-sensitive and could be exploited to infer user behaviours, raising security concerns [126]. Similarly, utility providers at the macro-scale level are often unwilling to share data due to competitive and privacy-related reasons.

FL is gaining popularity in residential-level STLF, where smart meter data is inherently diverse and varies significantly across different households. Many existing studies have applied FL to residential data by employing clustering techniques to manage data heterogeneity and group clients based on consumption patterns. These clustering-based approaches create multiple federated models tailored to different clusters, as demonstrated by Singh et al. [127], who combined FL and transfer learning to enhance forecasting accuracy by clustering households with similar electricity usage profiles. Although effective, the accuracy of these models heavily relies on the quality of clustering and remains sensitive to data anomalies, which can negatively impact model stability.

Other clustering-based FL models for STLF have incorporated various data attributes. For example, researchers have implemented federated long short-term memory (LSTM) models that use socioeconomic clustering to categorise users based on load characteristics [128]. Additionally, studies like [129] have explored non-clustered LSTM training for individual households but reported challenges related to data variability affecting model stability. Similarly, approaches involving bidirectional LSTM models combined with optics-based clustering have been developed to group users by region and heating type [130]. While clustering can help mitigate data diversity by segmenting clients into more homogenous groups, it has limitations. Static, predefined clusters struggle to capture transient variations in data, leading to increased communication and computational overheads.

In addition to handling different data types, new studies have added privacy-protecting methods like DP and clustering-based defence mechanisms to make FL models safer and more reliable. For instance, the FedBranched approach, introduced in [31], assigns clients to separate branches to improve adversarial resilience. At the same time, Layer-Based Anomaly Aware FedAVG (LBAA-FedAVG) selectively discards potentially adversarial updates to safeguard model robustness [131].

Limitations and Research Gap

Despite the advancements in clustering-based FL for residential STLF, there is limited research on applying FL at the substation level, which exhibits more stability and regularity than residential smart meter data. The consumption patterns at this level are aggregated and less susceptible

Table 2.4: Summary of FL Studies for STLF Using Clustering Approaches

Ref	Strengths	Weaknesses
[127]	Improves forecasting accuracy by clustering households with similar consumption patterns; reduces computational complexity.	Effectiveness is highly dependent on clustering quality; sensitive to data anomalies that can impact robustness.
[128]	Groups users based on socioeconomic data, enhancing model performance by considering load characteristics.	It relies on static clustering, which may not capture transient or dynamic variations in data distribution.
[129]	Demonstrated the potential for handling diverse household data without predefined clusters.	High variability in household data posed challenges for achieving stable model performance.
[130]	Effectively groups users by region and heating type for better accuracy.	Static clustering fails to address short-term data shifts and lead to over-segmentation and increased model complexity.
[31]	Enhances adversarial robustness by assigning clients to different branches for training.	Depends on clustering, leading to potential segmentation issues and higher communication overhead.
[131]	Selectively discards potentially adversarial updates to enhance model robustness.	Clustering-based mechanisms overlook non-clustered data patterns and increase computational costs.
[132]	Integrates differential privacy to protect model updates and enhance resilience.	It does not address data diversity; adding noise disproportionately affects small clusters, reducing accuracy.

to individual behavioural fluctuations. Hence, the necessity for clustering diminishes, and alternative strategies for handling data heterogeneity become more appropriate.

Moreover, existing studies often overlook the need for effective aggregation methods that can manage diverse data without relying on clustering. Clustering-based approaches can lead to unnecessary segmentation and added complexity when applied to substation-level data, where a unified model would be sufficient. The strengths and limitations of current studies for STLF are given in Table. 2.4. Additionally, these approaches adequately address other critical challenges, such as:

- **Static Clusters and Increased Complexity:** Dependence on predefined static clusters struggles to capture transient variations in household data, limiting the model's adaptability to changing consumption patterns [129]. Training multiple models for different clusters also increases model complexity and computational overhead.

- **Adversarial Robustness:** Protecting the FL system from malicious clients injecting poisoned updates is crucial. Clustering has limitations in mitigating such risks, especially if an entire cluster is compromised. Moreover, the effectiveness of clustering-based models

is highly dependent on clustering quality and can be adversely affected by data anomalies [127].

- **Privacy Preservation:** While FL inherently provides privacy benefits, integrating mechanisms like DP is essential to prevent indirect leakage through model updates. Clustering complicates DP implementation, as noise addition disproportionately affects smaller clusters, degrading model accuracy [132].
- **Intermittent Client Participation:** In real-world conditions, client availability can be unpredictable. Clustering approaches often assume synchronous communication patterns, which may not hold in dynamic environments [119, 133].

2.8.2 FL for Vision Aided Wireless Communication

Vision-aided wireless communication has emerged as a transformative approach to improving network reliability and supporting proactive handover (PHO) in high-frequency wireless networks, such as those using millimetre-wave (mmWave) and sub-terahertz (THz) frequencies. While offering substantial bandwidths for high-throughput applications, these frequencies are more susceptible to physical obstructions, necessitating precise and efficient beamforming [134]. Integrating vision-based data and traditional wireless sensing through multi-modal fusion has shown potential in enhancing blockage prediction accuracy by leveraging a richer representation of the wireless environment [135, 136]. Multi-modal fusion combines data from various sources, such as cameras, radar, and channel state information (CSI), to provide comprehensive environmental insights that improve beam selection and PHO strategies [137].

Recent research highlights the growing role of DL frameworks in facilitating these multi-modal approaches for link blockage prediction. For instance, frameworks such as Vision Wireless (ViWi) integrate vision and wireless sensing data to offer holistic views of the network environment, improving PHO performance [135]. Vision-aided approaches have demonstrated a significant edge over traditional wireless-only methods regarding detection accuracy and maintaining link quality. However, these methods often depend on centralised data processing, which raises privacy concerns, incurs high communication overhead, and limits scalability, particularly in privacy-sensitive and large-scale deployments [138].

Research Gap

The inherent limitations of centralised processing have prompted the exploration of distributed learning methods like FL, which facilitate collaborative model training without sharing raw data, thereby preserving user privacy and reducing bandwidth requirements [21]. Nevertheless, standard FL methods, such as FedAVG, face challenges in handling the variability of client data and noise from diverse edge nodes, which can compromise model robustness and performance

in dynamic environments [80]. While promising, the application of FL in vision-aided wireless communication remains limited, particularly in integrating semantic extraction techniques. Semantic extraction in FL aims to derive meaningful and compact representations of the environment from raw sensor data, significantly reducing data volume and transmission costs while retaining critical information [137]. However, research in vision-assisted wireless networks has scarcely focused on semantic-aware FL approaches.

The few studies available do not fully address how to balance the complexity of extracting semantic information with the need for real-time adaptability and efficient on-device processing. Furthermore, the current literature does not explore the impact of adversarial factors on the accuracy and robustness of blockage prediction models. This gap is particularly significant as adversarial attacks can introduce noise and distort model updates, degrading the performance and reliability of PHO systems [22]. This highlights the need for advanced aggregation methods within FL to mitigate the impact of malicious or noisy updates while handling data heterogeneity and ensuring robust performance across diverse edge nodes.

2.8.3 FL for Human Activity Recognition

HAR is an active area of research, developing intelligent systems that can monitor and interpret human actions in real-time for various applications, from healthcare to smart environments [139]. HAR can be categorised into outdoor and indoor settings, each utilising distinct technologies tailored to their unique challenges. Outdoor HAR often relies on wearable sensors like accelerometers, gyroscopes, and heart rate monitors, capturing multi-modal data that provides comprehensive insights into physical activity [140, 141]. These devices are ideal for dynamic, mobile environments where portability and versatility are crucial. In contrast, indoor HAR leverages contactless sensing technologies, such as radio frequency (RF) signals, including CSI and RSSI [142, 143]. These systems offer non-intrusive monitoring by analysing how wireless signals interact with human movements, preserving privacy and ensuring user comfort [144, 145].

The rationale behind using wearable technologies outdoors lies in their flexibility and detailed data capture, whereas contactless sensing is preferred indoors for its non-invasive nature and scalability in private, enclosed spaces [146, 147]. Both approaches highlight the ongoing need for advanced processing techniques, such as FL, to ensure data privacy, efficiency, and accuracy in distributed HAR systems [148, 149]. This section thoroughly reviews the indoor and outdoor environments for HAR systems, highlighting the challenges and research gaps in current research.

Outdoor HAR

The outdoor settings for HAR present unique challenges, such as user mobility, multi-modality, battery constraints, and energy efficiency. FL inherently supports multi-modal data fusion by enabling data collected from different sensors, such as accelerometers, gyroscopes, and GPS units, to contribute to a single model without pooling raw data in one place [147, 148]. This capability enhances the robustness of HAR systems, as it allows for richer contextual information and more accurate activity recognition, especially in outdoor environments where data sources can vary widely. Additionally, FL frameworks inherently distribute computational tasks across multiple edge devices, enabling personalised model training that can adapt to the diverse and context-specific nature of user activities [147].

However, the real-world implementation of FL-based HAR systems faces substantial challenges, particularly due to the heterogeneous nature of data collected from various users and sensors. Non-IID is a common challenge that affects model generalisation and convergence [150]. Addressing these challeges, advanced frameworks such as ProtoHAR [150], and ClusterFL [151] have been proposed to manage data heterogeneity through strategic clustering and personalised model updates. Similarly, recent explorations into FL-based HAR systems reveal a concerted effort to leverage multi-modal data fusion for enhancing classification accuracy, particularly in fall detection scenarios [148]. This study proposed a novel approach that transforms time-series sensor data into images to detect anomalies. Additionally, this approach also leverages visual data from cameras and input-level data fusion within FL frameworks to achieve a classification accuracy of 89.76%. Similarly, novel FL via augmented knowledge distillation (FedAKD) was designed for the collaborative training of heterogeneous DL models [152]. FedAKD exhibited superior communication efficiency compared to the FedAVG algorithm, with a 200-fold increase. It also achieved 20% higher accuracy than other knowledge distillation-based FL methods.

Research Gap

Despite the substantial progress in applying FL to outdoor HAR systems, several gaps are identified, such as:

- **Energy Efficiency:** Existing studies largely employ traditional DL models, which are not optimal for resource-constrained, battery-operated devices. Hence, architectural changes like the use of neuromorphic computing or hybrid models like Spike-LSTM can significantly reduce the computational complexity.
- **Handling Data Heterogeneity:** Though frameworks such as ProtoHAR and ClusterFL address non-IID data to some extent, they primarily focus on clustering and personalisation strategies. These approaches often increase computational complexity and are not suitable for real-time applications.

- **Communication Overhead:** While FL reduces the need to share raw data, the frequent transmission of model updates can still strain the limited bandwidth and battery life of wearable devices.

Addressing these gaps requires the development of innovative frameworks that seamlessly integrate energy-efficient neuromorphic models, advanced data fusion techniques, and adaptive FL protocols.

Indoor HAR settings

Indoor HAR utilises a range of sensor-based approaches, moving from traditional wearable devices to advanced contactless sensing technologies. These systems aim to classify and identify human actions through observations captured by diverse sensors, such as RF sensing and vision-based methods [144, 153]. Vision-based HAR, which depends on high-resolution cameras and computer vision techniques for effective activity tracking, encounters notable challenges regarding privacy concerns, including unauthorised data collection and sensitivity to environmental variables such as lighting conditions and background distractions [146]. These limitations directly shift the focus towards sensor-based approaches that prioritise maintaining a higher level of user privacy.

As a result, contactless RF sensing, particularly through channel state information (CSI), has garnered significant attention as a non-invasive, privacy-preserving substitute for wearable and vision-based systems. The primary advantage of RF-based HAR lies in its ability to monitor activities without requiring individuals to wear or carry sensors, enhancing convenience and scalability. CSI-based systems capture detailed physical layer information, such as amplitude and phase per sub-carrier, to generate unique signatures for various activities [143, 154]. This characteristic makes CSI-based HAR particularly effective in indoor settings, where the phenomenon of multipath propagation and signal scattering caused by human bodies creates distinct patterns that aid in accurate activity classification [155].

To this extent, limited studies have been conducted on HAR leveraging FL. For instance, the authors in [149] propose a novel method for HAR using wireless signals that address data privacy and the non-IID nature of distributed datasets. This scheme introduced a cross-domain federated learning framework (CDFL) that leverages transfer learning and achieves high accuracy rates in ultrasonic signal-based gesture recognition tasks (over 90% for a 5-category task with simulated data and 88% for a 10-category task with minimal real data), promising to ease the data collection burden while preserving privacy.

Similarly, the authors in [156] propose FedHAR as a customised FL framework that addresses significant challenges in HAR, such as label scarcity, real-time processing, and heterogeneous data. It leverages distributed learning to keep data local and employs a semi-supervised online learning strategy to handle unlabeled data effectively. The framework incorporates a hierarchical attention architecture for feature alignment and a semi-supervised learning loss to

integrate gradients from labelled and unlabelled clients. After extensive testing on two public datasets, FedHAR has shown superior performance compared to existing methods, with a significant improvement of about 10% across various metrics when fine-tuning models for unlabeled clients, highlighting its effectiveness in real-world HAR applications. However, this study used wearable sensing data and overlooked the energy efficiency and communication overhead during training.

The authors in [157] propose a novel method for HAR using a 3D convolutional neural network (3DCNN) applied to sensory data, translating the temporal and frequency bio-signal values into voxel intensities. This method enhances privacy and security by employing bitwise XOR encryption and FL, achieving significant accuracy with minimal loss when data is encrypted. Similarly, [158] introduces Hydra, a hybrid-model FL framework designed for devices with varying computational capabilities. It uses BranchyNet technology to create a hybrid model, allowing heterogeneous devices to train different model parts suited to their capabilities. Hydra clusters devices based on data similarity to improve co-training efficiency and introduces a sample selection algorithm and a large-to-small knowledge distillation technique to enhance model accuracy. Extensive experiments demonstrate Hydra's superior performance compared to state-of-the-art methods. However, both [157, 158] used wearable sensing data and did not consider communication overhead and energy efficiency.

In [159], a cluster FL-based algorithm is proposed for activity classification using wearable devices. The proposed algorithm demonstrated better model generalisation with diverse datasets and improved the system's overall performance. In [160], a wearable sensor-based FL model is presented for indoor HAR using the FedAVG algorithm. This work used the publicly available HAR datasets and demonstrated the utility of using FL for activity recognition. Similarly, novel FL via augmented knowledge distillation (FedAKD) was designed for the collaborative training of heterogeneous DL models [94]. FedAKD exhibited superior communication efficiency compared to the FedAVG algorithm, with a 200-fold increase. It also achieved 20% higher accuracy than other knowledge distillation-based FL methods.

Research Gap

Finally, Table. 2.5 summarises the contributions and limitations of the related work. Although FL presents a promising solution to privacy and scalability challenges in HAR, the integration with CSI-based systems remains underexplored. Moreover, the potential of utilising raw signal data to extract features and provide contextual information in HAR systems, particularly in an FL framework, has yet to be fully realised. This approach can revolutionise understanding and interpreting complex human activities in various environments. Furthermore, conventional approaches often involve converting wireless signals into spectrograms, which, while effective for visual feature extraction, result in significant computational overheads and increased communication burdens when transferring model updates [149, 165]. Additionally, the issue of commu-

Table 2.5: Comparative analysis of recent advances in HAR with contributions and limitations.

Ref	Contributions	Limitations
[161]	Introduced a simple RSSI-based HAR using K-means for activity classification.	Suffers from low accuracy, lacks privacy-preserving mechanisms, and does not consider energy efficiency or communication overhead.
[162]	Developed a gesture recognition system emphasising simplicity and ease of use.	Relies on centralised data processing, which raises privacy concerns and increases communication costs.
[155]	Applied ML algorithms for vital sign and activity classification using CSI data.	Centralised model poses privacy risks, lacks FL integration, and does not address communication or energy efficiency challenges.
[163, 164]	Demonstrated feature-level fusion in CSI-based activity classification for improved accuracy.	Limited by a theoretical approach, lacks practical privacy solutions, and omits considerations for energy efficiency and communication overhead.
[149]	Explored cross-domain FL to address HAR, considering the impact on communication overhead.	Did not incorporate feature fusion, missing richer contextual data analysis, and lacks energy efficiency measures.
[157]	Translated temporal and frequency bio-signal values into voxel intensities and used encryption to ensure security.	Focused on wearable sensor data, did not consider energy efficiency, and involved high communication overhead.
[158]	Created hybrid models allowing co-training with varying computational capabilities.	Utilised wearable sensor data lacked energy efficiency considerations and did not address communication overhead.
[159]	Utilised clustering to enhance HAR accuracy, leveraging wearable sensors.	Did not address communication efficiency or energy constraints, potentially limiting scalability in real-world applications.
[94]	Introduced a robust FL framework via knowledge distillation, optimising model learning.	Primarily focused on wearable sensor data; not directly applicable to CSI-based HAR and lacks energy and communication efficiency measures.

nication overhead, a critical factor in the scalability and efficiency of HAR systems, especially when integrated with FL, has not been fully explored in the current body of research. Developing and implementing efficient model-sharing techniques within FL-based HAR systems could significantly mitigate this challenge, paving the way for more sophisticated, real-time activity recognition solutions. Furthermore, most sensors are expected to be battery-operated, and frequently sharing large model parameters during training will drain the battery. Therefore, the potential of combining FL with advanced feature fusion and model compression strategies to enhance efficiency and maintain high accuracy in indoor HAR remains underexplored. For instance, while QAT offers a pathway to reduce communication overhead, its application to fed-

erated CSI-based HAR remains limited. Furthermore, the majority of studies do not address the need for adaptive aggregation algorithms capable of handling heterogeneous data effectively, leading to potential model instability.

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

The literature review highlights key findings, research gaps, and challenges in FL that align with the three core challenges identified in **Section 1.3**. Data diversity and multi-modal fusion **C1** in FL emerges as a significant obstacle due to the diverse data distributions and computational capabilities of ENs [33]. The data from ENs embodies unique local patterns or biases stemming from varying environments, user behaviors, and data collection methods [34]. This heterogeneity leads to suboptimal performance, where the global model struggles to generalise effectively across all clients, causing issues like model drift and unstable convergence [38,39]. On the other hand, multi-modal fusion in FL seeks to leverage diverse data sources, such as text, images, and audio, to enhance model performance [51]. This approach is particularly valuable in applications requiring rich contextual information, including healthcare, wireless communications, and smart cities [52]. However, integrating multi-modal data introduces complexities due to inherent diversity in data formats and feature representations. Issues such as modality alignment, data heterogeneity, and increased communication overhead persist, impeding effective model convergence [53, 54].

Significant gaps still need to be addressed despite the development of various techniques to address data diversity and multi-modal fusion. For instance, clustering-based methods group clients with similar data characteristics to mitigate non-IID issues [44,45], but struggle to handle transient data variations and often increase computational costs. Personalised FL customises global models for individual clients, improving local performance, but can be resource-intensive and needs to scale better [46, 47]. Normalisation techniques adjust local model updates before aggregation to reduce disparities [48], and domain adaptation aligns feature distributions across domains [49]. However, there is a need for scalable frameworks that can seamlessly integrate multi-modal data and address statistical heterogeneity without compromising efficiency.

Adversarial robustness and privacy preservation **C2** are significant concerns in FL, where the decentralised nature of training makes the system vulnerable to various attacks [17], [16]. Performance attacks like data poisoning involve malicious clients injecting corrupted data to introduce biases or errors into the global model, such as label flipping or backdoor attacks [73–75]. Model poisoning attacks involve attackers manipulating model updates to skew the global model towards incorrect patterns [77]. Techniques like HE, SMPC, and DP offer solutions but involve several trade-offs. For instance, HE enables computations on encrypted data, maintaining data privacy during model training [83, 84], but introduces significant computational over-

head. SMPC supports collaborative computations without data sharing, enhancing privacy in distributed settings [87, 88]. However, it incurs high communication costs. DP protects against inference attacks by introducing random noise to model updates, safeguarding sensitive information [82, 90]. However, noise injection degrades the model's performance. Additionally, defence mechanisms against model and data attacks include anomaly detection [97, 98], robust aggregation methods like Krum and Bulyan [18, 103], model pruning [108], and regularisation techniques [113, 114]. However, these methods often involve trade-offs between privacy preservation, computational overhead, and model utility.

Energy efficiency and computational constraints **C3** pose significant challenges in deploying FL on resource-constrained devices like IoT nodes. High model complexity imposes substantial computational and memory demands on client devices, particularly those with limited processing capacity. Managing model complexity ensures performance, efficiency, and inclusivity across heterogeneous clients as FL scales to larger models and datasets [28]. While techniques like model pruning [108, 109], quantisation [115, 116], and knowledge distillation [39] reduce complexity and communication overhead, their integration into FL frameworks for real-time applications remains underexplored. Additionally, techniques to address communication overhead include compression and sparsification of model updates [116, 117], client selection and scheduling [82], asynchronous federated learning [33]. However, balancing model complexity and communication efficiency remains a challenge. Complex models improve accuracy but increase communication costs, while simplifying models may impact performance. Innovative strategies are required to balance these demands, particularly for resource-constrained edge devices dynamically.

In summary, addressing the core challenges in FL requires innovative frameworks that handle data diversity and multi-modal fusion without increasing computational overhead, enhance privacy preservation and adversarial robustness without sacrificing model utility, and optimise model complexity and communication efficiency for deployment on resource-constrained devices. By tackling these gaps, this thesis aims to advance FL's applicability across critical domains such as smart grids, wireless communication, and healthcare, contributing scalable, robust, and energy-efficient solutions.

Chapter 3

Similarity Driven Truncated Aggregation (SDTA)

This chapter introduces the SDTA algorithm, a novel cluster-free aggregation mechanism that addresses the challenges **C1** and **C2** as discussed in **Section 1.3**. SDTA leverages cosine similarity to align client model updates, effectively managing data diversity without clustering. In conjunction with cosine similarity, the truncated aggregation mitigates the impact of noisy or adversarial updates, enhancing the resilience of the global model against potential attacks. Additionally, SDTA integrates DP to protect sensitive client data during training, ensuring privacy preservation without significantly compromising model performance. The proposed algorithm is tested for STLF at the substation level. This setting offers unique opportunities and challenges due to the inherent diversity and stability of the data compared to residential energy forecasting data. This study demonstrates that the proposed SDTA effectively managed diverse data sources, improving forecasting accuracy and enhancing the robustness and privacy of the federated model, all without relying on clustering techniques.

3.1 Introduction

The global shift towards a sustainable energy future is driving a need for reliable, efficient, and accurate methods in energy management. As countries transition to cleaner and renewable energy sources like solar, wind, and hydro, the variability and intermittency inherent in these sources introduce new complexities in power grid operations [166, 167]. These challenges necessitate accurate STLF, which helps grid operators maintain supply-demand equilibrium, support economic dispatch, and optimise generation in real-time [30]. The importance of precise forecasting extends to operational and environmental impacts, where balancing load reduces emissions and enhances system reliability [168]. However, the increasing complexity of the electricity market, characterised by multiple stakeholders such as producers, distributors, and consumers, coupled with regulatory pressures, has driven interest in sophisticated forecasting

techniques capable of addressing diverse operational demands [169].

STLF techniques have evolved considerably, with traditional statistical methods like autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) paving the way for more advanced data-driven methods [123, 124]. With the advent of big data and artificial intelligence, deep learning (DL) algorithms have been increasingly utilised for STLF, offering a robust means to model the complex and nonlinear load profiles associated with modern energy consumption patterns [30]. However, DL-based forecasting often requires substantial historical data, which raises privacy and security concerns when managed in a centralised framework. Centralised data collection not only poses privacy risks but also leads to increased communication costs and limited access to secure data silos, especially for sensitive data at the micro-level (e.g., individual buildings or households) [126].

FL has emerged as a distributed alternative to address these issues, allowing multiple entities to train a global model collaboratively without directly sharing sensitive data [11, 170]. In STLF, FL has been primarily applied at the residential level, where clustering techniques are frequently used to group clients with similar load profiles, thereby creating multiple federated models for different clusters. For instance, Singh et al. [127] combined clustering with federated and transfer learning to improve forecasting by segmenting households based on similar consumption patterns. Although this approach effectively manages high-variability residential data, it is highly sensitive to data anomalies and heavily dependent on clustering quality, which can impact model robustness and stability.

The reliance on clustering in FL has led to several limitations. For instance, clustering-based approaches typically require static groupings, unable to capture transient data variations in household-level STLF [128, 129]. Additionally, various studies have explored enhancements such as adversarial robustness and privacy protection. For example, Manzoor et al. [31] proposed FedBranched, an FL model that groups clients to enhance resilience against adversarial attacks. However, these approaches are unsuitable for substation-level data, which is inherently more stable and diverse than residential data, thus limiting the need for extensive cluster-specific models. Furthermore, clustering introduces overhead in terms of communication and computational complexity, which hinder model convergence, mainly when applied under privacy constraints, as noise addition from DP disproportionately affects smaller clusters [132].

3.1.1 Contributions

Addressing challenges **C1** and **C2**, this chapter proposes a novel SDTA framework tailored for substation-level STLF. Unlike traditional clustering-based approaches, SDTA avoids unnecessary segmentation by focusing on aligning client updates through layer-wise cosine similarity, thereby allowing for a unified model that accommodates data diversity without directly accessing client data. A distinctive feature of SDTA is its use of a truncated mean, which mitigates the influence of extreme or anomalous updates and reduces overfitting. This approach ensures

that the model remains robust to outliers and adversarial updates while enhancing convergence stability, particularly in the context of substation data, which exhibits more regularity compared to residential data.

Furthermore, SDTA incorporates DP mechanisms to protect client privacy without significantly compromising model accuracy. By balancing the trade-offs between privacy and accuracy, SDTA offers resilience against adversarial attacks, such as model sign inversion, without excluding entire client contributions. Through extensive simulations and testing on real-world substation data, this chapter demonstrates that SDTA outperforms standard FL algorithms like FedAVG and FedDist in scenarios that involve DP budgets, adversarial attacks, and partial client participation. The key contributions of this chapter are highlighted as:

- This study introduces SDTA, which combines similarity measures, filtering process, and DP mechanisms to enhance the robustness FL training process. The proposed techniques use layer-wise cosine similarity to align client updates and identify anomalous updates. Unlike traditional clustering-based approaches, SDTA measures the alignment of model updates without accessing client data. This allows it to effectively aggregate contributions from diverse clients while minimising the influence of outliers. Additionally, by employing truncated mean aggregation to filter the extreme values, SDTA enhances robustness against adversarial attacks and reduces the risk of overfitting.

- SDTA incorporates DP mechanisms to protect model updates during training, ensuring a balance between privacy and model accuracy. The proposed framework is thoroughly evaluated under varying privacy levels, low, medium, and high, each corresponding to different amounts of noise added to the model updates. Its performance is benchmarked against traditional methods like FedAVG and FedDist, demonstrating its robustness across diverse privacy requirements.

- SDTA enhances model robustness against adversarial updates by utilising similarity-based filtering and truncated aggregation at the layer level. It effectively mitigates the impact of adversarial updates, such as model sign inversion attacks, without relying on clustering or discarding entire client contributions.

- Extensive simulations on real-world substation data are performed to compare the performance of SDTA with established FL algorithms such as FedAVG and FedDist. Evaluation encompasses various challenging scenarios, including DP budget, adversarial attacks, and partial client participation due to random client dropout.

3.2 System Model and Preliminaries

This section will discuss the system model and some preliminary knowledge of the FL training process and aggregation mechanism.

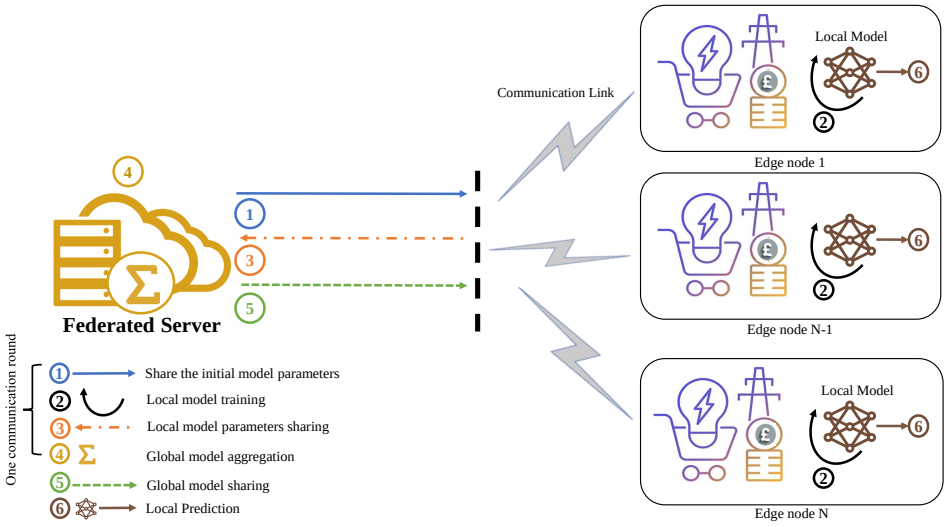


Figure 3.1: FL training process in each communication round for STLF.

3.2.1 Model Training Process for SDTA

This study considers a cross-silo FL architecture for STLF, which consists of a central federated server (FS) and N edge nodes (ENs), each corresponding to a substation within the distribution network as depicted in Fig. 6.1. The EN are denoted by index i where $i \in \{1,2,\ldots,N\}$, holds a distinct local dataset $\mathscr{D}_i$. These datasets are defined as $\mathscr{D}_i = \{(\mathbf{X}_{ij}, \mathbf{y}_{ij})\}_{j=1}^{\mathscr{D}_i}$, with $\mathscr{D}_i$ indicating the number of samples at the i-th EN. Each sample comprises a feature vector $\mathbf{X}_{ij}$ capturing historical load data and temporal features alongside a corresponding load demand label $\mathbf{y}_{ij}$.

The primary objective of the FL framework is to train a global model ω collaboratively by minimising the overall loss across all ENs, without sharing the raw data from each EN. This can be formulated as equ. (2.1). For regression tasks such as STLF, MAE is employed as the loss function, which is robust to outliers and provides a clear interpretation of forecasting errors. The local loss function at EN i given in equ. (2.2) is modified as:

$$\ell_i(\omega, \mathscr{D}_i) = \frac{1}{\mathscr{D}_i} \sum_{j=1}^{\mathscr{D}_i} \left| \mathbf{y}_{ij} - \hat{\mathbf{y}}_{ij}(\omega) \right|, \tag{3.1}$$

where $\mathbf{y}_{ij}$ and $\hat{\mathbf{y}}_{ij}(\omega)$ are the actual and forecasted load demands, respectively.

The FL training process begins with the FS distributing the initial global model ω^0 to all ENs at the start of round $t = 0$. Each EN updates this model using its local dataset $\mathscr{D}_i$. The update involves computing the gradient of the local loss function with respect to the model parameters:

$$\nabla_\omega \ell_i(\omega^t, \mathscr{D}_i), \tag{3.2}$$

where $\ell_i(\omega^t, \mathscr{D}_i)$ is the local loss function at EN i for model parameters ω^t at round t. Note that ω^t is used for both the global model sent to ENs and as the initial point for local updates,

emphasising the consistency of model parameters across the network. After computing the gradient, each EN i updates its local model weights, applying the learning rate η_i:

$$\omega_i^{t+1} = \omega^t - \eta_i \nabla \ell_i(\omega^t, \mathscr{D}_i), \tag{3.3}$$

where ω_i^{t+1} represents the updated local model parameters at EN i after local training in round t. This updated local model ω_i^{t+1} is then returned to the FS for aggregation to obtain the new global model. The details of the aggregation process and algorithms involved are discussed in Section 3.2.4.

3.2.2 Steps in FL Training Process

FL model training is an iterative process that involves several communication rounds between the FS and EN. The entire process is divided into six steps, as shown in Fig. 6.1. The first four steps in the FL training process represent a communication round. In contrast, the updated global model sharing and performance evaluation are done in step five, and actual predictions are made in step six. The detailed explanation of each step is given below:

Step 1: During the first communication round, the FS initialises the global model parameters ω^0 and broadcasts them to all ENs to start the training process. In subsequent communication rounds, the FS shares the aggregated global model parameters ω^t with ENs obtained from the previous training round. The ENs use the global model parameters as the initial point for their local model training.

Step 2: Each EN receives a copy of the global model parameters and trains a local model using its local data $\mathscr{D}_i$. At this stage, the LSTM model is trained using the preprocessed data at each EN. The details of data pre-processing and local model training using the LSTM are discussed in Section 3.2.3. The objective of the local loss function is defined in equ. (3.1).

Step 3: Since each EN trains its model using local data, they obtain distinct local model parameters ω_i^{t+1}. Therefore, the ENs return their updated local model parameters to the FS for aggregation.

Step 4: After obtaining the local model parameters for each EN, the FS performs the model aggregation to update the global model. This study proposed the SDTA algorithm for model aggregation and compared the results with FedAVG and FedDist. The four steps mentioned above represent a communication round, which is repeated until the convergence of the global model.

Step 5: Once the aggregation is done, the FS shares the updated global model ω^{t+1} with all ENs for local use.

Step 6: This is the final stage in FL, where the outdated model of each EN is replaced with the updated global model received from the FS and ready for load forecasting. The global model is trained collaboratively throughout the process without sharing any data.

Table 3.1: Feature variables for local model training

Weather Information
1. Dry bulb temperature
2. Dew points
Time Factor
3. Day of week
4. Hour of the day
5. Holiday/ weekend flag
Historic load profile
6. Previous day average load (24-h average load)
7. Previous day same-hour load (24-h lagging load)
8. Previous week same day same hour load (168-h lagging load)

3.2.3 Local Learning

To obtain an optimal global model, the training process for the EN model is similar to traditional ML model training, but it requires multiple communication rounds. In FL, data pre-processing is even more critical than in conventional model training because EN data has distinct trends and patterns that make the training process more challenging. Therefore, it is crucial to preprocess the forecasting data to develop a robust model. Preprocessing involves feature engineering, extracting input variables from the given data for local training, and feature scaling to normalise the input variables. Normalising the input variable reduces the dominance of features with relatively larger values, which improves the overall convergence time of the local model.

When dealing with a supervised regression problem like STLF, the quality of the model's performance hinges on the features selected during data processing. Historical load profiles and weather information are key input variables for accurate load forecasting. Additionally, contextual calendar information was considered, as it plays a pivotal role in the process. To achieve this, time factor is incorporated the time factor and extracted eight distinct feature variables are listed in the Table. 3.1.

The first two features represent the weather information derived from the previous hour of the day. As mentioned in [171], the weather parameters are key for accurate energy demand forecasting. Temperature, in particular, is essential since it considerably influences energy usage. Furthermore, distinct weather parameters are highly correlated, so including one or more results in multicollinearity [171], [172]. The time factor features are one-hot encoded variables where the day of the week ranges from 1 to 7 (1 for Monday and 7 for Sunday), an hour of the day varies from 1 to 24. A holiday flag is either 1 (public holidays) or 0, which were introduced based on the previous studies [173], [174]. Moreover, the historical load profile represents the lagging load data to consider the influence of demand on the previous values [175].

In conventional model training, scaling brings all input variables on a normalised scale to

reduce the dominance of features with larger values. It helps in improving the convergence time and performance of the model. This study applies Min-Max normalisation to weather information and historical load profiles. It is worth noting that feature scaling is done on EN, which transforms the input x to scaled value x_o as follows:

$$x_o = \frac{x - Min(x)}{Max(x) - Min(x)}, \tag{3.4}$$

where x is the input feature, $Min(x)$ and $Max(x)$ are the minimum and maximum values of a given feature, whereas the x_o is the transformed value which ranges between 0 and 1 [176].

As mentioned earlier, STLF is converted into a supervised regression problem employing multiple data types, as tabulated in Table. 3.1. Once EN receives the initial model parameters from FS, a LSTM is built to perform day-ahead load forecasting hourly, using local data. To ensure fairness, each EN has the same architecture, including an LSTM layer, two dense layers with 100 and 50 neurons, respectively and one fully connected layer. The activation function used for the fully connected layer is the rectifier linear unit (ReLU), and ADAM is used as an optimiser. Hyperparameter tuning on individual EN can improve the system's overall performance, resulting in massive computational costs, which is undesirable. Furthermore, it is still an open challenge due to the distributed nature of the data. Therefore, grid search is adopted on centralised data and obtains all the hyperparameters used by the local clients to reduce the system's complexity.

3.2.4 Model Aggregation Mechanism

The model aggregation mechanism at FS is pivotal in combining local model parameters from the ENs to update the global model. This process is iterative and continues until the global model converges. The global objective function involving N clients or EN is given in equ. (2.1). FedAVG, a prevalent aggregation method, synthesises the local updates into the global model by averaging the local model parameters. The FedAVG update to the global model ω at round $t+1$ is given in equ. (2.19):

One of the most effective and extensively used aggregation methods to solve the optimisation problem given in equ. (2.1) is FedAVG, which considers all local model parameters (weights and biases) in aggregation. The FedAVG algorithm presented in [11] extends the federated stochastic distance (FedSGD) algorithm, which aims to minimise the global loss function. SGD can be applied naively to the federated optimisation problem, where a single batch gradient calculation is done per communication round. This approach is computationally efficient but requires many communication rounds to obtain an optimal global model. In FedSGD, the EN completes a single-step gradient descent in each communication round and shares the acquired weights with FS. The FS performs weight aggregation proportionally to the number of local training samples, and this process can be thought of as a gradient descent step on the global

Algorithm 1: Federated Averaging (FedAVG) Algorithm for STLF.

Result: Final global model ω^T for STLF
Input: Local datasets $\{\mathscr{D}_i\}$, total number of clients N, fraction C of clients participating per round, number of communication rounds T, local batch size B, number of local epochs E, learning rate η;
Initialise global model parameters ω^0;
for $t = 0$ *to* $T-1$ **do**
    Select a random subset S_t of clients, where $|S_t| = C \times N$;
    $D = \sum_{i \in S_t} \mathscr{D}_i$;
    for *each client* $i \in S_t$ ***in parallel*** **do**
        Send the global model ω^t to client i;
        $\omega_i^{t+1} \leftarrow \text{LOCALTRAINING}(i, \omega^t)$;
    end
    Update global model equ. (2.19).
end
Procedure LOCALTRAINING(i, ω^t);
    Initialise local model $\omega_i \leftarrow \omega^t$;
    for $e = 1$ *to* E **do**
        for *each batch* $B \subseteq \mathscr{D}_i$ **do**
            $\omega_i \leftarrow \omega_i - \eta \nabla \ell_i(\omega_i, B)$;
        end
    end
    return $\omega_i^{t+1} \leftarrow \omega_i$ to the central server;

model. However, the slow global model convergence is one of the major drawbacks of FedSGD.

Therefore, this limitation is addressed in the FedAVG algorithm by introducing the concept of local epochs and batch size. In each communication round, multiple local epochs are executed on the subset of local data (small batches), reducing the frequent communication of FS and EN. Global model initialisation is the major step in the FL training process as it defines the model's architecture in terms of the number of layers and neurons. Once the global model is shared with EN, local training is done using the data on each client. The updated model parameter is shared with FS where the model aggregation is weighted, averaging as given in Algorithm 1. However, FedAVG is a coordinate-wise averaging technique that may lead to sub-optimal solutions, especially when diverging neurons in specific clients and increasing training time. Unlike FedAVG and FedDist, the proposed SDTA mechanism specifically addresses the challenges posed by data diversity and potential adversarial updates by incorporating similarity-based aggregation and outlier removal through truncation.

3.2.5 Federated Distance Algorithm

FedDist algorithm is a dissimilarity measurement approach that considers the Manhattan distance between the neurons of global and local models having a similar coordinate [177]. The

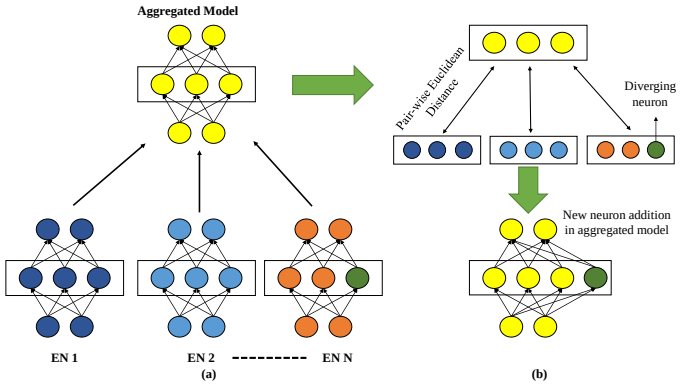


Figure 3.2: FedDist neuron generation process, where the diverging neuron is identified by computing the pair-wise Manhattan distance between the neurons of the EN model and FS model. The diverging neuron is added to an aggregated model if the distance is greater than the given threshold, as shown in (b).

divergence in neurons is caused by heterogeneous or non-IID data, which could be useful for updating the global model to make it more robust. These client-specific diverging neurons are incorporated into the global model as new neurons. This approach effectively deals with sparse data, where particular features are only present in a small subset of clients or data points, resulting in a more generalised model. Furthermore, layer-wise training is initiated to enable the neuron of the next layer to learn from the previous layer. The outstanding layers are frozen at this stage, and subsequent training is started. The first step is obtaining a global model, shown in Fig. 3.2 (a). The diverging node is identified using the Manhattan distance and added to the aggregating model. The detail of FedDist is given in Algorithm 2. Initially, the FS broadcasts the global model ω^t to each EN having local model ω_i^t. The EN starts local training and computes the pairwise Manhattan distance, or L1 norm, given by the following equation:

$$dist(N_1, N_2) = |N_1^{w1} - N_2^{w1}| + ... + |N_1^{wC} - N_2^{wC}|, \tag{3.5}$$

where N_1, N_2 are two different neurons and w_i is the i^{th} of C weights of neurons. The process of identifying diverging neurons is divided into three steps mentioned in the Algorithm 2. In the first step, the pair-wise Manhattan distance is calculated based on the equ. 3.5 to obtain the distance cost function represented by $\prod$. After the Manhattan distance calculation, the mean μ and standard deviation σ of the distance matrix are calculated for thresholding. The new neuron is added based on the thresholding process in the second step. If the distance exceeds the given threshold, the system freezes the layer, and a layer-wise update is started on the unfrozen layers. This process continues until all layers are treated.

Algorithm 2: Federated Distance Algorithm for STLF

Result: Final global model ω^T for STLF
Input: Total Communication Rounds T, Number of Clients N, Number of Layers L, Local Dataset $\mathscr{D}_i$, Fraction C of Clients participating, Learning Rate η;
Initialize global model parameters ω^0;
for $t = 1$ *to* T **do**
  Select a subset S of clients, where $|S| = C \times N$;
  for *each* $i \in S$ ***in parallel*** **do**
    Send global model ω^t to client i;
    $\Delta\omega_i \leftarrow \text{LocalTraining}(i, \omega^t)$;
  end
  $\omega^{t+1} \leftarrow \frac{1}{\sum_{i \in S} \mathscr{D}_i} \sum_{i \in S} \mathscr{D}_i \Delta\omega_i$; `// Federated Averaging`
  for *each layer* $l = 1$ *to* $L-1$ **do**
    for *each client* $i = 1$ *to* N **do**
      Compute neuron pair distance for layer l: $\Pi^l_{t=1} \leftarrow \text{distance}(\omega^l_t, \omega^l_{it})$;
    end
    Compute mean μ^l and standard deviation σ^l for each neuron distance Π^l_t;
    newNeuron $\leftarrow$ False;
    for *each neuron distance* d *in* Π^l_t **do**
      $dist\_threshold \leftarrow 3 \cdot \mu^l_d + \sigma^l_d + \text{penalty}(t)$;
      if $mean(d) > dist\_threshold$ **then**
        Append new neuron to layer l: ω^l_t;
        newNeuron $\leftarrow$ True;
      end
    end
    if *newNeuron* **then**
      for *each client* $i = 1$ *to* N **do**
        Update client i's model for layer l and above;
        Freeze updates for layer l and below;
      end
    end
  end
  Update global model for the next round: $\omega^{t+1} \leftarrow \omega^t$;
end

3.2.6 Differential Privacy in FL

In FL, it is essential to protect the privacy of local data, which remains on devices while only model updates are shared with the server. However, these updates can still leak sensitive information. To mitigate this risk, DP ensures a formal privacy guarantee [132]. (ε, δ)-DP offers a rigorous framework to quantify privacy preservation during distributed data processing, especially for STLF. Here, $\varepsilon > 0$ is a measure of the distinguishability between two neighbouring datasets, $\mathscr{D}_i$ and $\mathscr{D}'_i$, after the application of a privacy-preserving mechanism. A smaller value of ε indicates stronger privacy, implying that the outputs from similar datasets will be indistin-

guishable, reducing the risk of privacy leakage. Conversely, larger values of ε suggest a higher risk of exposing information from individual data points [82, 90].

