



Jinsheng Yuan

Federated Learning: Contextual Efficiency Optimisation and
Adversary

Faculty of Engineering and Applied Sciences

Doctor of Philosophy
Academic Year: 2021–2024

Supervisors: Prof. Weisi Guo, Dr. Yang Xing
March 2025



Faculty of Engineering and Applied Sciences

Doctor of Philosophy

Academic Year: 2021–2024

Jinsheng Yuan

Federated Learning: Contextual Efficiency Optimisation and
Adversary

Supervisors: Prof. Weisi Guo, Dr. Yang Xing
March 2025

This thesis is submitted in partial fulfilment of the
requirements for the degree of Doctor of Philosophy.

© Cranfield University 2025. All rights reserved. No part of
this publication may be reproduced without the written
permission of the copyright owner.

Academic Integrity Declaration

I declare that:

- the thesis submitted has been written by me alone.
- the thesis submitted has not been previously submitted to this university or any other.
- that all content, including primary and/or secondary data, is true to the best of my knowledge.
- that all quotations and references have been duly acknowledged according to the requirements of academic research.

I understand that to knowingly submit work in violation of the above statement will be considered by examiners as academic misconduct.

Abstract

Artificial Intelligence (AI) has been widely studied, applied, and achieved impressive successes in various sectors. Among these successes, Federated Learning (FL) stands out as an innovative paradigm that enables distributed devices to collaboratively train machine learning models without transferring sensitive data to central servers. This decentralised paradigm, crucial for privacy preservation, also introduces significant practical challenges, particularly due to the heterogeneity among participating clients, which include variations in computational power, distinct data distributions, and diverse user behaviours, all of which complicate the training process in efficiency and convergence. Existing FL frameworks often apply a uniform learning configuration across all clients, neglecting individual device characteristics and user contexts, inevitably leading to inefficient utilisation of hardware resources and increased energy consumption within the federation. Moreover, optimisation techniques intended to exhaust computation potential of the system, can inadvertently expose systems to vulnerabilities, compromising security and reliability at both software and hardware levels.

This thesis addresses the critical gap of the contexts of FL including heterogeneous hardware specifications and optimisation technique employments, by proposing adaptive, context-aware methodologies that optimise efficiency, while analysing and mitigating associated security risks within federated learning environments. First, a Mixed-Precision Over-the-Air Federated Learning (MP-OTA-FL) framework is introduced, allowing clients to operate at different precision levels based on their hardware capabilities. Experimental results show that MP-OTA-FL significantly improves energy efficiency and task performance, particularly for resource-constrained devices. Second, a novel Retrieval-Augmented Generation (RAG)-based user profiling framework is proposed, dynamically optimising client precision decisions based on contextual factors such as hardware constraints and user preferences, achieving higher global model accuracy and user satisfaction. Finally, the thesis explores hardware vulnerabilities introduced by FL efficiency optimisations, presenting a remote Rowhammer attack vector exploiting physically induced perturbations on client devices, demonstrating practical security risks under realistic conditions. Collectively, these contributions provide comprehensive strategies for contextually optimised, secure, and efficient FL deployment, balancing performance and

security considerations. The findings offer foundational guidance for developing robust FL systems tailored to diverse and resource-limited real-world scenarios.

Keywords:

Mixed-Precision, Quantization, Over-the-Air, Rowhammer, RAG, Energy Efficiency, Hardware Security

Acknowledgements

*Though titles and findings may echo, each odyssey towards PhD is solitary,
no two vessels are borne by the same currents or weather the same storm.*

Leaving behind the familiar, this PhD journey has been profound, unique, transformative, and deeply personal. On this meaningful voyage, I was never alone; countless wonderful souls illuminated my path, lending strength and wisdom at every turning tide.

Vividly, I recall my childhood declaration that I'd become a scientist when I grew up. Even though I'm not sure I truly meant it back then, somehow, here I find myself at the culmination of an adventure I never fully anticipated, much like Bilbo reflecting on old prophecies with Gandalf upon their return journey.

Above all, heartfelt thanks to my supervisors, Prof. Weisi Guo, and Dr. Yang Xing. Vast was your guidance, your insights my stars, patiently steering me through uncertainties. Always, your mentorship shaped my research profoundly and influenced who I've become as a scholar and as a person.

To my family, you've been my anchor, the quiet strength holding me steady through rough seas. Your endless love, trust, and sacrifices gave me courage when waves crashed hardest. My gratitude for your unwavering belief in me is beyond words.

Friends and colleagues, you've brightened this voyage in countless ways. Always, your laughter, shared coffees, midday conversations, and after-work stories built a shelter of encouragement that I could return to, especially when storms of doubt approached. Thank you for believing in me, even in those moments when I found it hard to believe in myself.

Special appreciation goes to those who've directly contributed to this thesis: Dr. Zhuangkun Wei, whose collaboration was invaluable in shaping the mixed-precision OTA aggregation scheme; Dr. Yun Tang, whose creative partnership brought to life the website essential for my work on RAG-based User Profiling; and Yuhang Hao, whose engaging discussions illuminated the complexities of DRAM architectures and hardware testing for my exploration into Rowhammer Attacks via PPO-generated interference.

Odysseus might have stayed contentedly with Penelope, but there's a Bilbo living within me.

I think I'm quite ready for another adventure.

Contents

Academic Integrity Declaration	i
Abstract	ii
Acknowledgements	iv
Contents	v
List of Publications	ix
List of Figures	x
List of Tables	xi
List of Equations	xii
List of Abbreviations	xiii
1 Introduction	1
1.1 Background	1
1.1.1 Client Device and Data Heterogeneity	2
1.1.2 Operation Contexts	2
1.1.3 Efficiency Optimisation Techniques in Federated Learning	3
1.1.3.1 Software-based Optimisation Techniques	3
1.1.3.2 Hardware-based Optimisation Techniques	4
1.1.3.3 Communication-centric Optimisation Techniques	4
1.1.3.4 Potential Induced Vulnerabilities	4
1.2 Research Gaps	4
1.2.1 Underutilisation of Client Heterogeneity	4
1.2.2 Absence of Context-Aware Efficiency Strategies	5
1.2.3 Vulnerability Introduced by Efficiency Optimisations	6
1.2.3.1 Software-based Optimisation Vulnerabilities	6
1.2.3.2 Hardware-based Optimisation Vulnerabilities	7
1.3 Aims and Objectives	7
1.3.1 Aim and Research Questions	7
1.3.2 Objectives	7
1.4 Thesis Organisation	8

2	Literature Review	10
2.1	Client Heterogeneity	10
2.1.1	Statistical Heterogeneity	10
2.1.2	System Heterogeneity	11
2.1.3	Model Heterogeneity	11
2.2	Contribution Assessment and Incentive Mechanisms	12
2.2.1	Contribution Assessment	12
2.2.2	Incentive Mechanisms	12
2.3	Personalization and End-User Satisfaction	13
2.4	Efficiency Optimization Techniques	14
2.4.1	Communication Compression	15
2.4.2	Client Selection and Scheduling	15
2.4.3	Computation Efficiency Optimisation	16
2.5	Security and Privacy Vulnerabilities	17
2.5.1	Malicious Participation	18
2.5.2	Man-in-the-Middle Attacks	19
2.5.3	Evasion Attacks	20
2.6	Critical Reflections	20
3	Mixed-Precision Federated Learning via Multi-Precision Over-the-Air Aggregation	22
3.1	Introduction	22
3.2	System Setup	25
3.2.1	Federated Learning	25
3.2.2	Over-The-Air (OTA) Computation for FL	25
3.2.3	Approximate Computing and Low-Precision ML	26
3.3	Mixed-Precision OTA Federated Learning	27
3.3.1	Multi-Precision Over-The-Air Aggregation	27
3.3.1.1	Channel Estimation at Clients	29
3.3.1.2	Uplink Design	29
3.3.1.3	Downlink Design	29
3.3.2	Approximate Computing via Quantization	30
3.3.3	Energy Consumption Estimation	30
3.4	Experiments	31
3.4.1	Settings	31
3.4.1.1	Data	31
3.4.1.2	Quantization Schemes	31
3.4.1.3	Performance Metrics	32
3.4.1.4	Energy Efficiency	32
3.4.2	Results	32
3.4.2.1	Energy Efficiency Estimation	32
3.4.2.2	Federated Training and Server Performance	32
3.4.2.3	Client Performance	34
3.5	Conclusion	35

4	RAG-based User Profiling for Precision Planning in Mixed-precision Over-the-Air Federated Learning	36
4.1	Introduction	36
4.2	Related Works	38
4.2.1	Mixed-Precision Federated Learning	38
4.2.2	RAG-LLMs	38
4.3	Methodology	39
4.3.1	Framework Overview	39
4.3.2	RAG-based User Profiling	43
4.3.2.1	Contextual Factors	43
4.3.2.2	RAG Database and LLM Integration	43
4.3.2.3	User Profiling Pipeline	43
4.3.3	Context-Quantization Evaluation	44
4.4	Experiments	45
4.4.1	Experimental Setup	45
4.4.2	Results	46
4.4.2.1	User Satisfaction versus Energy Cost	46
4.4.2.2	Global Model Performance	47
4.5	Conclusion	47
5	Rowhammer Attack via PPO Generated Physical Interference on Federated Learning Clients	48
5.1	Introduction	48
5.2	Background and Related Work	50
5.2.1	DRAM Architecture and Rowhammer Attack	50
5.2.2	Federated Learning	52
5.2.2.1	Security Threats in Federated Learning	53
5.2.2.2	Efficiency Optimization	54
5.2.3	Physical Adversarial Attacks	54
5.3	Attack Vector Formulation	55
5.3.1	Inspiration	55
5.3.2	Evaluation Metrics	57
5.3.2.1	Notations	57
5.3.2.2	Reinforcement Learning Performance Metrics	58
5.3.2.3	Bit-Flip Validation	58
5.3.3	Target System	59
5.3.3.1	Model Structure	59
5.3.3.2	Dataset	59
5.3.3.3	Server Setups	59
5.3.3.4	Clients Setups	59
5.3.4	RL Agent Design	60
5.3.4.1	Observation and Action Spaces	60
5.3.4.2	Reward Formulation	60
5.3.4.3	Policy Network Architecture	61
5.3.4.4	Training Protocol	61
5.4	Experiments	61

5.4.1	Experimental Setups	61
5.4.2	Rowhammer Test Bench	64
5.4.3	PPO Agent Training	64
5.4.3.1	Experimental Setup	64
5.4.3.2	Baseline Methods	65
5.4.3.3	Evaluation Metrics	65
5.4.3.4	Results and Analysis	65
5.4.4	Bit-Flip Validation	66
5.4.4.1	Estimated Row Activation Counts	67
5.4.4.2	Estimated Bit Flips	67
5.5	Discussion	68
5.5.1	Practical Implications and Feasibility	68
5.5.2	Limitations and Open Challenges	68
5.5.3	Security Considerations for Federated Learning Systems	69
5.5.4	Potential Mitigation Strategies	69
5.5.5	Broader Impact and Future Directions	69
5.6	Conclusion	70
6	Discussion	71
6.1	Overview of Thesis Contributions	71
6.2	Leveraging Client Heterogeneity in FL	71
6.3	Context-Aware Optimisation via User Profiling	72
6.4	Emergent Vulnerabilities from Efficiency Optimisation	72
6.5	Integration and Real-world Implications	72
6.6	Limitations	73
6.7	Summary	73
7	Conclusion and Further Work	74
7.1	Summary of Research Outcomes	74
7.2	Contributions to Knowledge	75
7.3	Practical Implications	75
7.4	Further Work	75
7.5	Closing Remarks	76