Formally, the definition of (ε, δ)-DP a randomised mechanism $\mathscr{M} : X \to R$, mapping a domain X to a range R, satisfies (ε, δ)-DP if for any two neighboring datasets $\mathscr{D}_i, \mathscr{D}_i' \in X$ and for all measurable sets $S \subseteq R$, the following inequality holds [82]:

$$\Pr[\mathscr{M}(\mathscr{D}_i) \in \mathscr{S}] \leq e^{\varepsilon} \Pr[\mathscr{M}(\mathscr{D}_i') \in \mathscr{S}] + \delta. \tag{3.6}$$

This definition ensures that the inclusion or exclusion of any individual data point in $\mathscr{D}_i$ does not significantly change the outcome of the mechanism $\mathscr{M}$, thus protecting the privacy of individual data points. In our proposed SDTA mechanism, DP is applied during the local model update phase by adding Gaussian noise to the updates before they are sent to the federated server, ensuring robust privacy protection while maintaining model performance.

Gaussian Mechanism for Differential Privacy

For numerical data in FL, one of the commonly used methods to ensure (ε, δ)-DP is the Gaussian mechanism, which involves adding Gaussian noise to the outputs of the function [82, 90]. This mechanism is particularly suitable for preserving privacy while performing continuous-valued operations, such as gradient updates in FL. Let $n \sim \mathscr{N}(0, \sigma^2)$ represent the Gaussian noise added to the local updates before they are transmitted to the FS. The noise distribution ensures the resulting updates maintain the (ε, δ)-DP guarantee. To achieve this, the noise variance σ^2 must be carefully chosen based on the privacy budget ε and δ [82]. According to the DP theory, the noise scale σ is determined as $\sigma \geq \frac{c\Delta s}{\varepsilon}$, where Δs is the sensitivity of the function s, defined as given as $\Delta s = \max_{\mathscr{D}_i, \mathscr{D}_i'} \|s(\mathscr{D}_i) - s(\mathscr{D}_i')\|$. The sensitivity Δs represents the maximum change in the output of the function s when a single data point is modified in the dataset. The constant c is computed as $c \geq \sqrt{2\ln\left(\frac{1.25}{\delta}\right)}$, and ensures that the noise scale is sufficient to preserve privacy within the given bounds of ε and δ. Introducing DP into the proposed SDTA mechanism allows for robust privacy preservation during model aggregation in FL while still facilitating collaborative model improvement. However, the choice of ε and σ remains a critical area for further research as it directly influences privacy and performance.

3.2.7 Adversarial Attacks

In FL, malicious clients can pose significant threats by trying to corrupt the global model ω^t that all ENs share. One of the most common adversarial strategies is model poisoning, where attackers alter their local model updates to diminish the global model's performance [132]. This can result in decreased accuracy on specific tasks or embedding backdoor vulnerabilities.

Threat Model: Model Leakage and Poisoning

This threat model considers a scenario where a malicious attacker controls a subset of clients. The attacker aims to poison the global model ω^t by crafting malicious updates $\Delta\omega_k^t$ that cause the model to perform poorly on specific tasks while still appearing effective on other data samples to avoid detection. Let ω^{t-1} represent the global model at iteration $t-1$, and $\Delta\omega_k^t$ be the malicious update from client k. The attacker's objective can be formulated as $\text{Attack}(\omega^{t-1}, \Delta\omega_k^t) = \max \ell(\omega^t(\omega^{t-1} + \Delta\omega_k^t), \mathscr{D}_{\text{test}})$, where ℓ is the loss function measuring the model's performance on the test dataset $\mathscr{D}_{\text{test}}$, and ω^t is the global model obtained by aggregating the updates from all clients, including the attacker's poisoned update $\Delta\omega_k^t$ [132].

We focused on the model flipping attack, where the attacker alters the sign of their local update before sending it to the FS. In this attack, the adversarial client submits the following update:

$$\Delta\omega_k^t = -\nabla_\omega \ell_k(\omega^t, \mathscr{D}_k). \tag{3.7}$$

By flipping the gradient sign, the attacker attempts to reverse the effect of their local update, thus misguiding the global model ω^t during aggregation. This type of attack can subtly degrade the performance of the global model without being immediately detected.

3.3 Proposed SDTA

In FL, aggregating model updates from ENs is critical for maintaining the performance of the global model. Traditional aggregation methods, such as FedAVG, assume equal contributions from all ENs, making them susceptible to noisy updates or outliers from ENs with diverse local data. These misaligned updates can significantly degrade the performance, especially in heterogeneous data environments [133]. To address these challenges, we propose the SDTA algorithm, which selectively aggregates EN updates by measuring their similarity, ranking them, and discarding the least aligned updates before aggregation. The proposed SDTA algorithm involves three steps, which include i) layer-wise similarity computation and ranking, ii) truncation of misaligned updates, and iii) layerwise aggregation.

The following subsections provide a detailed description of each step, including the mathematical formulations and the optimisation of the truncation percentage.

3.3.1 Layer-wise Similarity Computation and Ranking

At each communication round t, the FS receives local model updates $\Delta\omega_i^{t,l}$ for each layer $l \in \{1,2,\ldots,L\}$ from all N ENs. The local update $\Delta\omega_i^{t,l}$ is the difference between the updated local model weights and the previous global model weights at layer l given as $\Delta\omega_i^{t,l} = \omega_i^{t,l} - \omega^{t,l}$. To measure the alignment of updates from different ENs, the FS computes the cosine similarity

between every pair of ENs (i, j) for each layer l mathematically given as:

$$S_{(i,j)}^{t,l} = \frac{\langle \Delta\omega_i^{t,l}, \Delta\omega_j^{t,l} \rangle}{\|\Delta\omega_i^{t,l}\|_2 \, \|\Delta\omega_j^{t,l}\|_2}, \tag{3.8}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors, and $\| \cdot \|_2$ denotes the Euclidean norm (2-norm) of a vector.

The cosine similarity $S_{(i,j)}^{t,l}$ measures the directional alignment between the updates of EN i and EN j for layer l. A higher value of $S_{(i,j)}^{t,l}$ (closer to 1) indicates that the updates are more aligned, while a lower value (closer to -1) indicates misalignment. The FS constructs a similarity matrix $\mathbf{S}^{t,l} \in \mathbb{R}^{N \times N}$ for each layer l, where each element $S_{(i,j)}^{t,l}$ captures the pairwise similarity between ENs. Cosine similarity is chosen over other distance metrics due to its sensitivity to the directional alignment of updates rather than their magnitude. This is crucial in FL, where the magnitude of updates may vary significantly across ENs due to data heterogeneity, while the direction of updates is a more reliable indicator of alignment [132].

Average Similarity Score Calculation

For each EN i at layer l, the FS calculates the average similarity score $\bar{S}_i^{t,l}$ with respect to all other ENs:

$$\bar{S}_i^{t,l} = \frac{1}{N-1} \sum_{\substack{j=1 \\ j \neq i}}^{N} S_{i,j}^{t,l}. \tag{3.9}$$

This score represents how well the updates from EN i align with the updates from other ENs at layer l. A higher $\bar{S}_i^{t,l}$ indicates that EN i's update is more consistent with the majority of the ENs, whereas a lower value suggests that the update is divergent or noisy.

Ranking of Edge Nodes

After computing the average similarity scores $\bar{S}_i^{t,l}$ for each EN i at layer l, the FS ranks the ENs for each layer l based on their scores $\bar{S}_i^{t,l}$ in decreasing order. Specifically, the FS arranges the ENs in decreasing order of their average similarity scores as follows:

$$\bar{S}_{(i_1)}^{t,l} \geq \bar{S}_{(i_2)}^{t,l} \geq \cdots \geq \bar{S}_{(i_N)}^{t,l}, \tag{3.10}$$

where $i_1, i_2, \ldots, i_N$ represent the indices of the ENs ordered by their similarity scores. This ranking allows the FS to prioritise the updates from ENs that are more aligned with the overall update direction.

3.3.2 Truncation of Misaligned Updates

To enhance the robustness of the global model, the SDTA algorithm discards a fraction of ENs with the lowest similarity scores for each layer. Specifically, for each layer l, a predefined truncation percentage $Z\%$ is applied, and the FS discards the bottom $Z\%$ of ENs based on their similarity rankings. The number of discarded ENs is calculated as:

$$N_{\mathrm{Tr}}^{l} = \left\lfloor \frac{Z}{100} \times N^{l} \right\rfloor, \tag{3.11}$$

where N^l is the total number of ENs at layer l. The remaining $N_{\mathrm{R}}^{l} = N^{l} - N_{\mathrm{Tr}}^{l}$, ENs will contribute to the aggregation process. This truncation step ensures that updates from misaligned or potentially malicious ENs are excluded, thereby mitigating the influence of outliers or noisy data on the global model.

3.3.3 Layer-wise Aggregation and Global Model Update

After truncation, the FS aggregates the updates from the remaining ENs for each layer using a simple averaging rule. For each layer l, the aggregated update is computed as:

$$\Delta\omega^{t,l} = \frac{1}{N_{\mathrm{R}}^{l}} \sum_{i \in \mathscr{I}_{\mathrm{R}}} \Delta\omega_{i}^{t,l}, \tag{3.12}$$

where $\mathscr{I}_{\mathrm{R}}$ denotes the set of ENs that were not truncated for layer l. The global model is then updated using the aggregated updates for each layer l:

$$\omega^{t+1,l} = \omega^{t,l} + \eta\Delta\omega^{t,l}, \tag{3.13}$$

where η is the learning rate. By aggregating only the updates from the most aligned ENs, the global model update becomes more robust to noise and heterogeneity in the data across ENs. The pseudocode for the proposed SDTA algorithm is given in Algorithm. 3.

3.3.4 Optimisation of Truncation

In the proposed SDTA algorithm, the truncation percentage $Z\%$ is critical in balancing between excluding noisy or misaligned updates from ENs and retaining the most important contributions. Properly tuning $Z\%$ is crucial for improving the robustness and performance of the global model in FL setup.

Problem Formulation

The objective of the optimisation process is to find the optimal truncation percentage $Z^{*}\%$, which minimizes the global loss function $\ell(\omega^{t+1})$. In this context, the global loss function $\ell(\omega^{t+1})$

Algorithm 3: Similarity-Driven Truncated Aggregation (SDTA)

Result: Optimised global model ω^T
Input: Local datasets $\mathscr{D}_i$ for EN i, Total number of ENs N, Number of layers L in the model, Number of communication rounds T, Truncation percentage $Z\%$
Initialise global model weights ω^0
for $t = 1$ *to* T **do**
  for *each EN* $i = 1$ *to* N ***in parallel*** **do**
    Send global model ω^t to EN i
    EN i performs local training and computes local model update $\Delta\omega_i^{t,l}$ for each layer l
    ENs send local updates $\Delta\omega_i^{t+1,l}$ to the FS
  end
  for *each layer* l **do**
    for *all pairs of ENs* (i, j), $i \neq j$ **do**
      Compute cosine similarity $S_{i,j}^l$ for the l-th layer updates, as in Eq. (4.14)
    end
    Calculate average similarity score $\bar{S}_i^l$ for each EN i, as in Eq. (3.9)
  end
  Rank ENs in decreasing order of similarity scores $\bar{S}_i^l$
  Apply layer-wise truncation $Z\%$ of ENs with the lowest similarity as per Eq. (3.11)
  Aggregate the remaining updates for each layer l, as in Eq. (3.12)
  Update global model $\omega^{t+1,l} = \omega^{t,l} + \eta\Delta\omega^{t,l}$ for each layer l
end
Return final global model ω^T

is defined as the MAE, a common evaluation metric for STLF. The optimisation problem is mathematically formulated as follows:

$$\min_{Z\in[0,Z_{\max}]} \ell(\omega^{t+1}(Z)), \tag{3.14}$$

subject to the constraint:

$$0 \leq Z \leq Z_{\max}, \tag{3.15}$$

where $\ell(\omega^{t+1}(Z))$ is the global loss after applying truncation percentage $Z\%$, $Z = 0$, $Z_{\max}$ is the maximum allowable truncation percentage, set to 50%, ensuring that at least half of the EN updates are retained during aggregation.

Grid search is applied to efficiently explore the search space of possible truncation percentages $Z\%$. Instead of using the entire dataset for each EN in every communication round, a representative subset of the dataset is used for a limited number of communication rounds. This enables us to estimate the trend of the global loss function $\ell(\omega^{t+1})$ without incurring the computational cost of a full-scale training process. The grid search procedure is outlined as follows:

- Each EN uses a random subset of its local dataset $\mathscr{D}_i$ during the grid search process. This subset is selected to be representative of the full dataset, ensuring that the performance trends observed in the subset align with those observed over the entire dataset. The SDTA algorithm is run for a fixed number of communication rounds, T_s, using these subsets to evaluate the impact of different truncation percentages $Z\%$ on the global loss function.

- A predefined set of truncation percentages is chosen to cover the search space:

$$Z \in \{0\%, 10\%, 20\%, 30\%, 40\%, 50\%\}.$$

  This range provides a comprehensive evaluation of model performance across varying truncation levels.

- After the global model $\omega^{t+1}(Z)$ is obtained for each $Z\%$, the global loss function $\ell(\omega^{t+1}(Z))$ is evaluated on the validation dataset $\mathscr{D}_{\text{val}}$.

- By tracking the global loss function for different truncation percentages, the behaviour of the model as a function of $Z\%$ can be observed. This analysis allows us to determine how sensitive the model performance is to various truncation levels, helping us identify the truncation percentage that minimizes the global loss.

Finally, the optimal truncation percentage $Z^*\%$ is selected as the value that minimises the global loss function. This ensures that the SDTA algorithm retains only the most aligned EN updates, excluding noisy or misaligned contributions, to optimize the global model performance. Using subsets of the dataset and a limited number of communication rounds allows for a computationally efficient method to identify the optimal truncation percentage. This approach ensures that the loss function trend is captured accurately while reducing the computational overhead typically associated with running a full grid search across all EN datasets.

3.4 Simulation Setup

In this work, our focus was on developing an efficient STLF at the substation level. We adopted the cross-silo FL architecture and considered the small number of trusted users for model training. Since this study is intended to provide a practical and scalable framework for STLF, a diverse dataset with historical load profiles, weather, and contextual information was needed to evaluate the performance. Therefore, the ISO-New England dataset is used, containing the hourly reading of the historical load profile at the substation level with weather information [178]. The details of data distribution are discussed in the subsequent subsection.

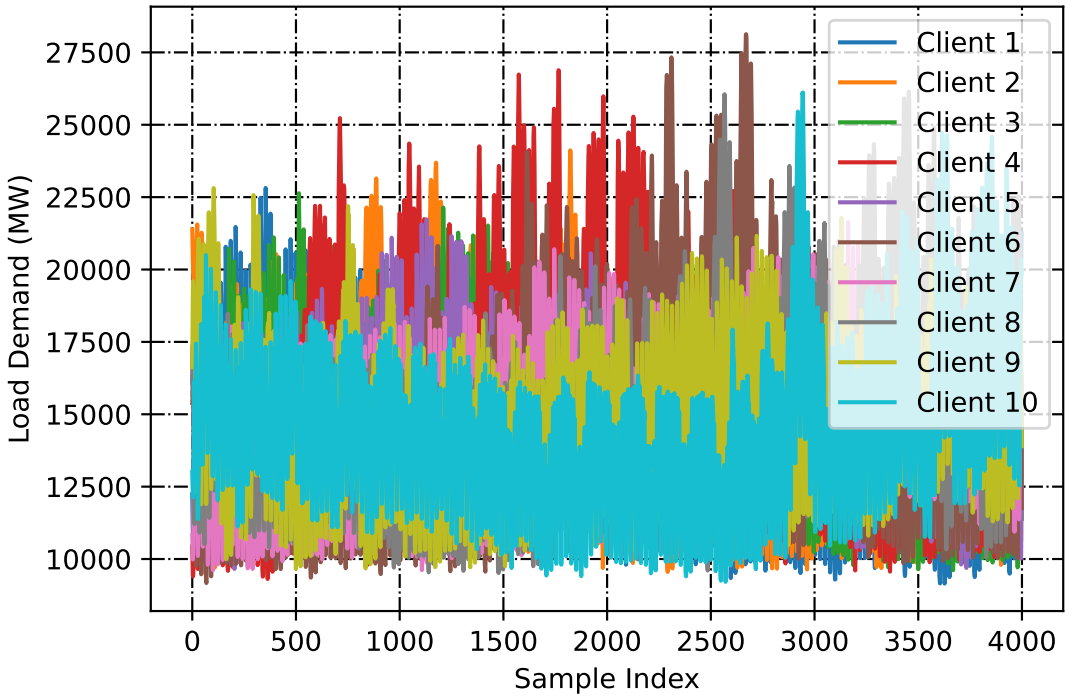


Figure 3.3: A sample plot of the dataset for all ENs showing varying peaks.

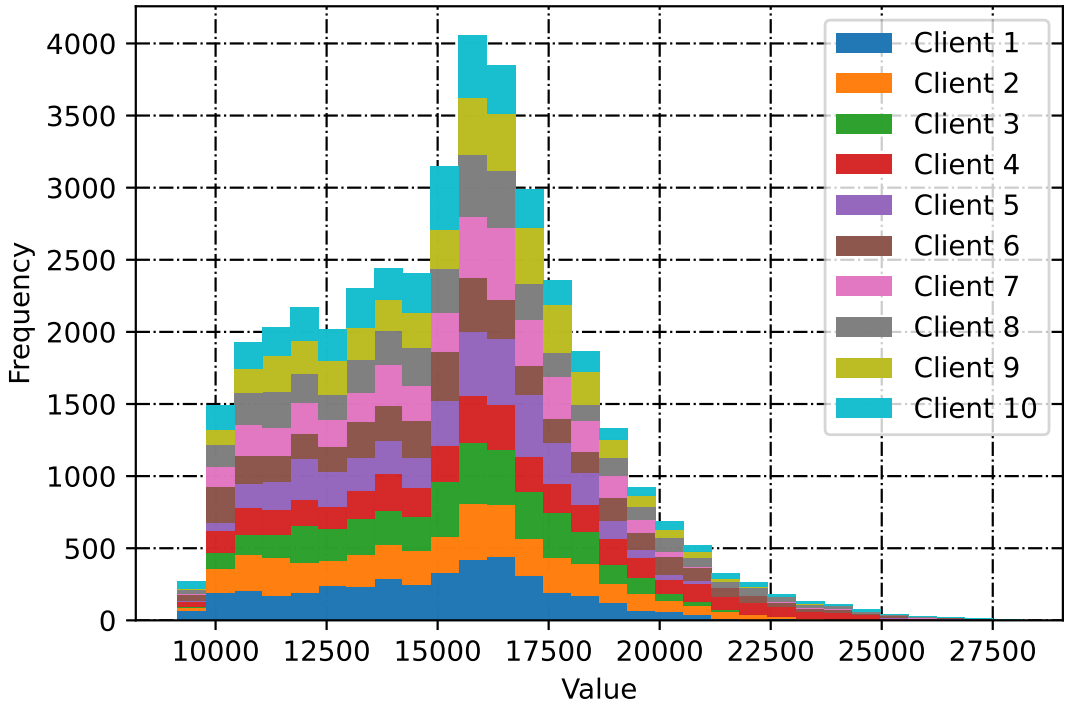


Figure 3.4: The histogram showing right-skewed data distribution ENs.

3.4.1 Dataset Description

This study used the ISO-New England dataset, which provides comprehensive hourly load profiles at the substation level, spanning five years from January 2009 to December 2013.

The dataset includes weather-related variables and contextual information essential for accurate STLF. We partitioned the dataset into 10 subsets to simulate FL settings, with each subset representing an edge node (EN) capable of local training. Each EN receives data for six months (26 weeks), with 80% allocated for training and 20% for testing. Finally, a combined global test set is created consisting of all local test sets, offering a diverse representation of energy consumption patterns to test the robustness of the proposed scheme.

The underlying rationale for this structured data distribution is two-fold: first, it ensures each client accesses a diverse data spectrum; second, it provides insight into how data distribution influences the global model's performance. This distribution approach mirrors real-world FL environments where data diversity is a key challenge. Before starting the training process, extensive data preprocessing is performed to extract handcrafted features, enhancing the effectiveness of local training. For a detailed explanation of feature extraction and data preprocessing, please refer to Section 3.2.3.

A sample plot of the dataset is shown in Fig. 3.3 to demonstrate the diversity of the data. Additionally, Fig. 3.4 visualises the data distribution across clients using histograms, revealing a right-skewed distribution that deviates from the normal distribution typically assumed in statistical models. The skewness suggests that many observations fall below the mean, reflecting peak loads in shorter intervals at substations. This inherent data diversity justifies the need for specific statistical tests to assess variance across clients, as such variance can affect the global model's accuracy.

3.4.2 Data Diversity Test

The dataset is right-skewed and deviates from a normal distribution; hence, two statistical tests, Levene's and Fligner-Killeen, were performed to check for diversity [179, 180].

Levene's Test

Levene's Test is a widely used method for testing the homogeneity of variances across multiple groups. It is particularly effective when the assumption of normality is violated, making it more suitable for the right-skewed data observed in our study. The test assesses whether the variance of load data differs significantly between clients, which is critical for FL systems since inconsistent variance can lead to biased global model updates. The mathematical formulation for Levene's Test is as follows [179]:

$$W = \frac{(N-k)}{(k-1)} \cdot \frac{\sum_{i=1}^{k} N_i (Z_{i\cdot} - Z_{\cdot\cdot})^2}{\sum_{i=1}^{k} \sum_{j=1}^{N_i} (Z_{ij} - Z_{i\cdot})^2}, \tag{3.16}$$

where N is the total number of data points, k is the number of groups (in this case, ENs), $Z_{ij} = |X_{ij} - \tilde{X}_i|$ is the absolute deviation of observation X_{ij} from its group median $\tilde{X}_i$, $Z_{i\cdot}$ is the mean

deviation for group i, and $Z_{..}$ is the overall mean deviation. The null hypothesis (H_0) of Levene's Test states that the variances are equal across all groups. A low p-value (typically < 0.05) leads us to reject the null hypothesis, indicating significant variance differences between groups. We performed pairwise Levene's Tests across all clients for our dataset and computed the p-values. The resulting matrix is presented in Table. 3.2, highlighting the statistical significance of variance differences between each pair of clients. The diagonal values in the matrix are NaN (Not a Number), as these represent the variance comparison of a client with itself, which is not meaningful. The matrix demonstrates substantial diversity across ENs, as most client pairs exhibit p-values far below the 0.05 threshold, suggesting significant variance differences. This indicates that some clients contribute disproportionately large or small updates to the global model, potentially introducing bias in the aggregated model.

Table 3.2: Pairwise Levene's Test p-value Matrix for ENs

	EN 1	**EN 2**	**EN 3**	**EN 4**	**EN 5**	**EN 6**	**EN 7**	**EN 8**	**EN 9**	**EN 10**
EN 1	NaN	1.000e-13	7.06e-05	1.16e-62	1.81e-09	1.97e-36	6.38e-03	2.33e-13	0.609	2.83e-02
EN 2	1.00e-13	NaN	3.94e-30	1.57e-23	3.66e-41	2.27e-09	2.85e-24	0.674	1.16e-15	2.08e-06
EN 3	7.06e-05	3.94e-30	NaN	5.40e-91	4.11e-02	3.52e-58	0.212	5.30e-28	4.67e-04	4.72e-09
EN 4	1.16e-62	1.57e-23	5.40e-91	NaN	1.99e-107	2.70e-04	1.33e-81	1.00e-19	9.75e-67	1.33e-43
EN 5	1.81e-09	3.66e-41	4.11e-02	1.99e-107	NaN	3.38e-71	9.97e-04	1.21e-37	2.70e-08	1.14e-14
EN 6	1.97e-36	2.27e-09	3.52e-58	2.70e-04	3.38e-71	NaN	7.49e-51	1.27e-07	2.30e-39	1.90e-23
EN 7	6.38e-03	2.85e-24	0.212	1.33e-81	9.97e-04	7.49e-51	NaN	8.04e-23	2.51e-02	2.46e-06
EN 8	2.33e-13	0.674	5.30e-28	1.00e-19	1.21e-37	1.27e-07	8.04e-23	NaN	4.28e-15	1.24e-06
EN 9	0.609	1.16e-15	4.67e-04	9.75e-67	2.70e-08	2.30e-39	2.51e-02	4.28e-15	NaN	7.34e-03
EN 10	2.83e-02	2.08e-06	4.72e-09	1.33e-43	1.14e-14	1.90e-23	2.46e-06	1.24e-06	7.34e-03	NaN

Fligner-Killeen Test

The second test performed is the Fligner-Killeen Test, which involves transforming the data based on ranks and medians within each group, making it highly effective for non-normal data distributions like those observed in our dataset. This method calculates a test statistic by ranking the observations and evaluating the deviations from the median ranks within each group. The formula for the test statistic is given as [180]:

$$X^2 = \frac{12}{N(N+1)} \left[\sum_{i=1}^{k} N_i \left(R_{i\cdot} - \frac{N+1}{2} \right)^2 \right] - \text{Correction Factor}, \tag{3.17}$$

where N represents the total number of data points across all groups, k is the number of groups (in our case, ENs), N_i denotes the number of observations in each group, and $R_{i\cdot}$ is the sum of ranks for the i-th group.

The p-values obtained from our pairwise Fligner-Killeen tests, presented in Table 3.3, reveal key insights about the variance diversity between different ENs. Several EN pairs, such as EN 1 and EN 4 (p-value = 9.38e-66) and EN 4 and EN 10 (p-value = 3.88e-44), exhibit highly significant variance differences, further reinforcing the diversity discovered by Levene's Test.

These results underscore the need to account for these differences in the model aggregation process.

Table 3.3: Pairwise Fligner-Killeen Test p-value Matrix for ENs

	EN 1	**EN 2**	**EN 3**	**EN 4**	**EN 5**	**EN 6**	**EN 7**	**EN 8**	**EN 9**	**EN 10**
EN 1	NaN	9.188e-14	7.36e-06	9.38e-66	9.47e-12	1.87e-35	1.06e-03	8.58e-13	4.23e-01	3.55e-02
EN 2	9.19e-14	NaN	4.21e-33	7.27e-28	1.33e-45	8.50e-10	5.15e-26	5.10e-01	5.15e-17	4.06e-06
EN 3	7.36e-06	4.21e-33	NaN	1.07e-93	1.82e-02	2.24e-59	1.83e-01	6.78e-29	1.41e-04	3.36e-09
EN 4	9.38e-66	7.27e-28	1.07e-93	NaN	1.13e-108	3.50e-05	5.60e-85	4.28e-21	2.20e-70	3.88e-44
EN 5	9.47e-12	1.33e-45	1.82e-02	1.13e-108	NaN	5.98e-73	1.49e-04	8.67e-39	5.69e-10	5.63e-15
EN 6	1.87e-35	8.50e-10	2.24e-59	3.50e-05	5.98e-73	NaN	4.71e-51	6.22e-08	1.24e-39	8.27e-24
EN 7	1.06e-03	5.15e-26	1.83e-01	5.60e-85	1.49e-04	4.71e-51	NaN	1.10e-22	1.17e-02	2.74e-06
EN 8	8.58e-13	5.10e-01	6.78e-29	4.28e-21	8.67e-39	6.22e-08	1.10e-22	NaN	2.36e-15	1.56e-06
EN 9	4.23e-01	5.15e-17	1.41e-04	2.20e-70	5.69e-10	1.24e-39	1.17e-02	2.36e-15	NaN	7.26e-03
EN 10	3.55e-02	4.06e-06	3.36e-09	3.88e-44	5.63e-15	8.27e-24	2.74e-06	1.56e-06	7.26e-03	NaN

3.4.3 Performance Metrics

To assess the performance of the STLF problem, the most commonly used metrics in traditional ML are mean absolute percentage error (MAPE) and mean absolute error (MAE), which are discussed in 2.2.1 and given in equ. (2.19 and 2.20). Together, these metrics provide a comprehensive understanding of model performance. MAE focuses on the absolute error without considering the magnitude of the actual values, while MAPE accounts for the relative error, offering a percentage-based view of forecasting accuracy.

In addition to error metrics, evaluating computational complexity is critical to understanding the scalability and efficiency of the proposed framework, particularly in resource-constrained environments. The computational complexity is quantified using floating point operations (FLOPs), which measures the operations required during each communication round as discussed in Section 2.2.3 as given in equ. (2.14). Local training F_{Local}, includes the forward and backward passes of the DL model during training. For an LSTM model used in this study, the computational cost is proportional to the number of neurons, layers, and training samples processed at each EN. Let S represent the sample size, H the hidden units in LSTM, and L the number of layers, than F_{local} is calculated as:

$$F_{\text{Local}} = O(S \times H^2 \times L). \tag{3.18}$$

Aggregation F_{Agg} is the cost of combining the local update depending on the number of participating ENs (N) and the model size (ω, representing the number of parameters):

$$F_{\text{Agg}} = O(N \times \omega). \tag{3.19}$$

Finally, algorithm-specific operations (F_{Algo}) depend on the type of aggregation mechanism used. For instance, the cost is limited to coordinate-wise averaging of model weights for Fe-

dAVG. However, FedDist has additional computations for distance calculations and neuron addition, and the proposed SDTA includes cosine similarity computations and truncation operations, which adds an extra cost. For L layers and N participating clients, the complexity is:

$$F_{\text{FedDist}} = O(2 \times N \times \omega \times L), F_{\text{SDTA}} = O(L \times N^2). \tag{3.20}$$

3.4.4 Performance Evaluation Strategy

We conducted all experimental procedures in a simulated environment to evaluate the performance of our proposed framework thoroughly under different scenarios. These simulations assessed the global generalisation capabilities and the personalised performance of models trained using FL. The global performance was evaluated using a global test set, while the personalised performance was assessed using the locally fine-tuned models at each EN. We calculated the MAPE and MAE for each model using the test data specific to each EN, ensuring a robust comparison across varying conditions.

3.4.5 Simulation Scenarios:

- **FL vs Centralised Learning:** In the first scenario, the basic performance of the proposed SDTA algorithm is evaluated in an FL setup. This was compared against centralised learning and two commonly used FL algorithms, FedAVG and FedDist. This comparison enabled us to investigate the effectiveness of SDTA in handling the challenges of diverse data distributions across ENs. .

- **Incorporating DP**: We incorporated DP based on the (ε, δ)-differential privacy framework to ensure data privacy during model updates. This mechanism controls the amount of noise added to model updates, ensuring that individual client data remains protected. The privacy budget ε controls the level of privacy, which has an inverse relation with privacy level. The larger ε values provide low noise, resulting in weaker privacy but better performance, while smaller ε values offer stronger privacy guarantees at the cost of accuracy. The parameter δ, set to 10^{-5}, accounts for the small probability that the privacy loss might exceed ε. We tested three scenarios where low privacy ε ranges between 6 and 8, providing lower noise, thus a better model performance with weaker privacy guarantee [82]. In medium privacy, the ε range is kept between 0 and 1, balancing privacy and accuracy. Finally, high privacy, where ε is kept at 0.1, offers strong privacy protection at the expense of reduced model performance.

- **Testing Under Diverse Conditions:** Beyond privacy and security, we explored the resilience of our proposed algorithm under various challenging conditions, such as random

client selection and adversarial attacks. In random client selection, a scenario is simulated where a subset of ENs are randomly selected in training, testing the algorithm's ability to maintain accuracy even when fewer clients contribute updates. This scenario is crucial in practical applications where communication failures or client unavailability may occur. Furthermore, we intentionally dropped some EN with similar patterns and used transfer learning to fine-tune them to reduce the communication cost. We also tested the algorithm's performance in adversarial attacks, where some clients send corrupted updates to disrupt the model. This scenario tested the proposed SDTA's ability to detect and mitigate the impact of such adversarial behaviour, comparing its performance with FedAVG and FedDist.

By incorporating DP mechanisms and adversarial defence strategies, our performance evaluation strategy ensures that the SDTA framework is robust and resilient to privacy threats and operational challenges in FL environments.

3.5 Results and Discussions

To evaluate the effectiveness of the proposed SDTA framework for STLF, we conducted a comprehensive comparative analysis against two widely employed FL algorithms: FedAVG and FedDist. This evaluation aims to demonstrate how SDTA addresses the challenges inherent in FL, such as data diversity and privacy preservation while maintaining high model performance. This study treated the STLF task as a multivariate regression problem, utilising handcrafted features derived from historical load profiles and weather data to predict hourly energy consumption.

Local training was conducted using a DNN model, as Section 3.2.3 outlined. The DNN model is purposefully kept simple to reduce computational complexity and minimise model size, ensuring scalability. For the comparative analysis, the performance of the global federated model is benchmarked against a centralised learning model, while the personalised models are compared to local training on individual ENs. During the local training phase, each EN refined its model independently using its local dataset, with no communication or parameter sharing among the nodes.

The hyperparameters for local and FL models were standardised to ensure a fair and consistent evaluation. In the FL setup, each EN trained locally for 20 epochs per communication round, with 500 communication rounds in total. The ADAM optimiser was employed with a local batch size 32, ensuring efficient convergence across nodes with heterogeneous data distributions.

In obtaining the optimal percentage of truncation value %Z, a grid search is performed for predefined values of Z as discussed in Section 3.3.4. We used a subset of data and trained the model for 100 communication rounds to check the trend of a learning curve using our proposed SDTA algorithm. The results for our initial analysis are presented in Fig. 3.5, and the curve is

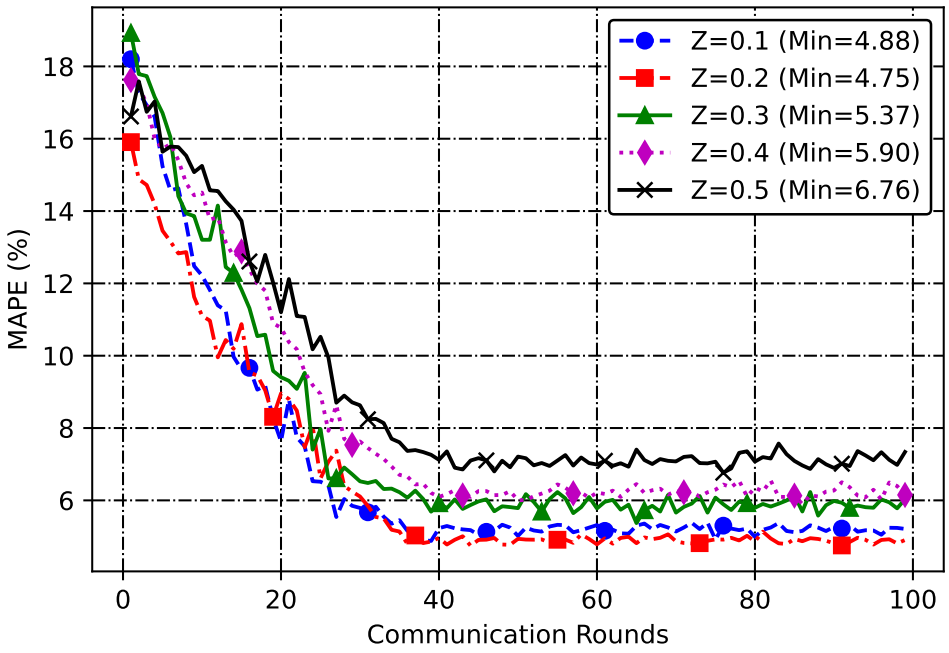


Figure 3.5: FL learning curve for percentage truncation value Z and MAPE using the subset of training data.

plotted against different percentages of Z values. The results showed that the optimal value for Z is 0.2, i.e., 20%, where the MAPE value is minimised. Therefore, a 20% truncation ratio is adopted in all subsequent experiments to enhance model alignment and mitigate the impact of data heterogeneity across ENs.

3.5.1 Centralised vs FL learning

In the initial phase of our evaluation, we compared the performance of centralised learning to that of FL, focusing on assessing the capabilities of the proposed SDTA framework. To isolate the impact of aggregation techniques, an ideal FL environment is assumed where all ENs updates are noise-free, allowing us to evaluate actual performance. Table 3.4 summarises the results, including key metrics such as MAE, MAPE, and the number of communication rounds required for model convergence.

As expected, centralised learning with direct access to aggregated data yielded the best performance, achieving an MAE of 290.4 and a MAPE of 1.96%. The number of communication rounds is not applicable (NA) for centralised learning, as the data is processed centrally without needing federated updates. However, centralised learning is rarely feasible in real-world applications due to strict data privacy regulations and the logistical challenges of aggregating data from disparate sources.

In the federated setting, the performance of FedAVG and FedDist underscores the challenges posed by data diversity across ENs. FedAVG recorded an MAE of 464.03 and a MAPE of

Table 3.4: Comparative results of centralised training and globalised models using different aggregation techniques.

Algorithm	MAE	MAPE	No of Rounds
Centralised	290.4	1.96	NA
FedAVG	464.03	3.11	225
FedDist	426.03	2.89	239
Proposed-SDTA	**380.69**	**2.63**	**209**

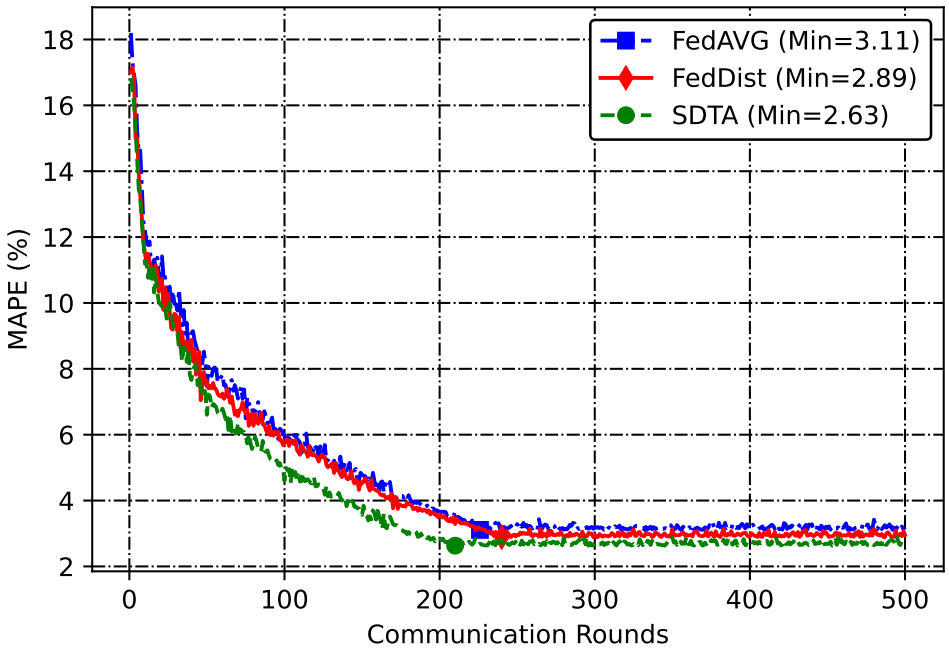


Figure 3.6: FL learning curve based on MAE as loss function for each communication round during the training process.

3.11%, reflecting the difficulties in aggregating models trained in diverse data. FedDist slightly improved performance by incorporating a distance-based weighting mechanism, with an MAE of 426.03 and a MAPE of 2.89%. The proposed SDTA framework, however, demonstrated a clear improvement over both FedAVG and FedDist, achieving an MAE of 380.69 and a MAPE of 2.63%. This can be attributed to SDTA's layer-wise cosine similarity scoring and truncated aggregation, which enhance model alignment and mitigate the impact of data diversity across ENs. Moreover, SDTA converged faster, requiring only 209 communication rounds compared to 225 for FedAVG and 239 for FedDist, reflecting its superior efficiency in federated environments.

The results in Fig. 3.6 present the learning curve of loss function MAPE in each communication round generated using the global test set during the training process. This curve exhibits a classical learning behaviour with a steep reduction in MAPE in initial rounds until it reaches the

Table 3.5: Complexity analysis of federated algorithms per round.

Algorithm	F_{Local} (FLOPs)	F_{Agg} (FLOPs)	F_{Algo} (FLOPs)	F_{Total} (FLOPs)
FedAVG	93.84×10^9	$586,500$	0	93.8406×10^9
FedDist	93.84×10^9	$586,500$	5.238×10^6	93.846×10^9
SDTA	93.84×10^9	$469,200$	5.238×10^6	93.845×10^9

slow monotonic decrease after a few initial rounds. The results show that the proposed SDTA has a minimum MAPE of 2.63% and converges early compared to the FedAVG and FedDist algorithms. Furthermore, the rapid decrease in MAPE loss shows the quick learning capabilities of our proposed algorithm in a federated setup.

3.5.2 Complexity Analysis Under Normal Conditions

The computational complexity of FL algorithms significantly impacts their efficiency and scalability, especially in resource-constrained distributed systems. To analyse this, we evaluate the complexity of three algorithms based on FLOPs as discussed in Section. 2.2.3. The analysis considers local training, model aggregation, and additional operations specific to each algorithm as given equ. (2.14). The existing model architecture comprises an LSTM layer, two dense layers (100 and 50 neurons), and a fully connected layer, yielding 58,650 parameters. Each client trains locally on a dataset of 4,000 samples for 20 epochs, with a batch size 32. For local training, the complexity per sample is 2 $\times$ Parameters = 117,300 FLOPs (forward and backward passes). Across all samples and epochs, the local training complexity per client is 9.38 billion FLOPs, and for 10 clients, this totals 93.84 billion FLOPs per communication round.

FedAVG involves only local training and aggregation, hence the computation cost Parameters $\times$ N = 586,500 FLOPs for 10 clients. In contrast, FedDist includes pairwise distance computation for neurons, incurring an additional cost of 5.23 million FLOPs. SDTA computes cosine similarities for layer-wise updates and ranks clients for truncation. While the similarity computation adds 5.23 million FLOPs, SDTA reduces aggregation costs to 469,200 FLOPs by truncating 20% of the least similar clients. A summary of the complexity analysis is presented in Table. 3.5.

These results were expected where the FedAVG has the lowest complexity due to its simple aggregation strategy. FedDist, while slightly more complex, incorporates layer-wise distance computations to improve robustness to heterogeneous updates. SDTA, on the other hand, balances computational cost and robustness by truncating low-quality updates, achieving a marginal reduction in aggregation overhead compared to FedDist. This analysis highlights that while FedAVG is the most computationally efficient, the proposed SDTA offers enhanced robustness against noisy or adversarial updates with only a minimal increase in complexity.

Table 3.6: Convergence time comparison for the algorithms under comparison.

Algorithm	Rounds	Training Time per Round	Total Training Time (seconds)
SDTA	209	62.31	13,007.79
FedDist	239	64.86	15,501.54
FedAVG	225	59.83	13,462.25

3.5.3 Convergence Analysis

The convergence time for FL algorithms is critical for evaluating their practical feasibility, especially in resource-constrained environments. This study compared the total training time required for SDTA, FedDist, and FedAVG to reach their respective minimum MAPE values under normal conditions. The average training time per communication round for SDTA, FedDist, and FedAvg was recorded as 62.31, 64.86, and 59.83 seconds, respectively. Table 3.6 presents the total convergence time for each algorithm based on the number of communication rounds required to achieve the minimum MAPE. SDTA achieved convergence in 209 rounds, while FedDist and FedAVG required 239 and 225 rounds, respectively. The results highlight that SDTA achieves the lowest MAPE in the shortest convergence time, reflecting its ability to balance computational efficiency with robustness to data heterogeneity. FedDist, despite incorporating a distance-based weighting mechanism, required longer convergence. This analysis demonstrates that SDTA offers a competitive advantage in terms of convergence efficiency, even with its added complexity for outlier rejection and layer-wise similarity computation.