List of Publications

- [1] K.-F. Chu, H. Yuan, J. Yuan, W. Guo, N. Balta-Ozkan and S. Li, ‘A survey of artificial intelligence-related cybersecurity risks and countermeasures in mobility-as-a-service,’ *IEEE Intelligent Transportation Systems Magazine*, vol. 16, no. 6, pp. 37–55, 2024. DOI: 10.1109/MITS.2024.3427655.
- [2] J. Yuan, Z. Wei and W. Guo, ‘Mixed-precision federated learning via multi-precision over-the-air aggregation,’ in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, 2025, pp. 1–7. DOI: 10.1109/WCNC61545.2025.10978511.
- [3] J. Yuan, Y. Tang and W. Guo, *Rag-based user profiling for precision planning in mixed-precision over-the-air federated learning*, 2025. arXiv: 2503.15569 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2503.15569>.
- [4] J. Yuan, Y. Hao, W. Guo, Y. Wu and C. Gu, *Remote rowhammer attack using adversarial observations on federated learning clients*, 2025. arXiv: 2505.06335 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2505.06335>.

List of Figures

2.1	Example of Attack Vectors with Federated AI MaaS System	17
3.1	Evolution of Federated Learning Systems Towards MP-OTA-FL	23
3.2	Architecture of MP-OTA-FL for Intelligent Transport Applications	24
3.3	Global Model Training Accuracy of MP-OTA-FL	33
3.4	Accuracy-Energy Trade-offs of Clients with MP-OTA-FL	34
4.1	Influence of Contextual Factors on User Satisfaction and Contribution in Federated Learning	38
4.2	Overview of the User-in-the-Loop Quantization Planning Framework . . .	41
4.3	Comparison of Personalised and Unified Quantization Standards: User Satisfaction versus Energy Efficiency	46
5.1	PPO Rowhammer Attack Vector	49
5.2	DRAM Organization	50
5.3	DRAM access pathways and processes of FL	56

List of Tables

3.1	GTSRB Classification Performance of Common Models across Quantization Levels	27
3.2	Estimated Energy Consumption per Sample for ResNet-50 Forward Pass and Relative Savings Compared to 32-bit	32
4.1	Examples of Contextual Factors and Inferable Factors	42
4.2	Word Accuracy of the Global Model after 100 Communication Rounds by Classes with Different Strategies.	42
4.3	Smart Voice Assistant Data Distribution	45
5.1	Common ASR models with their sizes and computation precision.	63
5.2	Rowhammer Attack Thresholds under Different Patterns	63
5.3	Repeated Tensor Overlap Rate (RUR, %) across all models and sparsity levels.	65
5.4	Cluster Density (CD, %) across all models and sparsity levels.	66
5.5	Estimated Row Activations Based on Empirical Clustering (Top 0.05% Sparsity Level)	67

List of Equations

1	FedAvg Aggregation	25
2	OTA Aggregation Process	26
3	Non-Commutative Property of Quantised Modulations	26
4	OTA Modulation Function	28
5	Channel Estimation Function between Client and Server	29
6	Base Band Transmission Function	29
7	Client Received Signal	29
8	Aggregated Model Recover Function	30
9	FPGA Energy Cost Estimation Function	31
10	Total Reward Calculation for Context-Quantization Evaluation	44
11	Total Penalty Calculation for Context-Quantization Evaluation	44
12	Satisfaction Score Calculation for Context-Quantization Evaluation	45
13	Optimal Quantization Level	45
14	Repeated Update Rate (RUR)	58
15	Cluster Density (CD)	58
16	Maximum number of updates	59
17	Expectation of activations	59
18	Reward function of PPO agent	60
19	Perceptibility Penalty	61
20	DRAM bandwidth	62

List of Abbreviations

ACT	Activation Command
AI	Artificial Intelligence
ASR	Automatic Speech Recognition
AxC	Approximate Computing
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DMA	Direct Memory Access
DP	Differential Privacy
DQN	Deep Q-Network
DRAM	Dynamic Random Access Memory
DRL	Deep Reinforcement Learning
DSP	Digital Signal Processor
ECC	Error-Correcting Code
EMD	Earth Mover's Distance
FL	Federated Learning
FPGA	Field Programmable Gate Array
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
GTSRB	German Traffic Sign Recognition Benchmark
HMM	Hidden Markov Model
HPC	High Performance Computing
IoT	Internet of Things
LLM	Large Language Model
MaaS	Mobility-as-a-Service
MAC	Multiply-Accumulate
MC	Memory Controller
MDP	Markov Decision Process
ML	Machine Learning
MMU	Memory Management Unit
MP-OTA-FL	Mixed-Precision Over-the-Air Federated Learning
NLP	Natural Language Processing
OTA-FL	Over-the-Air Federated Learning
PPO	Proximal Policy Optimization
PRE	Precharge Command
QAM	Quadrature Amplitude Modulation
RAG	Retrieval-Augmented Generation
RDMA	Remote Direct Memory Access

REF	Refresh Command
RL	Reinforcement Learning
SISO	Single-Input Single-Output
TRR	Target Row Refresh
TRPO	Trust Region Policy Optimization

Chapter 1

Introduction

1.1 Background

The rapid advancement and remarkable success of Artificial Intelligence (AI) over the past decade, from image classification [1] to Large Language Models (LLM) [2], have been primarily driven by two key factors: significant leaps in computational hardware performance and the continuous generation of vast amounts of data across all sectors of society, both of which, follow well-established scaling laws. Compute power continues to grow exponentially in accordance with trends like Moore’s Law, while the volume and diversity of data expand at an accelerating pace, enabling increasingly large models to achieve superior performance across a wide range of tasks. However, as data are increasingly produced at the edge, in smartphones, IoT devices, and other distributed infrastructures, centralised data aggregation is becoming both impractical and often undesirable due to privacy, bandwidth, and latency constraints, leading to the emergence of new learning paradigms capable of leveraging decentralised data such as Federated Learning (FL) [3]. FL enables multiple clients to collaboratively train a global model without the need to exchange their private data, by transmitting only model updates to a central server for aggregation. This decentralised approach not only addresses critical privacy concerns but also leverages the computational capabilities of edge devices, thereby reducing reliance on centralised data processing infrastructures. FL’s inherent privacy preservation aligns closely with increasing regulatory demands, such as the General Data Protection Regulation (GDPR) [4], which impose stringent restrictions on data sharing and aggregation practices. Consequently, FL has become highly relevant in sectors like healthcare, finance, and smart cities, where data privacy and governance are paramount. This distributed approach significantly reduces communication overhead and latency, facilitating real-time or near-real-time learning tasks.

Despite these promising attributes, the practical implementation of FL faces significant challenges, primarily stemming from diverse user behaviour patterns, as well as the heterogeneity inherent in the rapid hardware evolution, where AI systems are configured with diverse processors, accelerators, memory, storage, and networking capabilities. Such client heterogeneity poses severe obstacles to achieving optimal performance, fairness, and efficiency in FL systems [5].

1.1.1 Client Device and Data Heterogeneity

Practical Federated Learning (FL) deployments exhibit considerable heterogeneity among participating client devices, significantly influencing FL algorithm design and system performance. Client devices such as high-end smartphones, IoT sensors, and legacy hardware demonstrate considerable variations in computational power, memory capacity, battery life, and network connectivity [6], [7]. Traditional FL algorithms frequently adopt homogeneous assumptions, leading to inefficient resource utilisation. This may result in excessive computational burdens on resource-limited devices and underutilisation of more powerful clients, causing prolonged training times, degraded global model performance, or increased dropout rates [7], [8].

Moreover, simplistic client selection strategies, commonly random or uniform participation, fail to account for individual device characteristics. Such approaches neglect significant opportunities for optimisation, including context-aware client selection based on computational capacity, battery status, or network reliability. Recent studies suggest that adaptive client selection mechanisms leveraging contextual client attributes can substantially enhance FL efficiency and fairness [9]. However, comprehensive integration and scalability assessments of these methods within realistic, heterogeneous federated environments remain limited.

Beyond device-level heterogeneity, the diversity in client data—often termed statistical heterogeneity—poses considerable challenges for FL training. Variations in data distribution, volume, and quality across client devices can significantly impact model convergence, accuracy, and fairness. Such heterogeneity frequently results in biased global models, predominantly favouring client subsets with abundant or high-quality data while marginalising others [10]. Techniques such as personalised federated learning, hierarchical model aggregation, and transfer learning have been introduced to mitigate these challenges, yet practical implementations, scalability assessments, and evaluations within heterogeneous real-world contexts require further exploration [11].

Additionally, hierarchical and clustered federated learning structures, grouping clients based on shared characteristics or complementary capabilities, represent a promising yet underexplored direction. These structures could enable tailored model aggregation strategies, optimising resource allocation and enhancing computational efficiency at subgroup levels. Systematic investigation into the implementation, scalability, and effectiveness of hierarchical federated learning approaches within heterogeneous device networks is a critical gap in the existing literature [12].

1.1.2 Operation Contexts

FL deployment contexts involve multiple operational considerations, including computational efficiency, resource constraints, and user behaviour. Resource constraints, particularly prevalent among edge devices, significantly complicate FL participation. Limited battery life, restricted computational resources, and constrained storage capacities frequently hinder device engagement in resource-intensive training and communication tasks, leading to diminished user experience and increased device attrition from FL networks [7]. Addressing these constraints necessitates context-aware computational strategies capable of dynamically adapting model complexity and training intensity to individual device contexts.

Energy-efficient FL strategies have predominantly focused on simple client-level optimisations, such as energy-aware client selection or static model adjustments. However, sophisticated dynamic approaches—such as multi-objective optimisation or reinforcement learning strategies—capable of balancing complex trade-offs among accuracy, computational efficiency, and energy consumption remain underexplored in heterogeneous federated settings [13]. These advanced techniques offer significant potential for comprehensive solutions to resource constraints while ensuring high global accuracy and fairness.

Computation offloading represents another critical operational context within FL, offering potential resource relief to constrained devices by leveraging nearby edge or cloud infrastructures [6]. While computation offloading is well-established within edge computing, its integration into FL workflows with considerations for heterogeneity, privacy preservation, and adaptive decision-making remains insufficiently explored. Robust methodologies for secure and adaptive offloading, addressing privacy risks, communication overheads, and latency implications, constitute an essential area for further research.

Additionally, user behaviour and preferences introduce complexity into FL systems. Individual usage patterns, data availability, and user engagement significantly affect data quality, quantity, and representativeness. Active users may contribute disproportionately larger and higher-quality datasets, causing inherent imbalances within federations. Variability in data quality due to differing device settings or user interaction levels further exacerbates disparities, impacting model accuracy and fairness [14].

User privacy preferences critically impact participation rates and data contributions. Users concerned about privacy may restrict participation or impose stringent privacy conditions such as differential privacy, influencing the diversity and robustness of training data. Thus, sophisticated privacy-preserving methods like secure multiparty computation, homomorphic encryption, federated distillation, and tailored differential privacy implementations are required, aligned with varied user expectations and regulatory compliance [4], [15].

Addressing these multifaceted operational challenges demands innovative, adaptive strategies integrating computational efficiency with user sensitivities. Developing context-aware optimisation and scheduling tailored explicitly to nuanced user attributes and device constraints is essential for effective, fair, and robust FL outcomes.

1.1.3 Efficiency Optimisation Techniques in Federated Learning

Efficiency optimisation in Federated Learning encompasses software, hardware, and communication-centric techniques designed to minimise computational and communication overhead, crucial for enabling FL in resource-constrained environments.