3.5.4 Personalised Learning vs Local Learning

In the next phase, we evaluated the effectiveness of personalised models compared to local training. Fig. 3.7 compares MAPE across different ENs for personalised and local models. Each EN trained a model for local learning using only its local dataset, resulting in significant variation in MAPE. The lack of collaboration in local learning led to a higher average MAPE, with individual ENs exhibiting substantial performance differences based on their specific data characteristics.

In contrast, the personalised models derived from fine-tuning the global models produced by FedAVG, FedDist, and SDTA exhibited more consistent performance across ENs, achieving the lowest MAPE values across nearly all clients. For example, at EN 3, the MAPE for the proposed SDTA was approximately 1.81%, compared to 2.55% for local learning, highlighting the advantage of FL in leveraging shared knowledge from diverse ENs.While FedAVG and FedDist also benefited from personalisation, SDTA outperformed these methods due to its effective aggregation mechanism, which better captures underlying patterns across diverse datasets. By aligning model updates through cosine similarity and truncating outliers, SDTA produces a more robust global model that, when fine-tuned locally, yields superior personalised models. Finally, the

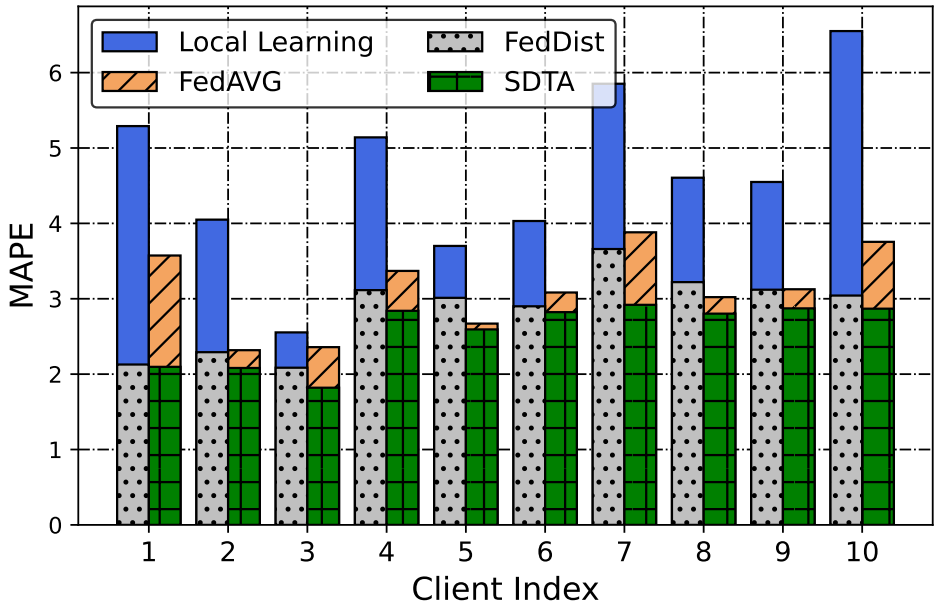


Figure 3.7: Comparative bar graph of personalised learning and local learning.

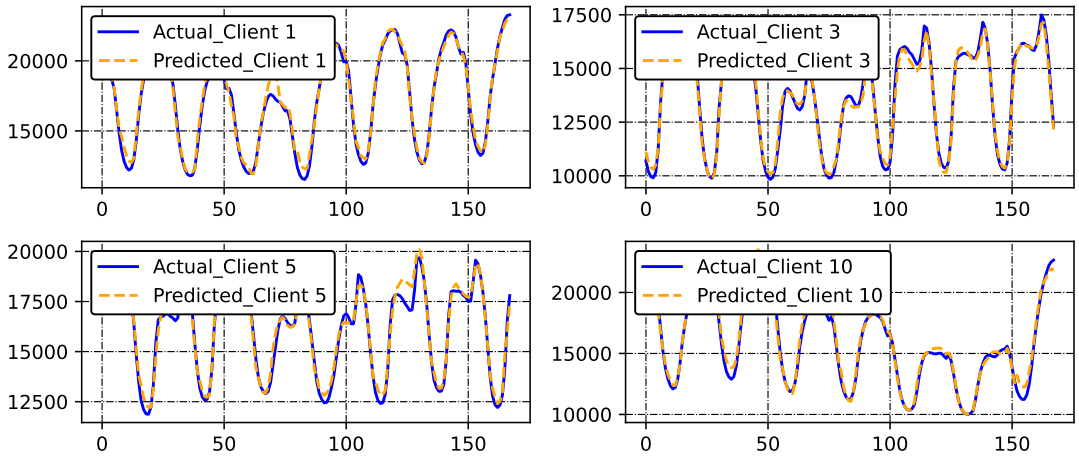


Figure 3.8: Comparison of actual vs predicted values using the personalised SDTA model.

results in Fig. 3.8 compare actual and predicted curves based on the personalised SDTA model and show a nearly good fit for actual vs. predicted values.

3.5.5 Comparison Under Differential Privacy

The experimental evaluation assesses the comparative performance of FedAVG, FedDist, and the proposed SDTA algorithm under varying DP settings. The analysis explores three privacy scenarios defined by the parameter ε, with results measured using MAPE over 500 communication rounds.

At the low privacy level $\varepsilon = 8.0$, where noise addition is small, SDTA achieves a minimum MAPE of 3.09%, surpassing FedDist 3.41% and FedAVG 3.66% as shown in Fig. 3.9.

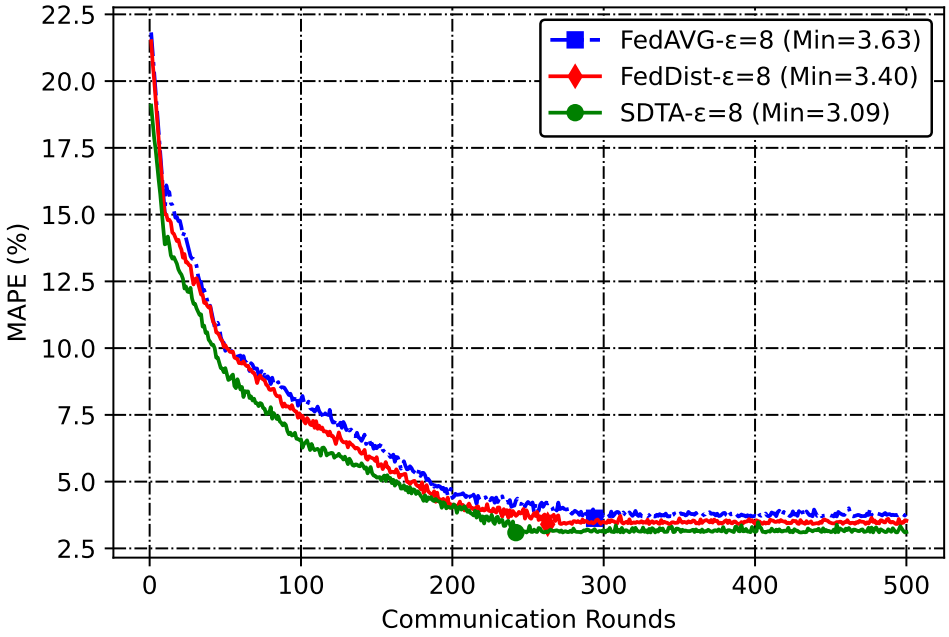


Figure 3.9: FL learning curve based on MAPE, low privacy level $\varepsilon = 8.0$ for each communication round during the training process.

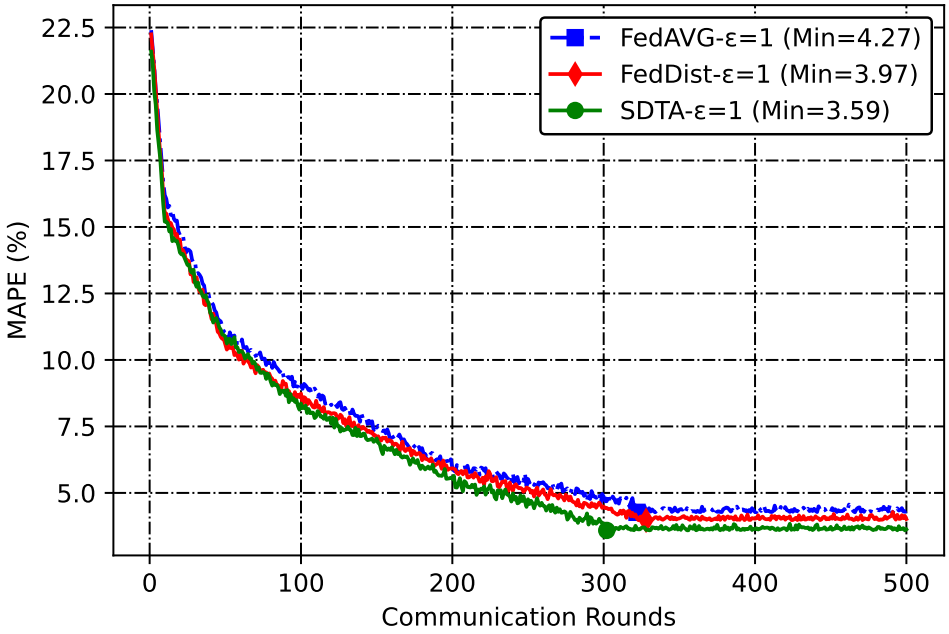


Figure 3.10: FL learning curve based on MAPE, medium level $\varepsilon = 1$ for each communication round during the training process.

This shows an improvement of approximately 9.4% and 15.6% over FedDist and FedAVG, respectively. All aggregation algorithms perform better with relaxed privacy settings with slight degradation to baseline results. However, in the medium privacy setting ($\varepsilon = 1.0$), SDTA outperforms the other algorithms, achieving a MAPE of 3.59% as shown in Fig. 3.10. In contrast,

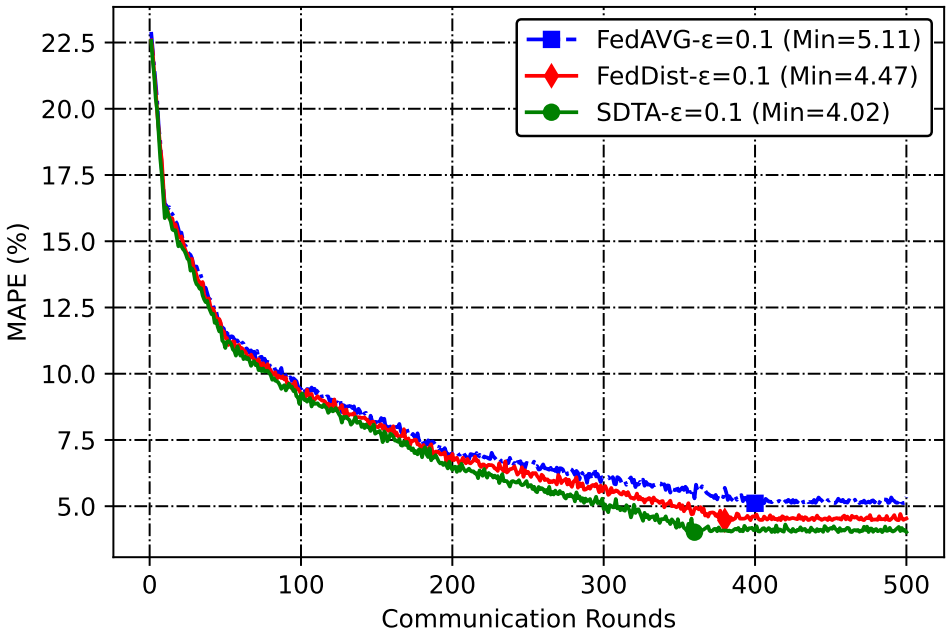


Figure 3.11: FL learning curve based on MAPE, high level $\varepsilon = 0.1$ for each communication round during the training process.

FedDist records 3.94%, and FedAVG results in 4.27%. This relative improvement of proposed SDTA over FedDist, and FedAVG demonstrates the ability to balance privacy and accuracy more effectively than the baseline methods. This trend highlights the adaptable proposed aggregation mechanism and maintains a performance advantage even as privacy constraints moderate.

Finally, in the high privacy setting $(\varepsilon = 0.1)$, SDTA achieves a minimum MAPE of 4.02%, outperforming FedDist, which records 4.45%, and FedAVG, which registers 5.05%. The results of this analysis are shown in Fig. 3.11. The results represent an improvement of 9.7% and 20.4% over FedDist and FedAVG, respectively, demonstrating the robustness of SDTA under strict privacy constraints. The superior performance indicates that incorporating similarity-driven aggregation effectively mitigates the impact of high noise levels associated with strong privacy guarantees.

3.5.6 Comparison Under Adversarial Conditions and Client Dropout

To assess the robustness of the proposed SDTA against adversarial conditions, a model sign-flipping attack is simulated, compromising 40% of ENs by inverting the sign of their model updates. Robustness against such attacks is critical in FL systems to ensure reliability and trustworthiness, especially when dealing with malicious clients or corrupted data. The results, depicted in Fig. 3.12 and Fig. 3.13, underscore the significant resilience of the proposed algorithm compared to baseline methods FedAVG and FedDist. For instance, the results in Fig. 3.12 present the convergence behaviour of SDTA, FedDist, and FedAVG plotted across 500 communication rounds in the presence of a sign-flipping attack affecting 40% of ENs. The learning

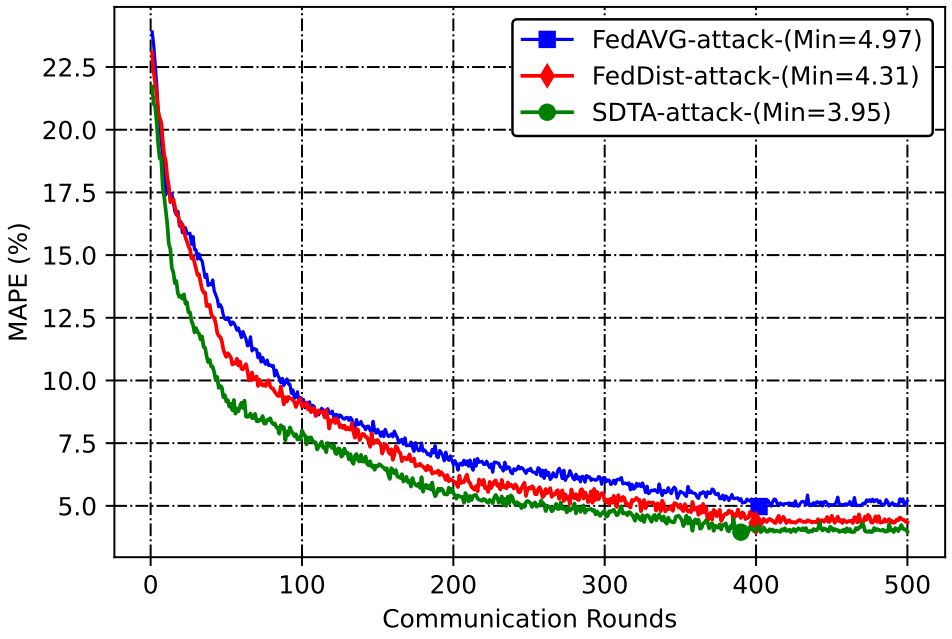


Figure 3.12: FL learning curve for model sign inversion attack, where 40% ENs are compromised.

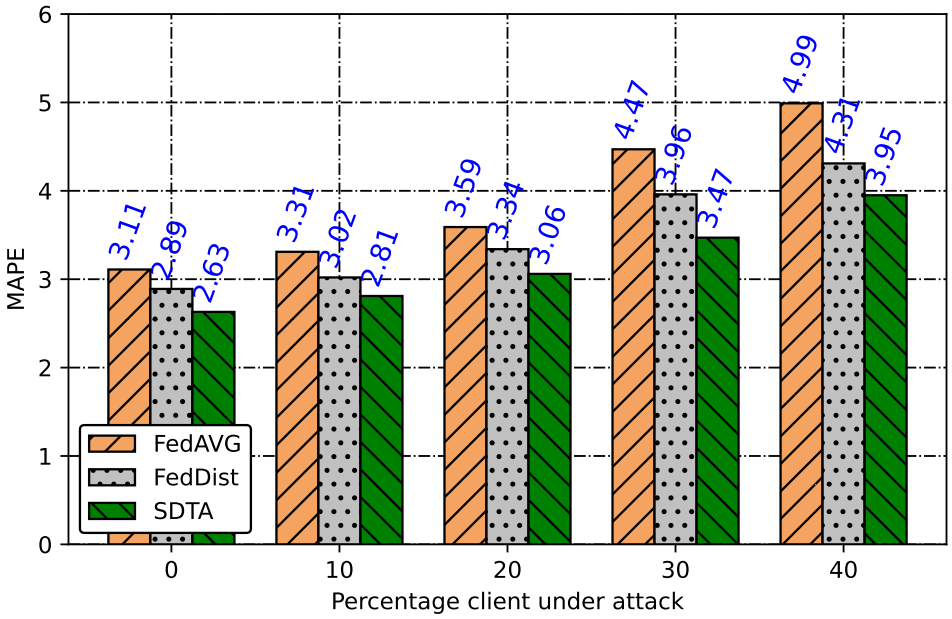


Figure 3.13: Performance comparison of FedAVG, FedDist, and SDTA under varying proportions of compromised EN (0-40%).

curves reveal that SDTA achieves a MAPE of 3.95%, which surpasses the performance of FedDist, with a MAPE of 4.31%, and FedAVG, achieving a final MAPE of 4.97%. These results indicate a substantial improvement in prediction accuracy, with SDTA outperforming FedAVG by 20.5% and FedDist by 8.4%.

These results further validate the robustness of the proposed SDTA against adversarial at-

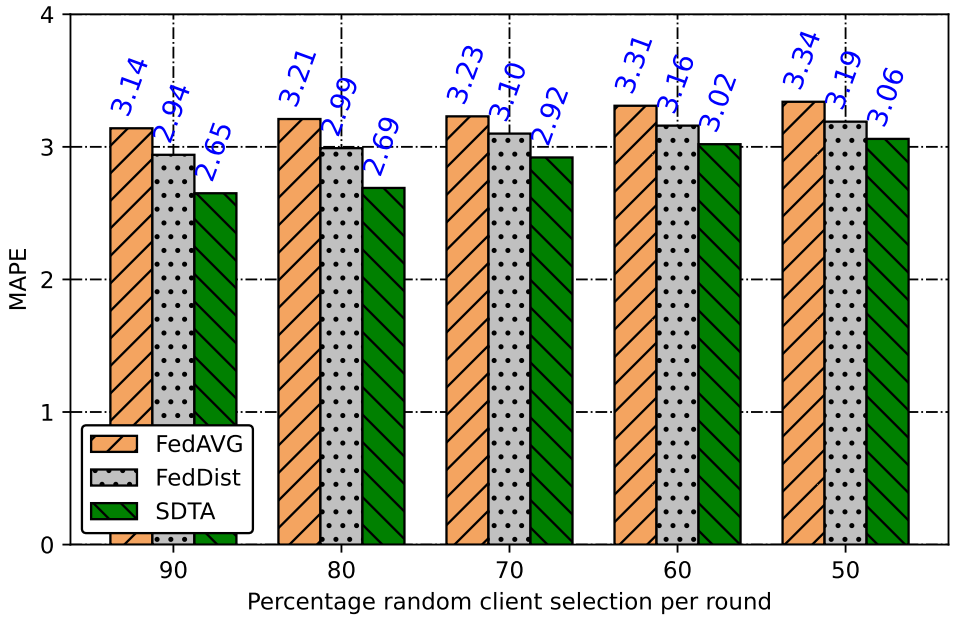


Figure 3.14: Impact of random client selection, comparing the MAPE values for FedAVG, FedDist, and SDTA across different client participation rates (90-50%).

tacks, particularly when a substantial proportion of ENs are compromised. The comparison in Fig. 3.13 provides additional insights by analysing the effect of different proportions of compromised ENs on model performance, covering attack scenarios ranging from 0% to 40%. Under non-adversarial conditions (0% attack), all approaches exhibit comparable performance, with MAPE values ranging between 2.6% and 3.1%, establishing a clear baseline for further comparison. However, as the proportion of compromised nodes rises, the pattern of performance degradation becomes distinct across the methods. FedAVG experiences the steepest decline, with MAPE value increased from 3.11% to 4.99% at the highest attack intensity, underscoring its vulnerability to adversarial manipulation. In contrast, SDTA maintains remarkable resilience, with only a slight increase in MAPE to 3.95% under the highest attack level (40%), representing a modest 1.32 percentage point increase from its baseline performance.

Finally, the impact of partial client participation is evaluated through random client selection, where the selection rates ranged from 90% to 50% per communication round. Random EN selection reduces communication and computation costs, making FL more scalable and efficient in real-world applications where full client participation may not be feasible due to resource constraints or network limitations. The results are presented in Fig. 3.14, demonstrate that with a 90% selection rate, SDTA achieves the lowest MAPE of 2.65%, establishing the near-optimal performance benchmark. At the same time, FedDist and FedAVG follow with MAPE values of 2.94% and 3.14%, respectively. As client selection rates decrease, all approaches show a gradual rise in MAPE, reflecting increased sensitivity to reduced participation. These results show that, when the percentage of client selection decreases, the performance of the global model also decreases.

3.5.7 Key Lesson Learnt

The impact of key factors on the SDTA framework is systematically analysed. EN participation affects both model performance and energy efficiency. While full participation enhances generalisation, random selection (e.g., 50%) significantly reduces energy consumption while maintaining competitive accuracy, highlighting the need for adaptive client selection. Privacy considerations, evaluated using DP, reveal a trade-off between accuracy and privacy as shown in Fig. 3.9-3.11. Despite performance degradation at higher privacy levels, SDTA outperforms FedAVG and FedDist by up to 20.4%, demonstrating its robustness in privacy-constrained settings. The comparison between local, centralised, and FL Table 3.4 highlights the infeasibility of centralised learning due to privacy and computational constraints, despite its superior performance (MAPE = 1.96%). FL approaches, particularly SDTA (MAPE = 2.63%), balance accuracy and efficiency, requiring fewer communication rounds than FedAVG and FedDist. The learning complexity analysis Table 3.5 confirms that SDTA introduces minimal additional computation while effectively mitigating noisy updates and adversarial attacks. Under a 40% adversarial attack, SDTA reduces the error rate by 20.5% over FedAVG, demonstrating its resilience against adversarial threats as given in figures 3.12-3.13. These findings establish SDTA as a scalable, privacy-preserving, and energy-efficient FL aggregation method, well-suited for heterogeneous and adversarial FL environments. Future improvements can focus on adaptive client selection, enhanced adversarial robustness, and optimising communication efficiency.

3.6 Summary

This chapter introduced the SDTA framework to address challenges **C1** and **C3**, as discussed in **Section 1.3**. Traditional FL aggregation methods, such as FedAVG, are often inadequate in scenarios involving data diversity, stringent privacy requirements, and adversarial threats due to their inability to effectively manage diverse and potentially malicious updates. The proposed SDTA algorithm selectively aggregates updates from ENs using a similarity-driven scoring and truncation mechanism on each layer. This method of selective aggregation reduces the effects of data diversity, noisy updates, and adversarial attacks. As a result, SDTA is particularly well-suited for heterogeneous and privacy-sensitive FL environments.

Simulation results demonstrated that SDTA outperforms existing FL aggregation methods across various scenarios. In an ideal FL environment without noise, SDTA achieved a MAPE of 2.63%, surpassing FedAVG and FedDist while requiring fewer communication rounds for convergence, thus demonstrating greater efficiency. Under DP constraints, SDTA showed minimal performance degradation, achieving a MAPE of 4.02% with $\varepsilon = 0.1$, outperforming FedDist and FedAVG by 9.7% and 20.4%, respectively. The resilience of the proposed scheme was further validated under model sign inversion attacks on 40% of ENs, where SDTA achieved a MAPE reduction of 20.5% over the FedAVG algorithm. The framework also proved effective

in situations with partial client participation, maintaining stable performance even as the client selection rate decreased from 90% to 50%. This resilience makes SDTA particularly suitable for practical applications where client availability is intermittent due to resource constraints or network issues.

Chapter 4

Semantic-Aware Federated Blockage Prediction (SFBP)

This chapter focuses on energy efficiency (both computation cost and communication overhead) and introduces SFBP, addressing the challenges **C1** and **C3** as discussed in **Section 1.3**. The core idea of the proposed framework is multi-modal data fusion and semantic information extraction. Additionally, the proposed framework performs similarity-driven FedAVG (SD-FedAVG), improving resilience against adversarial updates and reducing communication and computational overhead. Moreover, semantic awareness and SD-FedAVG also address latency issues inherent in FL training and inference. The proposed framework is tested on a wireless communication problem where multi-modal vision and wireless sensing data are used for proactive blockage prediction to assist handover. Therefore, this study aims to advance energy-efficient and secure FL framework integration in complex, dynamic environments like wireless networks.

4.1 Introduction

The rapid evolution of digital technologies has significantly increased the demand for higher data rates, lower latency, and enhanced energy efficiency in wireless communication networks. Fifth-generation (5G) networks were developed to address these needs, enabling services such as ultra-reliable low-latency communication (URLLC), massive machine-type communication (mMTC), and enhanced mobile broadband (eMBB) [22, 181]. However, emerging applications like autonomous vehicles, augmented and virtual reality, Industry 4.0, and smart healthcare impose even stricter bandwidth, latency, and reliability requirements, challenging the limits of 5G technology [182]. These demands have catalysed exploration into beyond 5G (B5G) and sixth-generation (6G) networks, envisioned to adapt seamlessly to rapidly changing data and connectivity needs [183].

A key enabler for B5G and 6G networks is the utilisation of higher frequency bands, such as millimeter-wave (mmWave) and sub-terahertz (THz) frequencies [184]. These bands of-

fer substantially larger bandwidths, making them ideal for supporting high-throughput applications. However, these frequency bands are more susceptible to physical obstructions and require precise beamforming with large antenna arrays to maintain stable connections [134]. Achieving precise beamforming introduces significant training overhead for optimal beam selection, hindering the low-latency and high-reliability requirements of many applications [185, 186]. Traditional solutions, such as adaptive beam codebooks and compressive sensing techniques, provide limited improvements, especially in dynamic environments where rapid adjustments are necessary [187, 188]. Recent research is focused on multi-modal sensing with machine learning (ML) techniques to address these challenges. By leveraging diverse inputs like GPS data, camera images, and radar, these solutions enhance beam selection and predict potential signal blockages [135, 136]. While these methods improve blockage prediction accuracy, their reliance on centralised data processing presents significant scalability challenges and risks of privacy breaches. Additionally, their limited sensing range constrains effectiveness in large, complex environments, limiting real-time adaptability [183].

An emerging approach to overcome these limitations involves multi-modal distributed sensing, where edge nodes with sensors collaborate to provide a richer, more comprehensive view of the wireless environment. This distributed sensing extends beyond individual base stations (BS), improving prediction accuracy in dynamic environments by combining data from multiple sources [22]. However, managing the large volumes of data generated and synchronising it across nodes in real-time poses significant challenges. Extracting compact, meaningful environment semantics from raw sensor data at the edge is a crucial solution. Semantic representations retain critical information while significantly reducing data volume and bandwidth usage, addressing the bottlenecks of traditional methods [137].

FL has emerged as a pivotal technology to harness the potential of distributed sensing, enabling collaborative model training without sharing raw data and thus preserving user privacy [21]. However, deploying FL in high-frequency wireless networks presents significant challenges. Managing noisy updates from diverse nodes, adversarial attacks, data heterogeneity, and maintaining model stability under dynamic conditions remain critical obstacles [80]. These factors can significantly degrade model performance, particularly in non-uniform or highly variable data environments. Recent studies have explored these challenges in different applications, highlighting the need for robust aggregation mechanisms and adaptive strategies to ensure reliable performance [189, 190].

4.2 Contributions

Motivated by recent advances in vision-aided wireless communication (VAWC) [135], this chapter introduces a novel framework called SFBP, tailored for next-generation wireless networks. The key innovation of this framework lies in combining edge-based semantic extraction with

FL to enable proactive blockage prediction and seamless proactive handover (PHO). To enhance edge processing efficiency, the SFBP framework employs MobileNetV3 [191], a lightweight computer vision (CV) model for semantic extraction. This model converts raw images into compact, privacy-preserving representations suitable for edge environments. The proposed framework introduced the SD-FedAVG algorithm to deal with noisy updates caused by adversarial attacks and data variability. This enhancement ensures better alignment of model updates, improving the robustness and stability of the global model even in scenarios with data variations across edge nodes. Prior research has not explored the combination of semantic information and FL for predicting blockages in vision-assisted systems. A key contribution of this work is the detailed analysis of the impact of model prediction discrepancies, specifically false positives (FP) and false negatives (FN), on the performance of PHO. These discrepancies, often caused by noisy updates during the FL training process, can critically affect PHO success rates. Additionally, we present a comparative analysis of the proposed SFBP framework against traditional centralised and FL approaches, focusing on metrics like energy efficiency, latency, and communication overhead. The publicly available Vision Wireless (ViWi) dataset [135] is utilised to benchmark our results, ensuring that findings are grounded in realistic and reproducible scenarios. The key contributions of this study are highlighted as follows:

- This study introduces the SFBP framework, leveraging bimodal vision and wireless sensing data and a lightweight CV model MobileNetV3 as a feature extractor for edge-based semantic extraction. This approach significantly reduces communication costs and inference latency, making it suitable for real-time, resource-constrained environments. The results are benchmarked against the more complex YOLOv5 model, illustrating the trade-offs between computational efficiency and prediction accuracy.

- A modified aggregation algorithm, SD-FedAVG is introduced, incorporating a cosine similarity measure to align model updates. This adaptive aggregation mechanism effectively mitigates the impact of noisy updates caused by adversarial attacks, ensuring robustness.

- Thorough analysis of the effects of noise on model updates and data variability on blockage prediction accuracy is done. Additionally, the impact of FP and false negatives FN on PHO success rates is examined, offering insights into the resilience of FL in practical deployments.

- Comparative analysis of the SFBP framework is presented, focusing on the energy efficiency of raw data transfer during centralised training and model parameter sharing, both with and without semantic information. Energy efficiency refers to the average electrical energy consumption for transferring raw data or model parameters over a wireless link, measured in kilowatt-hours per gigabyte (kWh/GB). Additionally, we evaluate the latency of centralised versus on-device inference using the publicly available ViWi dataset [135].

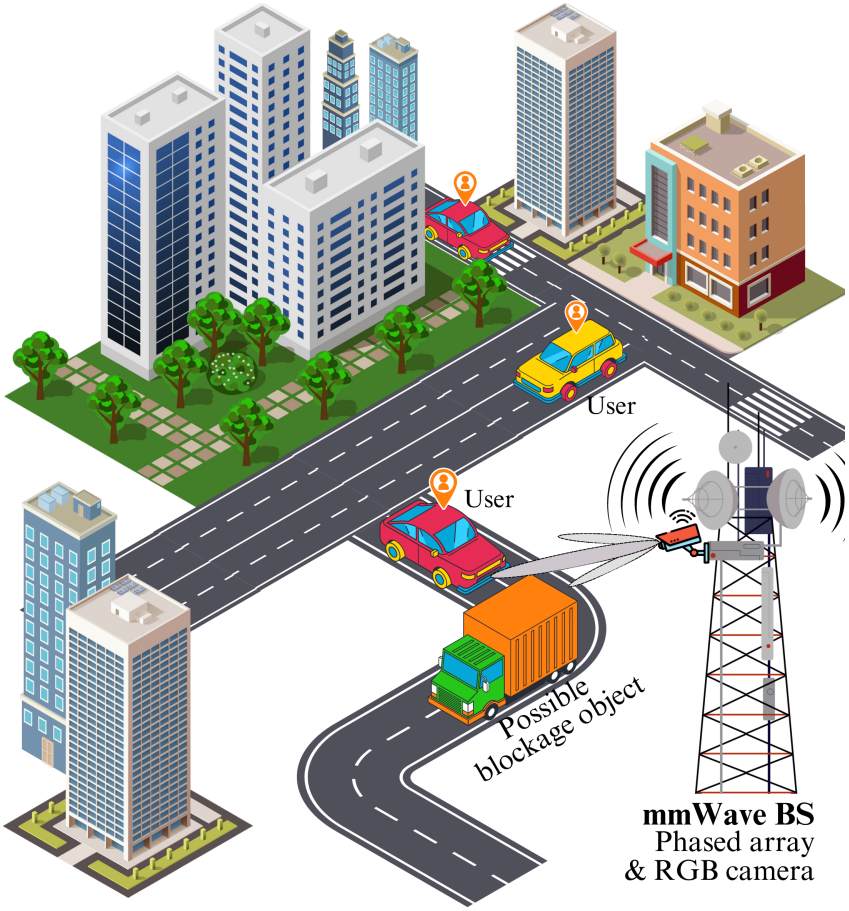


Figure 4.1: The mmWave BS is equipped with vision sensors that serve mobile users in an urban setting. Users can experience link blockages while passing large objects (e.g., buses).

4.3 System Model for Blockage Predicition

In this study, we consider a mmWave communication system where each BS is equipped with vision sensors to capture high-resolution images of the surrounding environment, as illustrated in Fig. 4.1. The system consists of N distributed nodes, each denoted by the index $n \in \{1,2,\ldots,N\}$. The mmWave BS is equipped with M-element antenna arrays, using orthogonal frequency division multiplexing (OFDM) with K subcarriers. A predefined beamforming codebook $\mathscr{F} = \{f_m\}_{m=1}^{Z}$ is used, where $f_m \in \mathbb{C}^{M\times 1}$ represents the beamforming vector, and Z denotes the total number of beamforming vectors. The received downlink signal for user n at subcarrier k and time t is represented as:

$$y_{n,k}[t] = h_{n,k}^{T}[t] f_m x[t] + n_{n,k}[t], \tag{4.1}$$

where $h_{n,k}[t] \in \mathbb{C}^{M\times 1}$ denotes the channel between the BS and user n at the k-th subcarrier and time t. The term $x[t]$ is the complex transmitted symbol, and $n_{n,k}[t]$ is the Gaussian noise, modeled as $n_{n,k}[t] \sim \mathscr{CN}(0,\sigma^2)$.

The channel h_k in equation (4.1) is a general one that can be shown mathematically as the

sum of multiple path propagation, which is shown here [137]:

$$h_{n,k}[t] = h_{n,k}^{\mathrm{LOS}}[t] + h_{n,k}^{\mathrm{NLOS}}[t], \tag{4.2}$$

where $h_{n,k}^{\mathrm{LOS}}[t]$ is the LOS component, and $h_{n,k}^{\mathrm{NLOS}}[t]$ is the NLOS component. LOS communication is crucial in mmWave systems due to its higher channel gains compared to NLOS paths [192]. The presence or absence of the LOS link is determined by the binary variable $b_n[t]$:

$$b_n[t] = \begin{cases} 1 & \text{LOS blocked for user } n \\ 0 & \text{LOS not blocked for user } n \end{cases} \tag{4.3}$$

Consequently, the effective channel model at time t for user n can be written as:

$$h_{n,k}[t] = (1 - b_n[t]) h_{n,k}^{\mathrm{LOS}}[t] + h_{n,k}^{\mathrm{NLOS}}[t]. \tag{4.4}$$

4.3.1 Problem Formulation for Blockage Prediction

Link blockages in urban wireless environments are often caused by dynamic objects like vehicles or pedestrians, leading to interruptions in LOS communication. Predicting these blockages requires analysing sequences of images captured by the vision sensors at each BS. For each user n, the goal is to predict future link blockages using a sequence of images and beam indices. At any given time t, the sequence of observations for user n over a window of r instances is represented as:

$$\mathscr{S}_n[t] = \{(X_n[i], b_n[i])\}_{i=t-r+1}^{t}, \tag{4.5}$$

where $X_n[i] \in \mathbb{R}^{W \times H \times C}$ is the RGB image captured by the vision sensor at time instance i. Here, W, H, and C represent the image width, height, and number of color channels, respectively, and $b_n[i]$ denotes the corresponding beamforming vector from the predefined codebook. The objective is to predict the blockage status $s_n[t]$ over a future observation window of r' time instances, defined as:

$$s_n[t] = \begin{cases} 1 & \text{if } b_n[t] = 1 \text{ for } t \in \{t+1, \ldots, t+r'\} \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

where $s_n[t] = 1$ indicates a predicted blockage within the observation window. The goal is to utilise $\mathscr{S}_n[t]$ to estimate $s_n[t]$ with high success probability, represented as $\mathscr{P}(\hat{s}_n[t] = s_n[t] | \mathscr{S}_n[t])$, where $\hat{s}_n[t]$ is the estimated status obtained using the DL model $\ell_{\omega}(\mathscr{S}_n[t])$. The DL is trained using the dataset $\mathscr{D} = \{(\mathscr{S}_n, s_n)\}_{n=1}^{N}$, where N represents the number of users, and each instance includes observed sequences $\mathscr{S}_n$ and corresponding ground truth s_n. Thus, the prediction

function ℓ_ω is optimised to maximise the likelihood:

$$\ell_{\omega^*} = \underset{\ell_\omega}{\operatorname{argmax}} \prod_{n=1}^{N} \mathbb{P}(\hat{s}_n = s_n | \mathscr{S}_n), \tag{4.7}$$

ensuring the best estimation of blockage status for each user based on the observed sequence $\mathscr{S}_n$.

4.3.2 Proactive Handover Mechanism

The key idea behind the blockage prediction is to perform PHO for seamless connectivity. For user n, we consider two BSs: the current serving BS c and a neighbouring BS c'. The sequence of observations at each BS for user n is represented as $\mathscr{S}_n^{(c)} = \{(X_n^{(c)}[i], b_n^{(c)}[i])\}_{i=t-r+1}^{t}$ and $\mathscr{S}_n^{(c')} = \{(X_n^{(c')}[i], b_n^{(c')}[i])\}_{i=t-r+1}^{t}$. The handover decision variable $d_n^{cc'}$ is defined as:

$$d_n^{cc'} = \begin{cases} 1 & \text{if } (s_n^{(c)} = 1,\ s_n^{(c')} = 0) \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

where $d_n^{cc'} = 1$ signifies the need for handover, while $d_n^{cc'} = 0$ indicates that no handover is necessary. To simplify, we assume that $s_n^{(c')} = 0$, thus the condition simplifies to:

$$d_n^{cc'} = \begin{cases} 1 & \text{if } s_n^{(c)} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

where the HO decision solely depends on the link status of the user. Therefore, to account for the prediction inaccuracies caused by the ML model, HO success metric $\mathscr{H}$ will consider both false positives and false negatives. This metric is mathematically represented as:

$$\mathscr{H} = \begin{cases} 1 & \text{if } (\hat{s}_n^{(c)}, s_n^{(c)}) \in \{\text{TP}, \text{TN}\} \\ 0 & \text{if } (\hat{s}_n^{(c)}, s_n^{(c)}) \in \{\text{FP}, \text{FN}\}, \end{cases} \tag{4.10}$$

where TP, TN, FP, and FN are true positive, true negative, false positive and false negative, respectively. $\mathscr{H} = 1$ indicates successful handover based on accurate blockage prediction, while $\mathscr{H} = 0$ represents failure due to incorrect prediction.

4.4 Proposed SFBP Approach

This section introduces the proposed SFBP framework, which aims to ensure seamless connectivity of high-mobility mmWave wireless communication systems by anticipating potential

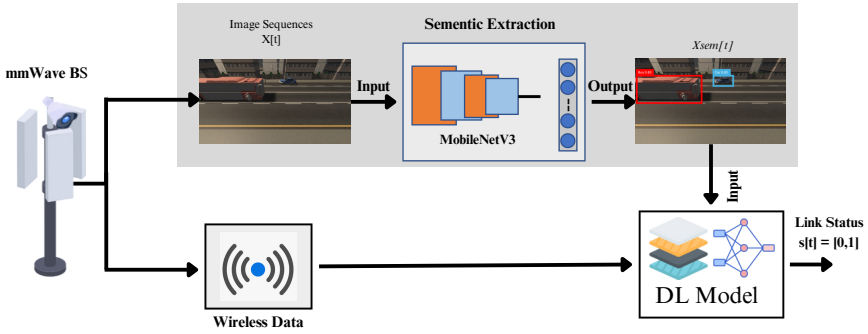


Figure 4.2: A block diagram of proposed SFBP for training a DL model on edge node.

blockages in advance. Traditional approaches often rely on co-located sensing and centralised processing, which limits their applicability in real-world systems. Our framework leverages distributed sensing using data from multiple nodes with vision sensors across the network. This distributed approach expands sensing coverage and captures diverse environmental perspectives, effectively addressing LOS and NLOS conditions.

However, the increase in data volume from distributed sensing introduces challenges in data storage, processing, and synchronisation between nodes and the BS. To mitigate these issues, we employ edge processing techniques and extract critical environment semantics using a lightweight CV model MobileNetV3 [191]. By focusing on essential information, such as the presence and locations of objects in the scene, we significantly reduce the data traffic between distributed nodes and the BS, alleviating storage and transmission burdens. Furthermore, the SFBP framework incorporates FL to train blockage prediction models collaboratively across distributed nodes without data sharing, thus preserving privacy and reducing communication overhead. By sharing only model parameters, FL minimises the need for data synchronisation and handles data heterogeneity across different nodes. The SFBP framework provides a streamlined and effective solution, enabling the wireless system to understand its surroundings and maintain uninterrupted connectivity through PHO decisions. The entire process is divided into three key steps: (i) semantic extraction, (ii) blockage prediction, and (iii) PHO, which are detailed in the subsequent sections.

4.4.1 Semantic Information Extraction

The first step in the proposed SFBP framework is extracting semantic information from raw images, represented as X_{sem}. This process involves identifying relevant features in the environment, which are later used for training the blockage prediction model. Selecting a suitable DL model for semantic feature extraction is essential at this stage, as it must balance accuracy and computational efficiency, especially for edge processing on distributed nodes.

For semantic information extraction, we utilise MobileNetV3, a lightweight and efficient DNN model designed explicitly for edge processing. Instead of building and training Mo-

bileNetV3 from scratch, we integrated a pre-trained MobileNetV3 into our architecture with slight modifications. This decision offers two key advantages: (i) improved detection performance through transfer learning and (ii) faster training convergence. Transfer learning allows MobileNetV3 to leverage prior knowledge from large and diverse datasets, such as the COCO dataset [193], which includes object classes typically found in diverse outdoor environments. As a result, the model delivers robust detection performance without requiring extensive additional training.

At the inference stage of MobileNetV3, a sequence of input images $\mathbf{X}$ is processed; it extracts key semantic features such as object classes and bounding box coordinates. This capability allows for the efficient conversion of raw images into compact semantic representations $X_{sem} \in \mathbb{R}^{[N\times(6\times1)]}$, where N is the number of detected objects. Each object's vector includes critical details like top-left coordinates (x_1, y_1), bottom-right coordinates (x_2, y_2), and centre point (x_c, y_c). This semantic information, in combination with wireless sensing data, is used to train the DL model for blockage prediction. The schematic diagram of model training is shown in Fig. 4.2.

We also compare YOLOv5, a state-of-the-art object detection model for semantic information extraction, known for its accuracy in segmenting urban environments. While YOLOv5 offers high precision, it requires significantly more computational resources, making it less suitable for edge nodes with limited processing power. This efficient semantic extraction process enables the SFBP framework to focus on transmitting only critical information from the distributed nodes to the BS if needed, reducing data transmission overhead and preserving bandwidth while maintaining high accuracy in blockage prediction.

4.4.2 Federated Learning for Blockage Prediction

This subsection provides a detailed explanation of FL, a key component in our blockage prediction approach. FL enables distributed training of models directly on edge nodes (ENs) without sharing raw data, preserving privacy and reducing communication overhead. In our proposed system, each BS functions as an EN, represented by the set $i \in \{1, 2, \ldots, N\}$, where N denotes the number of client ENs actively participating in the training process.