1.1.3.1 Software-based Optimisation Techniques

Software methods include model pruning, quantisation, sparse updates, and precision management. Periodic averaging combined with quantised communications, such as FedPAQ, significantly enhances communication efficiency by reducing message sizes and frequencies [16]. Layer-wise pruning and quantisation (FedLP-Q, FedLP) systematically reduce model parameters and computational demands while preserving accuracy [17].

FedPaI further advances pruning by achieving extreme sparsity at model initialisation, accelerating convergence [18].

Quantisation strategies have evolved notably, with ultra-low-bit quantisation frameworks like FedQNN substantially decreasing computational demands [19]. Bit-freezing techniques adaptively manage computational precision during training, reducing errors from quantisation. Studies indicate that low-precision training, such as 8-bit precision, can match full-precision performance with robust aggregation [20]. Mixed-precision quantisation (FedMPQ) assigns precision levels tailored to device capabilities, optimising efficiency and accuracy trade-offs [21].

1.1.3.2 Hardware-based Optimisation Techniques

Hardware optimisations leverage specialised processors, Remote Direct Memory Access (RDMA), and page-locked memory to enhance computational and communication efficiency. Dedicated hardware accelerators and clustered edge computing architectures, exemplified by MEC-AI HetFL, dynamically offload computation and manage asynchronous updates, improving participation from resource-constrained devices [22].

1.1.3.3 Communication-centric Optimisation Techniques

Communication optimisation techniques like FedSZ integrate lossy and lossless coding schemes to substantially reduce communication payload sizes without compromising model integrity [23]. Message quantisation and streaming methods, as implemented in frameworks such as NVIDIA FLARE, address bandwidth bottlenecks, enabling scalable federated learning deployments [24].

1.1.3.4 Potential Induced Vulnerabilities

Efficiency optimisation techniques may inadvertently introduce vulnerabilities such as reduced robustness to adversarial attacks, biases from mixed-precision aggregations, and increased susceptibility to communication errors. Addressing these vulnerabilities is vital for secure and resilient federated learning.

1.2 Research Gaps

While Federated Learning (FL) presents a compelling framework for collaborative learning with privacy preservation, several critical research gaps hinder its widespread and effective deployment, particularly concerning client heterogeneity and the pursuit of computational efficiency. These gaps span the effective utilisation of device diversity, nuanced exploration of optimisation trade-offs, context-aware computational strategies, and robustness considerations arising from efficiency-driven optimisations.

1.2.1 Underutilisation of Client Heterogeneity

Current FL research often adopts simplified assumptions, treating the federated environment as homogeneous. This perspective overlooks the extensive variability found in

real-world deployments, such as differences in computational power, memory capacity, battery life, and network connectivity among diverse client devices [25], [7]. Such simplification leads to suboptimal resource allocation, often overburdening resource-limited clients while simultaneously underutilising more capable devices, resulting in prolonged training periods, reduced global model performance, and increased dropout rates [8].

Moreover, current FL protocols often select clients based on simplistic criteria, such as random selection or uniform participation, which fail to account for individual device characteristics and capabilities. Recent efforts in context-aware client selection have demonstrated preliminary benefits, suggesting that informed and adaptive client selection methods, such as those considering available computational resources, energy status, or network conditions, can substantially improve training efficiency and fairness [9]. However, systematic investigation into integrating these methods into scalable FL architectures remains limited.

Additionally, existing FL solutions seldom explore hierarchical or clustered approaches extensively. Hierarchical FL, where clients are grouped based on similarity or complementary characteristics, may allow better resource allocation and improved computational efficiency by enabling tailored training protocols at subgroup levels [12]. Nonetheless, the practical implementation, scalability, and effectiveness of hierarchical approaches in real-world heterogeneous settings are still inadequately studied.

The heterogeneous nature of FL client devices presents complex and interrelated trade-offs among model accuracy, computational load, energy consumption, and communication overheads [26]. Current FL strategies often lack sophisticated mechanisms to dynamically balance these conflicting criteria in a client-specific or context-aware manner. While previous works have introduced partial solutions, such as adaptive model pruning, compression, or asynchronous update schemes to systematically explore and manage these multifaceted trade-offs remain scarce.

Furthermore, the interactions among trade-offs can be highly non-linear and context-dependent. For instance, improving computational efficiency by aggressive model compression might disproportionately degrade accuracy on devices with limited local data, while minimally impacting those with abundant high-quality data [13]. Consequently, advanced optimisation methods, such as multi-objective optimisation, reinforcement learning, and context-aware Bayesian optimisation, offer promising avenues for managing such complex interactions but have not been adequately explored or integrated within FL frameworks.

1.2.2 Absence of Context-Aware Efficiency Strategies

While techniques such as mixed-precision computing and dynamic quantisation have demonstrated significant success in centralised machine learning settings by effectively balancing computational efficiency and model accuracy [27], their adaptation to decentralised and heterogeneous FL scenarios is significantly underexplored. Context-aware computational strategies, such as adaptive precision allocation based on device-specific resource availability, could deliver substantial computational and energy savings without severely impacting model performance.

Additionally, dynamic computation offloading, leveraging nearby edge or cloud infrastructure to complement resource-constrained clients during intensive training tasks,

represents another promising but inadequately explored strategy in the context of FL. Although computation offloading is well-established in edge computing literature, its integration into FL workflows, particularly in a context-aware and privacy-preserving manner, remains largely uninvestigated [6]. Establishing methodologies for secure and adaptive offloading decisions, including assessing privacy risks, communication overheads, and latency implications, constitutes a crucial research gap that needs to be addressed.

Another critical yet underexplored aspect of context-aware efficiency involves accurately assessing individual client contributions. Client contributions in FL environments are challenging to quantify due to diverse client contexts, heterogeneous data quality, quantity, and device performance. Accurate evaluation and reward mechanisms based on client contributions could enhance participation incentives, fairness, and overall model performance. However, current methodologies lack robust mechanisms for reliably assessing and leveraging client-specific contributions within resource-constrained and heterogeneous settings.