Each EN maintains a local dataset, denoted as $|D_i|$, representing a subset of data at the i-th BS. The total volume of vision-sensing data across all nodes is represented as $D = \sum_{i=1}^{N} |D_i|$. Traditional ML solutions often require transmitting these datasets to a centralised server, which can result in high communication costs and latency due to the large data volumes. To address this, we leverage FL using the semantic features obtained from MobileNetV3 at each EN to train local models. Once local training is complete, each EN shares only its model parameters ω_i with the Federated Server (FS), significantly reducing communication costs. This is an iterative process which continues for several communication rounds until the model converges and the desired number of communication rounds is reached. A comprehensive representation

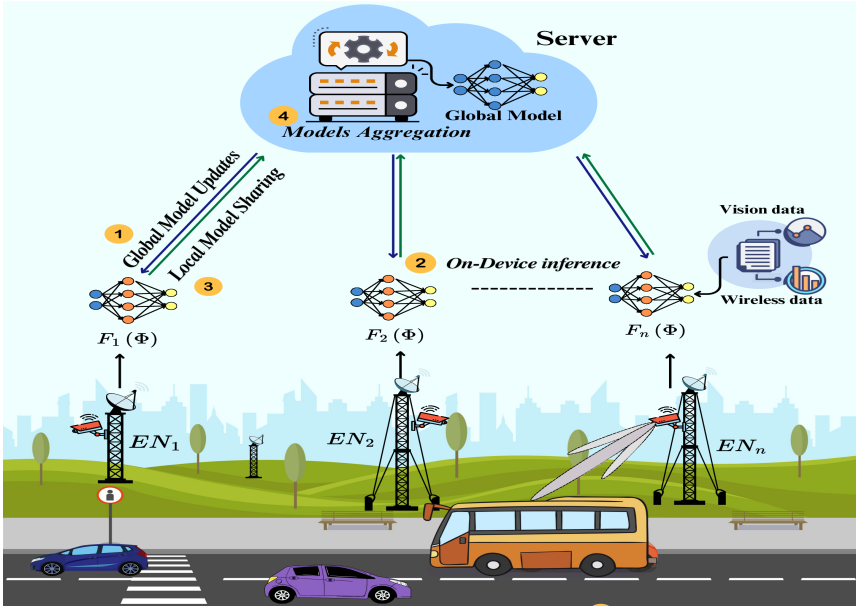


Figure 4.3: A block diagram of proposed SFBP framework with multiple BS equipped with vision and wireless sensing capabilities. Each EN has local data processing capabilities.

of the training process encompassing semantic extraction and DNN for blockage prediction is illustrated in Fig. 6.1.

Local Training

The local training process at each EN involves a 3-layer DNN, trained using the semantic information extracted by MobileNetV3. Each DNN is structured with layers comprising 132, 64, and 32 neurons, respectively, and utilises a rectified linear unit (ReLU) as the activation function to introduce non-linearity. A dropout layer is added after each dense layer to mitigate overfitting, and a sigmoid function is used in the output layer for the binary classification of blockage states. Each EN trains its model using its local dataset $|D_i|$ and updates its model parameters $\omega_{n,t}$. The training process is uniform across all ENs, ensuring consistent model structures throughout the network. The common goal of local learning is to minimise the local loss function $\ell_n(\omega)$ given in eq. (2.1) where X_{sem} is the input features, and y are the the target labels.

Global Model Aggregation

Once the local learning phase is completed, the ENs share their locally optimised model parameters ω_n with the FS, which aggregates these updates to create a global model given in eq. (2.19). The goal is to find the optimal global model parameters ω^* by minimising this global cost function:

$$\omega^* = \arg\min_{\omega} \ell(\omega). \tag{4.11}$$

Similarity driven model aggregation

A novel aspect of our approach is the SD-FedAVG, which aims to improve the robustness of model aggregation, particularly in noisy updates during training. The proposed SD-FedAVG enhances the aggregation process by weighting each EN's contribution based on the alignment between its local update and the global model. This is achieved using cosine similarity, which helps prioritise updates that align closely with the overall training direction, improving the stability and performance of the global model. The global model parameters ω^t are updated using:

$$\omega^{t+1} = \sum_{i=1}^{N} w_i \omega_i^{t+1}, \tag{4.12}$$

where w_i is the weight assigned to the i-th EN and is computed based on the similarity between its local update and the global model:

$$w_i = \begin{cases} \frac{\exp(\text{cos\_sim}_i)}{\sum_{j=1}^{N} \exp(\text{cos\_sim}_j)}, & \text{if cos\_sim}_n \geq \tau, \\ 0, & \text{otherwise.} \end{cases} \tag{4.13}$$

The cosine similarity between the local update ω_i^{t+1} and the global model ω^{t+1} is given by:

$$\text{cos\_sim}_i = \frac{\langle \omega_i^{t+1}, \omega^{t+1} \rangle}{\|\omega_i^{t+1}\| \|\omega^{t+1}\|}, \tag{4.14}$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product, and $\| \cdot \|$ represents the Euclidean norm. A threshold τ is applied to filter out noisy updates, helping maintain the integrity of the global model by excluding updates that deviate significantly from the overall direction.

Noise Detection and Thresholding

A noise detection mechanism is implemented at the FS to ensure robust aggregation. This mechanism identifies potentially noisy updates by analysing the distribution of cosine similarity scores across all ENs. The mean $\mu_{\cos}$ and standard deviation $\sigma_{\cos}$ of the similarity scores are calculated as follows:

$$\mu_{\cos} = \frac{1}{N} \sum_{n=1}^{N} \text{cos\_sim}_n, \quad \sigma_{\cos} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (\text{cos\_sim}_n - \mu_{\cos})^2}.$$

The threshold τ is determined using:

$$\tau = \mu_{\cos} - \alpha \sigma_{\cos},$$

where α is set to 1 in our implementation. This choice is based on the empirical rule (68-95-99.7 rule), which suggests that approximately 68% of the data points lie within one standard deviation from the mean [194]. By setting $\alpha = 1$, we ensure that around 68% of the clients with similarity scores near the mean can contribute to the global model while significantly deviating updates are excluded. If the variance $\sigma_{\cos}$ exceeds a predefined threshold σ_{noise}, SD-FedAVG is applied. Otherwise, standard FedAVG is used, with weights based on the relative size of each EN's dataset:

$$w_i = \frac{|\mathscr{D}_n|}{\sum_{j=1}^{N} |\mathscr{D}_j|} \tag{4.15}$$

The overall FL process includes several communication rounds to refine the global model iteratively, and the pseudo-code is given in Algorithm 4. Once training is complete, the global model is distributed back to the ENs, enabling each EN to use the trained model for real-time blockage prediction.

4.4.3 Proactive HO Mechanism

Predicting blockages in high-frequency wireless networks is highly beneficial as it allows for the proactive handling of LOS obstructions via HO. With the assistance of a vision-based blockage prediction model, the proactive HO algorithm can initiate and complete the HO process to maintain seamless connectivity. In support of this, the proposed SFBP framework discussed in Section 4.4.1 and 4.4.2 is employed for blockage prediction, using the setting discussed in Section 4.3. This setup encompasses two neighbouring high-frequency BSs operating within identical wireless conditions. Additionally, both BSs are equipped with vision and wireless sensors, providing raw data of the wireless system's surroundings.

The HO process is tightly coupled with the accuracy of blockage prediction, which is why our approach utilises a two-phase methodology: (i) Training Phase and (ii) Inference Phase. The training phase is conducted offline using the FL approach discussed in the Section 4.4.2. During this phase, multiple BSs collaboratively train a global model that captures the spatial and temporal dynamics of blockages in the network. The training is distributed by leveraging FL, allowing each BS to refine its local model based on region-specific data while maintaining privacy.

Afterwards, the optimal global model is deployed on each BS for real-time inference. The key objective of this study is to predict blockage proactively using environment semantics that will enable the PHO decision metric $\mathscr{H}$ given in the equation (4.10). Here are some assumptions for our HO mechanism: (i) there is a BS denoted by $B^{(c')}$ available for HO to serve the user from $B^{(c)}$, (ii) the time to block is always greater than the HO time, which ensures the successful HO. It is important to note that one of our previous studies thoroughly investigated the relationship between blockage, HO time, and their impact on quality QoS. Hence, these assumptions have

Algorithm 4: Dynamic Aggregation with SD-FedAVG

Result: Trained global model parameters ω^T
Input: Communication rounds T, number of clients N, $\alpha = 1$, noise threshold σ_{noise};
Initialise global model ω^0;
for *each communication round* $t = 1$ *to* T **do**
  Receive local parameters ω_i^{t+1} from each client i;
  for *each client* i **do**
    Compute cos_sim_i between ω_i^{t+1} and ω^t;
  end
  Compute $\mu_{\cos}$ and $\sigma_{\cos}$;
  Set threshold $\tau = \mu_{\cos} - \alpha\sigma_{\cos}$;
  if $\sigma_{cos} > \sigma_{noise}$ **then**
    Apply SD-FedAVG:
    for *each client* i **do**
      if $cos\_sim_i \geq \tau$ **then**
        $w_i = \frac{\exp(\text{cos\_sim}_i)}{\sum_{j=1}^{N} \exp(\text{cos\_sim}_j)}$;
      end
      else
        $w_i = 0$;
      end
    end
  else
    Apply Standard FedAVG:
    for *each client* i **do**
      $w_i = \frac{|\mathscr{D}_i|}{\sum_{j=1}^{N} |\mathscr{D}_j|}$;
    end
  end
  Aggregate global model:
  $\omega^{t+1} = \sum_{i=1}^{N} w_i \omega_i^{t+1}$;
  Send ω^{t+1} back to each client i;
end

simplified our analysis [184].

Note: Despite our goal of achieving accurate blockage prediction and timely handovers, real-world deployments present inherent challenges that can impact performance. Factors such as data heterogeneity across base stations, channel noise during federated updates, adversarial attacks like model poisoning, and hardware constraints due to varying sensor capabilities can affect the consistency and reliability of the global model. If not properly managed, these issues may lead to increased latency or reduced reliability in the handover mechanism. Therefore, a comprehensive evaluation of the model's robustness and adaptability under various conditions is essential to optimising its performance.

Table 4.1: Data distribution of ViWi Dataset.

	Training Samples	**Testing Samples**	**Total Samples**
EN-1	1173	297	1470
EN-2	1653	387	2040
EN-3	1174	316	1490
Total	4000	1000	5000

4.5 Simulation Setup

This section outlines the simulation process, including the dataset used, the metrics applied to evaluate the performance, and the comparison scenarios considered for the proposed SFBP framework.

4.5.1 Dataset Description

The performance of the proposed SFBP framework is evaluated using the publicly available ViWi dataset [135], which provides a rich set of vision and wireless data for studies in mmWave communications. ViWi is generated using Wireless Insite, a ray-tracing tool, and Blender 3D to simulate realistic urban environments. The dataset consists of synchronous wireless and vision-sensing samples across various scenarios, differentiated by camera location (distributed or co-located) and view (direct or blocked).

This study focuses on a distributed scenario with three vision sensors mounted on different BSs, named EN-1, EN-2, and EN-3. These sensors have unique fields of view (FoV) and capture data related to a highly mobile user (a car) and a blocking object (a bus). The dataset includes 5,000 samples, with 4,000 samples dedicated to training and 1,000 for testing the model. The dataset categorizes the user's link status $b[t]$ into two classes: LOS (not blocked) and NLOS (blocked), based on the user's received signal strength and position relative to the obstacles. The distribution of samples across the three edge nodes, highlighting the data heterogeneity, is presented in Table 4.1.

4.5.2 Performance Metrics

A rigorous evaluation of the predictive model and the PHO algorithm is essential to demonstrate the effectiveness of the SFBP framework. The following metrics are used to assess the system's performance comprehensively.

Predictive Model Evaluation

The blockage prediction task is treated as a binary classification problem. While accuracy is a

commonly used metric measuring the ratio of correct predictions to total predictions, it can be misleading in imbalanced datasets due to the accuracy paradox. To ensure a balanced evaluation, we also consider Precision, Recall, and F1-score as discussed in Section. 2.2.2. These metrics provide a deeper insight into the model's predictive power, particularly in handling minority classes (NLOS cases). Using semantic information from MobileNetV3 and YOLOv5, comparisons are made between centralised and distributed learning setups.

PHO Algorithm Evaluation

The effectiveness of the PHO mechanism is directly linked to the accuracy of the blockage prediction model. To evaluate this, we introduce a modified HO failure rate metric H, accounting for FP and FN:

$$H = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{1}_{\{(\hat{s}_{ni}^{(c)},s_{ni}^{(c)})\in \mathrm{FP}\cup\mathrm{FN}\}}\right), \tag{4.16}$$

where N is the total number of samples used for testing. Here, $\{(\hat{s}_n^{(c)},s_n^{(c)})\}$ includes TP, TN, FP, and FN. The metric captures the impact of incorrect blockage predictions (FP and FN) on the HO success rate, providing a holistic measure of the PHO mechanism's performance.

Energy Efficiency and Latency Analysis

In addition to prediction accuracy, energy efficiency and latency are critical factors for the practical deployment of the SFBP framework. Energy efficiency refers to the average electrical energy consumption for transferring raw data or model parameters over a wireless link, measured in kilowatt-hours per gigabyte (kWh/GB) as discussed in Section. 2.2.3 given in equ. (2.18).

Finally, latency is evaluated based on the time required for image capture (t_{cap}), data transmission (t_{tx}), and inference (t_{inf}). With a 10 Gbps mmWave backhaul link and a camera frame rate of 26 frames per second (fps), the overall latency D is calculated as:

$$D = t_{cap} + t_{tx} + t_{inf}. \tag{4.17}$$

This metric evaluates the advantage of on-device inference provided by MobileNetV3, especially compared to the more computationally intensive YOLOv5 model.

4.5.3 Simulation Scenarios

To validate the robustness and effectiveness of the SFBP framework, the following scenarios are considered:

- **Performance under data heterogeneity:** Evaluating the model's accuracy across varying data distributions to assess the impact of data diversity on the global model.

- **Robustness to noisy updates:** Assessing the impact of noisy updates on model convergence and how the SD-FedAVG mechanism mitigates the effects of FP and FN, thus maintaining a low HO failure rate.
- **Adversarial attack resilience:** Simulating simple adversarial attacks to evaluate the model's robustness, comparing the standard FedAVG and the proposed SD-FedAVG.
- **Energy and latency comparison:** Comparing energy estimates and latency between centralised data sharing, semantic sharing, and the proposed scheme.

4.6 Results and Discussion

In this section, we present a detailed analysis of the simulation results to evaluate the effectiveness of the proposed SFBP framework. The results are compared with both centralised learning and FL without semantic extraction to highlight the performance gains and trade-offs in terms of predictive accuracy, HO failure rate, energy efficiency, and inference latency. The simulations are conducted using the ViWi dataset, with three base stations (EN-1, EN-2, EN-3) acting as federated clients, where training is performed for 40 communication rounds. Additionally, we assess the SFBP framework under varying conditions, including ideal learning, data heterogeneity, and noisy updates. Moreover, we explore how the proposed SD-FedAVG mitigates the impact of noisy updates and adversarial attacks. Finally, we conduct an energy efficiency comparison and provide insights into latency reduction, highlighting the practical advantages of on-device inference in edge nodes. The results are presented in three main subsections: predictive model performance, HO algorithm performance, and communication cost and latency comparison.

4.6.1 Predictive Model Performance

This section compares the performance of the proposed model under various scenarios. This includes ideal learning conditions where centralised and distributed learning are compared. To further evaluate the robustness of the proposed framework, model evaluation is done using varying data distribution and noisy updates during the model updates.

Centralised vs FL Comparison

We adopted a simplistic approach for our initial analysis, thus establishing an ideal learning environment using the entire sample size in the ViWi dataset. We subsequently compared the outcomes with those of centralised learning and FL without semantics, and the results are shown in Table 4.2. These results also compare YOLOv5 and MobileNetV3, and the simulations ran

Table 4.2: Comparison results of centralised learning, FL without semantics and proposed SFBP with MobileNetV3.

Technique	Accuracy	Precision	Recall	F1-Score
Centralised	99	98	99	99
FL-baseline	98.5	98	98.9	98.2
SFBP-YOLOv5	97.5	98	97	98
SFBP-MobileNetV3	**97.1**	**97.6**	**97.8**	**97.8**

several times to get reliable average results. As anticipated, the centralised model training performed better. However, it is worth noting that our SFBP framework with MoblileNetV3 was not far behind, even though it does not involve data sharing.

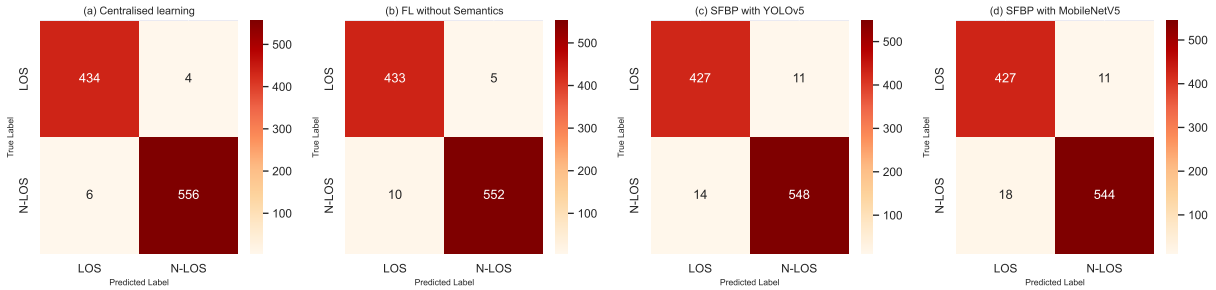


Figure 4.4: Confusion matrix for (a) centralised, (b) FL-baseline, (c) SFBP-YOLOv5 and (d) proposed SFBP-MobileNetV3, where diagonal values represent correct predictions and off-diagonal values represent FP and FN, respectively.

The confusion matrices of centralised learning, FL without semantics, SFBP-YOLOv5 and our proposed SFBP-MobileNetV3 are presented in Fig. 4.4. The diagonal values of the confusion matrix represent the TP and TN, whereas the off-diagonal values represent the FP and FN, respectively. These matrices give us a clear picture of how each method performed, where the centralised learning achieved an impressive accuracy of about 99%. FL without semantics was not far behind, with a global model accuracy of 98.5%, followed by the SFBP-YOLOv5 with 97.56%, and the proposed SFBP-MobileNetV3 with an accuracy of 97.12%. We also reported the precision, recall, and F1 scores for all three scenarios to further validate the effectiveness of our proposed scheme. Although centralised learning performs the best, our proposed SFBP framework achieves comparable results while benefiting from reduced computation cost and latency, which will be discussed in the following section.

Comparison for Data Variations

We adjusted the data size to create an imbalanced dataset in this comparison and used SFBP-MobileNetV3 only, as there is no significant difference between the object detection results of YOLOv5 and MobileNetV3. Our goal is to see the impact of data variation on the performance

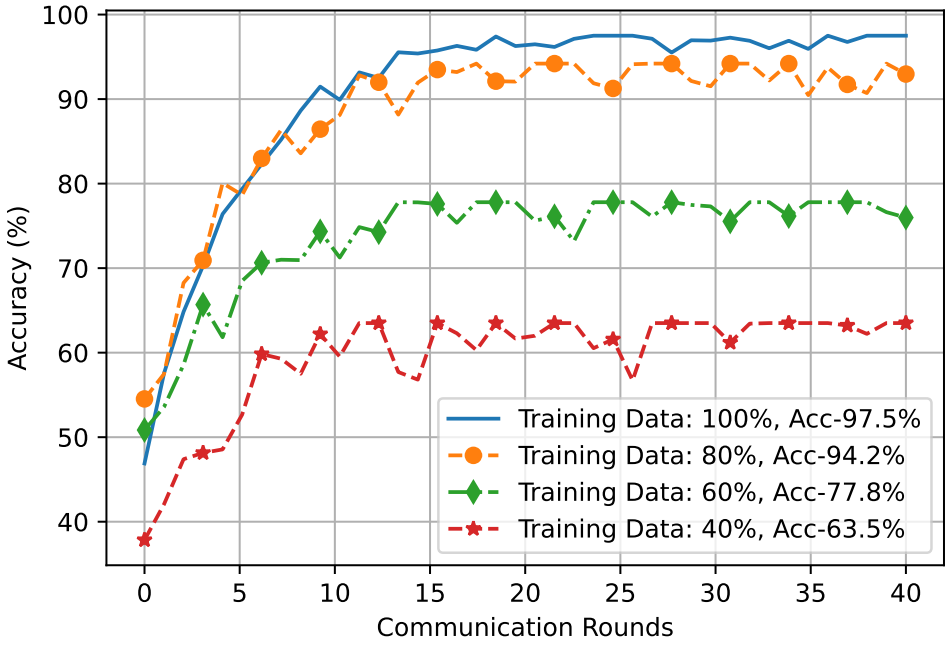


Figure 4.5: Learning curve plotted for data variations using the global test for each communication round.

of the predictive model, which has a direct effect on the PHO mechanism. The proposed model was trained using varying percentages of the dataset on each EN, and a learning curve was generated using the global test, depicted in Fig. 4.5. The results display typical learning behaviour, with a sharp increase in the initial rounds followed by steady improvement. However, using a smaller training set caused a significant decrease in performance. For example, with only 60% of the training data, the maximum accuracy over 40 communication rounds was approximately 77.8%, compared to 97.5% under ideal learning conditions.

Comparison for Noisy Updates

In this study, we evaluated the impact of noise on model aggregation in FL by varying the privacy budget (ε) across five levels: $\varepsilon = 0.1$, $\varepsilon = 0.5$, $\varepsilon = 1$, $\varepsilon = 2$, and $\varepsilon = 5$. For each ε, Gaussian noise was added to the local gradients before aggregation, with noise variance (σ^2) determined by ε according to the formula $\sigma^2 = \frac{2\ln(1.25/\delta)}{\varepsilon^2}$, where δ was fixed at 10^{-5} [195]. A smaller ε resulted in higher noise variance, which directly impacted the ability of the model to converge. Additionally, we created a simple adversarial attack where the EN-3 intentionally added the noise to poison the model updates. The learning curve for noisy updates and adversarial attacks is shown in Fig. 4.6. The results in Fig 4.6, compares the learning accuracy of FedAVG and the proposed SD-FedAVG across different noise levels (ε) over 40 communication rounds. Under various noise conditions, ranging from $\varepsilon = 5$ (low noise) to $\varepsilon = 0.1$ (high noise), SD-FedAVG consistently performs better than FedAVG, particularly at $\varepsilon = 5$ as shown in Fig. 4.6 (a), where SD-FedAVG achieves an accuracy of 95.1% compared to FedAVG's 94.5%. As noise increases

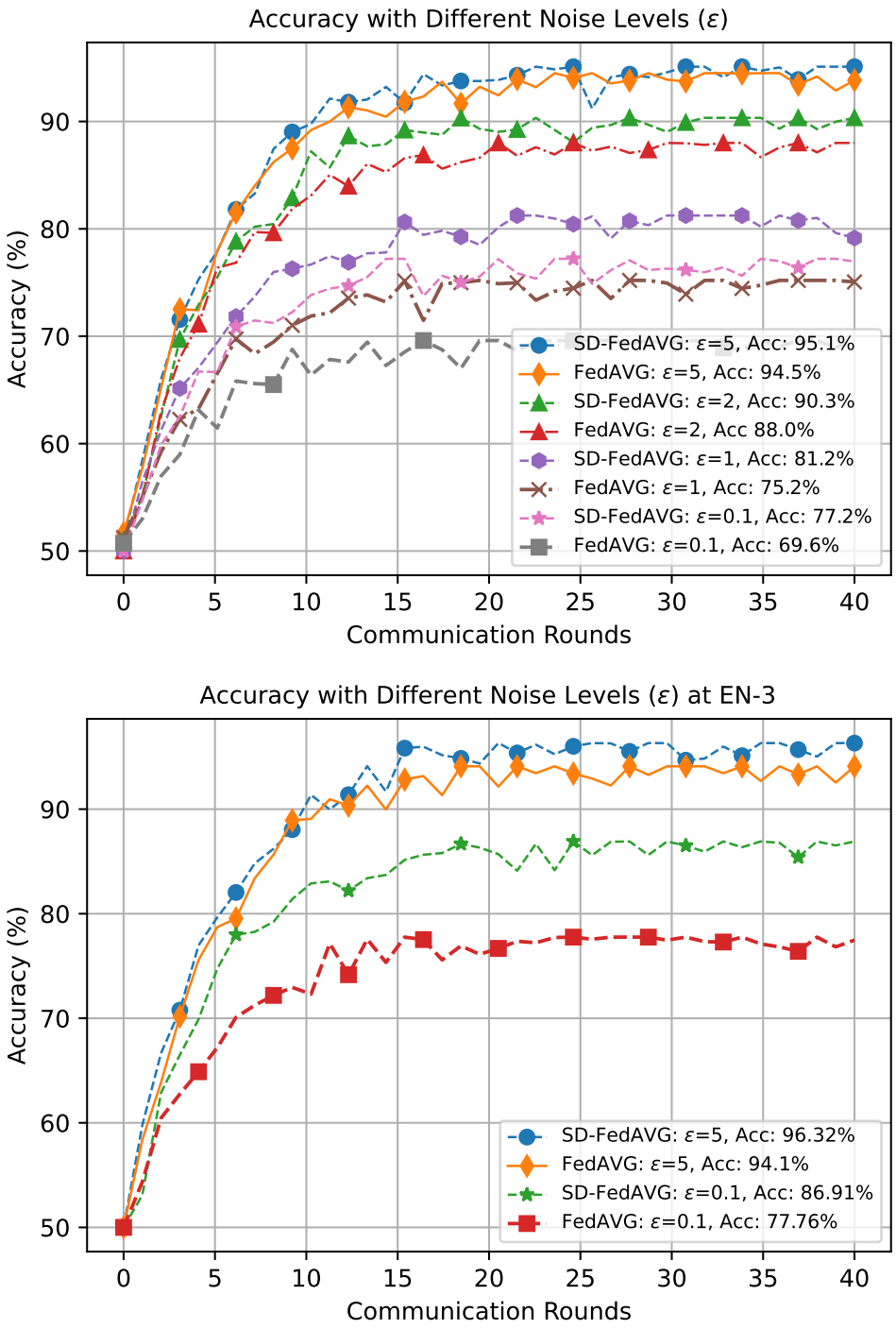


Figure 4.6: Plot (a) shows the variable noise level controlled by ε and plot (b) represents the learning curve for the adversarial attack on EN-3 during the training process.

(ε decreases), the performance gap widens, especially at $\varepsilon = 0.1$, where SD-FedAVG achieves 77.2% accuracy versus 69.6% for FedAVG. Similarly, for an adversarial attack on EN-3, the trend remains consistent as shown in Fig. 4.6 (b), with SD-FedAVG performing better than FedAVG, particularly at higher noise levels. For instance, at $\varepsilon = 0.1$, the accuracy achieved by SD-FedAVG is approximately 86.91% compared to FedAVG, which achieves an accuracy

of 77.76%. Finally, the results in Fig. 4.7 show the confusion matrix for worst-case scenarios for both noisy updates and adversarial attacks. The results are obtained using the global model trained under variable noise conditions controlled by ε, where in our case, the value of $\varepsilon = 0.1$ for both cases. The results show that the proposed SD-FedAVG aggregation mechanism performed better than the FedAVG algorithm with low FP and FN. These results directly impact the PHO mechanism, which will be discussed in the next subsection.

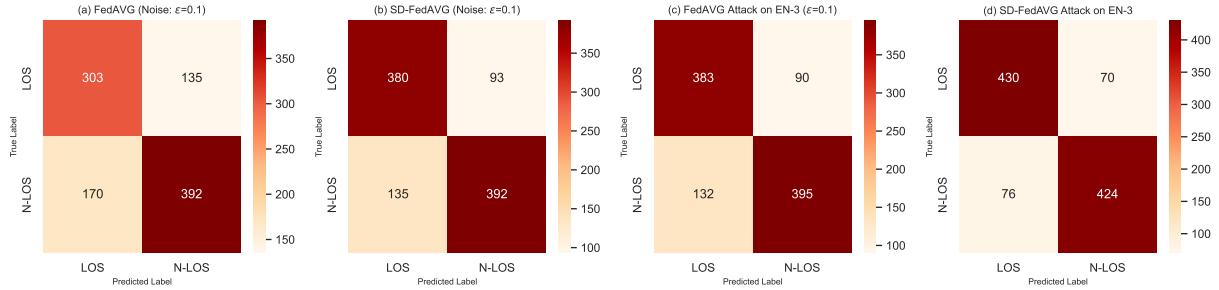


Figure 4.7: Confusion matrix for (a) FedAVG and (b) SD-FedAVG with $\varepsilon = 0.1$, (c) FedAVG, and (d) SD-FedAVG with $\varepsilon = 0.1$ on EN-3.

4.6.2 PHO Performance Evaluation

The effectiveness of the PHO mechanism is intricately linked to the accuracy of the blockage prediction model. This section evaluates the PHO performance under various conditions, including ideal learning scenarios, diverse data distribution, noisy updates, and adversarial attacks. These conditions thoroughly analyse the robustness of the prediction model and the resulting HO failure rates.

Ideal Learning Conditions

In the first scenario, we assume ideal learning conditions, where the model is trained with the complete dataset, and no noise is introduced. Under these conditions, we compare the handover failure rates (H) for centralised FL without semantics and the proposed SFBP framework. The results are shown in Table 4.3, which show that the FN is comparatively higher, suggesting that the algorithm is more likely to miss the optimal timing for an HO, potentially leading to service disruptions or reactive HO that could degrade the user experience. However, the overall impact is very low. For instance, the HO failure rate is increased by 1.9% and 1.5% when the proposed framework is compared with centralised learning. These results were expected; however, there is a huge gain in energy efficiency, which we will discuss in the next subsection and provide a trade-off comparison.

Model	FN	FP	% HO failure (H)
Centralised	8	2	1
FL-baseline	9	6	1.5
SFBP-YOLOv5	14	11	2.5
SFBP-MobileNetV3	**18**	**11**	**2.91**

Table 4.3: Comparison of HO failure rates (H) for, centralised, FL-baseline (without semantics), SFBP-YOLOv5 and SFBP-MobileNetV3 based on the performance of blockage prediction model.

Scenario	FP	FN	% HO failure rate (H)
Training data 40%	155	210	36.5
FedAVG (noisy update)	134	170	30.4
SD-FedAVG (noisy update)	93	135	22.8
FedAVG (EN-3)	90	132	22.2
SD-FedAVG (EN-3)	70	76	14.6

Table 4.4: HO failure rates for worst-case scenario, i.e., data distribution of 40%, $\varepsilon = 0.1$ for noisy update and attack on EN-3.

Data Distribution Variations

The first worst-case scenario involves training the model with only 40% of the dataset at each EN, leading to imbalanced data across the network. The results, shown in Fig 4.5, indicate a significant drop in model accuracy, which results in FN and FP values of 155 and 210, respectively. This results in an HO failure rate of 36.5%, in which approximately 21% of users would experience service disruptions due to missed HO triggers, highlighting the challenges posed by data heterogeneity FL.

Noisy Updates and Adversarial Attack

Noisy updates and adversarial attacks can severely degrade the performance predictive model. Hence, we chose the worst-case scenario with a small value of $\varepsilon = 0.1$. Additionally, we evaluated the impact of an adversarial attack on EN-3, where malicious updates were introduced. Using the results reported in Fig. 4.7, we created a Table. 4.4, summarising the HO failure rates. The results show that SD-FedAVG significantly improves the performance of the proactive HO mechanism by reducing the HO failure rate by up to 7.6% under both noisy updates. This improvement highlights the effectiveness of the cosine similarity-based aggregation mechanism in filtering out noisy or malicious updates, ensuring robust blockage prediction and PHO decision-making in mmWave communication systems.

4.6.3 Analysis of Communication Cost and Latency

The comparative analysis of energy efficiency in terms of energy estimates, as expressed in equ. (2.18), is presented in Table 4.5. For centralised learning, the energy estimation equation is defined as $E_{est} = [E(\alpha \times t_c) + (\beta \times P_{trn})]$, where E is the total number of epochs during the training process. Throughout the analysis, parameters α and β are consistently kept at values of 0.003 and 0.0001, respectively. For the centralised scenario without semantic compression, the number of epochs E is set to 40, the computation time per epoch (t_c) is 300 seconds, and the data size (P_{trn}) is 386100 KB. This results in an estimated energy cost (E_{est}) of approximately 74.61 W.

Introducing semantic compression significantly reduces the computational and communication overhead. With semantic compression, the centralised model requires 72 epochs, each with a computation time of 102 seconds and a reduced data transmission size of 38100 KB. Consequently, the total energy estimate decreases considerably to approximately 25.842 W. In the FL setups, the energy estimation equation is adapted as $E_{est} = [N(\alpha \times t_c) + (N+1)(\beta \times P_{trn})]R$, where N denotes the number of edge nodes (ENs), and R represents the number of communication rounds. The additional term $(N+1)$ in the communication cost accounts for model sharing from the central server to the edge nodes. For the FL-baseline model, computational and communication parameters are notably lower, yielding an estimated energy cost of approximately 19.92 W.

The proposed SFBP method further optimises the model by leveraging semantic extraction techniques such as YOLOv5 and MobileNetV3, significantly decreasing both model size and computational complexity. With a model size reduction from 300 KB (FL-baseline) to just 97 KB (SFBP), the SFBP approach achieves an energy estimate of approximately 8.392 W. As shown in Table 4.5, the proposed SFBP method achieves substantial improvements in energy efficiency, demonstrating approximately 88.75% and 57.87% energy reduction compared to the centralised method without semantic compression and the FL-baseline, respectively. The energy efficiency of the proposed SFBP method improves by approximately 67.53% compared to centralised learning with semantic compression. This additional reduction highlights the efficacy of integrating semantic extraction with FL, significantly minimising both computational and communication overhead. These results underscore the suitability of SFBP for deployment in resource-constrained and latency-sensitive environments.

Finally, the proposed SFBP framework uses on-device inference, which results in a significant reduction in latency. This study assumes a 10 Gbps mmWave backhaul link with a camera frame rate of 26 frames per second (fps) [196]. The overall latency is calculated using equation (4.17) where t_{cap}, t_{tx} is an image capture and the transmission time, whereas t_{inf} is the model inference time. In this study, t_{cap} is 38.5 ms, t_{tran} is 4.4 as given in [184, 196]. The t_{inf} for FL-baseline is 28 ms, SFBP-YOLOv5 is 16 ms and proposed SFBP-MobileNetV3 is 12.5 ms; therefore, the overall delay in equation (4.17) is 70.9 ms for centralised learning. For on-device

Table 4.5: Comparison of efficiency in energy estimates for centralised learning (with and without semantic compression), FL-baseline (with semantics), and proposed SFBP.

Model	**Computation cost**	**Communication cost**	E_{est}
Centralised	$E(\alpha \times t_c)$ $40(0.003 \times 300)$ 36	$\beta \times P_{tran}$ 0.0001×386100 38.61	74.61
Centralised (Semantic)	$E(\alpha \times t_c)$ $72(0.003 \times 102)$ 22.032	$\beta \times P_{tran}$ 0.0001×38100 3.81	25.842
FL-baseline	$N(\alpha \times t_c)$ $3(0.003 \times 42)$ 0.378	$(N+1)(\beta \times P_{tran})$ $4(0.0001 \times 300)$ 0.12	19.92
SFBP	$N(\alpha \times t_c)$ $3(0.003 \times 19)$ 0.171	$(N+1)(\beta \times P_{tran})$ $4(0.0001 \times 97)$ 0.0388	8.329

inference, the t_{tran} is zero, and the overall inference time for FL-baseline, SFBP-YOLOv5, and the proposed scheme is 66.4, 54.5 and 51 ms, respectively.

4.6.4 Discussion on Model Performance, Privacy, and Energy Trade-offs in SFBP

The performance evaluation of SFBP highlights its effectiveness in blockage prediction and PHO under diverse conditions, including data heterogeneity, noisy updates, and adversarial attacks. The impact of EN participation, privacy constraints, and energy efficiency trade-offs are analysed, demonstrating the practical advantages of semantic extraction in FL settings.

Impact of EN Participation and Data Heterogeneity

The effect of varying the number of ENs and training data sizes on model performance was assessed. Table 4.4 demonstrates that reducing training data to 40% per EN leads to a significant drop in blockage prediction accuracy, with an HO failure rate of 36.5% compared to 2.91% under ideal conditions. This highlights the need for balanced data distribution to maintain predictive accuracy in FL environments. However, SFBP with MobileNetV3 mitigates this degradation by preserving semantic features, allowing the model to retain performance even under limited data availability.

Privacy Considerations and Robustness to Noisy Updates

The impact of privacy constraints was evaluated by introducing DP noise at varying levels of ε. As seen in Fig. 4.6, SFBP with SD-FedAVG consistently outperforms FedAVG, particularly under strict privacy budgets ($\varepsilon = 0.1$), where SFBP achieves 77.2% accuracy compared to

FedAVG's 69.6%. Furthermore, the proposed SD-FedAVG mechanism effectively mitigates adversarial attacks, with a HO failure rate reduction of 7.6% compared to FedAVG in the presence of malicious updates (Table 4.4). These results validate SFBP's resilience to privacy noise and adversarial perturbations while preserving model accuracy.

Trade-offs Between Centralised, FL, and Personalised Models

The comparison in Table 4.2 highlights that centralised learning achieves the highest accuracy (99%) but at the cost of high communication overhead and privacy risks. The FL baseline (without semantics) achieves 98.5% accuracy, while SFBP-MobileNetV3 achieves 97.1%, demonstrating a minimal performance trade-off in exchange for significant energy efficiency gains. This trade-off is further evident in handover failure rate analysis in Table 4.3, where SFBP performs slightly worse than centralised learning (2.91% vs. 1%) but achieves major reductions in computation and communication costs.

Energy Consumption and Latency Analysis

A detailed energy efficiency evaluation in Table 4.5 demonstrates that SFBP achieves an 88.75% reduction in energy consumption compared to centralised learning and a 57.87% reduction compared to the FL baseline. This improvement is due to semantic compression, which reduces model size from 300 KB (FL-baseline) to 97 KB (SFBP), leading to lower transmission costs and computational complexity. Additionally, the latency analysis shows that on-device inference with SFBP-MobileNetV3 achieves a total delay of 51 ms, a 28% reduction compared to centralised inference (70.9 ms), making it suitable for real-time mmWave blockage prediction applications.

4.7 Summary

This chapter presented the SFBP framework, integrating lightweight computer vision CV techniques with FL for proactive blockage prediction in mmWave/THz networks, addressing challenges **C1** and **C3**. The key innovation of this work lies in the use of semantic information extracted by lightweight CV models, specifically MobileNetV3, to decentralise the training process. Through extensive simulations on the ViWi dataset, the framework demonstrated its robustness and practicality in real-world scenarios. A comparative analysis between MobileNetV3 and YOLOv5 for semantic information extraction revealed that while YOLOv5 provides marginally better object detection accuracy (97.5% compared to MobileNetV3's 97.1%), MobileNetV3 significantly reduces the computational cost and inference latency, making it more efficient for edge processing. Specifically, the inference latency using MobileNetV3 was decreased to 51 ms, compared to 66.4 ms for YOLOv5 and 70.9 ms for centralised learning, high-

lighting the advantages of lightweight models for edge processing in high-mobility networks. The SFBP framework also significantly reduced communication costs, cutting data transmission by 88.75% compared to centralised learning and by 57.87% compared to FL without semantic extraction (FL-baseline). Furthermore, the proposed SFBP framework has shown better results compared to centralised semantic compression with the improvement of 67% in energy efficiency. Another key contribution is the detailed analysis of HO failure rates, focusing on the impact of FP and FN in the predictive model. The framework's performance under noisy updates and adversarial attacks revealed increased HO failure rates due to reduced prediction accuracy. To address this, the proposed SD-FedAVG aggregation mechanism effectively mitigates the impact of noisy updates and adversarial attacks, reducing the HO failure rate by up to 7.6% under worst-case conditions.

Chapter 5

Hybrid Neuromorphic Federated Learning

This chapter introduces the HNFL framework to address the challenges **C1** and **C3**, identified in **Section 1.3**. HNFL is designed to tackle the challenges of multi-modal data fusion and energy efficiency in edge computing environments, particularly for applications involving resource-constrained IoT devices and sensors. By integrating SNNs with LSTM networks, the proposed framework implements a hybrid model called Spiking-LSTM (S-LSTM). This hybrid approach leverages the event-driven processing capabilities of SNNs and the sequence modelling strengths of LSTMs, enabling efficient on-device processing, reducing communication overhead, and enhancing scalability and security in FL systems. The proposed HNFL framework is applied to HAR applications using wearable sensing data, a domain that demands real-time processing, personalisation, and privacy preservation. Through this application, we demonstrate that HNFL not only outperforms traditional DL methods in accuracy but also achieves significant improvements in energy efficiency, making it a promising solution for next-generation IoT and edge computing applications in healthcare and lifestyle management.

5.1 Introduction

HAR has gained prominence across diverse fields, such as healthcare, smart living environments, and sports, enabled by wearable sensors that provide a continuous stream of contextual data. The rapid adoption of wearable technologies, such as smartwatches and fitness trackers, has transformed HAR by allowing real-time monitoring and proactive health and lifestyle management. In particular, HAR is valuable for remote patient monitoring, elderly care, and performance tracking in athletes, where accurate and timely recognition of activities can offer critical insights and support [139].

Traditional HAR systems primarily rely on centralised DL models as shown in Fig. 5.1, which aggregate data on a central server for analysis. Although these models deliver high accuracy, they encounter several limitations, such as high communication costs, potential privacy risks, and scalability issues due to increased data volumes. In addition, the computational bur-

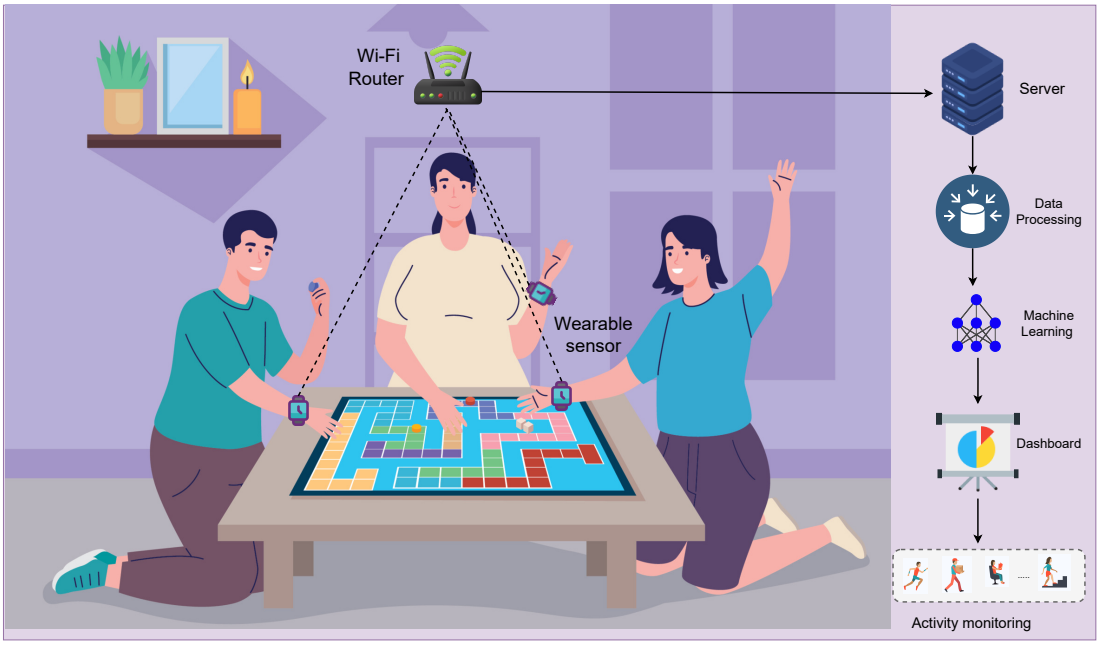


Figure 5.1: Conceptual framework of centralised indoor HAR using wearable sensors

den of DL-based HAR on centralised servers leads to latency challenges, particularly for real-time applications [197, 198]. Moreover, the dependency on data transfer to a centralised server raises privacy concerns, especially in health and lifestyle applications where data sensitivity is paramount [148]. FL has emerged as a decentralised solution to address these challenges, enabling collaborative model training across multiple edge devices without raw data exchange. This paradigm shift enhances user privacy and lowers communication costs by distributing computation across devices [21]. However, implementing FL in HAR systems introduces new issues, particularly for edge devices with limited processing power. The high computational demands of DL models often exceed the capabilities of wearable devices, prompting the need for more energy-efficient frameworks.