1.2.3 Vulnerability Introduced by Efficiency Optimisations

Efficiency-driven optimisations in Federated Learning (FL), while crucial for managing resource constraints, can inadvertently introduce vulnerabilities that affect both model performance and system security. Such optimisations generally fall into two broad categories: software-based and hardware-based. Each category presents distinct vulnerabilities that remain insufficiently studied in current FL literature.

1.2.3.1 Software-based Optimisation Vulnerabilities

Software-based optimisations in FL primarily involve algorithmic techniques aimed at reducing computational and communication overhead. Common examples include aggressive model compression, gradient quantisation, sparsification methods, and asynchronous update strategies [26], [28]. While these approaches successfully reduce resource demands, they also introduce several potential vulnerabilities.

Aggressive model compression or quantisation, for example, reduces model redundancy and precision, thereby potentially decreasing the robustness of the global model against noisy or malicious updates. Such compressed or sparse model representations may be more susceptible to adversarial manipulations, including backdoor or poisoning attacks [29]. Similarly, gradient sparsification and asynchronous communication, while minimising communication overhead, can amplify statistical heterogeneity and delay convergence, increasing the system's susceptibility to straggler effects and exacerbating risks from unreliable or adversarially compromised clients [28].

Moreover, software-based efficiency measures that dynamically adapt client participation or training intensity based on resource availability or communication reliability can inadvertently create opportunities for malicious actors. For instance, adversaries could strategically manipulate local resource states or network conditions to evade detection by adaptive anomaly detection mechanisms or to exploit efficiency-driven adaptive training protocols [14]. Comprehensive research into robust aggregation techniques, secure adaptive scheduling, and resilient anomaly detection mechanisms specifically tailored to software-based efficiency optimisations remains a largely unaddressed area.

1.2.3.2 Hardware-based Optimisation Vulnerabilities

Hardware-based optimisations leverage advancements at the hardware level, such as page-lock memory (pinned memory), Direct Memory Access (DMA). Although these strategies can improve efficiency to varying levels, they similarly expose FL systems to potential vulnerabilities that have received minimal attention in current research.

For instance, memory optimisation techniques, such as page-lock memory, which significantly improve efficiency in both training and inferencing for deep learning, could also serve as a guide for external aggressors on DRAM related attacks. Additionally, techniques like DMA/RDMA, improve memory operation efficiency by enabling internal or external devices to bypass the CPU and directly access memory, also degrade the security level of the FL application.

Addressing these vulnerabilities requires integrated research efforts that jointly consider software-based and hardware-based optimisations. Developing comprehensive frameworks that include robust aggregation, secure adaptive mechanisms, and hardware-aware protections represents a critical step forward in achieving secure and efficient FL deployments in heterogeneous and resource-constrained environments.

1.3 Aims and Objectives

1.3.1 Aim and Research Questions

The central aim of this thesis is to understand how contextual factors in heterogeneous Federated Learning (FL) environments affect computational efficiency, optimisation trade-offs, and system security. Specifically, the thesis seeks to address the following research questions:

1. What are the contextual factors and their corresponding effects in federated learning systems?
2. How to utilise the heterogeneity of clients in federated learning for better efficiency or task accuracy?
3. How to effectively and dynamically assess and integrate individual client contributions and user preferences into federated learning of clients with diverse contexts?
4. How do efficiency optimisation techniques influence the robustness of federated learning systems, and what new adversarial vulnerabilities arise from software and hardware optimisation techniques?

1.3.2 Objectives

To address the above research questions, the specific objectives designed are:

1. **Literature Review of Contextual Factors in Federated Learning:** Conduct a comprehensive literature review to identify contextual factors in federated learning, including client heterogeneity, contribution assessment, efficiency optimisation techniques and security vulnerabilities, highlighting existing research gaps.

2. **Mixed-Precision Federated Learning:** Develop a Mixed-Precision Over-the-Air Federated Learning (MP-OTA-FL) framework enabling clients to participate in federated learning while operating at multiple computation precision, and evaluate efficiency and accuracy trade-offs.
3. **RAG-based User Profiling for Contextual Optimisation:** Develop and validate Retrieval-Augmented Generation (RAG)-based user profiling methods, leveraging contextual client data to improve precision planning, optimise resource utilisation, and improve overall federated learning model performance.
4. **Analysis of Hardware-based Vulnerabilities:** Investigate both software and hardware-based efficiency optimisation techniques, including page-lock memory, huge pages, and RDMA, assessing their potential exploitation through novel Rowhammer attack scenarios and evaluating associated security risks.

1.4 Thesis Organisation

This thesis systematically explores contextual efficiency optimisation strategies in Federated Learning and analyses related adversarial vulnerabilities. It presents comprehensive solutions that integrate efficiency and security considerations within heterogeneous client scenarios. The thesis is organised based on publications and the structure is as follows:

Chapter 1: Introduction presents the foundational concepts of Federated Learning, outlines motivations for context-aware efficiency optimisation, identifies the interplay with adversarial vulnerabilities, and specifies the research aims, objectives, and structure.

Chapter 2: Literature Review of Contextual Factors in Federated Learning a comprehensive literature review to identify contextual factors in federated learning, including client heterogeneity, contribution assessment, efficiency optimisation techniques, and security vulnerabilities. Part of the content of this chapter was revised and contributed as a section in publication [1], as listed in the List of Publications.

Chapter 3: Mixed-Precision Federated Learning via Multi-Precision Over-the-Air Aggregation directly addresses contextual efficiency optimisation by introducing the Mixed-Precision Over-the-Air Federated Learning (MP-OTA-FL) framework. This methodology enables clients of heterogeneous capabilities to operate at appropriate computation precision through multi-precision aggregation to enhance their trade-offs between efficiency and accuracy, especially for those with most limited resources. This chapter was revised into an independent manuscript and published in IEEE Wireless Communications and Networking Conference, as listed as listed in the List of Publications.