Neuromorphic computing, particularly with SNNs, offers a promising solution by emulating the event-driven nature of biological neural systems to reduce energy consumption. Unlike conventional DL models, SNNs operate asynchronously and event-driven, making them well-suited for real-time, low-power applications on resource-constrained edge devices [199]. However, while SNNs provide efficient data processing, they lack the sequence modelling strengths required for time-series data, which is integral to HAR. To address this limitation, this study proposes a hybrid S-LSTM model that combines the energy efficiency of SNNs with the sequence processing capability of LSTMs, creating a hybrid neuromorphic approach for federated HAR.

The S-LSTM model operates within an FL framework to process multi-modal sensor data, improving activity recognition accuracy while managing energy efficiency and privacy. The key contributions of this chapter include:

- The study introduced a HNFL framework specifically designed for time series HAR data.

The proposed S-LSTM combines the strengths of SNNs and LSTM and multi-modal data fusion to improve activity recognition accuracy while ensuring privacy and reducing computational demands.

- This study rigorously evaluates the S-LSTM model using two publicly available datasets covering various environmental settings and activity scenarios. The study compares the performance of the S-LSTM model with that of traditional LSTM, spike CNN (S-CNN), and simple CNN models. The results highlight the strength and robustness of the proposed HNFL framework and demonstrate how multi-modal data fusion can significantly improve HAR accuracy within a FL paradigm.

- This research analyses how randomly selecting clients affects the performance of the HAR model and provides valuable insights into finding the best balance between computational and communication efficiency in relation to the accuracy of the HAR model. The findings offer a strategic framework for selecting clients, which can help develop more effective and efficient FL implementations in HAR systems.

5.2 Preliminaries and System Model

This section provides a detailed discussion of the fundamental principles of FL and SNNs, along with the hybrid S-LSTM model used for HAR in distributed settings.

5.2.1 FL Framework for HAR

The system model for outdoor HAR is shown in Figure 5.2. Each participant or node holds only a subset of the dataset, ensuring privacy. The learning process is collaborative, with a central FS coordinating the training process and aggregating the model parameters from all participants (N) to refine the global model. The training continues until the global model reaches a specified level of accuracy or a predetermined number of iterations.

Without losing the generality, for each participant $i = \{1,\ldots,N\}$, a localised dataset is denoted as $|\mathscr{D}_i| \equiv \mathscr{D}$, with $\mathscr{D}_i$ representing the subset of the dataset at the i-th device, and the cumulative dataset expressed as $\mathscr{D} = \sum_{i=1}^{N} |\mathscr{D}_i|$. In this schema, given the model parameter ω and a local loss function $\ell(\omega_i, x)$ applicable to any data sample x, the local empirical loss at the i-th participant is given in eq. (2.2). The core job of FL is to optimise a global loss function $\ell(\omega)$ on the FS, which is mathematically defined in eq. (2.1). FL is an iterative process that necessitates each participant to compute the local gradient at each time iteration $t = \{1,\ldots,T\}$, following the equation:

$$\omega_i t = \tilde{\omega}_i(t-1) - \eta \nabla \ell_i(\tilde{\omega}_i(t-1), x_i(t)), \tag{5.1}$$

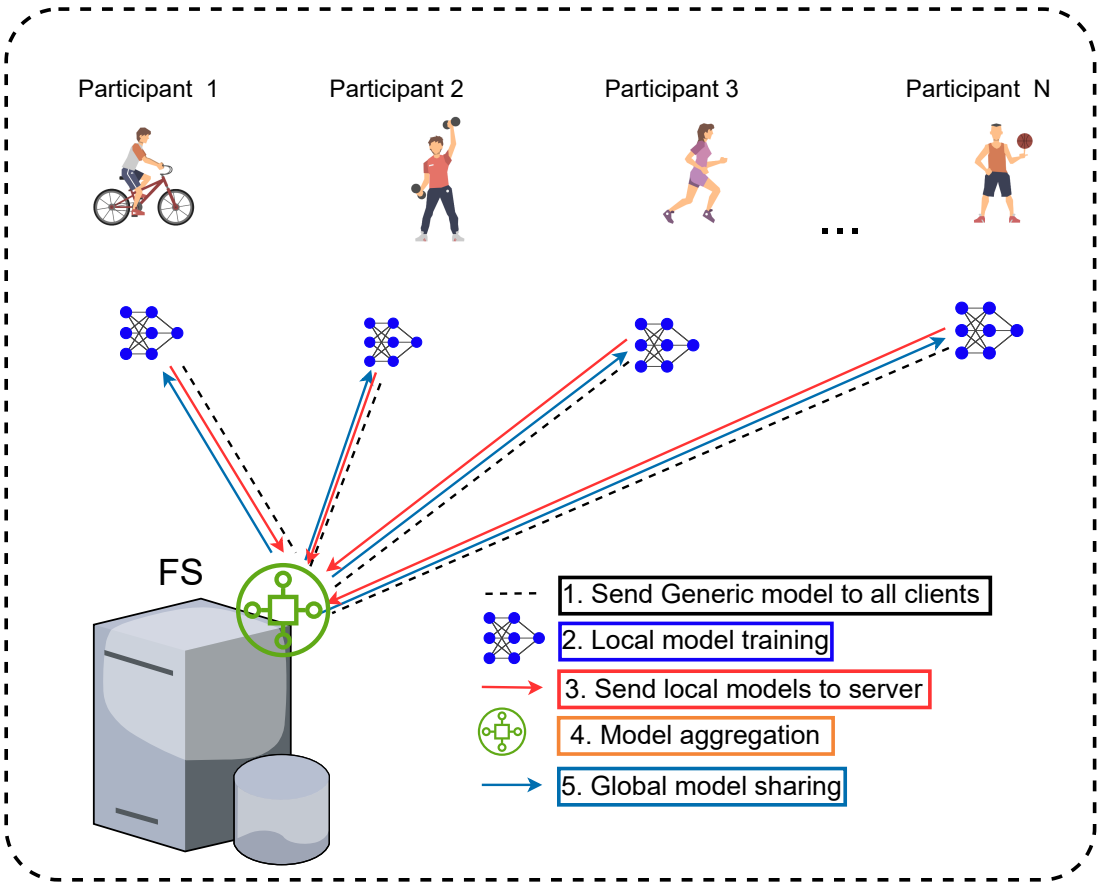


Figure 5.2: Conceptual FL framework for HAR using wearable sensing in the outdoors.

where $\omega_i(t)$ denoting the model parameters for the i-th participant, η as the learning rate, $\tilde{\omega}_i(t-1)$ as the model parameters from the previous iteration, and $\nabla\ell_i(\tilde{\omega}_i(t-1), x_i$ as the gradient of the loss function ℓ with respect to the model parameters for the data point $x_i(t)$. Upon receiving all local updates, the FS conducts model aggregation, expressed as given in eq. (2.19).

5.2.2 Spiking Neural Network

SNNs are inspired by biological neural networks that use discrete events 'spike' for information processing, as shown in Figure 5.3. Unlike ANNs, which process information in a continuous manner, SNNs utilise discrete events to encode and transmit information, embodying a more energy-efficient [200]. The operation of SNNs relies on the idea of event-driven computation. This means that the neurons in the network stay inactive until incoming spikes trigger them. This mechanism makes the computational model highly efficient because only a few neurons are active at any given time, achieving sparsity [201]. Additionally, the binary nature of spike signals (0s and 1s) facilitates processing with low-precision arithmetic, further reducing the computational load. These characteristics collectively endow SNNs with the capacity for high-efficiency processing, particularly on resource-constrained edge devices, a feature that sets them apart from conventional DL models [200, 201].

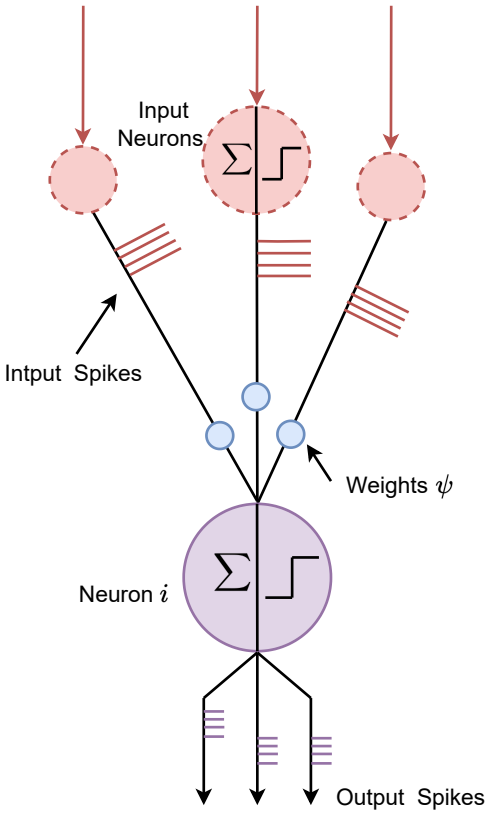


Figure 5.3: Spiking neurons propagation process.

In every spiking neuron, the membrane potential accumulates spike signals received from preceding neurons. This potential varies over time t, potentially increasing, decreasing, or remaining unchanged based on the incoming spikes. When the membrane potential surpasses a predefined threshold v_{th}, it triggers the neuron to emit a spike signal, which is then propagated to subsequent layers in the network. Following spike generation, the neuron undergoes a refractory period during which its membrane potential is temporarily invariant to further spikes, ensuring a period of inactivity post-firing. This dynamic spike generation and transmission process is modelled using the Leaky Integrate-and-Fire (LIF) model. The LIF model is celebrated for its simplicity and effectiveness in capturing the essential characteristics of neuronal spiking behaviour, making it a standard framework for simulating spiking neural networks [202].

LIF neuron model provides a fundamental abstraction of biological neuron dynamics, closely resembling an electrical circuit consisting of a capacitor Q, a resistor Z, a power source V, and an input current J. In this analogy, the neuron's membrane potential $V_i^{(l)}(t)$ is equivalent to the voltage across the capacitor, where the membrane capacitance Q determines the charge-storing capacity, and the resistance Z represents the leakage of the accumulated charge. The neuron integrates incoming synaptic inputs over time, similar to how a capacitor accumulates charge when an external current $J_i(t)$ is applied. The governing equation of the LIF neuron at layer l

and neuron index i can be mathematically expressed as [203]:

$$\tau_q \frac{dV_i^{(l)}(t)}{dt} = -(V_i^{(l)}(t) - V_{res}) + ZJ_i(t), \tag{5.2}$$

where $\tau_q = Q \cdot Z$ represents the membrane time constant, controlling the rate of decay of the membrane potential. The term $V_i^{(l)}(t)$ denotes the instantaneous membrane potential, V_{res} is the resting potential to which the neuron resets after firing, and $J_i(t)$ represents the input current at time t, corresponding to the sum of pre-synaptic inputs. In the absence of external input, the neuron's potential decays over time due to passive leakage, akin to a capacitor slowly discharging through a resistor. To capture this behavior in a discrete-time simulation, the membrane potential update rule is reformulated as [203, 204]:

$$V_i^{(l)}(t) = \beta V_i^{(l)}(t-1) + \sum_j \omega_{ij} o_j^{(l-1)}(t), \tag{5.3}$$

where β $(0 < \beta < 1)$ is the leakage factor governing how much past information is retained, j represents the index of neurons in the previous layer $(l-1)$, w_{ij} is the synaptic weight from neuron j to neuron i, and $o_j^{(l-1)}(t)$ is the binary output of neuron j at time t. The neuron accumulates the weighted sum of its inputs until the membrane potential surpasses a predefined threshold v_{th}, triggering a spike event [203]:

$$o_i^{(l)}(t) = \begin{cases} 1, & \text{if } V_i^{(l)}(t) \geq v_{th} \\ 0, & \text{otherwise} \end{cases} \tag{5.4}$$

This behavior mirrors the breakdown voltage of a capacitor in an electrical circuit, where the accumulated voltage exceeds a critical level, leading to a rapid discharge. The neuron's membrane voltage is subsequently reset to V_{res}, preparing it for the next cycle of integration. In computational neuroscience and neuromorphic computing, this event-driven thresholding mechanism plays a crucial role in enabling energy-efficient SNNs by reducing unnecessary computations. The LIF model thus serves as a biologically inspired and computationally efficient framework for implementing artificial spiking neurons, bridging the gap between neuroscience and hardware-efficient deep learning architectures.

The inherent challenge in training SNNs lies in the non-differentiability of the spike function, a hurdle for traditional gradient descent methods, like backpropagation, which rely on continuous and differentiable activation functions [205]. The main problem with the step function is that its gradient is either zero or undefined, which makes it impossible to update weights during training. To solve this issue, surrogate gradient methods are used. These methods employ a smooth, differentiable approximation of the step function during the backward pass, which makes it possible to compute the gradient. This allows the network to be trained using tra-

ditional techniques while still using the unique spiking behaviour of SNNs during inference. The surrogate piece-wise linear function, aligning with the previously established notation, is mathematically represented as [200, 203]:

$$\frac{\partial o_i^{(l)}(t)}{\partial V_i^{(l)}(t)} = \gamma \max\left(0, 1 - \frac{|V_i^{(l)}(t) - v_{th}|}{v_{th}}\right), \tag{5.5}$$

where γ is a scaling factor controlling the SNNs' update magnitude and v_{th} is the threshold voltage for spiking. The backpropagation method in SNNs mirrors that of ANNs, except for using a surrogate function to approximate the non-differentiable threshold function. Hence, the weight update rule on the local participant for layer l at a given time is mathematically represented as [204]

$$\omega_{ij}^{(l)}(t) = \omega_{ij}^{(l)}(t-1) - \eta \frac{\partial \ell(t)}{\partial \omega_{ij}^{(l)}(t-1)}, \tag{5.6}$$

where $\omega_{ij}^{(l)}(t-1)$ represent the model parameters at $t-1$ for neuron j in layer l to neuron i. Numerous strategies have been developed to effectively utilise the capabilities of SNNs and overcome their inherent challenges, especially the non-differentiability of spike operations. Among these strategies, Spike-Timing-Dependent Plasticity (STDP) is particularly useful for unsupervised learning, as it leverages the temporal dynamics of spikes and is well-suited for handling unlabelled data [206].

On the other hand, supervised learning scenarios benefit from the Backpropagation Through Time (BPTT) technique, which modifies traditional backpropagation to allow for error correction over sequential time steps [200], thus facilitating the analysis of spatiotemporal data. Additionally, surrogate gradient learning introduces a means to approximate the non-differentiable spike function with a continuous, differentiable surrogate, enabling the use of gradient-based optimisation techniques [200]. Parallelly, reward-modulated STDP integrates principles from reinforcement learning, applying rewards to modulate STDP in a feedback-driven learning process [207]. Our investigation focuses on the supervised training of spiking neural networks with diverse datasets. This has led to the development of a hybrid model that combines BPTT with surrogate gradient methods, as given in [208]. This approach addresses two pivotal challenges: It overcomes the spike function's non-differentiability, a significant obstacle for conventional gradient-based optimisation, and it equips the model to learn temporal patterns in HAR data proficiently. By extending BPTT to facilitate backward error propagation through time, the model optimises a loss function ℓ, thereby improving its predictive accuracy. The formulation for the gradient of the loss ℓ with respect to synaptic weights ω_{ij} and the neuron's membrane potential is given as:

$$\frac{\partial \ell(t)}{\partial \omega_{ij}} = \sum_t \frac{\partial \ell(t)}{\partial V_i^{(l)}(t)} \cdot o_j^{(l-1)}(t), \tag{5.7}$$

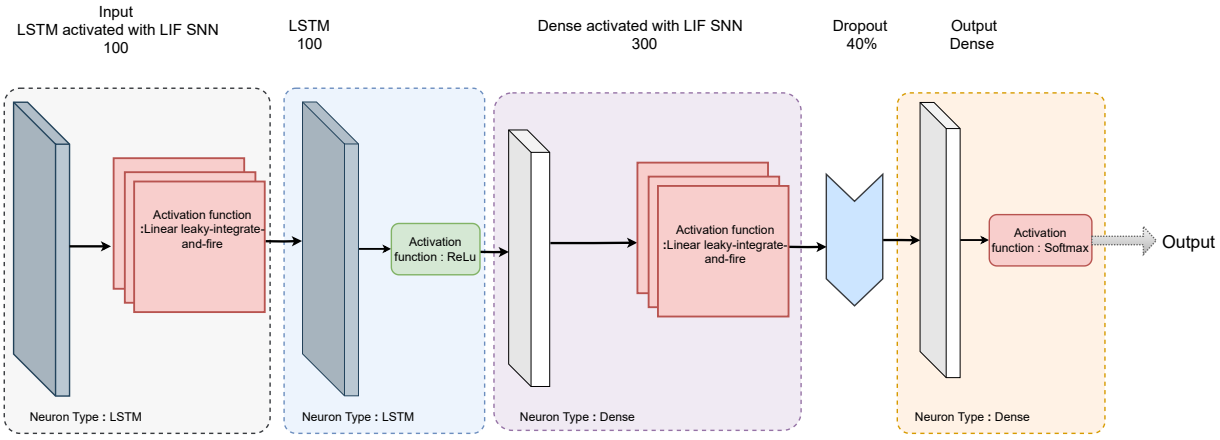


Figure 5.4: Proposed hybrid S-LSTM model where input LSTM layer activated by LIF.

$$\frac{\partial \ell(t)}{\partial V_i^{(l)}(t)} = \frac{\partial \ell(t)}{\partial o_i^{(l)}(t)} \cdot \frac{\partial o_i^{(l)}(t)}{\partial V_i^{(l)}(t)} + \beta \frac{\partial \ell(t)}{\partial V_i^{(l)}(t+1)}, \tag{5.8}$$

where $o_j^{(l-1)}(t)$ signifies the output from neuron j in the preceding layer $l-1$ at time t, influencing the input to neuron i in layer l. This formulation underscores our system's ability to navigate the complexities inherent in training SNNs, enabling detailed learning of temporal dependencies critical for HAR applications.

5.2.3 Proposed S-LSTM Model

Our proposed S-LSTM model seamlessly combines LSTM units with the spiking behaviour of LIF neurons, as shown in Figure 5.4. Initially, the framework employs an LSTM layer composed of 100 neurons dedicated to analysing the input data. This particular layer is designed to return sequences, capturing the essential temporal correlations present within the data. Subsequently, the spiking layer with LIF replaces the conventional activation functions. The spiking layer has a trainable threshold that determines neuron firing and uses a surrogate gradient to approximate the gradient during backpropagation due to the non-differentiable nature of spiking behaviour. Following the spiking neural layer, an additional LSTM layer, also comprising 100 neurons, further refines the data sequences. This is followed by a dense layer, integrating 300 neurons, which also adopts LIF neurons. Additionally, a dropout layer is added to mitigate overfitting, followed by a fully connected output layer. The output layer uses a SoftMax activation function, producing a probability distribution over the possible activity classes. The training process of federated S-LSTM for HAR is given in Algorithm 5.

Algorithm 5: Federated S-LSTM training with surrogate gradient and BPTT.

Input: Initial model parameters ω_0, clients $i = \{1, 2, \ldots, N\}$
Output: Trained model parameters ω_t
Procedure: Initialisation
for each client i **in parallel do**
    $\mathscr{D}_i \leftarrow$ local dataset
    $\omega_i \leftarrow \omega_0$ {Initialise local model}
end for
Procedure: FL training
for round $t = \{1, 2, \ldots, T\}$ **do**
    for each client $i = \{1, 2, \ldots, N\}$ **in parallel do**
        $\Delta\omega_i \leftarrow \text{LocalTraining}(\omega_i, \mathscr{D}_i)$
    end for
    $\omega_{t+1} \leftarrow \text{ServerUpdate}(\{\Delta\omega_i\}_{i=1}^{N})$ {Aggregate updates}
    broadcast ω_{t+1} to clients
end for
Procedure: LocalTraining($\omega, \mathscr{D}$)
Initialise local parameters ω_i, learning rate η
for each time step $t = \{1, \ldots, T\}$ **do**
    Compute local gradient using surrogate gradient and BPTT {Based on Equ (5.5)–(5.9)}
    Update local model parameters $\omega_i(t)$
end for
return model update ω
Procedure: ServerUpdate($\{\Delta\omega_i\}_{i=1}^{N}$)
$\omega \leftarrow$ **aggregate**($\{\Delta\omega_i\}$)
return updated model parameters ω_{t+1}

5.3 Simulation Setup

This section provides a detailed explanation of the methodologies we employed, offering an in-depth exploration of the datasets used, the criteria used to evaluate performance, and the metrics used to determine the effectiveness of our proposed model. Our discussion thoroughly analyses the datasets, highlighting the unique challenges and considerations of HAR with wearable sensor data. Furthermore, it discusses the performance evaluation strategy and metrics used in the study.

5.3.1 Dataset Description

Despite being a well-researched topic, evaluating HAR using smartphone data is a recent and active area of research. However, using wearable sensing data for HAR offers both opportunities and challenges. The complexity of such datasets arises from various factors, including sensor configurations, sampling rates, accessibility, and diversity of the collected data. Furthermore, distinct activity patterns among different classes lead to significant class imbalances, making HAR an ideal domain for evaluating the effectiveness of neuromorphic federated learning ap-

proaches under diverse conditions. In our pursuit of datasets that offer reproducibility, diversity, and realism, we found two publicly accessible datasets. The first, known as the UCI dataset [209], is a staple in HAR research due to its widespread use in benchmarking. However, the UCI dataset is collected in a highly controlled laboratory setting, and its relatively small sample size presents certain limitations. To broaden the scope of our analysis, we incorporated the Real-World dataset [210], which was collected in unconstrained outdoor environments. Further details on each dataset, including its unique characteristics and challenges, are discussed below.

UCI dataset

The UCI dataset was collected using Samsung Galaxy S II smartphones worn by a diverse group of 30 participants of various ages and genders. These individuals engaged in six everyday activities, including walking, walking upstairs, walking downstairs, sitting, standing, and lying down. The activities were performed under varied conditions, with the smartphones positioned on the left wrist and at a location of the participant's choosing to simulate real-world usage scenarios. The embedded sensors in the smartphones, specifically the accelerometers and gyroscopes, were instrumental in capturing data on triaxial linear acceleration and angular velocity, achieving a sampling rate of 50 Hz. This dataset was extensively processed to remove noise and enhance signal quality using advanced filtering techniques [209]. A total of 17 distinct features were meticulously extracted from each signal, covering a broad spectrum of time and frequency domain characteristics, such as signal magnitude, jerk, and the application of the fast Fourier transform (FFT).

The collected signals were segmented into discrete windows lasting 2.56 seconds each to facilitate a detailed analysis, with a 50% overlap between consecutive windows [209]. This segmentation process yielded 561 unique features for each window, drawn from various statistical and frequency-domain analyses. The comprehensive dataset encompasses over 10,299 instances, thoughtfully divided into training (70%) and testing (30%) subsets to support robust model evaluation. However, the dataset was intentionally partitioned into five distinct subsets to simulate localised datasets for individual participants, mirroring real-world federated learning scenarios where data distribution is inherently uneven. Each participant's data was further divided, allocating 80% for training and 20% for testing purposes, with the testing portions aggregated to create a global test set. This organised approach emphasises the adaptability of the dataset in federated learning research and demonstrates the meticulous technique used for data preprocessing and merging sensor data. The preprocessing methods, such as noise reduction and segregating sensor signals into significant components, highlight the dataset's usefulness in capturing and analysing intricate human activities through multi-sensor integration.

Real-World Dataset

While the UCI dataset is widely used in HAR research, it has limitations because it was collected in a controlled laboratory setting. Additionally, due to the small sample size, the true potential of FL could not be explored. Therefore, we opted for a more realistic dataset collected by Sztyler and Stuckenschmidt [210]. This dataset comprises data from 15 individuals (eight males and seven females) embodying a broad spectrum of daily activities such as climbing stairs, jumping, lying, standing, sitting, running, and walking. Distinctively, this dataset captures accelerometer, GPS, gyroscope, light, magnetic field, and sound levels. We used the accelerometer and gyroscope data in which sensors were placed on seven strategic body locations: the chest, forearm, head, shin, thigh, upper arm, and waist, thereby providing a comprehensive view of bodily movements during various activities.

The data collection utilised standard smartphones and a smartwatch affixed to these body positions, recording at a frequency of 50 Hz. The devices used were synchronised using network time services to guarantee accuracy in time-stamping the collected data [210]. To mirror real-life usage scenarios as closely as possible, the devices were attached to the body using everyday items such as sports armbands and pockets. This method ensured comfort and realism, simulating how wearable devices are typically worn during daily activities. The data collection spanned various real-world locations, including urban and natural settings, to capture a diverse range of movement patterns and environmental influences on sensor data.

The innovative aspect of our study is multi-sensor fusion, incorporating HAR accelerometer and gyroscope data in federated settings. The preprocessing and feature extraction stages are pivotal in transforming the raw sensor data into a format suitable for analysis. The data was segmented into windows, each spanning one second and overlapping by half to ensure continuity and capture transitions between activities effectively [210]. Various time and frequency-based features were extracted from these windows, including applying a discrete Fourier transform to translate time-based signals into the frequency domain. Our data processing technique included the unique feature of computing gravity-based characteristics to determine the position of the wearable device on the body. A low-pass filter was applied to segregate the acceleration data into gravitational and body movement components, allowing us to compute the orientation of the device. This orientation data helped us understand the context of activity recognition. To avoid over-fitting and improve the generalisability of the model, we categorised these orientations into predefined groups [210].

The Real-World dataset was suitable for the HAR study because it was captured in a natural environment and had a realistic class imbalance. For example, the jumping activity only made up 2% of the data, while standing accounted for 14% of the total data. Moreover, the high-class imbalance and the availability of separated user data made it a perfect fit for a comprehensive study on FL approaches for HAR.

5.3.2 Performance Metrics

HAR is a multi-class classification where multiple metrics are used to assess the effectiveness of the DL model. The commonly used metric accuracy, the proportion of correctly identified activities among all classifications, is a fundamental metric. However, the utility of accuracy is often limited in datasets characterised by a significant imbalance among classes. This imbalance can lead to the accuracy paradox, where models might exhibit high accuracy by predominantly predicting the majority class, thus neglecting the nuanced detection of less frequent activities. We employ additional metrics, precision, recall, and the F1-score to address these limitations and provide a more comprehensive evaluation as discussed in Section. 2.2.2.

All experimental procedures were conducted in a simulated environment for a comprehensive evaluation. This allowed us to evaluate its effectiveness using two strategies: global performance evaluation and personalized model assessment. The global assessment evaluates the model's capability across the entire dataset by leveraging a global test set to infer its generalization potential. On the other hand, personalised evaluations are conducted at an individual participant level, using local data to refine the global model and tailor predictions more closely to individual patterns. This bifurcated approach enables a comparison between the model's universal applicability and its customized effectiveness. In addition, we have expanded our evaluation criteria to include energy efficiency, which is a crucial factor in FL situations where computational and communication resources are limited. The energy efficiency metric is based on the computational requirements of local training and the amount of data transmitted during each communication round, which is discussed in Section. 2.2.3, and mathematically given in equ. (2.18).

5.4 Results and Discussion

In this chapter, the proposed hybrid neuromorphic S-LSTM model is rigorously evaluated using two distant publicly available datasets (UCI and Real-World), focusing on applying HAR indoors and outdoors. This study aims to assess the performance of the proposed S-LSTM in terms of accuracy, energy efficiency, and adaptability within the context of wearable sensor data fusion. The evaluation begins with the UCI dataset, which provides a controlled indoor environment to benchmark the capabilities of the proposed S-LSTM model against traditional DL architectures such as LSTM and CNN. The UCI dataset, with its structured activities and controlled settings, offers an ideal scenario for scrutinising the nuanced differences between the models. It allows us to explore the effectiveness of the S-LSTM model in capturing and learning from the temporal sequences inherent in human activities. Following the UCI dataset, we extend our evaluation to the Real-World dataset, which presents a more challenging and varied set of outdoor activities, thus testing the robustness and generalisability of our model in scenarios closer to everyday human behaviours.

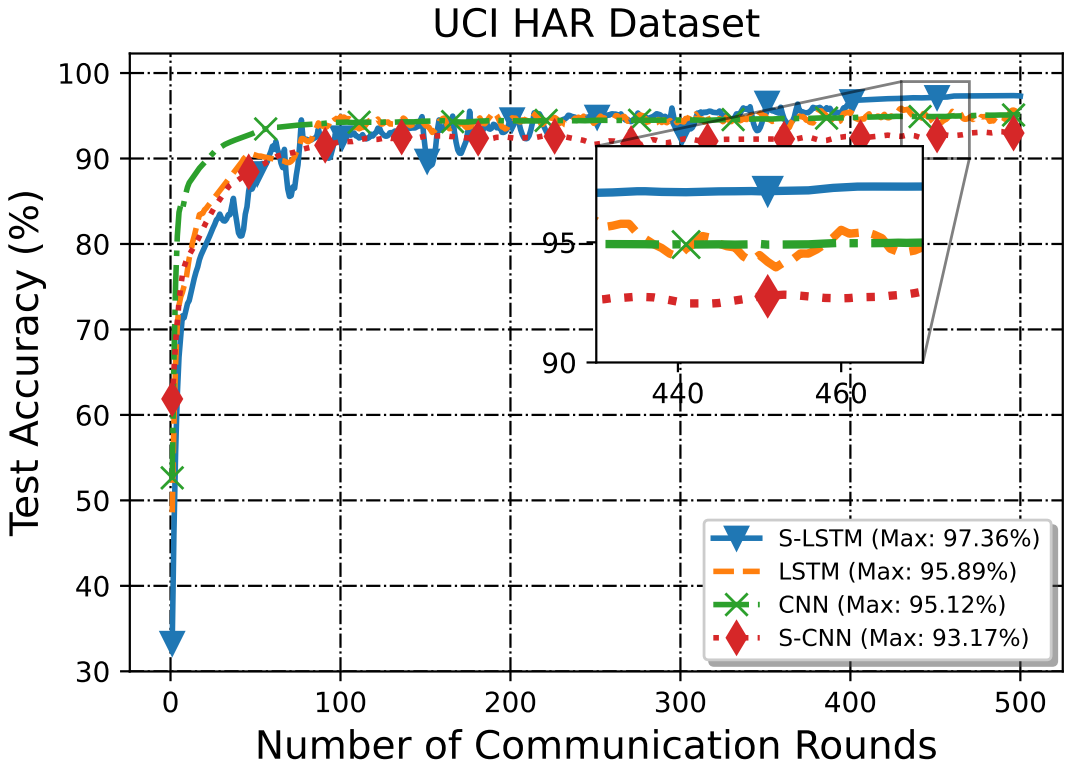


Figure 5.5: Learning curve representing the accuracy for UCI-dataset obtained using global test set.

5.4.1 UCI Results

As previously discussed, the UCI dataset covers various indoor human activities, making it a suitable option to evaluate our model's effectiveness in a controlled setting. This dataset is divided among five participants acting as the edge nodes capable of model training without data sharing. To analyse the results, we used an 80-20 split for training and testing the model on each participant, creating separate local training and testing sets. Later, we aggregated the test set of each participant to create a global test set, which allowed us to assess the robustness of the proposed architecture.

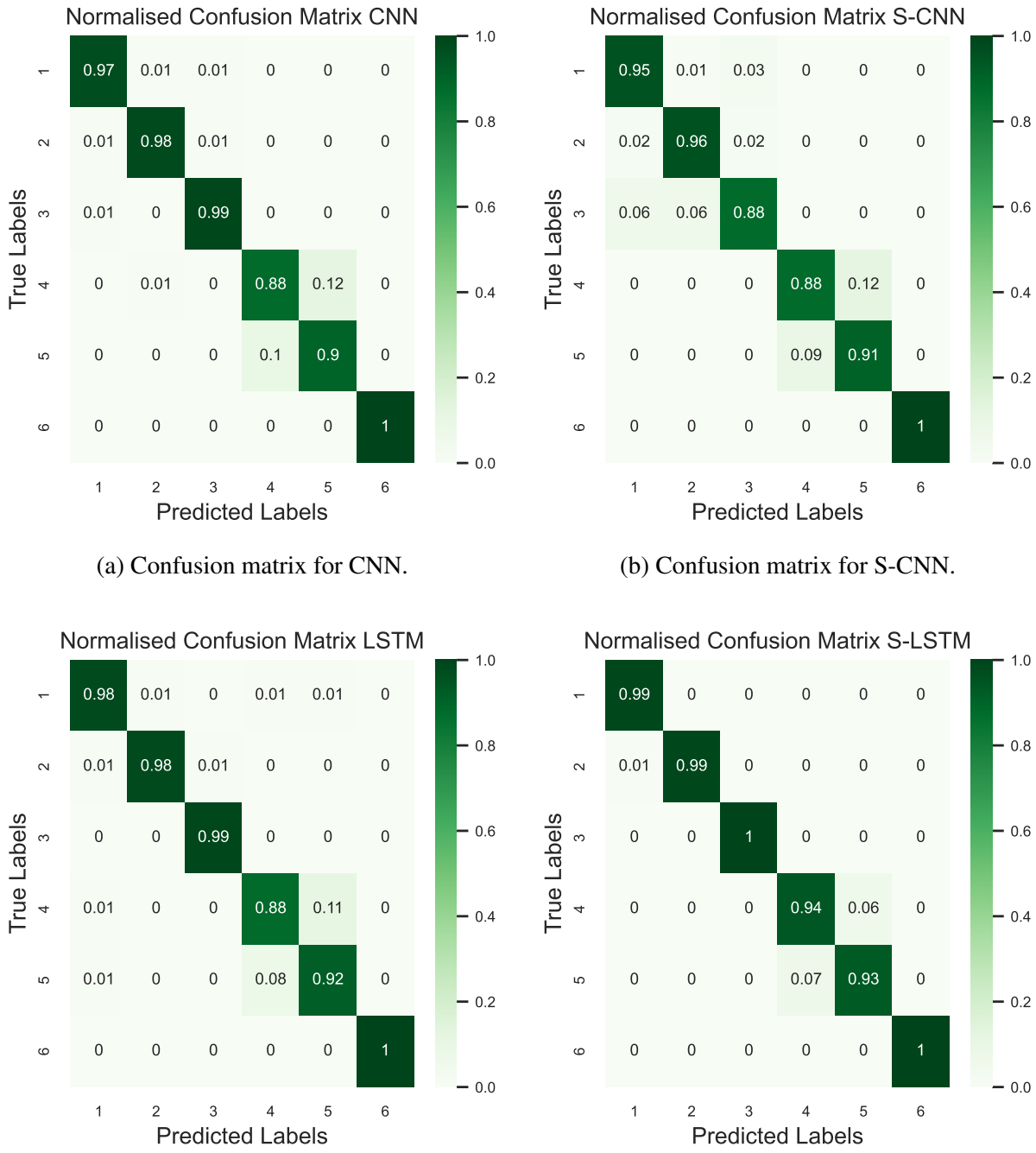
It is important to note that this evaluation is done in simulations, where each model training spans 500 communication rounds in a federated setting. Each model is trained for 3 epochs locally and shares the model parameters with FS, where FedAVG is adopted for model aggregation. In our first comparison, the results in Figure. 5.5, represent the accuracy of the learning curve obtained using a global test set in the training process. This learning curve exhibits a typical pattern of a rapid increase in accuracy with more steady-state behaviour as training approaches its performance limits. It is noteworthy that the hybrid S-LSTM model performed better than both LSTM and CNN models, reaching a maximum accuracy of 97.36%. The LSTM model came in a close second with a peak accuracy of 96.30%, while the CNN and S-CNN models had slightly lower results, peaking at 95.14% and 93.25%, respectively.

Confusion matrices provide a comprehensive predictive accuracy metric across activities to analyse model performance effectively on the UCI HAR dataset. The results in Figure 5.6, provide an insight into the predictive accuracy of the CNN, S-CNN, LSTM, and S-LSTM models for individual classes. Each figure represents a normalised confusion matrix, where the diagonal elements indicate the percentage of correct predicted labels for each class. It is worth noting that class 6 (lying) had a true positive rate of 100% across all models, as it is an easily distinguishable activity. The results in Figure 5.6 (d) show that the S-LSTM model performed well and distinguished between classes 1, 2, and 3, with true positive rates exceeding 0.99%. Additionally, the model showed impressive proficiency in distinguishing other classes like 4 (sitting) and 5 (standing), with a misclassification rate of only 0.06%. This demonstrates its enhanced ability to identify the subtle temporal patterns that differentiate these activities. The CNN and S-CNN models displayed slightly higher misclassification rates of 0.12% and 0.11%, respectively, for classes 4 and 5, while the LSTM showed a rate of 0.11%. These models performed well but were not as proficient as the S-LSTM model in distinguishing between the relevant classes.

The results presented in Table 5.1, provide a detailed comparison of the performance metrics, including precision, recall, and F1-score. The models were evaluated based on their classification capabilities for six common daily activities. The proposed S-LSTM model outperforms others, with high F1-scores of 0.99% for activities like walking and walking upstairs and a perfect F1-score of 1.00 for other activities like walking downstairs and lying. The uniform success in identifying the lying activity across all models can be attributed to the unique motion patterns associated with this activity. However, the analysis has identified a challenge in distinguishing between sitting and standing activities. The proposed S-LSTM model has a small advantage with F1-scores of 0.93% for sitting and 0.94% for standing. Moreover, the CNN and S-CNN models have lower efficacy, with F1-scores ranging between 0.88 and 0.90. This pattern suggests that the motion characteristics of sitting and standing are similar, posing challenges for models, especially CNN and S-CNN, in accurately distinguishing between these two activities.

Table 5.1: Comparative results of global models for CNN, S-CNN, LSTM, and S-LSTM for the UCI dataset trained in a federated environment. Here **P, R, F1** represents precision, recall and F1-score.

Class	**CNN**			**S-CNN**			**LSTM**			**S-LSTM**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Walking	0.98	0.97	0.98	0.93	0.95	0.94	0.99	0.98	0.99	0.99	0.99	0.99
Walking up	0.98	0.98	0.98	0.93	0.96	0.95	0.99	0.99	0.99	0.99	0.99	0.99
Walking down	0.98	0.99	0.98	0.93	0.88	0.91	0.99	0.99	0.99	1.00	1.00	1.00
Sitting	0.89	0.88	0.88	0.90	0.88	0.89	0.91	0.90	0.91	0.93	0.94	0.93
Standing	0.89	0.90	0.89	0.89	0.91	0.90	0.91	0.92	0.91	0.94	0.93	0.94
Lying	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00



(a) Confusion matrix for CNN.

(b) Confusion matrix for S-CNN.

(c) Confusion matrix for LSTM.

(d) Confusion matrix for S-LSTM.

Figure 5.6: The confusion matrix for four DL models compared in this study for UCI HAR dataset, where the index number represents the activity. The labels corresponding to the activities are (1) walking, (2) walking upstairs, (3) walking downstairs, (4) sitting, (5) standing, and (6) lying.

5.4.2 Real-World Dataset Results

In the Real-World dataset, simulations were conducted over 500 communication rounds with the participation of 15 edge nodes. During each communication round, every participant was trained

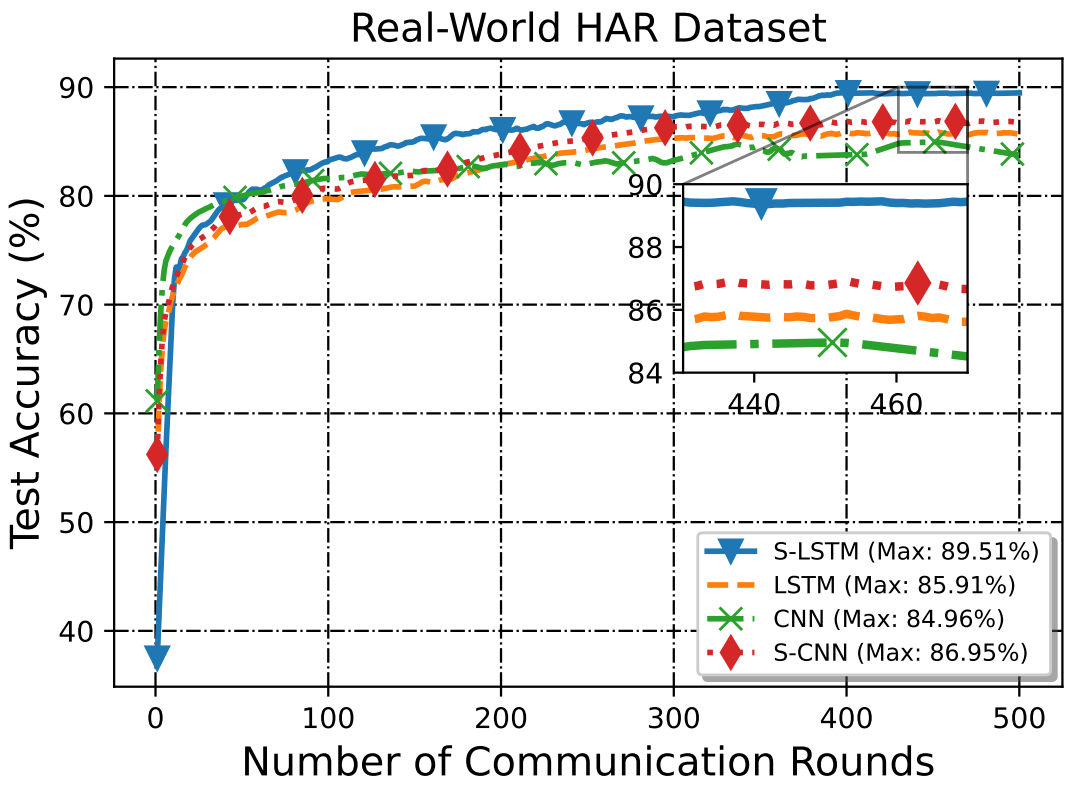
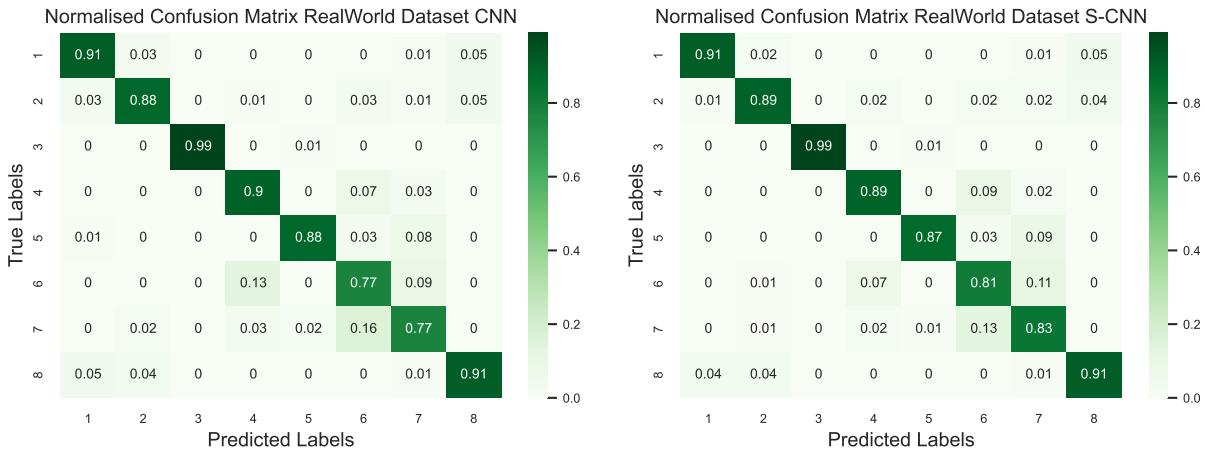


Figure 5.7: Learning curve of accuracy obtained using a global test set for Real-World dataset spanning 500 communication rounds.