Chapter 4: RAG-based User Profiling for Precision Planning in Mixed-Precision Over-the-Air Federated Learning explores the integration of Retrieval-Augmented Generation (RAG)-based user profiling techniques to infer contextual information such as user behaviour, resource sensitivity, and potential contribution. This approach enables fine-grained optimisation of computational precision for better server model performance and client efficiency and training contribution, extending the previously proposed MP-OTA-FL framework. This chapter was revised into an independent manuscript and accepted for publication in IEEE Vehicular Technology Conference, with pre-print on ArXiv, as listed as listed in the List of Publications.

Chapter 5: Rowhammer Attack via Physical Interference on Federated Learning Clients examines how hardware-based efficiency optimisations, such as page-lock memory, huge pages, and RDMA, can facilitate potential attacks, introduces a novel RL based threat vector targeting the server hardware via inducing Rowhammer attacks. This chapter was revised into an independent manuscript and submitted to ArXiv as a pre-print, as listed in the List of Publications.

Chapter 6: Discussion synthesises findings from previous chapters, offering integrated frameworks for contextually optimised efficiency, and optimisation related adversarial vulnerabilities. It presents a comprehensive discussion on interpretation, contribution, and limitation of the findings.

Chapter 7: Conclusion and Future Work summarises the key contributions, evaluates how objectives were met, and outlines future research directions.

Chapter 2

Literature Review

Federated Learning (FL) is a distributed machine learning paradigm in which a central server coordinates many clients, such as mobile or IoT devices, to jointly train a model without sharing raw data [3]. In contrast to traditional centralised learning, FL keeps data decentralised for privacy, but introduces new challenges due to the diverse contexts in which clients operate. FL has gained tremendous attention for its potential in large-scale, privacy-preserving applications [14]. However, the effectiveness of FL depends on various contextual factors inherent to real-world settings. Key factors include client heterogeneity in terms of differences in data distributions and device capabilities across clients, contribution assessment methods to evaluate and incentivise participants' contributions, end-user satisfaction in meeting individual users' needs and fairness expectations, efficiency optimization techniques to reduce communication and computation overheads, and security vulnerabilities from malicious participants or external attackers. In this chapter, I review the state-of-the-art research on each of these aspects, focusing primarily on recent advances, while acknowledging earlier milestone works, to illustrate how these factors impact federated learning and corresponding research gaps.

2.1 Client Heterogeneity

2.1.1 Statistical Heterogeneity

One of the defining challenges in FL is heterogeneity among participating clients. Unlike controlled data centre environments, federated clients may have vastly different data and system characteristics. Statistical heterogeneity refers to the scenario where clients' local datasets are not identically distributed (non-IID) and may be imbalanced in size. This is the norm in FL, where users generate personal data with unique distributions, leading to divergent local model updates and degraded global model performance if not properly handled. In fact, federated averaging (FedAvg) [3], the seminal FL algorithm based on iterative model averaging, was shown to be robust to mild non-IID data in some cases, but in highly skewed data scenarios the convergence can stall or the model may favour dominant data sources. Early studies identified that naively minimizing the average loss can disproportionately sacrifice performance on some clients [30]. Numerous algorithms have been proposed to tackle statistical heterogeneity. For example, FedProx [5] introduced a proximal term in clients' objective functions to keep local model updates closer

to the global model, thereby stabilizing training on heterogeneous data. FedProx generalises FedAvg and demonstrated significantly more robust convergence under both non-IID data and variable compute budgets, improving accuracy by reducing the “drift” of local models [5]. Other methods like SCAFFOLD [31] use control variates to correct for client-drift in stochastic gradients, and have theoretical convergence guarantees even when client data distributions diverge.

2.1.2 System Heterogeneity

Client heterogeneity is not only about data, there is also system heterogeneity. Clients differ in computational power, memory, network connectivity, and energy capacity. Some devices may train faster while others lag behind or occasionally drop out. FL in cross-device settings often involves hundreds or millions of ephemeral clients on slow or unreliable network connections, where communication is often the primary bottleneck [14]. This variability makes synchronous aggregation challenging: if the server waits for all devices, training slows; if it proceeds with a subset, accuracy may be impacted by bias. Practical FL systems address this by selecting only a fraction of clients per round and often by dropping or bounding stragglers’ effect, that is to wait for a time limit then proceeding with available updates. Recent research explores fully asynchronous FL to avoid idle waiting, where clients send updates on their own schedule and the server updates the model iteratively upon receiving each or a batch of updates. Asynchronous methods can mitigate stragglers but need to handle stale updates and convergence stability. Another angle is to cluster clients by hardware capability or geography such as hierarchical FL, assigning those with better connectivity to more frequent communication, or using intermediate aggregation servers to accommodate slower clients [32].

2.1.3 Model Heterogeneity

Another important subclass of heterogeneity is model heterogeneity. In some applications, different clients may require or employ different model architectures. For instance, edge devices with limited resources might use a smaller model than the global model, or different organizations might train models for related but not identical tasks. Traditional FL assumes a common model architecture for aggregation to make sense, so heterogeneity in models demands new approaches. One solution is knowledge transfer techniques, such as federated distillation, rather than averaging model parameters layer by layer, clients distill their knowledge to a common format that can be aggregated [33]. Another approach is to train a multi-model system, where a global representation is shared but each client has some personalised layers. In cross-silo FL with fewer but more powerful organizational clients, sometimes parties agree on a subset of model layers to share while keeping others private, addressing architectural differences.

Handling heterogeneity is crucial for successful federation at scale. Embracing heterogeneity can even be beneficial, by allowing clients to train locally suitable models and share only parts of them, FL can become more flexible and achieve higher overall performance [34]. However, achieving effective collaboration in the presence of heterogeneous conditions remains a central research challenge.