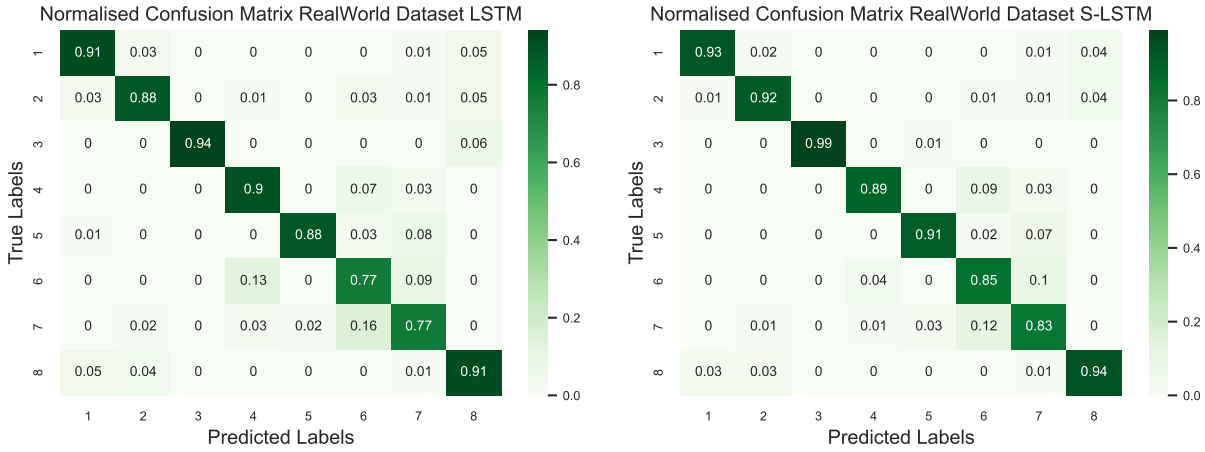
for five local epochs. The dataset was split into 80-20 for each participant to train and test, respectively. The global test set was generated by aggregating the test sets of all participants. The performance of the S-LSTM model was evaluated using this global test set. Figure 5.7 illustrates the accuracy learning curve during training. The Real-World dataset's learning curve displayed a sharp increase in accuracy during the initial rounds, followed by a steadier response towards the end of the simulation. The results demonstrated that S-LSTM and S-CNN perform better than their traditional counterparts, LSTM and CNN, with the highest accuracy of 89.69% and 86.90%, respectively. In contrast, the LSTM and CNN models achieved accuracies of 85.85% and 84.97%, respectively. This comparison of learning curves highlighted the superior performance of hybrid spiking models and their potential to capture complex human activities in less controlled outdoor environments.

To further consolidate the discussion, the confusion matrix of the four models considered in this study is illustrated in Figure 5.8. More specifically, the results in Figure 5.8 (d) provide the classification ability of S-LSTM for eight outdoor activities in the Real-World HAR dataset. The results show that the S-LSTM model is proficient in classifying more prominent activities, with a score of 0.96% for jumping, 0.86% for running, and 0.91% for walking. However, the confusion matrix also reveals that the S-LSTM model encountered challenges in classifying activities with subtle motion patterns, such as climbing down 0.88%, climbing up 0.87%, and standing 0.80%.



(a) Confusion matrix for CNN.

(b) Confusion matrix for S-CNN.



(c) Confusion matrix for LSTM.

(d) Confusion matrix for S-LSTM.

Figure 5.8: The confusion matrix CNN, S-CNN, LSTM and S-LSTM models for Real-World data set. The index represents the activity, where the label corresponding to the activities are: (1) climbing down, (2) climbing up, (3) jumping, (4) lying, (5) running, (6) sitting, (7) standing, (8) walking.

Misclassifications mainly occur between activities that share similar motion characteristics. For instance, climbing down was occasionally misclassified as walking, climbing up was confused with running, and sitting was sometimes classified as standing.

The results in Table 5.2 extensively analyse various models trained using the Real-World HAR dataset in a federated learning context. The proposed S-LSTM has consistently demonstrated superior performance among the models, surpassing the other models in precision, recall, and F1-score. For instance, when classifying the walking activity, the S-LSTM model achieved a precision of 0.94%, a recall of 0.93%, and an F1-score of 0.94%, which sets a benchmark that surpasses other models in this study. The S-LSTM model performs better in accurately identifying and classifying complex temporal activity patterns. This is evident from its remarkable

Table 5.2: Comparison of different DL techniques for Real-World dataset. Here **P, R, F1** represents precision, recall, and F1-score, respectively.

Class	**CNN**			**S-CNN**			**LSTM**			**S-LSTM**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Climb down	0.90	0.91	0.90	0.92	0.91	0.91	0.90	0.91	0.90	0.94	0.93	0.94
Climb up	0.90	0.88	0.89	0.92	0.89	0.90	0.90	0.88	0.89	0.93	0.92	0.93
Jumping	0.92	0.91	0.91	0.89	0.92	0.94	0.99	0.94	0.96	0.94	0.93	0.94
Lying	0.84	0.90	0.87	0.89	0.89	0.89	0.84	0.90	0.87	0.95	0.89	0.92
Running	0.98	0.88	0.93	0.98	0.87	0.93	0.98	0.88	0.93	0.97	0.91	0.94
Sitting	0.73	0.77	0.75	0.74	0.81	0.77	0.73	0.77	0.75	0.78	0.85	0.82
Standing	0.77	0.77	0.77	0.75	0.83	0.79	0.77	0.77	0.77	0.79	0.83	0.81
Walking	0.91	0.91	0.91	0.92	0.91	0.91	0.91	0.91	0.91	0.93	0.94	0.93

F1-scores of 0.94% for climbing down and 0.93% for climbing up. These scores demonstrate the model's proficiency in distinguishing and categorising intricate motion sequences. It is worth noting that all models examined encountered challenges in differentiating between the 'sitting' and 'standing' activities. The F1 scores in these categories remained relatively modest, ranging from 0.73% to 0.85%. However, the S-LSTM model achieved marginally higher scores in these challenging classifications, confirming its robustness and the effectiveness of its learning architecture in complex scenario discrimination.

5.4.3 Energy Efficiency Comparison

To further enhance our comparative study, we implemented a strategy where only 50% of edge nodes would participate randomly during the training process. The idea behind randomly selecting clients is to reduce communication overheads and assess the trade-off between accuracy and communication cost. For comparison, the results shown in Figure 5.9 illustrate the accuracy of the learning curve obtained during the training process using a global test set of 500 rounds. This approach was used to determine how reducing client participation affects model performance and communication cost benefits. The results indicate that the proposed S-LSTM model achieved the highest accuracy of 88.11%, while the counterpart LSTM reached a maximum of 84.43%. The CNN and S-CNN models achieved accuracies of 83.98% and 85.20%, respectively. More specifically, the random client participation significantly reduced communication costs in this case, with only a slight variation in performance.

To further analyse the energy efficiency of the proposed hybrid model, rigorous evaluations were conducted by quantifying energy estimates as defined by Eq. (5.12). The energy efficiency is calculated for a single communication round with all participants and a randomly selected 50% of participants. For the simplicity of our analysis, the computation constant α and communication constant β were set to 0.003 and 0.0001, respectively, as adopted from past

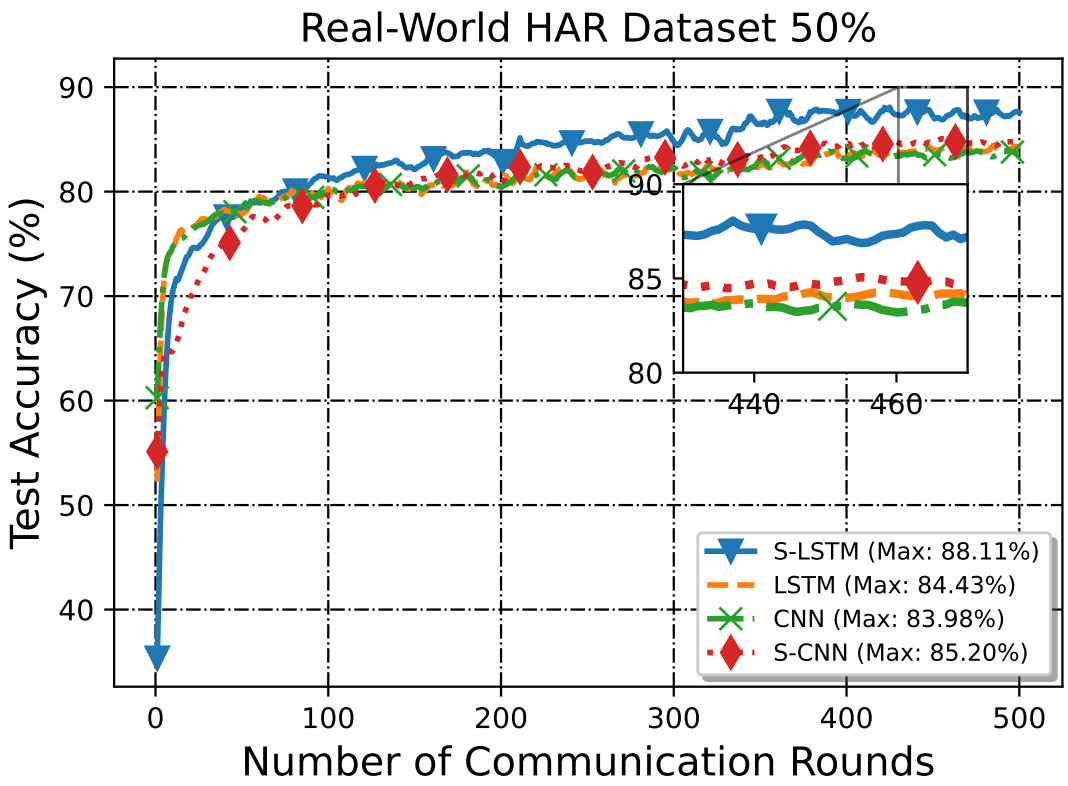


Figure 5.9: Learning curve for Real-World dataset, with 50% random client participant trained for 500 communication rounds.

literature [31, 32]. The results in Table 5.3 present the energy estimates E_{est}, which are dependent on both computation time and model parameters P_{trn}, under the specified client selection criteria. Notably, the S-LSTM model had a minimum computation time of 208 s with all participants, further reduced to 121 s with 50% client participation. Regarding energy consumption, the S-LSTM model demonstrated superior efficiency, with the lowest energy estimate of 6.10 watts. This represents a 24.41% decrease compared to the LSTM model, which had an energy estimate of 8.07 watts. The results confirm that selecting 50% of participants at random significantly improved the model's energy efficiency. However, it's important to consider the accuracy-efficiency trade-off when using DL models, especially for energy-constrained applications. Finding the optimal balance between accuracy and efficiency is critical for deploying these models.

5.4.4 Personalised Model Comparison

The previous results were obtained and discussed using the global model and test set. However, using local data, participants can customise the model to their needs. Therefore, the global model is fine-tuned using local data to create a personalised model. The personalised performance is then compared to the global model. The results in Figure 5.10 compare the global and person-

Table 5.3: Comparison of energy efficiency for different models with 100% and 50% client selection per communication round.

Model	Client Selection (%)	Computation Time t_{com} (s)	Energy Estimate E_{est} (W)
CNN	100	258	38.24
	50	143	15.73
S-CNN	100	252	29.38
	50	136	15.67
LSTM	100	220	8.07
	50	137	4.32
S-LSTM	100	208	6.10
	50	121	3.27

alised models on test accuracy across 15 participants. The x-axis represents the client number, while the y-axis plots the test accuracy. For each client, two bars represent the global and personalised model accuracy. The results show that personalisation significantly improved the local test accuracy compared to the global model for all models under consideration. Specifically, the proposed S-LSTM personalised model achieved the highest average accuracy of 97.12% across clients, substantially higher than its 89.69% global accuracy. The other models showed similar trends of increased accuracy with personalisation, with averages of 95.39% (LSTM), 95.96% (S-CNN), and 95.10% (CNN).

5.4.5 Discussion on Performance, Scalability, and Energy Trade-offs

The evaluation of the hybrid neuromorphic S-LSTM model for HAR in federated environments highlights critical trade-offs in terms of model performance, computational efficiency, scalability, and learning paradigms. This section provides insights into the impact of EN participation, model personalisation, global vs. local learning trade-offs, and energy consumption.

Effect of Number of Clients on Model Performance

The number of participating ENs significantly influences model generalisability and convergence in FL. As demonstrated in the Real-World dataset experiments, reducing client participation to 50% per communication round led to only a minor accuracy degradation (from 89.69% to 88.11%) Fig. 5.9. However, this reduction lowered the energy consumption by 46.3%, highlighting an essential trade-off between model accuracy and energy efficiency. The UCI dataset results also confirmed that a larger number of active clients enhances convergence stability, as observed from the learning curves in Fig. 5.5. These findings suggest that selective client participation strategies can significantly optimize energy efficiency while maintaining model performance, making FL more practical for resource-constrained environments.

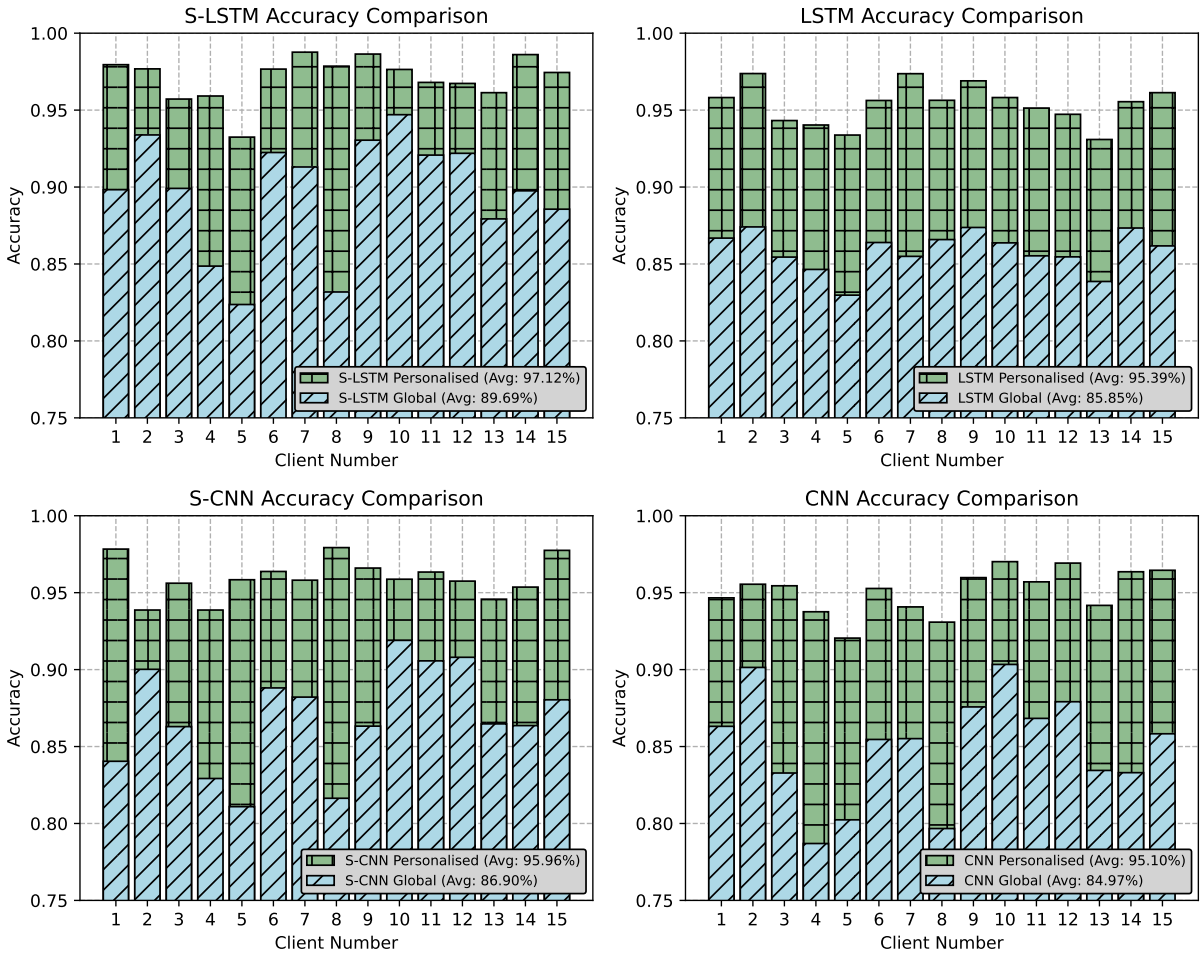


Figure 5.10: Accuracy comparison graph for global and personalised models for participants. The personalised accuracy was obtained after fine-tuning using the local dataset.

Trade-off Between Global, Local, and Personalised Learning

The comparison between global and personalised models revealed significant performance improvements when clients fine-tuned the global model on their local datasets. Fig. 5.10 shows that personalised S-LSTM models achieved an average accuracy of 97.12%, significantly surpassing the global model's 89.69% accuracy. This highlights the importance of client-specific adaptation in FL, where global models may not generalise well across diverse activity patterns. In contrast, local training without FL resulted in lower accuracy due to the limited availability of training data per client. These results demonstrate that a hybrid FL approach, where a global model is first trained and then personalised at each client, offers the best trade-off between generalisation and adaptation.

5.4.6 Energy Efficiency Across Learning Paradigms

Energy consumption plays a crucial role in wearable sensor-based HAR applications, where computational resources are often limited. The proposed S-LSTM model demonstrated the low-

est energy consumption, consuming 6.10 W with full client participation and only 3.27 W with 50% participation as given in Table. 5.3. Compared to LSTM (8.07 W), CNN (38.24 W), and S-CNN (29.38 W), the spiking-based S-LSTM model achieved a 24.41% reduction in energy consumption compared to LSTM, highlighting its efficiency for real-world deployment. Moreover, reducing the client selection rate from 100% to 50% further enhanced energy efficiency, proving that selective client participation strategies can significantly optimise energy usage without major performance degradation.

5.5 Summary

In this chapter, we proposed an HNFL framework that synergises the computational efficiency of SNNs with the dynamic temporal learning capabilities of LSTM networks for HAR using multimodel data from wearable sensors, addressing challenges **C1** and **C3**. This integrated S-LSTM model capitalises on LSTM layers to adeptly capture temporal dependencies within time-series sensor data while incorporating spiking layers to facilitate event-driven processing, thereby enhancing energy efficiency in federated settings. The model training leverages surrogate gradient learning and BPTT, facilitating supervised end-to-end learning. Our approach has been tested on two publicly available HAR datasets, UCI and Real-World. The UCI dataset is for indoor environments, while the Real-World dataset is for outdoor settings. Our evaluations demonstrated the superior performance of the proposed S-LSTM model in a FL paradigm compared to traditional models like LSTM, CNN, and S-CNN. The simulation results show concrete evidence that the S-LSTM model outperforms the LSTM model in accuracy. In controlled indoor environments represented by the UCI dataset, the S-LSTM model outperformed the LSTM model by 1.06%. However, its performance in the complex and diverse outdoor settings of the Real-World dataset was even more impressive, with a significant 3.84% increase in accuracy over the LSTM model. This improvement highlights the S-LSTM model's robustness and ability to handle the unpredictable nature of real-world human activities. Moreover, our findings highlighted a significant 32.30% improvement in energy efficiency compared to the LSTM model. Randomly selecting participants for model training improves energy efficiency but affects accuracy. However, personalising the global model by fine-tuning it with local data can significantly increase performance. On average, this approach improved accuracy by 9% across participants.

Chapter 6

Federated Fusion and Model Quantisation

Our final chapter introduces FedFusionQuant (FFQ), a novel FL framework designed to further reduce computational and communication overhead in distributed learning systems, addressing the challenges **C2** and **C3**, as discussed in **Section 1.3**. The proposed FFQ framework incorporates signal processing, feature fusion and model compression through QAT. Additionally, the proposed framework employed a customised FedDist algorithm for adaptive parameter tuning based on neuron dissimilarity measures to effectively mitigate overfitting. Moreover, incorporating QAT allows the model to maintain high accuracy while substantially reducing model size. This framework showcases how strategic model compression and feature fusion can overcome the computational and communication constraints inherent in FL for HAR, paving the way for more efficient, real-time implementations in resource-constrained environments. This chapter highlights how edge based feature fusion and model compression techniques can be synergistically employed within FL to advance the scalability and practicality of HAR systems, particularly in resource-constrained environments.

6.1 Introduction

The advent of pervasive computing has significantly transformed indoor environments, integrating smart devices to create intelligent living spaces. This technological synergy has propelled the development of applications ranging from real-time monitoring and security surveillance to automation in building and industrial processes, facilitating scalable local autonomy [211, 212]. HAR and indoor localisation are at the forefront of this evolution, which have emerged as pivotal research domains with applications in smart healthcare, intelligent building control, the IoT, behaviour analysis, gesture recognition, and smart surveillance systems [153, 213, 214]. In particular, HAR plays a crucial role in elderly care for fall detection, enabling continuous monitoring and timely interventions to enhance patient safety and independence [144]. Additionally, indoor HAR optimises space utilisation and energy efficiency in smart buildings by providing contextual awareness of user location and occupancy.

Indoor HAR is fundamentally a classification problem that aims to identify human actions and movements based on observations captured through various sensors [144, 153]. Traditional HAR systems have employed vision-based methods utilising high-resolution cameras and computer vision techniques. While effective, these approaches raise significant privacy concerns due to the intrusive nature of video capture and are sensitive to environmental factors such as lighting conditions and background variability [146]. Wearable sensors, including accelerometers and gyroscopes, offer an alternative by directly capturing movement data. However, as explored in HNFL, wearable sensors, particularly in outdoor settings, require continuous physical interaction, which can pose challenges in scenarios like elderly care due to inconvenience and potential discomfort [144, 215].

In contrast, this chapter focuses on non-invasive HAR for indoor environments using RF sensing, which has gained popularity due to its unobtrusiveness and high privacy preservation. RF-based systems, particularly those leveraging CSI, provide sophisticated methods for monitoring indoor activities by capturing comprehensive wireless signal characteristics at the subcarrier level [145, 153]. The ubiquitous presence of Wi-Fi signals in indoor environments makes CSI-based HAR an economical and practical solution. Human movements induce unique scattering patterns in wireless signals, creating distinct CSI signatures that can be mapped to classify corresponding activities [143].

Despite the advantages, CSI-based HAR systems face challenges related to the unpredictability of wireless signals. Multipath propagation, environmental dynamics, and signal fluctuations introduce noise and inconsistencies in the captured data, affecting the accuracy and reliability of HAR models [143]. Moreover, traditional HAR systems rely on centralised data collection and processing, necessitating the sharing of raw data with a central server. This approach raises significant privacy concerns, especially under stringent regulations like the General Data Protection Regulation (GDPR) [146, 215]. Centralised processing also leads to inefficient bandwidth usage, increased network and storage costs, energy inefficiency, and latency issues that limit real-time performance and scalability.

The previous **Chapter 5** explored architectural innovations in FL to enhance energy efficiency and processing capabilities on edge devices. Specifically, HNFL demonstrated how integrating SNNs with LSTMs could improve HAR accuracy and energy efficiency for wearable sensors in outdoor settings. However, applying FL in non-invasive, indoor HAR using CSI data introduces new challenges, particularly regarding communication overhead and energy efficiency. In this chapter, we introduce a FL framework designed to enhance the efficiency, accuracy, and scalability of HAR systems using CSI data in indoor environments. Unlike HNFL, which focuses on architectural changes to improve on-device processing, FFQ addresses computational and communication efficiency by integrating signal processing, feature fusion, and model compression techniques.

Feature fusion combines statistical and differential features with processed CSI data, provid-

ing the model with the contextual understanding necessary for navigating varied environmental conditions [145,153]. Distinguishing itself from traditional FL approaches, FFQ adopts the FedDist algorithm [216], tailored to mitigate overfitting by adjusting the model parameters based on a dissimilarity measure of diverging neurons, a pivotal enhancement for handling heterogeneous data effectively.

To reduce the communication overhead in the federated training process, our framework introduces a model compression strategy that quantises the model parameters from a 32-bit float point to 16 and 8-bit integers precision. Additionally, we adopted the quantisation-aware training (QAT), which involves adjusting the bit precision of model computations and gradients from the conventional 32-bit float point to a more communication-efficient 16 and 8-bit format during local model training. This quantisation not only minimises the size of the model updates transmitted in each communication round but also ensures that local model training adapts to the reduced precision, maintaining accuracy while reducing bandwidth requirements. Furthermore, FFQ optimises the synthesis of model updates, ensuring that collective intelligence accurately reflects the diversity of human activities across disparate settings, thereby enhancing the overall accuracy and reliability of the HAR system. In summary, the main contributions of this work are highlighted as follows:

- This work introduces the FFQ framework, merging FL with edge-based preprocessing and feature fusion to enrich HAR models with a deep contextual understanding of CSI data. The proposed framework aims to extract and fuse the statistical and differential features to achieve a low false positive rate, particularly for fall detection.
- The study adopted the FedDist algorithm for global model aggregation tailored to address the challenge of overfitting in the context of highly heterogeneous data. This study also modified the divergence metric by adjusting model parameters based on a dissimilarity measure of diverging neurons, resulting in the effective handling of diverse data. The results of the modified FedDist are compared with FedAVG algorithm.
- The proposed FFQ framework incorporates QAT into its model compression strategy, significantly improving the efficiency of FL architectures by training local models directly in a reduced 16 and 8-bit precision format. This approach minimises the size of model updates, significantly reducing communication overhead while maintaining comparable accuracy with the state-of-the-art techniques.

6.2 Preliminaries and System Model

This section presents the generic FL training process for a typical classification problem for HAR. The conventional HAR model collects the raw data samples and transmits them to a centralised location for processing and model training. In contrast, the FL approach trains the

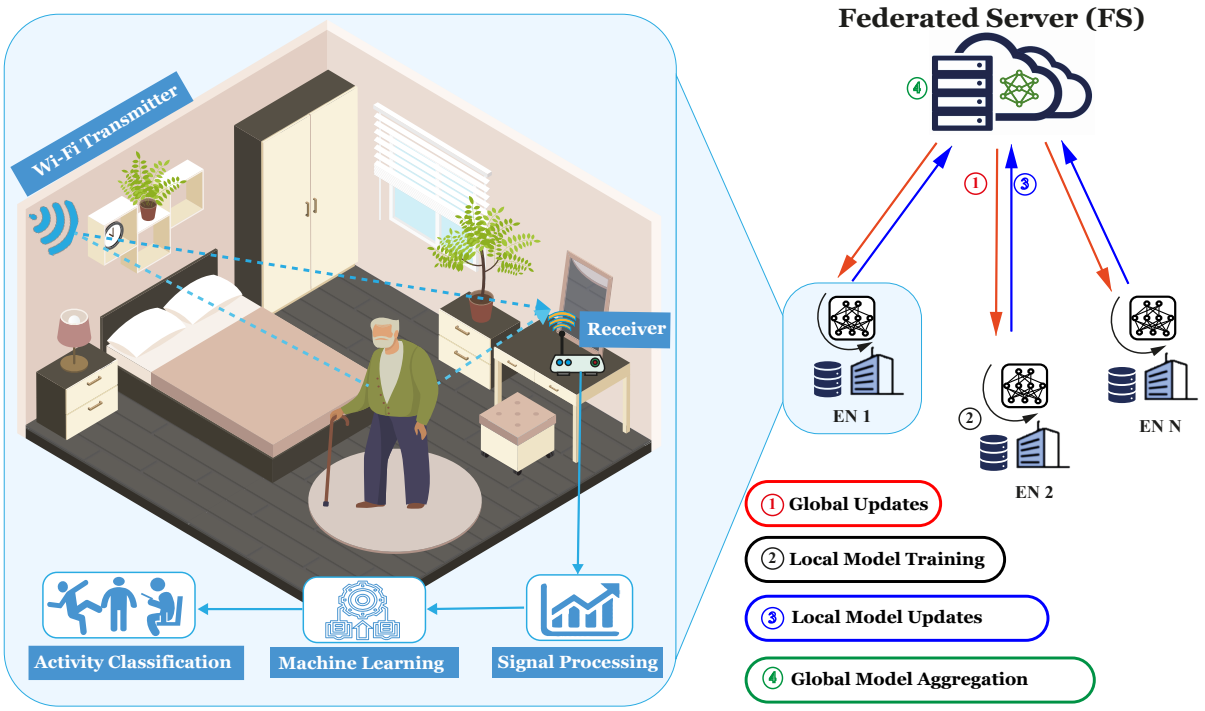


Figure 6.1: The conceptual framework for FL-based HAR using CSI.

model across many clients where the dataset is highly decentralised. This work considered a cross-silo FL architecture where the data is divided among a few trusted end clients, as shown in Fig. 6.1. Consider a FL system with N ENs representing clients, where each client i holds a local dataset $\mathscr{D}_i$. The global learning objective is to minimise the loss function $\ell(\omega)$ over the global model parameters ω, formulated as in eq. 2.1. The local loss function given in eq. 2.2, is modified as the average loss over the data points in $\mathscr{D}_i$, expressed as:

$$\ell_i(\omega) = \frac{1}{|\mathscr{D}_i|} \sum_{j \in \mathscr{D}_i} \ell(x_j, y_j; \omega), \tag{6.1}$$

here, $\ell(x_j, y_j; \omega)$ denotes the loss of the model parameterised by ω on a data point (x_j, y_j) from client i's dataset, with x_j being the feature vector and y_j the associated class label.

The FedAVG algorithm aggregates the model parameters on the FS during the training process across T communication rounds, each comprising E local epochs of training on each client, with a local batch size of B. The FedAVG process is mathematically represented through the following iterative process:

- **Initialisation**: The FS initialises the global model parameters ω^0 and distributes them to all EN. This process is represented by step ① as shown in Fig. 6.1.

- **Local Training**: For each communication round $t = \{1, 2, \ldots, T\}$, and each client i, the client updates the model parameters ω by performing E epochs of gradient descent on its

local dataset $\mathscr{D}_i$, using a batch size of B. This can be mathematically depicted as:

$$\omega_i^{t+1} = \omega_i^t - \eta \nabla \ell_i(\omega_i^t), \tag{6.2}$$

where η is the learning rate and $\nabla \ell_i(\omega_i^t)$ is the gradient of the loss function ℓ_i with respect to the model parameters ω at round t. The process of local learning is represented by step ②, as shown in Fig. 6.1.

- **Model Aggregation**: After local training, each client sends its updated model parameters ω_i^{t+1} back to the FS given as step ③. The server then aggregates these parameters to update the global model step ④ as shown in Fig 6.1. The aggregation is performed as a weighted average, considering the size of each client's dataset:

$$\omega^{t+1} = \frac{1}{\sum_{i=1}^{N} |\mathscr{D}_i|} \sum_{i=1}^{N} |\mathscr{D}_i| \omega_i^{t+1} \tag{6.3}$$

- **Global Model Update**: The updated global model parameters ω^{t+1} are then shared with all clients, marking the beginning of the next training round.

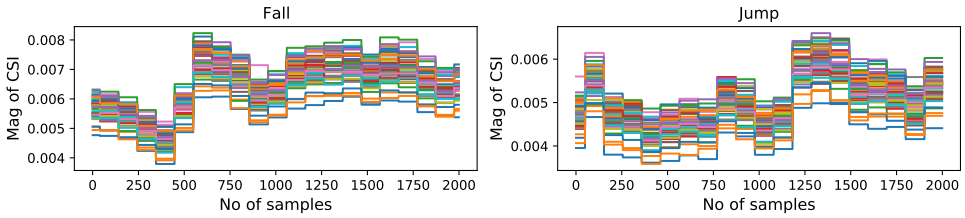
This iterative process of local training and global aggregation continues for T rounds, aiming to minimise the global loss function $\ell(\omega)$ and improve the model's performance on the multi-class classification task.

6.3 Proposed FFQ for HAR

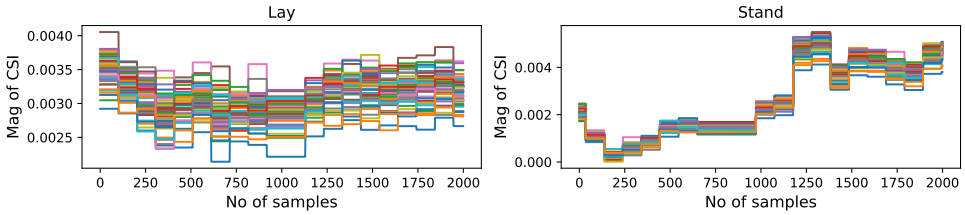
The section outlines our proposed FFQ framework for HAR, building on the foundational understanding of FL discussed earlier in section 6.2. The proposed framework includes extensive data pre-processing, feature engineering, feature-level fusion, and model compression, enhancing communication efficiency while maintaining high classification accuracy. We will discuss the methodologies employed in data pre-processing and feature engineering and the novel temporal feature extraction and feature fusion process. Additionally, we will elaborate on the local training, model compression strategy with QAT, and our custom model aggregation mechanism, which collectively contributes to the robustness and efficacy of the FFQ framework.

6.3.1 Signal Propagation and CSI Acquisition

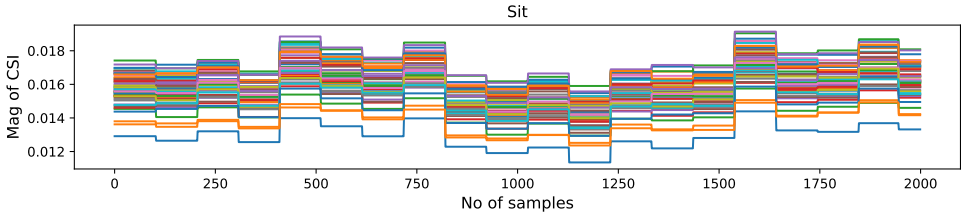
The first step in FL-based HAR is collecting fine-grain CSI data under the influence of daily routine activities. Wi-Fi signals undergo various effects due to their broadcast nature, such as reflection, refraction, scattering, and diffraction [217]. For example, static objects like floors, ceilings, walls, tables, and chairs reflect signals, while micro activities cause dynamic effects like



(a) Received CSI for fall and jump activities.



(b) Received CSI for laying and standing activities.



(c) Received CSI for sitting activity.

Figure 6.2: 2D plots of received CSI under the influence of different activities.

refraction, diffraction, and scattering. These dynamic activities weaken the signal and distort both the phase and amplitude of the received signal.

In this study, we implemented the IEEE 802.11a standard using software-defined radios (SDR) and used two universal software radio peripherals (USRP) as transmitters and receivers. The preamble (known data sequences) in the message packet is used for channel estimation, and the physical layer CSI is acquired under the influence of different activities. The details of implementing the IEEE 802.11a standard and data collection process are beyond the scope of this work. However, the implementation and process of raw data collection in a lab environment are discussed in our previous studies [217, 218]. The plots of the 2000 CSI samples under the influence of different activities are shown in Fig. 6.2. It can be observed that different activities have a significant variation in received CSI, which can be translated into a unique signature for different activities after careful signal processing.

6.3.2 Data Pre-processing and Feature Engineering

The collected raw CSI signal contains information on both phase and amplitude. The received signal phase is usually affected by carrier frequency offset (CFO) caused by a mismatch of

transmitter-receiver clocks. A small CFO causes significant variation in the phase of the received signal, which makes the phase changes caused by the body negligible [153]. Therefore, this work considered the amplitude information for indoor human activity classification. Although the amplitude of the received signal is a stable metric, the noise added by the external environment is still a big problem. The noisy CSI may not give distinct features; therefore, data denoising is one of the vital steps in HAR.

Signal Denoising using EMD

In this work, we used EMD for data denoising, given the non-linear and non-stationary nature of time series CSI data [153, 163]. The EMD decomposes the raw symbols, which consists of m samples for each client i, denoted as $X(t) = \{x_1, x_2, \ldots, x_m\}$. The EMD process enables the decomposition of $X(t)$ into a finite set of intrinsic mode functions (IMFs) and a residue, significantly enhancing the signal quality for further processing. For client i, the denoising process through EMD can be mathematically represented as:

$$X_i(t) = \sum_{k=1}^{n_i} IMF_{ik}(t) + r_i(t), \tag{6.4}$$

where $IMF_{ik}(t)$ are the intrinsic mode functions extracted from $X_i(t)$, n_i denotes the total number of IMFs for client i, and $r_i(t)$ represents the residue.

The signal post-denoising, denoted as $\hat{X}_i(t)$, is reconstructed by excluding the first few $IMF_{ik}(t)$, identified primarily as noise components:

$$\hat{X}_i(t) = \sum_{k=p+1}^{n_i} IMF_{ik}(t) + r_i(t), \tag{6.5}$$

where p is the last IMF considered to be noise. This modification ensures that the reconstructed signal $\hat{X}_i(t)$ is free of primary noise artefacts.

Feature Engineering

Feature engineering plays a crucial role in the subsequent analysis following the denoising of the time series data. We employ DWT for feature extraction, given its efficacy in capturing both frequency and time domain aspects of human activities, which vary across different activities. The denoised signal, denoted as $\hat{X}_i(t)$, undergoes decomposition into high and low-frequency components using DWT, facilitating the separation of the signal into different time scales. This process involves passing $\hat{X}_i(t)$ through a pair of low and high pass filters, followed by down-sampling to adhere to the Nyquist criteria. The extraction of meaningful features is achieved by applying a thresholding technique to the wavelet coefficients obtained from DWT, which is mathematically represented as [219]:

$$\tilde{\Phi}_{j,i} = \begin{cases} (\Phi_{j,i})\left(|\Phi_{j,i}| - \tau_j\right), & |\Phi_{j,i}| \geq \tau_j \\ 0, & |\Phi_{j,i}| < \tau_j \end{cases} \tag{6.6}$$

where $\Phi_{j,i}$ and $\tilde{\Phi}_{j,i}$ denote the original and denoised wavelet coefficients, respectively, at the j^{th} level of decomposition and i^{th} location. The threshold τ_j is determined based on the universal threshold τ_U, detailed as follows [220]:

$$\tau_U = \sigma\sqrt{2\log(L)}, \tag{6.7}$$

with L representing the length of the pre-processed CSI signal, and σ denoting the standard deviation of noise, estimated by:

$$\sigma = \frac{\text{median}(|Z_1|)}{0.6745}, \tag{6.8}$$

where Z_1 is the first level detail coefficient.

After completing the feature extraction process, the next step is to perform feature selection to reduce redundancy in the training data and avoid overfitting [221]. While there are many ways to select features, we chose LDA because the CSI data shows a normal distribution with features that are minimally correlated [221]. LDA aims to create a new subspace of the data by maximising the distance between classes (inter-class) and minimising the distance within the same class (intra-class), as defined by the equations below:

The between-class scatter matrix C_b is given by:

$$C_b = \sum_{j=1}^{k} d_j (m_j - c)(m_j - c)^T, \tag{6.9}$$

where d_j denotes the number of samples in the j^{th} class, m_j the mean vector of the j^{th} class, and c the global centroid. The within-class scatter matrix C_w is defined as:

$$C_w = \sum_{j=1}^{k} \sum_{x \in D_j} (x - m_j)(x - m_j)^T, \tag{6.10}$$

where D_j represents the set of all samples in the j^{th} class.

Furthermore, the total scatter matrix C_t is represented by:

$$C_t = \sum_{j=1}^{n} (x_j - c)(x_j - c)^T, \tag{6.11}$$

where x_j are the data points, and c is the global centroid. The lower dimensions of the input training data are achieved through the application of eigenvalue decomposition on the scatter matrices, which effectively eliminates noisy and redundant components.

6.3.3 Temporal Feature Extraction and Feature Fusion

This study extracts temporal features from the denoised signal $\hat{X}_i(t)$ to expand the feature space beyond the conventional LDA feature set. The goal is to capture the dynamic nature of human activities with statistical and first-order differential features from each signal window. A comprehensive set of statistical features, including mean, variance, standard deviation, skewness, and kurtosis, is computed from each window W in the denoised signal $\hat{X}_i(t)$ to capture the signal's intrinsic properties over time.

First-order differential features are calculated to encapsulate the temporal evolution of the signal, highlighting changes between consecutive samples. For a signal window W, the differential feature, $Diff_i^W$, is defined as:

$$Diff_i^W = \{x_{i+1} - x_i | x_i, x_{i+1} \in W\} \tag{6.12}$$

After extracting the statistical features (S_i^W) and differential features ($Diff_i^W$), a fusion process is performed to combine these temporal features with the raw CSI features selected through LDA (G_i'). This approach ensures a more complete characterisation of each activity, considering both the static and dynamic aspects of the signal. The fusion of LDA-selected features and temporal features yields the final feature vector H_i for each signal window W, formulated as:

$$H_i = G_i' \oplus S_i^W \oplus Diff_i^W \tag{6.13}$$

In this formulation, $\oplus$ symbolises the concatenation operation, blending the diverse feature sets into a unified representation H_i also named as combined temporal features (CTF), which is the input for the subsequent classification model. This fused feature vector H_i is poised to offer a robust foundation for accurate activity recognition, embodying the multifaceted nature of human motion patterns.

6.3.4 Local Training and Quantisation

A pivotal stage in our proposed FFQ framework is local training, wherein individual clients fine-tune the global model using their local data $\mathscr{D}_i$ as discussed in Section 6.2. This stage is crucial for adapting the global model to diverse local conditions and data characteristics. Following local training, the quantisation process is employed to optimise communication efficiency by compressing the updated model parameters before they are transmitted back to the FS.

Quantisation Process

The quantisation strategy aims to reduce the model parameter size, facilitating efficient communication across the FL network. This reduction is achieved by lowering the precision of the model parameters to a specified bit-length b without significantly compromising the model's

predictive performance. The simplified quantisation operation is represented mathematically as:

$$\hat{\omega}_i^{t+1} = Q(\omega_i^{t+1}, b), \tag{6.14}$$

where $Q(\cdot, b)$ denotes the quantisation function that compresses the parameters ω_i^{t+1}, with b indicating the target bit precision. This process balances the trade-off between model accuracy and communication cost.

Pre-Quantisation and Post-Quantisation

The quantisation process can be performed into two stages: pre-quantisation and post-quantisation. In **Pre-Quantisation**, the model parameters are compressed before local training. This phase involves standard training operations where the parameters ω_i^t are adjusted based on the local data characteristics. The goal during this phase is to adapt the global model to local nuances and improve its performance across diverse data distributions. **Post-Quantisation**: After the local training, the updated model parameters ω_i^{t+1} are quantised to reduce their size before model sharing. This step compresses the parameter values, resulting in $\hat{\omega}_i^{t+1}$, which are then sent to the FS. While this significantly reduces communication overhead, it introduces potential precision loss, which can degrade the model's performance.

Incorporating Quantisation-Aware Training

QAT is integrated into the local training phase to enhance the model's robustness to quantisation effects. QAT anticipates the impact of quantisation by incorporating a simulation of the quantisation process within the training algorithm. This approach allows the model to adapt to the reduced precision and maintain high accuracy compared to post-quantisation. The mathematical representation of QAT is as follows:

Model Quantisation during Training: The model parameters undergo a simulation of quantisation and subsequent de-quantisation within each training iteration, represented by:

$$\omega_{i,sim}^t = DQ(Q(\omega_i^t, b)), \tag{6.15}$$

where $DQ(\cdot)$ represents the de-quantisation function, reversing the quantisation effect for training purposes.

Parameter Update with Model Quantisation: This simulated environment enables the gradient descent optimisation to account for quantisation effects, mathematically expressed as:

$$\hat{\omega}_i^{t+1} = \omega_i^t - \eta \nabla \ell_i(\omega_{i,sim}^t), \tag{6.16}$$

where η is the learning rate, and $\nabla \ell_i$ denotes the gradient of the loss function with respect to the simulated quantised parameters. This equation is a modified version of the local update

with QAT given in Eq. 6.2. Incorporating QAT into local training effectively counters potential accuracy declines due to reduced parameter precision. It is worth mentioning that the post-quantisation effectively reduces the communication burden, but it has inherent limitations. For instance, reducing the bit-length of model parameters can lead to a loss of precision, negatively impacting the model's predictive performance. Additionally, the post-quantisation does not account for the reduced precision during the local training phase, meaning the model parameters are optimised without considering the eventual quantisation, which can result in suboptimal performance after quantisation.

6.3.5 Global Aggregation using FedDist

The FedDist algorithm improves the conventional aggregation process in FL by introducing a divergence analysis between quantised local and global model parameters. This selective aggregation strategy optimises model performance across highly diverse client data. Given quantised weights from the clients $\hat{\omega}_i^{t+1}$ and the global model $\hat{\omega}^t$, the Euclidean distance for divergence assessment is calculated as follows:

$$dist(\hat{N}_1, \hat{N}_2) = \sqrt{\sum_{i=1}^{C} (\hat{N}_1^{w_i} - \hat{N}_2^{w_i})^2}, \tag{6.17}$$

where $\hat{N}_1$ and $\hat{N}_2$ represent neurons in the local and global models, respectively, and C is the number of weights per layer. This is a layer-wise update process where the architecture of the global model is modified based on the diverging neuron. If the distance between the neurons of the local and global model is greater than the given threshold, the local node is considered as diverging; thus, layer-wise training is started. The layer having the diverging neuron on the client side is frozen, and weights for an unfrozen layer are sent to minimise the communication overhead. Once there is no diverging neuron, the system performs the simple weighted average as the FedAVG algorithm. This technique enables the identification of the client that constantly acts as an outlier and has a negative impact on the performance of the global model. This QAT FedDist algorithm is given in Algorithm 6.

6.4 Simulation Setup

This section explains the simulation setup created to thoroughly test the effectiveness of the proposed FFQ framework for HAR using CSI. Our goal is to validate the accuracy, robustness, and efficiency of the FFQ model under various conditions using experiments that mimic real-world scenarios in smart healthcare applications. Additionally, we describe a dataset consisting of real-world activities captured through advanced sensing technologies, followed by a discussion of the performance metrics that measure the performance of the proposed framework. Furthermore,

Algorithm 6: QAT-Enhanced Federated Distance Algorithm

Result: Final global model $\hat{\omega}^T$ after enhancement
Input: Total communication rounds T, number of clients N, number of layers L;
Initialise global quantised model parameters $\hat{\omega}^0$;
for $t = 0$ *to* $T-1$ **do**
  Client Updates:
  for *each client* $i = 1$ *to* N ***in parallel*** **do**
    Send global model parameters $\hat{\omega}^t$ to client i;
    $\hat{\omega}_i^{t+1} \leftarrow \text{ClientUpdate}(i, \hat{\omega}^t)$;
  end
  Global Aggregation:
  $\hat{\omega}^{t+1} \leftarrow \sum_{i=1}^{N} \frac{|\mathscr{D}_i|}{\sum_{j=1}^{N} |\mathscr{D}_j|} \hat{\omega}_i^{t+1}$;
  $\hat{\omega}^{t+1} \leftarrow \sum_{i=1}^{N} \frac{n_i}{N} \hat{\omega}_i^{t+1}$;
  Euclidean Distance:
  for $l = 1$ *to* $L-1$ **do**
    for $i = 1$ *to* N **do**
      Compute pair distance $\text{dist}(\hat{N}_1, \hat{N}_2)$;
    end
    Compute μ_l^t and σ_l^t for distances in layer l;
    Set newNeuron = False;
    for $c = 1$ *to* C **do**
      dist_threshold = $3 \cdot \mu_{dl}^t + \sigma_{dl}^t$;
      if $mean(c) > dist\_threshold$ **then**
        Append new neuron $\hat{\omega}_t^l$;
        Set newNeuron = True;
      end
    end
    if *newNeuron* **then**
      for $i = 1$ *to* N **do**
        $\hat{\omega}_i^l = \hat{\omega}^l$ `// Freeze layer` l `and below`
        Update layers $\hat{\omega}_i^{l+1}$ to $\hat{\omega}_i^L$ at client i;
      end
    end
  end
  $\hat{\omega}^{t+1} \leftarrow \hat{\omega}^t$ `// Global updates`
end
Procedure $\text{ClientUpdate}(i, \hat{\omega})$;
  Initialise local model parameters $\hat{\omega}_i \leftarrow \hat{\omega}$;
  for *epoch* $= 1$ *to* E **do**
    for *each batch b of size B from client i's data* **do**
      $\hat{\omega}_i \leftarrow \hat{\omega}_i - \eta \nabla F_i(\omega_{i,sim}^t; b)$;
    end
  end
  return $\hat{\omega}_i$;

we also outline our performance evaluation strategy, which includes accuracy, communication overhead and comparative analysis with state-of-the-art models.

6.4.1 Dataset Description and Data Partitioning

The raw dataset is collected in an indoor lab environment where an IEEE 802.11a standard is implemented using SDR. Two USRPs are used as transmitters and receivers and are kept at approximately 6m distance. Thirty volunteers from different age groups have participated in the data collection process. The implementation and data collection process details are beyond this paper's scope; however, the details can be seen in [217]. The dataset includes 80,000 samples across six typical daily activities: sitting, standing, running, falling, lying down, and jumping. Additionally, the line-of-sight component of CSI is also captured where no activity is performed. These activities were selected to encompass a range of human motions relevant to smart healthcare monitoring systems.

The raw data from this study comprises complex CSI for 52 subcarriers, capturing both amplitude and phase components. However, for the purpose of this analysis, only the amplitude information was extracted and utilised to develop features engineering for the HAR tasks as discussed in section 6.3. This selective data usage helps focus on the most relevant information for activity recognition, reduce computational complexity, and enhance model performance.

To simulate a realistic FL environment, the dataset was equitably distributed among five clients ENs. Each client was allocated 16,000 samples. A standard split of 80% training and 20% testing was employed to prepare the data for the federated training process. Hence, the training set of each EN consists of 12800 samples. Notably, the test sets from each client were combined to form a comprehensive global test set, where the total number of the test samples in the global test set is 16000. This global test set plays a crucial role in assessing the overall performance of the proposed FFQ model, ensuring that the evaluation reflects a wide array of scenarios and is not biased by any single client's data profile. Additionally, the data distribution among the participants was kept highly unbalanced to ensure the actual case for the FL scenario. Despite being a HAR task, the dataset used in this study is highly diverse and involves several participants, device limitations and high-class imbalance, which could be a good use case for FL applications.

6.4.2 Model Configurations and Hyperparameters

In this study, we employed a straightforward yet robust Deep Neural Network (DNN) for local learning. This model configuration is specifically chosen to balance the computational load and maintain high accuracy across various EN implementations. The local model consists of an input layer that accepts an N-dimensional vector with N as the number of features per sample. This architecture includes three fully connected layers with 128, 64, and 32 units, respectively, each followed by a rectified linear unit (ReLU) activation function. Finally, there is a dropout layer with a dropout rate of 0.3 and an output layer with a softmax activation function. Additionally, regularisation factor of 0.01 was applied with learning rate of t 10^{-5}.

Local Model Configuration

- **Input Layer:** The input layer accepts an N-dimensional vector with N as the number of features per sample.
- **Dense Layers:** The architecture includes three fully connected layers with 128, 64, and 32 units, respectively, each followed by a rectified linear unit (ReLU) activation function.
- **Dropout Layer:** A dropout layer with a dropout rate of 0.3 is implemented to help prevent overfitting during training.
- **Output Layer:** The final layer uses a softmax activation function to classify the input into various activity categories. It is optimised during training using a sparse categorical cross-entropy loss function.
- **Optimiser:** ADAM, known for its adaptive learning rate features, was chosen to facilitate efficient convergence.
- **Kernel Regulariser:** A regularisation factor 0.01 was applied to minimise overfitting by penalising large weights.
- **Learning Rate:** Set at 10^{-5} to provide a slow and steady update during backpropagation, minimising the risk of overshooting minima.

6.4.3 Performance Metrics

To evaluate the proposed FFQ framework, we use average accuracy as the baseline metric. However, accuracy alone can be misleading with imbalanced datasets, leading to the accuracy paradox, where a model appears effective but is biased towards the majority class. We consider Precision, Recall, and F1-score as discussed in Section. 2.2.2. Precision measures the correctness of positive predictions, which is crucial for high-stakes activities like fall detection. Recall assesses the model's ability to identify all relevant instances, which is vital in scenarios where missing a positive instance is critical. The F1-score provides a harmonic mean of precision and recall, balancing both metrics and being especially useful for imbalanced classes. To evaluate the FFQ framework, we conducted experiments in a controlled simulation environment using a global test set, assessing the model's performance across all participants. We also included an evaluation of energy efficiency, an important metric in FL environments where resource constraints are significant. We measured energy efficiency in terms of communication overhead by simplifying this equation: [222]:

$$E_{est} = R\left[\alpha \cdot t_{com} + N \cdot (\beta \cdot P_{trn})\right], \tag{6.18}$$

where α and β represent the energy cost coefficients for computation (per second) and communication (per kilobyte), respectively; R is the number of communication rounds; N denotes the number of participating clients; t_{com} is the computation time depending on the device; and P_{trn} is the payload size per communication round. In eq. (6.18), we are only interested in the communication overhead per round, hence, the simplified equation is as $E_r = N(\beta \cdot P_{trn})$.

6.5 Results and Discussion

This section provides a comprehensive evaluation of the proposed FFQ framework, specifically tailored for HAR. The dataset, distributed across five clients, each containing 16,000 samples, forms the basis of our experimental analysis. Specifically, each client utilises 12,800 samples for training purposes and the remaining 3,200 for testing. The test samples from each client are combined into a global test set comprising 16,000 samples in total, which is used to evaluate the performance of the global model under different experimental setups. For robust analysis, two distinct case studies are conducted, which include:

1. **Combined Temporal Fusion (CTF):** This setup explores the performance enhancement achieved by integrating statistical and differential features with basic raw data features. The evaluation is conducted across two scenarios: multi-class classification of daily activities and binary-class fall detection.

2. **CTF with QAT:** In addition to the basic CTF setup, this case study incorporates QAT to investigate the impact on communication overhead and examine the trade-off between accuracy and energy efficiency in contexts of both multi-class and binary-class problems.

These case studies assess the effectiveness of the FFQ framework in enhancing classification accuracy, minimising communication overhead, and ensuring data privacy and computational efficiency. Subsequent sections will provide detailed quantitative results from these studies, followed by a discussion of the implications of these findings. Additionally, we will identify potential areas for improvement and discuss how these findings can be applied to improve federated learning applications in smart healthcare.

6.5.1 Case:1 CTF for Multi-class Classification

The CTF provides additional contextual information, which aims to improve classification accuracy significantly. Here, we systematically evaluate the performance by comparing the results of the FedDist algorithm with FedAVG, with and without the incorporation of feature fusion. Analyses are conducted for multi-class classification scenarios encompassing seven daily activities and binary-class fall detection, which is critical for prompt and effective healthcare response.

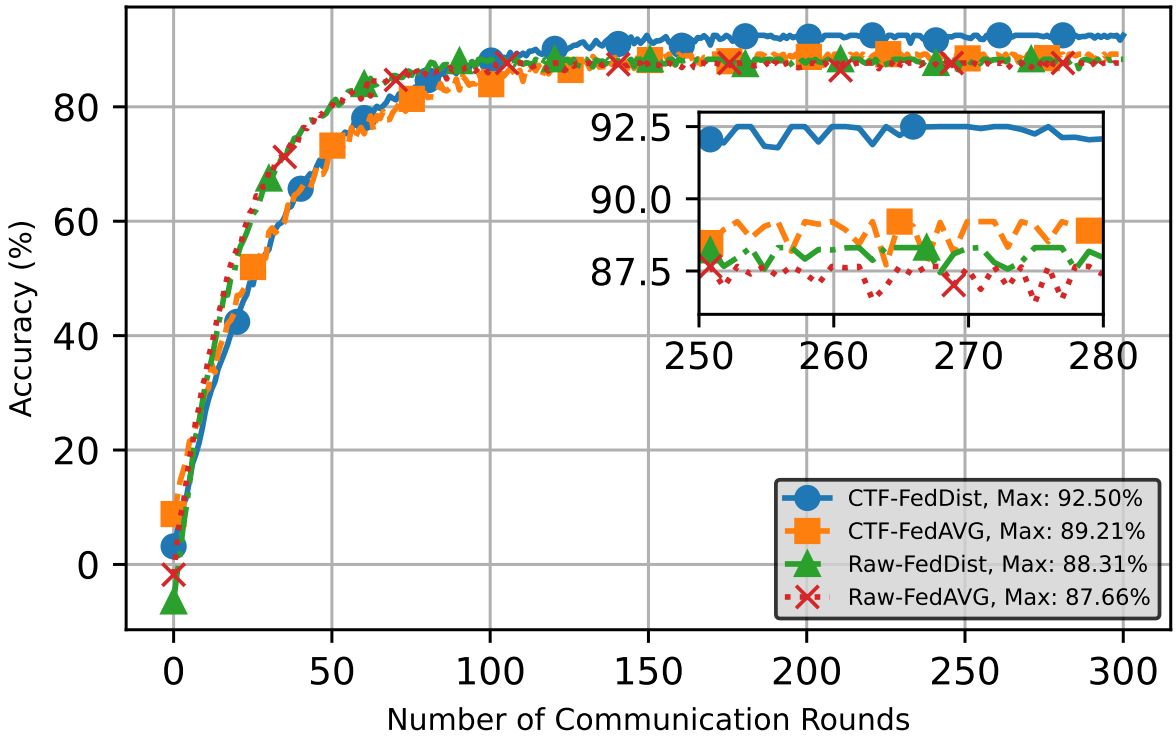
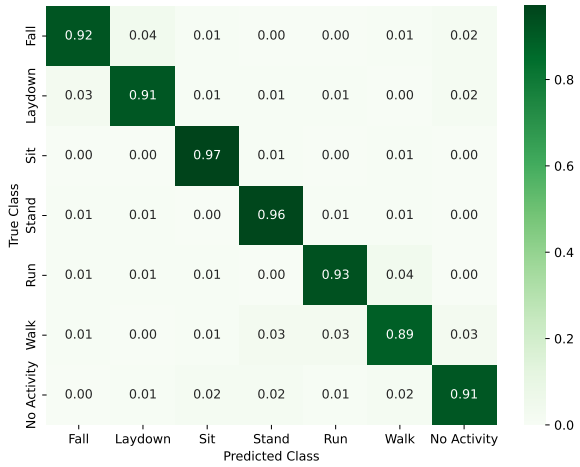


Figure 6.3: The comparative accuracy learning curve for CTF and simple feature for FedDist and FedAVG algorithm.

The results in Fig. 6.3 present the learning curves from the global model evaluated using the test set across 300 communication rounds, comparing the accuracy of FedDist and FedAVG algorithms under two feature configurations: CTF and simple amplitude (Amp) obtained using DWT and LDA. The CTF-FedDist setup achieved the highest accuracy, peaking at 92.50%, demonstrating the effectiveness of combining temporal features with the FedDist algorithm. In comparison, CTF-FedAVG, Amp-FedDist, and Amp-FedAVG reached maximum accuracies of 89.21%, 88.31%, and 87.66%, respectively. These results highlight the importance of contextual information and strategic model aggregation to improve the accuracy of multi-class HAR.

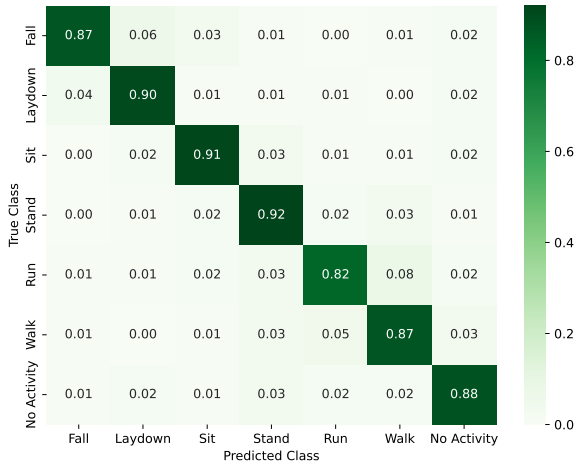
The results in Fig. 6.4 show a set of normalised confusion matrices representing the classification outcomes for FedDist and FedAVG. Each confusion matrix displays the predictive performance across six activity classes and a No Activity category. The CTF-FedDist matrix shown in Fig. 6.4 (a) indicates high diagonal values, which imply strong true positive rates for most activities, particularly for Fall, Sit, and No Activity. These results show that the CTF features combined with the FedDist algorithm lead to highly accurate classification. In comparison, the CTF-FedAVG matrix given by Fig. 6.4 (b) also demonstrates high accuracy but slightly lower true positive rates for Run and Walk, indicating a minor increase in confusion between these two activities.

Conversely, the Amp-FedDist Fig. 6.4 (c) and Amp-FedAVG Fig. 6.4 (d) matrices exhibit lower diagonal values across activities, with a notable decrease in correctly classified instances



(a) Confusion Matrix for CTF-FedDist.

(b) Confusion Matrix for CTF-FedAVG.



(c) Confusion Matrix for Raw-FedDist.

(d) Confusion Matrix for Amp-FedAVG.

Figure 6.4: Confusion matrices for CTF and raw amplitude-only feature for FedDist and FedAVG algorithms: (a) Confusion Matrix for CTF-FedDist, (b) Confusion Matrix for CTF-FedAVG, (c) Confusion Matrix for Raw-FedDist, (d) Confusion Matrix for Amp-FedAVG.

for Fall, a critical class for safety in healthcare scenarios. Additionally, across all matrices, the off-diagonal elements represent misclassifications, with the non-zero values indicating the proportion of each actual class being incorrectly predicted as each of the other classes. Additionally, the results also demonstrate that the Fall and Laydown are frequently confused due to the similar motion patterns captured using CSI. Similarly, activities like Walk and Run exhibit a degree of confusion, with off-diagonal values of 0.03 and 0.04, respectively. This misclassification is expected, as walking and running share similarities in their dynamic nature and can be challenging to distinguish based on certain features.

In Table 6.1, there is a comparison of performance metrics that shows the benefits of using feature fusion instead of simple amplitude-based (Amp) features in the FedDist algorithm. The CTF-FedDist configuration has a higher precision, recall, and F1 Score when compared to Amp-

Table 6.1: Performance comparison for CTF-FedDist and Raw-FedDist

	CTF-FedDist			Amp-FedDist		
Class	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Fall	0.939	0.920	0.929	0.926	0.870	0.897
Laydown	0.929	0.919	0.924	0.882	0.909	0.896
Sit	0.942	0.980	0.960	0.901	0.910	0.905
Stand	0.932	0.960	0.946	0.868	0.911	0.889
Run	0.939	0.930	0.935	0.882	0.828	0.854
Walk	0.908	0.890	0.899	0.853	0.870	0.861
NA	0.929	0.919	0.924	0.880	0.889	0.884

FedDist, especially for critical activity classes such as Fall, Run, and Stand. It is important to note that in the Fall classification, the precision and F1 Score of CTF-FedDist are significantly higher than Amp-FedDist. This highlights the significance of temporal feature fusion in accurately distinguishing complex activities. These results indicate that using CTF features in FL models can significantly improve HAR in practical healthcare systems.

6.5.2 Case:2 CTF for Fall Detection (Binary Classification)

In this scenario, the idea was to train a binary classifier to identify Falls from other daily routine activities. Similar to multi-class classification, the model is trained for 300 communication rounds. The global test set is used to evaluate the performance of the model, and results for the accuracy learning curve for binary classifier with and with temporal feature fusion are given in Fig. 6.5. The learning curve for fall detection for a binary classifier exhibits classical behaviour, with a sharp increase in initial results and a more steady state of behaviour after 100 communication rounds. The results show that the CTF feature set improves the model performance. Notably, the CTF-FedDist configuration achieves the highest accuracy, peaking at 97.21%, illustrating the benefits of integrating temporal context in the detection algorithm. Moreover, this performance is closely followed by the CTF-FedAVG, with a maximum accuracy of 95.67%. On the other hand, configurations that used simple amplitude-based features such as Amp-FedDist and Amp-FedAVG exhibited reduced peak accuracies of 93.88% and 91.39%, respectively.

The results in Fig. 6.6 show the confusion matrices for binary classification in Fall detection using CTF and amplitude-only features for FedDist and FedAVG algorithms. The matrices reveal that CTF features combined with FedDist, as shown in Fig 6.6 (a), yield the highest true positive rate for Falls at 0.98% and a true negative rate of 0.96%, outperforming the other configurations. In comparison, the amplitude-only feature sets given in Fig. 6.6 (c), (d) exhibit reduced performance, particularly in distinguishing non-fall activities, underscoring the significance of temporal features in achieving higher classification accuracy for critical fall detection tasks in FL environments.

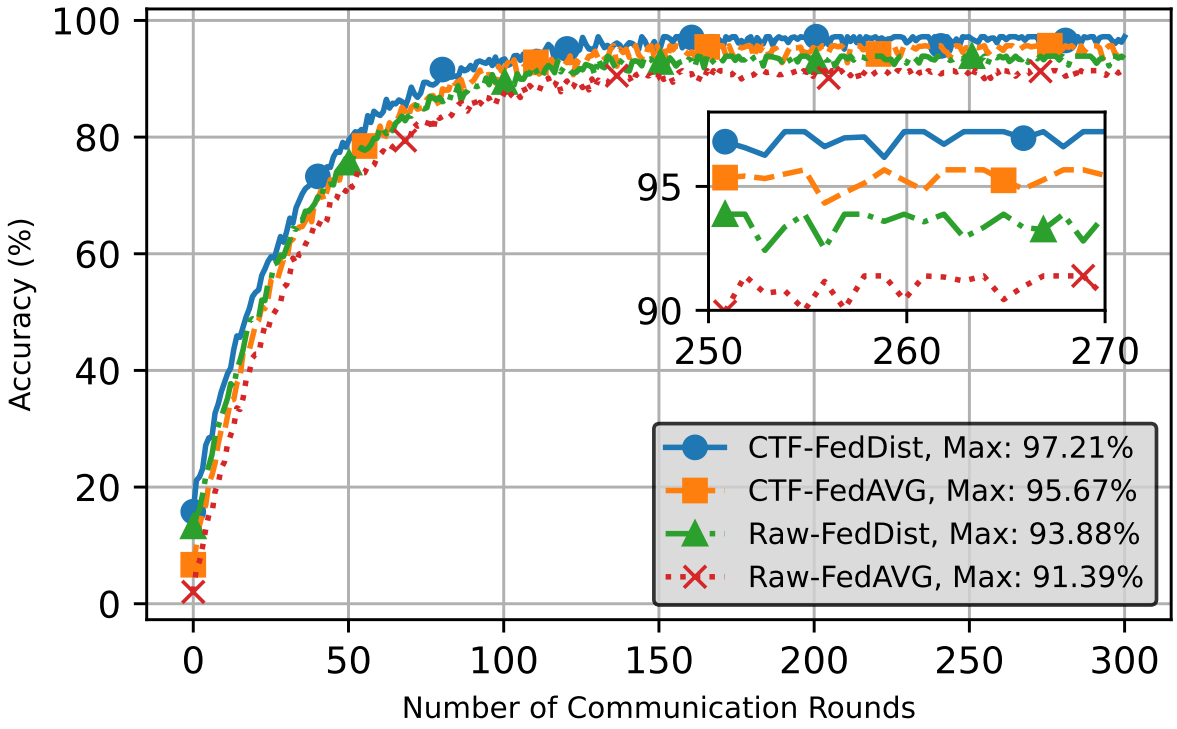


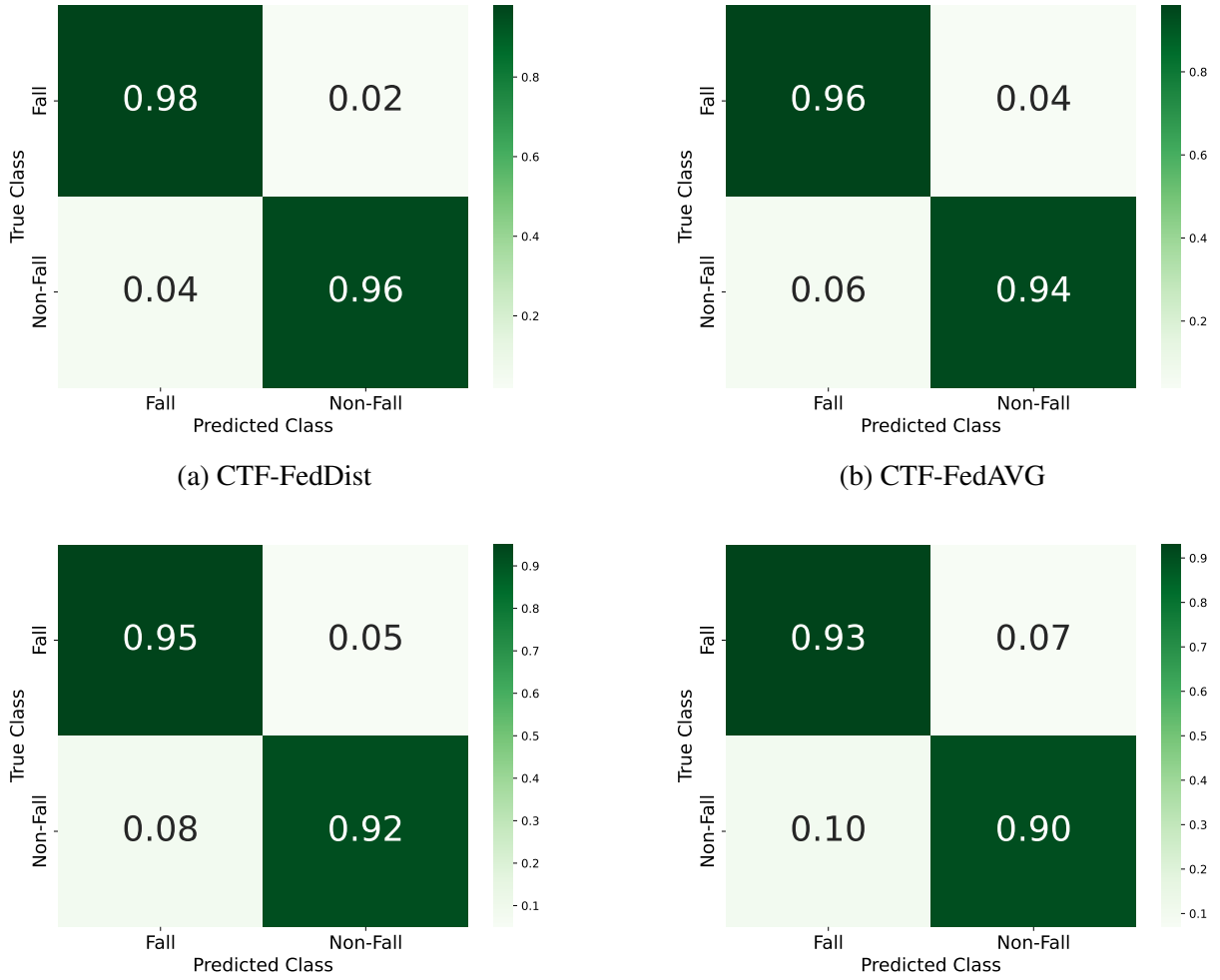
Figure 6.5: The comparative accuracy learning curve for CTF and simple feature for FedDist and FedAVG algorithm for fall detection.

6.5.3 Case:3 CTF with QAT

Building upon the insights from the previous two case studies, we now turned our attention to model compression through quantisation at varying bit precisions and incorporating QAT. This case study is driven by optimising the FFQ framework, particularly the FedDist algorithm, to balance performance (accuracy) and reduce communication overhead. Furthermore, this case study evaluates the impact of lowering model parameter precision from the standard 32-bit to 16-bit and further to 8-bit, both with and without QAT.

The results in Fig. 6.7, plot the model accuracy for multi-class activity classification against the number of communication rounds, demonstrating the convergence behaviour over 300 communication rounds. The baseline 32-bit FedDist model serves as a benchmark with a peak accuracy of 92.50%. When quantisation is applied, the QAT-FedDist models at 16-bit and 8-bit precision achieve peak accuracies of 90.89% and 89.66%, respectively, affirming the efficacy of QAT in mitigating the loss in accuracy typically associated with reduced bit representations. The Post-Quant models at corresponding quantisation levels exhibit a slight dip, with peak accuracies of 88.06% for 16-bit and 86.89% for 8-bit, underscoring the importance of QAT in optimising the model for lower-bit precision operations.

Similarly, Fig. 6.8 presents the accuracies of a learning curve for binary Fall detection, illustrating the performance for models trained with and without QAT. Benchmarking against



(a) CTF-FedDist

(b) CTF-FedAVG

(c) Raw-FedDist

(d) Raw-FedAVG

Figure 6.6: Confusion matrices for binary fall detection using CTF and amplitude-only features with FedDist and FedAVG.

the FedDist 32-bit precision model, which achieved an accuracy of 97.21%, the QAT-FedDist models at 16-bit and 8-bit precisions achieve comparable accuracies of 94.33% and 91.23%, respectively. In contrast, the Post-Quant with FedDist at 16 and 8-bit quantisation demonstrated lower average accuracy of 90.06% and 88.86%. Although a certain degree of accuracy reduction is anticipated due to model compression, applying QAT considerably mitigates this effect, showcasing its critical role in maintaining high model performance despite the reduced bit depth.

Finally, the energy efficiency of the proposed FFQ framework is measured by the communication overhead, which is quantified using the simplified version of Equ. (6.18), given as $E_r = N \cdot (\beta \cdot P_{trn})$. With the energy cost coefficient for communication, β, assumed to be 0.0001 [222], and the payload size per communication round, P_{trn}, equating to the size of the model. Notably, the original model, at 32-bit precision, has a size of 185.91 KB. Through quantisation, the model size is considerably reduced to 98.33 KB for the 16-bit representation and 52.31 KB for the 8-bit version.

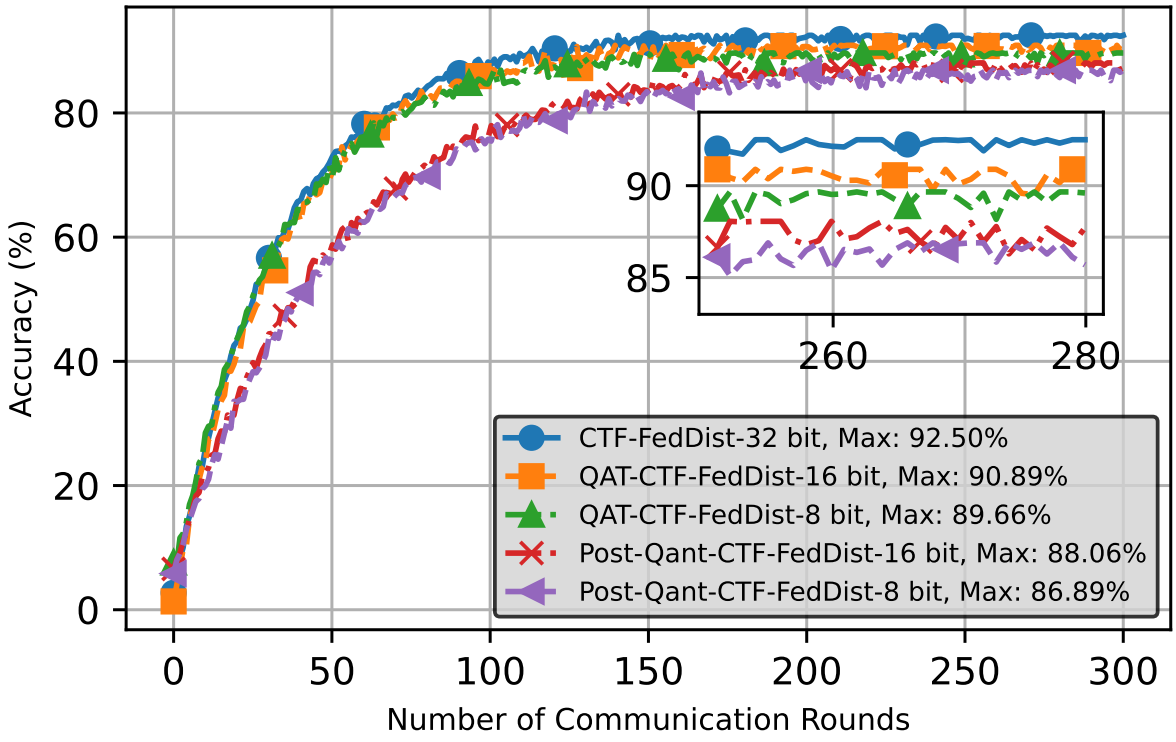


Figure 6.7: The comparative accuracy learning curve for post quantisation (Post-Qant) and QAT for Multi-class classification using FedDist.

These model size reductions result in a commensurate decrease in communication overhead and directly translate to a significant enhancement in energy efficiency, as shown in Fig. 6.9. The results show a reduction from 185.91 KB to 98.33 KB in the 16-bit quantised model, equating to approximately a 47% reduction in energy consumption per communication round. Similarly, the 8-bit quantised model, at 52.31 KB, implies a reduction of over 71% in energy costs compared to the original model size. This tangible improvement in energy efficiency underscores the practical relevance of our research in the context of FL setups. Additionally, the quantisation process, particularly when combined with QAT, not only preserves the accuracy of the FFQ framework but also significantly diminishes the energy required for model updates during the learning process.

6.5.4 Discussion on Model Convergence and Energy Efficiency

The evaluation of the FFQ framework across different case studies highlights the trade-off between model convergence and classification performance. The comparative learning curves indicate that model accuracy stabilises within 250–300 communication rounds, suggesting that further rounds may yield diminishing returns while increasing energy consumption. This observation is particularly relevant for optimising training efficiency in FL scenarios.

For **multi-class HAR classification** (Case 1), the learning curves in Fig. 6.3 show that Fed-

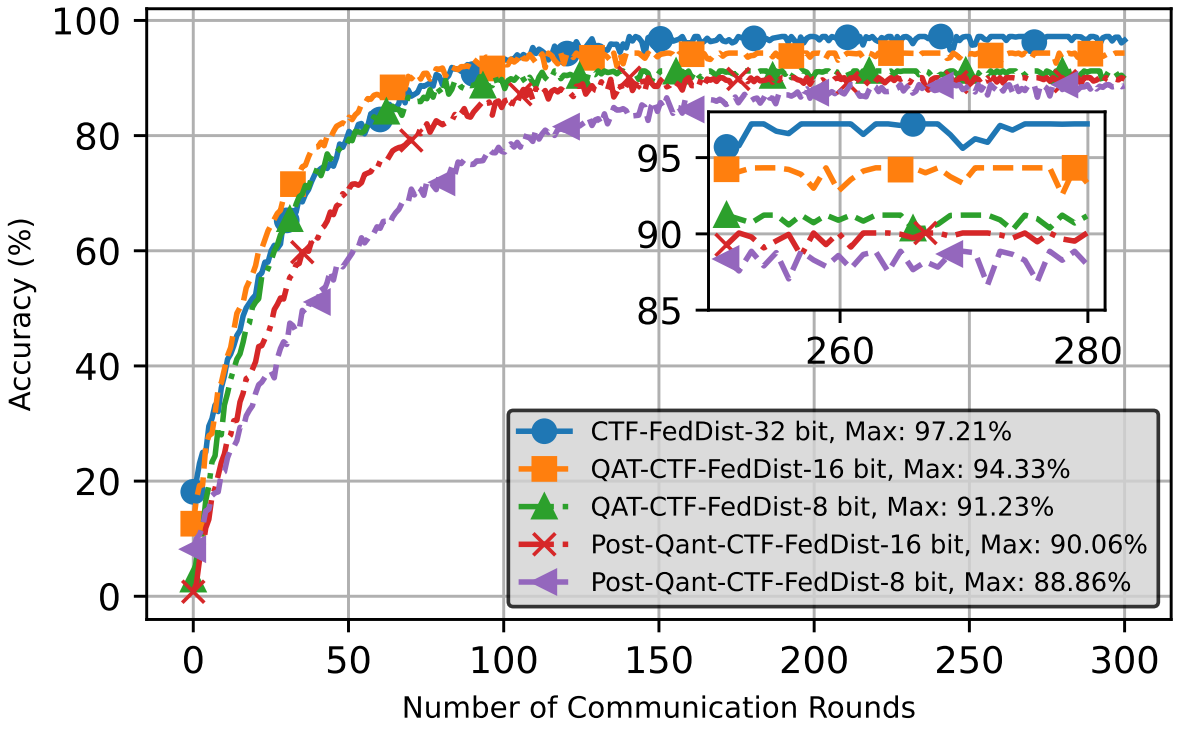


Figure 6.8: The comparative accuracy learning curve for Post-Qant and QAT for binary fall detection using FedDist.

Dist with CTF achieves the highest accuracy of 92.50% at 300 rounds, outperforming FedAVG. However, the learning curve suggests that optimal convergence is reached around 250 rounds, indicating that extending training beyond this point results in marginal improvements at the cost of increased communication overhead. Similarly, for **fall detection** (Case 2), the results in Fig. 6.5 reveal that the CTF-FedDist model reaches a peak accuracy of 97.21% at 300 rounds, with the accuracy curve flattening after approximately 250 rounds. This suggests that further training rounds contribute minimally to model improvement but significantly increase resource consumption.

In **QAT (Case 3)**, the learning curves in Fig. 6.7 and Fig. 6.8 highlight that model quantisation influences both convergence behavior and classification performance. The baseline 32-bit FedDist model achieves 92.50% accuracy for multi-class HAR and 97.21% for fall detection, whereas the 16-bit and 8-bit QAT models retain competitive performance with slight reductions (90.89% and 89.66% for multi-class, 94.33% and 91.23% for fall detection). Notably, quantised models demonstrate faster convergence, particularly at 16-bit precision, suggesting that reducing bit-depth can accelerate training while maintaining acceptable accuracy levels.

Implications for Energy Consumption

From an energy perspective, excessive training rounds increase computational and communication costs unnecessarily. Since the FFQ models exhibit accuracy saturation beyond 250 rounds,

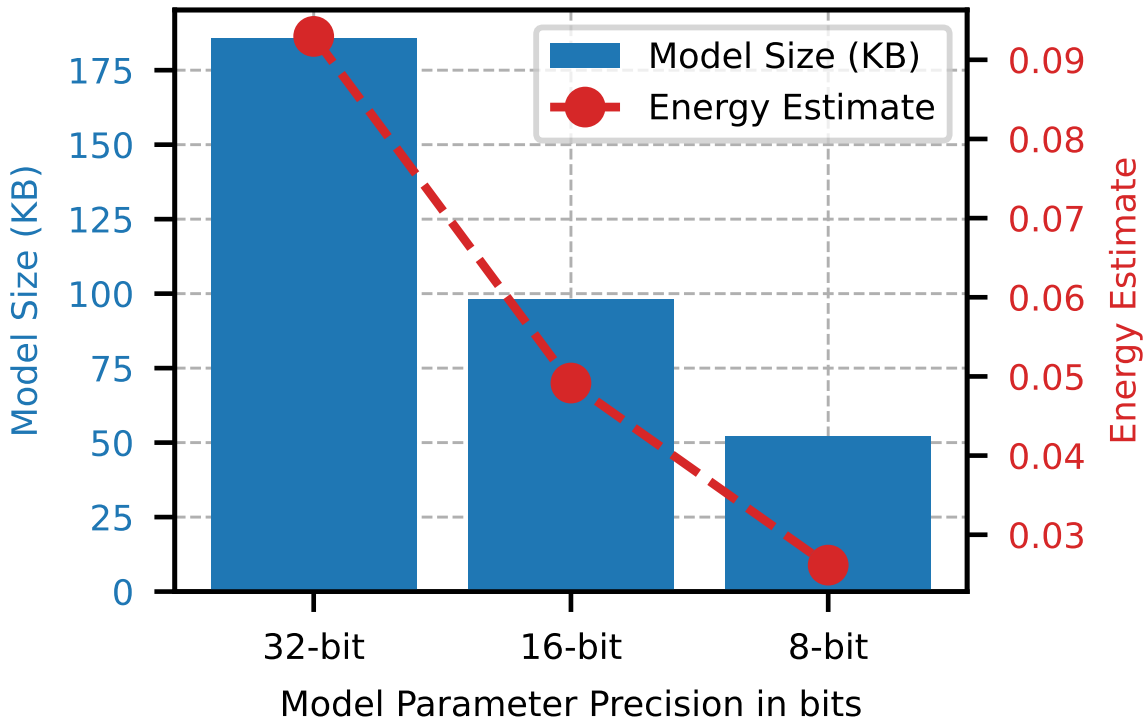


Figure 6.9: The comparison between the reduction in communication overhead, computed as energy estimates for different quantisation.

implementing an *adaptive stopping criterion* based on convergence metrics could significantly reduce energy consumption. Moreover, the quantisation process reduces the model size and communication overhead, making it more suitable for energy-constrained FL applications. The energy efficiency benefits of quantisation are evident from the reduced model sizes: 16-bit quantized models reduce communication overhead by approximately **47%**, while 8-bit models achieve a reduction of over **71%**. Given this trade-off, 16-bit QAT emerges as the most effective balance between accuracy retention and energy efficiency, as it achieves faster convergence with minimal degradation in model performance.

6.5.5 Summary of Key Findings

The results presented in this study validate the effectiveness of the proposed FFQ framework for HAR in federated settings, demonstrating its capability to balance accuracy, energy efficiency, and communication overhead. This section discusses key outcomes.

Impact of Client Participation on Model Performance and Energy Efficiency

The study maintained five federated clients for consistency, with each contributing to global model training. The results in Fig. 6.3 show that client participation combined with temporal

feature fusion (CTF) improves classification accuracy, reaching 92.50% for FedDist, compared to 88.31% with simple amplitude features. Energy efficiency was indirectly analysed through QAT, where reducing model precision to 16-bit and 8-bit lowered communication overhead by 47% and 71%, respectively (Fig. 6.9). While the number of clients was fixed, the findings suggest that optimising client selection strategies in FL can further enhance efficiency.

Trade-Off Between Local, Global, and Personalised Models

The results highlight the performance benefits of global FL over local models while ensuring privacy compared to centralised learning. CTF-FedDist achieved 92.50% accuracy, closely approaching centralised learning but without data sharing. Additionally, the use of personalised models (as explored in Chapter 5) can further refine local performance. While personalisation was not explicitly evaluated in this chapter, prior results suggest it can mitigate inter-client variability.

Energy Consumption at Different Stages

The study systematically analysed energy efficiency in communication overhead reduction through model quantisation. Fig. 6.9 shows that QAT significantly reduces energy costs without major accuracy loss, making it viable for real-world deployment. However, energy consumption related to semantic inference or model execution on-device (e.g., in pre-trained vision models, as applicable in Chapter 4) was not explicitly measured in this study. Future work could extend this evaluation by incorporating real-time device profiling.

6.6 Summary

This chapter presented the FFQ framework for indoor HAR using CSI, incorporating advanced pre-processing, feature fusion, and model quantisation strategies to address challenges **C2** and **C3**. Unlike traditional FL approaches for HAR, the FFQ framework employed the FedDist algorithm to adjust model parameters based on a dissimilarity measure of diverging neurons, effectively mitigating overfitting. Additionally, during local model training, the framework employed QAT to refine the precision of model computations and gradients from a standard 32-bit down to 16 or 8-bit formats. This strategy significantly reduces bandwidth requirements while maintaining an acceptable level of accuracy.

Our comprehensive evaluation demonstrates that the CTF significantly enhances activity classification accuracy, with an improvement of approximately 4.29% in multi-class HAR scenarios and up to 5.55% in binary fall detection tasks compared to models without feature fusion. However, the introduction of model compression revealed a trade-off between accuracy and energy efficiency. The QAT within the 16-bit quantised models resulted in a slight decrease

in accuracy, which is approximately 1.61% for multi-class tasks and 2.88% for fall detection scenarios, relative to their 32-bit counterparts. However, the 16-bit quantisation also achieved a 47% reduction in communication overhead. In the case of 8-bit quantisation, performance decrements of 3.84% for multi-class and 5.98% for binary classifications were observed, alongside an impressive 71% reduction in communication overhead.

Chapter 7

Conclusion and Future Work

This final chapter summarises the research presented in this thesis, highlighting the key contributions and their impact on addressing the three critical challenges in FL, as discussed in **Section 1.3**. These challenges include managing data diversity, ensuring robustness against adversarial threats, and optimising resource efficiency, requiring innovative real-world application solutions. This thesis has advanced FL by proposing novel frameworks that address these gaps, making it more effective in domains like energy forecasting, wireless communications, and HAR. This chapter also synthesises the contributions, explores their broader implications, and identifies future research directions to improve FL further. The goal is to provide a clear summary of the progress made and a roadmap for future work.

7.1 Summary of Contributions

In Chapter 3, a novel SDTA framework is proposed to address the challenges **C1** and **C2**. The SDTA overcome these challenges by leveraging layer-wise similarity measures, truncated mean aggregation to filter extreme values and DP to secure model updates. Statistical tests like Levene's and Fligner-Killeen confirmed significant heterogeneity among client datasets. SDTA performs layer-wise similarity measures to align the model updates from diverse clients and identify anomalous updates. Truncated mean aggregation filters the outliers in model parameters to mitigate the impact of noise and adversarial attacks. Under ideal FL conditions with non-IID data, SDTA achieved the MAPE of 2.63%, outperforming the FedAVG and FedDist with MAPE of 2.89% and 3.11%, respectively. This highlights the ability of the proposed SDTA to achieve higher accuracy and efficiency in environments with non-IID data distributions, effectively addressing the challenge **C1**. Furthermore, SDTA was further tested under DP constraints and adversarial scenarios to evaluate its robustness and privacy-preserving capabilities. Under strict DP settings with $\varepsilon = 0.1$, SDTA achieved a MAPE of 4.02%, surpassing FedAVG and FedDist by 20.4% and 9.7%, respectively. This validates the resilience of SDTA in balancing accuracy and privacy, a critical requirement in FL systems addressing challenge **C2**. Additionally, the

robustness of SDTA was further demonstrated under model sign inversion attacks affecting 40% of ENs. In this adversarial scenario, SDTA achieved a 20.5% MAPE reduction compared to FedAVG, showcasing its capability to detect and mitigate malicious updates effectively. Moreover, the framework exhibited remarkable stability under partial client participation, maintaining consistent performance even as the client selection rate dropped from 90% to 50%.

Chapter 4 introduced the SFBP framework, addressing challenges **C1** and **C3**. The proposed framework leveraged lightweight CV models, specifically MobileNetV3, for efficient edge-based semantic information extraction, enabling multi-modal fusion for proactive blockage prediction in mmWave/THz networks. MobileNetV3 demonstrated its suitability for edge processing by achieving a comparable object detection accuracy of 97.1%. MobileNetV3 reduced the inference latency to 51 ms compared to YOLOv5, with an inference latency of 66.4 ms, highlighting its efficiency for high-mobility networks. Additionally, the proposed framework effectively reduced communication cost by 88.75% compared to centralised learning and 57.87% compared to baseline FL, showcasing its ability to optimise resource efficiency in distributed settings, addressing challenge **C3**. The proposed framework integrated the SD-FedAVG aggregation mechanism to improve prediction accuracy, addressing challenge **C1**. Furthermore, this robust aggregation reduced the HO failure rates by up to 7.6% under worst-case conditions, demonstrating resilience and stability. Integrating multi-modal fusion with semantic-aware lightweight models and robust aggregation, the SFBP framework successfully balances computational efficiency, communication overhead, and prediction accuracy, making it highly applicable for next-generation wireless networks.

Chapter 5 introduced the HNFL framework, dealing with the challenges **C1** and **C3**. This hybrid model synergised the computational efficiency of SNNs with the dynamic temporal learning capabilities of LSTM networks for outdoor HAR using wearable sensor data. The proposed hybrid model integrated spiking layers for event-driven processing with LSTM layers adept at capturing temporal dependencies, addressing the challenge of multi-modal fusion. The HNFL framework is evaluated on two publically available datasets: the UCI for controlled indoor settings and the Real-World dataset for diverse and unpredictable outdoor scenarios. The simulation results showcased its robustness in handling both simple and complex environments. Specifically, the S-LSTM outperformed traditional LSTM models, achieving a 1.06% increase in accuracy for indoor environments and an impressive 3.84% improvement in outdoor settings. Moreover, energy efficiency, a critical component of challenge **C3**, was another notable outcome of this work. The S-LSTM model demonstrated a remarkable 32.30% improvement in energy efficiency compared to LSTM models, making it highly suitable for resource-constrained edge devices. Additionally, personalisation through fine-tuning the global model with local data further enhanced performance, yielding an average accuracy improvement of 9% across participants. This contribution exemplified the potential of hybrid neuromorphic architectures in FL and provided practical insights into addressing the dual challenges of data diversity and resource

efficiency in HAR applications.

Finally, Chapter 6 introduced the FFQ framework, which addresses **C2** and **C3** by leveraging advanced signal processing, feature engineering, and model compression techniques. Unlike the architectural innovations of the HNFL framework in Chapter 5, FFQ focuses on computational and communication efficiency through signal processing, feature fusion, and model compression. The proposed framework combines statistical and differential features with processed CSI data, providing a contextual understanding for navigating varied environmental conditions. Additionally, FFQ framework also includes a FedDist-based adjustment mechanism that reduces overfitting by aligning model updates according to neuron divergence. This approach ensures improved model robustness in federated settings. Moreover, the framework employed quantisation with QAT, reducing computation precision to 16-bit or 8-bit formats while maintaining competitive accuracy. Evaluation results demonstrated the effectiveness of FFQ in improving activity classification accuracy and energy efficiency. Feature fusion contributed to an improvement of 4.29% in multi-class HAR scenarios and 5.55% in binary fall detection tasks compared to models without fusion. However, model compression introduced a trade-off between accuracy and efficiency. While 16-bit quantisation resulted in minor accuracy reductions of 1.61% for multi-class and 2.88% for binary tasks, it achieved a 47% reduction in communication overhead. For 8-bit quantisation, the trade-off was more pronounced, with a 3.84% and 5.98% accuracy reduction, respectively, but an impressive 71% reduction in communication overhead. This contribution showcased how pre-processing, feature fusion and quantisation techniques can drive the adoption of FL in privacy-sensitive, non-invasive HAR systems, making it a valuable addition to real-world indoor applications.

7.2 Limitations and Future Research Direction

This thesis has addressed the three critical challenges in FL, offering innovative solutions for data diversity, adversarial robustness, and resource efficiency. Despite these contributions, several areas remain unexplored, paving the way for future research to build upon the current findings. This section outlines potential directions for enhancing FL frameworks by addressing the identified limitations, exploring emerging privacy-preserving techniques, and tackling new challenges. These directions aim to refine the scalability, robustness, and applicability of FL further across diverse real-world scenarios.

Privacy and security remain a critical challenge in FL, with DP being a widely used technique for safeguarding sensitive data. While DP effectively balances privacy and model utility, its reliance on noise injection can degrade accuracy, particularly in resource-constrained or highly sensitive applications. Emerging model encryption techniques and quantum key distribution (QKD) offer promising alternatives to address these limitations. HE enables computations on encrypted data, ensuring privacy throughout the training process. Similarly, QKD provides

a theoretically unbreakable method for secure communication between nodes. However, these techniques have significant computational and communication overheads, potentially limiting their scalability in practical FL deployments. Future research will conduct a thorough computational complexity analysis of these methods, exploring optimisations to make them viable for large-scale FL systems while maintaining a balance between security, efficiency, and accuracy.

Another critical limitation of this research lies in the scope of data and update mechanisms considered. The proposed frameworks were evaluated using data from a limited number of edge nodes, ensuring controlled heterogeneity. Additionally, the research primarily employed synchronous update mechanisms, where all participating nodes were required to complete local training before aggregation. While this approach simplifies implementation and analysis, it does not account for the straggler effect, delays caused by nodes with slower computational capabilities or unstable network connections. Such scenarios can significantly impact system performance and convergence time in real-world deployments. Future investigations should focus on scaling the analysis to more extensive and diverse edge networks and exploring asynchronous update strategies to mitigate the straggler effect. This would provide deeper insights into the robustness and efficiency of the proposed frameworks under realistic operational constraints.

Another limitation of the current research is the absence of an incentive mechanism to encourage active participation in model training, particularly for those contributing high-quality updates. In FL, the heterogeneity of client data and computational capabilities often leads to varying contributions to the global model's performance. Without a proper incentive mechanism, there is little motivation for clients to participate actively or to ensure the quality of their local updates. Incorporating incentive mechanisms could improve client engagement and enhance the system's overall robustness and efficiency by prioritising clients' contributions with valuable updates. Future research will explore dynamic incentive models that reward nodes based on the quality and relevance of their updates. These mechanisms will integrate game theory and reinforcement learning techniques to balance fairness and efficiency while maintaining privacy and resource constraints.

Another limitation of the current research lies in the assumption of homogeneous model architectures across all clients during aggregation. In real FL scenarios, edge nodes often have varying computational resources, sensor types, and data characteristics, making it impractical to enforce a single, unified model architecture across all participants. This constraint limits the flexibility and applicability of the existing frameworks, especially in settings involving diverse hardware or multi-modal data. Future research can also focus on developing aggregation methods that can seamlessly integrate updates from heterogeneous model architectures. Techniques such as meta-learning, cross-model compatibility layers, or shared latent space representations can be explored to enable effective aggregation while preserving the unique contributions of each client. This direction would enhance the scalability and adaptability of federated learning systems for diverse, real-world environments.

A critical limitation of the current frameworks is the absence of explainability mechanisms essential for interpreting and understanding the decisions made by models. In many real-world applications, such as healthcare, energy management, and wireless communications, providing insights into why a model made a specific prediction or decision is crucial for building trust and facilitating adoption. Explainability is particularly challenging in FL due to the decentralised nature of the system, where local data remains private, and only model updates are shared. Future research can explore developing lightweight and privacy-preserving explainability techniques tailored for FL. Methods such as federated feature attribution, model interpretability layers, or distributed explainable AI (XAI) frameworks could be explored to provide transparency without compromising client privacy. Incorporating explainability in FL will enhance user trust and enable stakeholders to identify potential biases or errors in the system, paving the way for more reliable and ethical applications.

Another promising area for future exploration is incorporating multi-modal transformer models into the FL paradigm. With their self-attention mechanisms, transformers have shown remarkable success in capturing complex relationships across diverse data modalities, such as text, images, and time-series signals. Integrating such models into FL frameworks could enhance the ability to process and fuse multi-modal data in a decentralised setting, enabling richer feature extraction and improved predictive performance. However, applying transformers in FL introduces new challenges, such as high computational and memory demands, which could strain resource-constrained edge devices. Future research will investigate the techniques to optimise transformer architectures for FL, such as parameter sharing, pruning, quantisation, and efficient attention mechanisms.

Acknowledgement for the use of AI

I acknowledge the use of the two language assistance tools, ChatGPT by OpenAI and Grammarly, in proofreading of this thesis. These tools were employed to enhance the clarity, coherence, and academic tone. Their contributions were limited to improving the language and flow of the text, while the intellectual content and findings of this work remain entirely my own.

Bibliography

[1] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[2] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Toward an intelligent edge: Wireless communication meets machine learning," *IEEE communications magazine*, vol. 58, no. 1, pp. 19–25, 2020.

[3] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," *arXiv preprint arXiv:2002.10619*, 2020.

[4] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3347–3366, 2021.

[5] 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.

[6] L. U. Khan, I. Yaqoob, N. H. Tran, S. A. Kazmi, T. N. Dang, and C. S. Hong, "Edge-computing-enabled smart cities: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 200–10 232, 2020.

[7] B. Custers, A. M. Sears, F. Dechesne, I. Georgieva, T. Tani, and S. van der Hof, *EU personal data protection in policy and practice*. Springer, 2019.

[8] B. M. Gaff, H. E. Sussman, and J. Geetter, "Privacy and big data," *Computer*, vol. 47, no. 6, pp. 7–9, 2014.

[9] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "yang2020federated," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2022–2035, 2020.

[10] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[11] 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. [Online]. Available: https://doi.org/10.48550/arXiv.1610.05492

[12] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[13] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140 699–140 725, 2020.

[14] Y. Deng, T. Han, and N. Ansari, "Fedvision: Federated video analytics with edge computing," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 62–72, 2020.

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

[16] H. N. C. Neto, J. Hribar, I. Dusparic, D. M. F. Mattos, and N. C. Fernandes, "A survey on securing federated learning: Analysis of applications, attacks, challenges, and trends," *IEEE Access*, vol. 11, pp. 41 928–41 953, 2023.

[17] 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*, vol. 10, no. 2, pp. 194–213, 2024.

[18] 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

[19] 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, 2024. [Online]. Available: https://www.mdpi.com/1999-5903/16/10/374

[20] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients - how easy is it to break privacy in federated learning?" in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell,

M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 16 937–16 947. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/c4ede56bbd98819ae6112b20ac6bf145-Paper.pdf

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

[22] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.

[23] B. Custers, A. M. Sears, F. Dechesne, I. Georgieva, T. Tani, and S. Van der Hof, *EU personal data protection in policy and practice*. Springer, 2019, vol. 29.

[24] "Nvidia clara federated learning." [Online]. Available: https://resources.nvidia.com/en-us-federated-learning/bvu-ea6hc0k?ncid=pa-srch-goog-8454

[25] S. Yang, B. Ren, X. Zhou, and L. Liu, "Parallel distributed logistic regression for vertical federated learning without third-party coordinator," *arXiv preprint arXiv:1911.09824*, 2019.

[26] 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.

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

[28] 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.

[29] X. Gu, F. Sabrina, Z. Fan, and S. Sohail, "A review of privacy enhancement methods for federated learning in healthcare systems," *International Journal of Environmental Research and Public Health*, vol. 20, no. 15, p. 6539, 2023.

[30] 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.

[31] 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.

[32] 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.

[33] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[34] B. S. Guendouzi, S. Ouchani, H. EL Assaad, and M. EL Zaher, "A systematic review of federated learning: Challenges, aggregation methods, and development tools," *Journal of Network and Computer Applications*, vol. 220, p. 103714, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804523001339

[35] Z. Lu, H. Pan, Y. Dai, X. Si, and Y. Zhang, "Federated learning with non-iid data: A survey," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19 188–19 209, 2024.

[36] D. Gao, X. Yao, and Q. Yang, "A survey on heterogeneous federated learning," *arXiv preprint arXiv:2210.04505*, 2022.

[37] A. B. de Luca, G. Zhang, X. Chen, and Y. Yu, "Mitigating data heterogeneity in federated learning with data augmentation," *arXiv preprint arXiv:2206.09979*, 2022.

[38] K. Stacke, G. Eilertsen, J. Unger, and C. Lundström, "Measuring domain shift for deep learning in histopathology," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 2, pp. 325–336, 2021.

[39] Y. H. Chan and E. C. Ngai, "Fedhe: Heterogeneous models and communication-efficient federated learning," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*, 2021, pp. 207–214.

[40] L. Qu, Y. Zhou, P. P. Liang, Y. Xia, F. Wang, E. Adeli, L. Fei-Fei, and D. Rubin, "Rethinking architecture design for tackling data heterogeneity in federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10 061–10 071.

[41] Y. Sun, J. Shao, Y. Mao, J. H. Wang, and J. Zhang, "Semi-decentralized federated edge learning for fast convergence on non-iid data," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 1898–1903.

[42] S. Rajendran, Z. Xu, W. Pan, A. Ghosh, and F. Wang, "Data heterogeneity in federated learning with electronic health records: Case studies of risk prediction for acute kidney injury and sepsis diseases in critical care," *PLOS Digital Health*, vol. 2, no. 3, p. e0000117, 2023.

[43] Y. Cheng, L. Zhang, and A. Li, "Gfl: Federated learning on non-iid data via privacy-preserving synthetic data," in *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2023, pp. 61–70.

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

[45] R. Saha, S. Misra, and P. K. Deb, "Fogfl: Fog-assisted federated learning for resource-constrained iot devices," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8456–8463, 2021.

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

[47] S. B. Guendouzi, S. Ouchani, and M. Malki, "Enhancing the aggregation of the federated learning for the industrial cyber physical systems," in *2022 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2022, pp. 197–202.

[48] T. Berghout, T. Bentrcia, M. A. Ferrag, and M. Benbouzid, "A heterogeneous federated transfer learning approach with extreme aggregation and speed," *Mathematics*, vol. 10, no. 19, 2022. [Online]. Available: https://www.mdpi.com/2227-7390/10/19/3528

[49] C.-H. Yao, B. Gong, H. Qi, Y. Cui, Y. Zhu, and M.-H. Yang, "Federated multi-target domain adaptation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2022, pp. 1424–1433.

[50] P. Tian, Z. Chen, W. Yu, and W. Liao, "Towards asynchronous federated learning based threat detection: A dc-adam approach," *Computers & Security*, vol. 108, p. 102344, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404821001681

[51] L. Che, J. Wang, Y. Zhou, and F. Ma, "Multimodal federated learning: A survey," *Sensors*, vol. 23, no. 15, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/15/6986

[52] Y.-M. Lin, Y. Gao, M.-G. Gong, S.-J. Zhang, Y.-Q. Zhang, and Z.-Y. Li, "Federated learning on multimodal data: A comprehensive survey," *Machine Intelligence Research*, vol. 20, no. 4, pp. 539–553, 2023.

[53] W. Huang, D. Wang, X. Ouyang, J. Wan, J. Liu, and T. Li, "Multimodal federated learning: Concept, methods, applications and future directions," *Information Fusion*, vol. 112, p. 102576, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253524003543

[54] P. Qi, D. Chiaro, and F. Piccialli, "Fl-fd: Federated learning-based fall detection with multimodal data fusion," *Information Fusion*, vol. 99, p. 101890, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253523002063

[55] T. Shaik, X. Tao, L. Li, H. Xie, and J. D. Velásquez, "A survey of multimodal information fusion for smart healthcare: Mapping the journey from data to wisdom," *Information Fusion*, vol. 102, p. 102040, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253523003561

[56] S. Malakar, S. D. Roy, S. Das, S. Sen, J. D. Velásquez, and R. Sarkar, "Computer based diagnosis of some chronic diseases: a medical journey of the last two decades," *Archives of Computational Methods in Engineering*, vol. 29, no. 7, pp. 5525–5567, 2022.

[57] T. Zhang and M. Shi, "Multi-modal neuroimaging feature fusion for diagnosis of alzheimer's disease," *Journal of Neuroscience Methods*, vol. 341, p. 108795, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165027020302181

[58] J. Duan, J. Xiong, Y. Li, and W. Ding, "Deep learning based multimodal biomedical data fusion: An overview and comparative review," *Information Fusion*, vol. 112, p. 102536, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253524003142

[59] X. Qu, Z. Liu, C. Q. Wu, A. Hou, X. Yin, and Z. Chen, "Mfgan: Multimodal fusion for industrial anomaly detection using attention-based autoencoder and generative adversarial network," *Sensors*, vol. 24, no. 2, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/2/637

[60] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705121000381

[61] 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*, vol. 150, pp. 272–293, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X23003333

[62] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016. [Online]. Available: https://doi.org/10.1137/130943170

[63] W. Huang, M. Ye, Z. Shi, G. Wan, H. Li, B. Du, and Q. Yang, "Federated learning for generalization, robustness, fairness: A survey and benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2024.

[64] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems*, I. Dhillon, D. Papailiopoulos, and V. Sze, Eds., vol. 2, 2020, pp. 429–450. [Online]. Available: https://proceedings.mlsys.org/paper_files/paper/2020/file/1f5fe83998a09396ebe6477d9475ba0c-Paper.pdf

[65] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," *arXiv preprint arXiv:2002.06440*, 2020.

[66] R. Pathak and M. J. Wainwright, "Fedsplit: an algorithmic framework for fast federated optimization," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 7057–7066. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/4ebd440d99504722d80de606ea8507da-Paper.pdf

[67] S. Li, Q. Qi, J. Wang, H. Sun, Y. Li, and F. R. Yu, "Ggs: General gradient sparsification for federated learning in edge computing," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–7.

[68] S. EK, F. PORTET, P. LALANDA, and G. VEGA, "A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison," in *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2021, pp. 1–10.

[69] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated meta-learning with fast convergence and efficient communication," *arXiv preprint arXiv:1802.07876*, 2018.

[70] J. Hamer, M. Mohri, and A. T. Suresh, "FedBoost: A communication-efficient algorithm for federated learning," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 3973–3983. [Online]. Available: https://proceedings.mlr.press/v119/hamer20a.html

[71] J. Chen, H. Yan, Z. Liu, M. Zhang, H. Xiong, and S. Yu, "When federated learning meets privacy-preserving computation," *ACM Comput. Surv.*, vol. 56, no. 12, Oct. 2024. [Online]. Available: https://doi.org/10.1145/3679013

[72] K. N. Kumar, C. K. Mohan, and L. R. Cenkeramaddi, "The impact of adversarial attacks on federated learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 2672–2691, 2024.

[73] E. Hallaji, R. Razavi-Far, and M. Saif, "Federated and transfer learning: A survey on adversaries and defense mechanisms," in *Federated and Transfer Learning*. Springer, 2022, pp. 29–55.

[74] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095070512300134X

[75] 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)*, 2021, pp. 852–863.

[76] E. Rosenfeld, E. Winston, P. Ravikumar, and Z. Kolter, "Certified robustness to label-flipping attacks via randomized smoothing," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 8230–8241. [Online]. Available: https://proceedings.mlr.press/v119/rosenfeld20b.html

[77] X. Zhou, M. Xu, Y. Wu, and N. Zheng, "Deep model poisoning attack on federated learning," *Future Internet*, vol. 13, no. 3, 2021. [Online]. Available: https://www.mdpi.com/1999-5903/13/3/73

[78] J. Geng, Y. Mou, Q. Li, F. Li, O. Beyan, S. Decker, and C. Rong, "Improved gradient inversion attacks and defenses in federated learning," *IEEE Transactions on Big Data*, pp. 1–13, 2023.

[79] Y. Sun, Z. Liu, J. Cui, J. Liu, K. Ma, and J. Liu, "Client-side gradient inversion attack in federated learning using secure aggregation," *IEEE Internet of Things Journal*, vol. 11, no. 17, pp. 28 774–28 786, 2024.

[80] B. Rao, J. Zhang, D. Wu, C. Zhu, X. Sun, and B. Chen, "Privacy inference attack and defense in centralized and federated learning: A comprehensive survey," *IEEE Transactions on Artificial Intelligence*, pp. 1–22, 2024.

[81] Z. Ye, W. Luo, Q. Zhou, Z. Zhu, Y. Shi, and Y. Jia, "Gradient inversion attacks: Impact factors analyses and privacy enhancement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–17, 2024.

[82] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. Vincent Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[83] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (Csur)*, vol. 51, no. 4, pp. 1–35, 2018.

[84] X. Yi, R. Paulet, E. Bertino, X. Yi, R. Paulet, and E. Bertino, *Homomorphic encryption*. Springer, 2014.

[85] P. Martins, L. Sousa, and A. Mariano, "A survey on fully homomorphic encryption: An engineering perspective," vol. 50, no. 6, Dec. 2017. [Online]. Available: https://doi.org/10.1145/3124441

[86] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," Cryptology ePrint Archive, Paper 2012/144, 2012. [Online]. Available: https://eprint.iacr.org/2012/144

[87] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 19–38.

[88] M. Al-Rubaie and J. M. Chang, "Privacy-preserving machine learning: Threats and solutions," *IEEE Security & Privacy*, vol. 17, no. 2, pp. 49–58, 2019.

[89] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Annual Cryptology Conference*. Springer, 2012, pp. 643–662.

[90] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[91] M. Yang, T. Guo, T. Zhu, I. Tjuawinata, J. Zhao, and K.-Y. Lam, "Local differential privacy and its applications: A comprehensive survey," *Computer Standards & Interfaces*, vol. 89, p. 103827, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0920548923001083

[92] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Communication-efficient federated distillation with active data sampling," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 201–206.

[93] 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.

[94] G. Gad and Z. Fadlullah, "Federated learning via augmented knowledge distillation for heterogenous deep human activity recognition systems," *Sensors*, vol. 23, no. 1, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/1/6

[95] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025520301201

[96] F. Mo, H. Haddadi, K. Katevas, E. Marin, D. Perino, and N. Kourtellis, "Ppfl: privacy-preserving federated learning with trusted execution environments," ser. MobiSys '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 94–108. [Online]. Available: https://doi.org/10.1145/3458864.3466628

[97] B. Weinger, J. Kim, A. Sim, M. Nakashima, N. Moustafa, and K. J. Wu, "Enhancing iot anomaly detection performance for federated learning," *Digital Communications and Networks*, vol. 8, no. 3, pp. 314–323, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864822000190

[98] S. Li, Y. Cheng, W. Wang, Y. Liu, and T. Chen, "Learning to detect malicious clients for robust federated learning," *arXiv preprint arXiv:2002.00211*, 2020.

[99] 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, 2023.

[100] 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*, ser. IH&MMSec '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 51–62. [Online]. Available: https://doi.org/10.1145/3437880.3460403

[101] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X22000784

[102] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X22001339

[103] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/f4b9ec30ad9f68f89b29639786cb62ef-Paper.pdf

[104] T. Idé, "Collaborative anomaly detection on blockchain from noisy sensor data," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2018, pp. 120–127.

[105] Y. Gou, R. Wang, Z. Li, M. A. Imran, and L. Zhang, "Clustered hierarchical distributed federated learning," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 177–182.

[106] 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)*, 2022, pp. 4333–4337.

[107] 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.

[108] Z. Chen, W. Yi, H. Shin, and A. Nallanathan, "Adaptive model pruning for communication and computation efficient wireless federated learning," *IEEE Transactions on Wireless Communications*, vol. 23, no. 7, pp. 7582–7598, 2024.

[109] S. Vahidian, M. Morafah, and B. Lin, "Personalized federated learning by structured and unstructured pruning under data heterogeneity," in *2021 IEEE 41st International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2021, pp. 27–34.

[110] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, "Model pruning enables efficient federated learning on edge devices," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 12, pp. 10 374–10 386, 2023.

[111] L. Yi, X. Shi, N. Wang, J. Zhang, G. Wang, and X. Liu, "Fedpe: Adaptive model pruning-expanding for federated learning on mobile devices," *IEEE Transactions on Mobile Computing*, vol. 23, no. 11, pp. 10 475–10 493, 2024.

[112] X. An, L. Shen, H. Hu, and Y. Luo, "Federated learning with manifold regularization and normalized update reaggregation," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 55 097–55 109. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/acf2b98eeb09b21968c2de6b1c6952e9-Paper-Conference.pdf

[113] 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*, ser. CIKM '22. New York, NY,

USA: Association for Computing Machinery, 2022, p. 862–873. [Online]. Available: https://doi.org/10.1145/3511808.3557475

[114] 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)*, 2022, pp. 61–70.

[115] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste, "Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks," *Journal of Machine Learning Research*, vol. 22, no. 241, pp. 1–124, 2021. [Online]. Available: http://jmlr.org/papers/v22/21-0366.html

[116] 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 (ICCV)*, Oct 2017.

[117] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Sparse binary compression: Towards distributed deep learning with minimal communication," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[118] 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.

[119] 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.

[120] 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.

[121] 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.

[122] 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.

[123] 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.

[124] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3125–3148, 2018.

[125] 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.

[126] 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.

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

[128] 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.

[129] 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.

[130] 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.

[131] 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.

[132] M. A. Husnoo, A. Anwar, N. Hosseinzadeh, S. N. Islam, A. N. Mahmood, and R. Doss, "A secure federated learning framework for residential short term load forecasting," *IEEE Transactions on Smart Grid*, 2023.

[133] C. Briggs, Z. Fan, and P. Andras, "Federated learning for short-term residential load forecasting," *IEEE Open Access Journal of Power and Energy*, vol. 9, pp. 573–583, 2022.

[134] G. Charan, A. Hredzak, C. Stoddard, B. Berrey, M. Seth, H. Nunez, and A. Alkhateeb, "Towards real-world 6g drone communication: Position and camera aided beam prediction," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 2951–2956.

[135] M. Alrabeiah, A. Hredzak, Z. Liu, and A. Alkhateeb, "Viwi: A deep learning dataset framework for vision-aided wireless communications," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.

[136] G. Charan, M. Alrabeiah, and A. Alkhateeb, "Vision-aided 6g wireless communications: Blockage prediction and proactive handoff," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10 193–10 208, 2021.

[137] G. Charan and A. Alkhateeb, "Computer vision aided blockage prediction in real-world millimeter wave deployments," in *2022 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2022, pp. 1711–1716.

[138] G. Charan, M. Alrabeiah, and A. Alkhateeb, "Vision-aided dynamic blockage prediction for 6g wireless communication networks," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–6.

[139] G. Diraco, G. Rescio, P. Siciliano, and A. Leone, "Review on human action recognition in smart living: Sensing technology, multimodality, real-time processing, interoperability, and resource-constrained processing," *Sensors*, vol. 23, no. 11, p. 5281, 2023.

[140] S. Kalabakov, B. Jovanovski, D. Denkovski, V. Rakovic, B. Pfitzner, O. Konak, B. Arnrich, and H. Gjoreski, "Federated learning for activity recognition: A system level perspective," *IEEE Access*, 2023.

[141] M. M. Islam, S. Nooruddin, F. Karray, and G. Muhammad, "Human activity recognition using tools of convolutional neural networks: A state of the art review, data sets, challenges, and future prospects," *Computers in Biology and Medicine*, p. 106060, 2022.

[142] S. M. Bokhari, S. Sohaib, A. R. Khan, M. Shafi *et al.*, "Dgru based human activity recognition using channel state information," *Measurement*, vol. 167, p. 108245, 2021.

[143] H. F. T. Ahmed, H. Ahmad, and C. Aravind, "Device free human gesture recognition using wi-fi csi: A survey," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103281, 2020.

[144] N. Damodaran, E. Haruni, M. Kokhkharova, and J. Schäfer, "Device free human activity and fall recognition using wifi channel state information (csi)," *CCF Transactions on Pervasive Computing and Interaction*, vol. 2, no. 1, pp. 1–17, 2020.

[145] G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. M. Ni, "We can hear you with wi-fi!" *IEEE Transactions on Mobile Computing*, vol. 15, no. 11, pp. 2907–2920, 2016.

[146] S. Sathyanarayana, R. K. Satzoda, S. Sathyanarayana, and S. Thambipillai, "Vision-based patient monitoring: a comprehensive review of algorithms and technologies," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 2, pp. 225–251, 2018.

[147] H. Yu, Z. Chen, X. Zhang, X. Chen, F. Zhuang, H. Xiong, and X. Cheng, "Fedhar: Semi-supervised online learning for personalized federated human activity recognition," *IEEE Transactions on Mobile Computing*, 2021.

[148] P. Qi, D. Chiaro, and F. Piccialli, "Fl-fd: Federated learning-based fall detection with multimodal data fusion," *Information Fusion*, p. 101890, 2023.

[149] K. Zhang, X. Liu, X. Xie, J. Zhang, B. Niu, and K. Li, "A cross-domain federated learning framework for wireless human sensing," *IEEE Network*, vol. 36, no. 5, pp. 122–128, 2022.

[150] D. Cheng, L. Zhang, C. Bu, X. Wang, H. Wu, and A. Song, "Protohar: Prototype guided personalized federated learning for human activity recognition," *IEEE Journal of Biomedical and Health Informatics*, 2023.

[151] 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.

[152] 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.

[153] S. Yousefi, H. Narui, S. Dayal, S. Ermon, and S. Valaee, "A survey on behavior recognition using wifi channel state information," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 98–104, 2017.

[154] S. Sigg, M. Scholz, S. Shi, Y. Ji, and M. Beigl, "Rf-sensing of activities from non-cooperative subjects in device-free recognition systems using ambient and local signals," *IEEE Transactions on Mobile Computing*, vol. 13, no. 4, pp. 907–920, 2013.

[155] Y. Wang, K. Wu, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 581–594, 2016.

[156] H. Yu, Z. Chen, X. Zhang, X. Chen, F. Zhuang, H. Xiong, and X. Cheng, "Fedhar: Semi-supervised online learning for personalized federated human activity recognition," *IEEE Transactions on Mobile Computing*, vol. 22, no. 6, pp. 3318–3332, 2023.

[157] C.-H. Pham, T. Huynh-The, E. Sedgh-Gooya, M. El-Bouz, and A. Alfalou, "Extension of physical activity recognition with 3d cnn using encrypted multiple sensory data to

federated learning based on multi-key homomorphic encryption," *Computer Methods and Programs in Biomedicine*, vol. 243, p. 107854, 2024.

[158] P. Wang, T. Ouyang, Q. Wu, Q. Huang, J. Gong, and X. Chen, "Hydra: Hybrid-model federated learning for human activity recognition on heterogeneous devices," *Journal of Systems Architecture*, vol. 147, p. 103052, 2024.

[159] X. Ouyang, Z. Xie, J. Zhou, J. Huang, and G. Xing, "Clusterfl: a similarity-aware federated learning system for human activity recognition," in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, 2021, pp. 54–66.

[160] Z. Xiao, X. Xu, H. Xing, F. Song, X. Wang, and B. Zhao, "A federated learning system with enhanced feature extraction for human activity recognition," *Knowledge-Based Systems*, vol. 229, p. 107338, 2021.

[161] A. M. Ashleibta, A. Taha, M. A. Khan, W. Taylor, A. Tahir, A. Zoha, Q. H. Abbasi, and M. A. Imran, "5g-enabled contactless multi-user presence and activity detection for independent assisted living," *Scientific Reports*, vol. 11, no. 1, pp. 1–15, 2021.

[162] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: A ubiquitous wifi-based gesture recognition system," in *2015 IEEE conference on computer communications (INFOCOM)*. IEEE, 2015, pp. 1472–1480.

[163] D. Zhang, H. Wang, and D. Wu, "Toward centimeter-scale human activity sensing with wi-fi signals," *Computer*, vol. 50, pp. 48–57, 2017.

[164] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Device-free human activity recognition using commercial wifi devices," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1118–1131, 2017.

[165] B. McMahan, E. Moore, D. Ramage, and S. Hampson, "Communication-efficient learning of deep networks from decentralised data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[166] J. D. Fernández, S. P. Menci, C. M. Lee, A. Rieger, and G. Fridgen, "Privacy-preserving federated learning for residential short-term load forecasting," *Applied energy*, vol. 326, p. 119915, 2022.

[167] S. Wang, L. Sun, and S. Iqbal, "Green financing role on renewable energy dependence and energy transition in e7 economies," *Renewable Energy*, vol. 200, pp. 1561–1572, 2022.

[168] Y. He, F. Luo, M. Sun, and G. Ranzi, "Privacy-preserving and hierarchically federated framework for short-term residential load forecasting," *IEEE Transactions on Smart Grid*, vol. 14, no. 6, pp. 4409–4423, 2023.

[169] Y. Gu, L. Quan, and F. Ren, "Wifi-assisted human activity recognition," in *2014 IEEE Asia Pacific Conference on Wireless and Mobile*. IEEE, 2014, pp. 60–65.

[170] C. Sabater, A. Bellet, and J. Ramon, "An accurate, scalable and verifiable protocol for federated differentially private averaging," *Machine Learning*, pp. 1–45, 2022.

[171] M. Wang, Z. Yu, Y. Chen, X. Yang, and J. Zhou, "Short-term load forecasting considering improved cumulative effect of hourly temperature," *Electric Power Systems Research*, vol. 205, p. 107746, 2022.

[172] A. J. del Real, F. Dorado, and J. Durán, "Energy demand forecasting using deep learning: applications for the french grid," *Energies*, vol. 13, no. 9, p. 2242, 2020.

[173] A. Lahouar and J. B. H. Slama, "Day-ahead load forecast using random forest and expert input selection," *Energy Conversion and Management*, vol. 103, pp. 1040–1051, 2015.

[174] R. Lu and S. H. Hong, "Incentive-based demand response for smart grid with reinforcement learning and deep neural network," *Applied energy*, vol. 236, pp. 937–949, 2019.

[175] A. Shaqour, T. Ono, A. Hagishima, and H. Farzaneh, "Electrical demand aggregation effects on the performance of deep learning-based short-term load forecasting of a residential building," *Energy and AI*, vol. 8, p. 100141, 2022.

[176] M. N. Fekri, H. Patel, K. Grolinger, and V. Sharma, "Deep learning for load forecasting with smart meter data: Online adaptive recurrent neural network," *Applied Energy*, vol. 282, p. 116177, 2021.

[177] E. Sannara, F. Portet, P. Lalanda, and V. German, "A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison," in *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2021, pp. 1–10.

[178] ISO-New England. [Online]. Available: https://www.iso-ne.com/markets-operations/system-forecast-status/three-day-system-demand-forecast

[179] Y. Zhou, Y. Zhu, and W. K. Wong, "Statistical tests for homogeneity of variance for clinical trials and recommendations," *Contemporary Clinical Trials Communications*, vol. 33, p. 101119, 2023.

[180] G. Pan, "On a levene type test for equality of two variances," *Journal of Statistical Computation and Simulation*, vol. 63, no. 1, pp. 59–71, 1999.

[181] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, "Wireless communications and applications above 100 ghz: Opportunities and challenges for 6g and beyond," *IEEE access*, vol. 7, pp. 78 729–78 757, 2019.

[182] C. De Alwis, A. Kalla, Q.-V. Pham, P. Kumar, K. Dev, W.-J. Hwang, and M. Liyanage, "Survey on 6g frontiers: Trends, applications, requirements, technologies and future research," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 836–886, 2021.

[183] C.-X. Wang, X. You, X. Gao, X. Zhu, Z. Li, C. Zhang, H. Wang, Y. Huang, Y. Chen, H. Haas *et al.*, "On the road to 6g: Visions, requirements, key technologies and testbeds," *IEEE Communications Surveys & Tutorials*, 2023.

[184] M. Al-Quraan, A. Khan, L. Mohjazi, A. Centeno, A. Zoha, and M. A. Imran, "Intelligent beam blockage prediction for seamless connectivity in vision-aided next-generation wireless networks," *IEEE Transactions on Network and Service Management*, 2022.

[185] J. G. Andrews, T. Bai, M. N. Kulkarni, A. Alkhateeb, A. K. Gupta, and R. W. Heath, "Modeling and analyzing millimeter wave cellular systems," *IEEE Transactions on Communications*, vol. 65, no. 1, pp. 403–430, 2016.

[186] S. Jayaprakasam, X. Ma, J. W. Choi, and S. Kim, "Robust beam-tracking for mmwave mobile communications," *IEEE Communications Letters*, vol. 21, no. 12, pp. 2654–2657, 2017.

[187] J.-S. Choi, W.-H. Lee, J.-H. Lee, J.-H. Lee, and S.-C. Kim, "Deep learning based nlos identification with commodity wlan devices," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3295–3303, 2017.

[188] N. H. Mahmood and H. Alves, "Dynamic multi-connectivity activation for ultra-reliable and low-latency communication," in *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2019, pp. 112–116.

[189] D. Deng, X. Wu, T. Zhang, X. Tang, H. Du, J. Kang, J. Liu, and D. Niyato, "Fedasa: A personalized federated learning with adaptive model aggregation for heterogeneous mobile edge computing," *IEEE Transactions on Mobile Computing*, pp. 1–15, 2024.

[190] A. Yazdinejad, A. Dehghantanha, H. Karimipour, G. Srivastava, and R. M. Parizi, "A robust privacy-preserving federated learning model against model poisoning attacks," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 6693–6708, 2024.

[191] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[192] G. Charan, A. Hredzak, and A. Alkhateeb, "Millimeter wave drones with cameras: Computer vision aided wireless beam prediction," *arXiv preprint arXiv:2211.07569*, 2022.

[193] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[194] A. Bluman, *Elementary Statistics: A step by step approach 9e*. McGraw Hill, 2014.

[195] 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.

[196] R. Nawaratne, D. Alahakoon, D. De Silva, and X. Yu, "Spatiotemporal anomaly detection using deep learning for real-time video surveillance," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 393–402, 2019.

[197] S. Hafeez, A. R. Khan, M. Al-Quraan, L. Mohjazi, A. Zoha, M. A. Imran, and Y. Sun, "Blockchain-assisted uav communication systems: A comprehensive survey," *IEEE Open Journal of Vehicular Technology*, 2023.

[198] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 351–360.

[199] Y. Wang, S. Duan, and F. Chen, "Efficient asynchronous federated neuromorphic learning of spiking neural networks," *Neurocomputing*, vol. 557, p. 126686, 2023.

[200] Y. Venkatesha, Y. Kim, L. Tassiulas, and P. Panda, "Federated learning with spiking neural networks," *IEEE Transactions on Signal Processing*, vol. 69, pp. 6183–6194, 2021.

[201] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, 2019.

[202] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[203] Y. Kim and P. Panda, "Optimizing deeper spiking neural networks for dynamic vision sensing," *Neural Networks*, vol. 144, pp. 686–698, 2021.

[204] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 51–63, 2019.

[205] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Frontiers in neuroscience*, vol. 12, p. 331, 2018.

[206] C. Lee, G. Srinivasan, P. Panda, and K. Roy, "Deep spiking convolutional neural network trained with unsupervised spike-timing-dependent plasticity," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 3, pp. 384–394, 2018.

[207] K. Xie, Z. Zhang, B. Li, J. Kang, D. Niyato, S. Xie, and Y. Wu, "Efficient federated learning with spike neural networks for traffic sign recognition," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9980–9992, 2022.

[208] S. A. Tumpa, S. Singh, M. F. F. Khan, M. T. Kandemir, V. Narayanan, and C. R. Das, "Federated learning with spiking neural networks in heterogeneous systems," in *2023 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2023, pp. 1–6.

[209] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz *et al.*, "A public domain dataset for human activity recognition using smartphones." in *Esann*, vol. 3, 2013, p. 3.

[210] T. Sztyler and H. Stuckenschmidt, "On-body localization of wearable devices: An investigation of position-aware activity recognition," in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2016, pp. 1–9.

[211] P. Fountas, K. Kolomvatsos, and C. Anagnostopoulos, "A deep learning model for data synopses management in pervasive computing applications," in *Intelligent Computing*, K. Arai, Ed. Cham: Springer International Publishing, 2021, pp. 619–636.

[212] Z. Lv and H. Song, "Mobile internet of things under data physical fusion technology," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4616–4624, 2020.

[213] M. Abdel-Basset, W. Ding, and L. Abdel-Fatah, "The fusion of internet of intelligent things (ioit) in remote diagnosis of obstructive sleep apnea: A survey and a new model," *Information Fusion*, pp. 84–100, 2020.

[214] Y. Wang, K. Wu, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 581–594, 2016.

[215] V. Bianchi, M. Bassoli, G. Lombardo, P. Fornacciari, M. Mordonini, and I. De Munari, "Iot wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8553–8562, 2019.

[216] E. Sannara, F. Portet, P. Lalanda, and V. German, "A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison," in *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2021, pp. 1–10.

[217] S. M. Bokhari, S. Sohaib, A. R. Khan, M. Shafi *et al.*, "Dgru based human activity recognition using channel state information," *Measurement*, vol. 167, p. 108245, 2021.

[218] A. R. Khan and S. Sohaib, "Cooperative noma, prototyping and experimental evaluation using sdr," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2872–2876, 2021.

[219] M. Weeks and M. Bayoumi, "Discrete wavelet transform: architectures, design and performance issues," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 35, no. 2, pp. 155–178, 2003.

[220] Q. Pan, L. Zhang, G. Dai, and H. Zhang, "Two denoising methods by wavelet transform," *IEEE transactions on signal processing*, vol. 47, no. 12, pp. 3401–3406, 1999.

[221] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," *Data classification: Algorithms and applications*, p. 37, 2014.

[222] A. R. Khan, H. U. Manzoor, F. Ayaz, M. A. Imran, and A. Zoha, "A privacy and energy-aware federated framework for human activity recognition," *Sensors*, vol. 23, no. 23, p. 9339, 2023.