2.2 Contribution Assessment and Incentive Mechanisms

2.2.1 Contribution Assessment

In federated networks with many self-interested parties, it is important to understand and fairly measure each client’s contribution to the global model. Contribution assessment methods aim to quantify how much a given client’s data or effort has improved the model. This has several purposes: (1) to incentivise clients via rewards or payments proportional to their contributions, (2) to identify and mitigate “free riders”, those participants who submit useless or low-quality updates but still wish to benefit from the final model, and (3) to ensure fairness and trust in the collaborative process [35]. A straightforward idea is to evaluate the global model’s performance with and without a particular client’s updates, but doing this exhaustively for all combinations of clients is combinatorially expensive. Researchers have thus turned to cooperative game theory concepts, notably the Shapley value, which provides a principled way to attribute a value to each player (client) based on marginal contributions [36]. In the FL context, the Shapley value of a client is computed by averaging its marginal improvement to the model over all subsets of other clients. Direct Shapley value calculation is infeasible for large numbers of clients due to its exponential computation complexity, recent works move towards approximated Shapley value calculation, such as GTG-Shapley [32], which uses gradient reconstruction to evaluate a client’s influence without retraining from scratch for each subset.

Another line of work focuses on detecting free-riders and malicious contributors. Free-riders may attempt to download the global model each round without actually expending local computation or may submit random updates. Methods to spot such behaviour include verifying sample updates or using peer monitoring in decentralised FL. Some approaches require clients to provide proof-of-work or proof-of-learning, such as a history-aware contribution tracking framework by [35], which flags suspicious clients by analysing their long-term contribution patterns across rounds, where a sudden drop or consistently zero contribution might indicate a free-rider or malfunction, and ultimately, establishing a robust reputation system, where clients build trust over time by consistent contributions, is an important future direction for federated systems.

2.2.2 Incentive Mechanisms

Alongside measuring contributions, incentive mechanisms are designed to encourage clients to participate actively and honestly in FL. Many incentive schemes model the interaction between the server and clients as a game or economic market. Some approaches use reward pools divided according to contribution scores such as Shapley values, so that clients with more valuable data or better effort are compensated more, fostering a virtuous cycle of participation. For instance, Kang et al [37] combine reputation systems with contract theory, where the server offers contracts with rewards to clients for a certain quality of contribution, and clients with higher reputation based on contribution history get better terms, which in turn incentivise honest effort. Yu et al [38] introduce a fairness-aware incentive scheme that dynamically adjusts rewards to balance not only the total contribution but also issues of equity among clients, ensuring that even clients with less data or slower devices have incentive to join without feeling exploited. Hierarchical incentive mechanisms have also been studied for multi-tier FL, in which mobile devices are

grouped under edge servers, where incentives are distributed at both the edge level and the cloud level [6].

A critical aspect is ensuring the incentive mechanism itself does not distort the learning. If clients are rewarded solely on immediate accuracy contribution, they might manipulate their behaviour or data to game the metric. Thus, robust contribution assessment often goes hand-in-hand with security and fairness, to detect false or low-quality contributions, as well as to ensure minority or lower-resource clients are not always undervalued. Recent research highlights the need for "truthful" mechanisms that encourage clients to report honestly about their capabilities and follow the protocol by designing rewards such that the best strategy for each client is to act truthfully [37], [38].

2.3 Personalization and End-User Satisfaction

Federated learning traditionally produces a single global model intended to serve all clients. However, given the diversity of users and data, a one-size-fits-all model may not be optimal for many participants. End-user satisfaction in FL refers to how well the outcome of the learning process meets the needs of each client or user. If some users experience poor model performance, such as a language model that fails to adapt to a user's dialect or a recommendation system that ignores a minority preference, these users will be dissatisfied and discouraged to continue participating. This concern has driven research in personalised federated learning and fairness in FL.

Personalization techniques allow each client to obtain a model that is tuned to its own data or preferences while still benefiting from knowledge aggregated from others. A simple approach is local fine-tuning. After the global model is learned, each client can continue training on its own data for a few epochs. This often yields a boost on that client's task, at the expense of diverging from the global model, which is fine if the goal prioritise local accuracy. More advanced approaches aim to incorporate personalization into the federated training itself. One class of methods is based on meta-learning. Fallah et al [39] proposed Per-FedAvg, a model-agnostic meta-learning approach for FL, in which a globally shared initialization is learned such that each client can cheaply adapt it to their local task with a few gradient steps. This approach keeps the benefits of collaboration, the strong starting model, but yields a personalised model for each user after adaptation. There are also multi-task learning formulations of FL, where the global objective is modified to minimise a weighted sum of individual client losses, allowing a bias towards better worst-case performance. Mohri et al [40] introduced Agnostic Federated Learning, which optimises for any target distribution over clients, effectively focusing on the worst-off client to ensure no client's performance is too low. Such distributionally robust optimization is one way to formalise fairness, seeking to maximise the accuracy of the least satisfied user, thus providing some guarantee of service to all participants. Another notable approach is Q-FFL (q-Fair Federated Learning) by [30], which generalises the objective to enforce a more uniform accuracy distribution across clients. By re-weighting client losses with a parameter q , one can tune the trade-off between overall accuracy and fairness, higher q places more emphasis on clients with worse performance, improving their outcomes at some cost to the best-performing clients. Empirical results showed that Q-FFL achieved a significantly more equitable performance spread, addressing the issue where standard FedAvg might give some clients near-perfect accuracy while

others are barely above chance [30].

Beyond algorithmic fairness, user satisfaction can be improved by giving users a degree of control or transparency. Some systems allow users to opt-out of certain updates or to locally adjust how their data is weighted. This ventures into the human-in-the-loop FL, which is still under-explored. Nonetheless, the prevalent direction in research is creating personalised models automatically. Other personalization strategies include, learning personal model layers or representations, where each client has some private model parameters in addition to the shared global parameters [41], clustering clients with similar data and learning a specialised model per cluster [42], and federated meta-learning frameworks where the system infers how to best personalise for a new client based on experience with previous ones.

Personalization is tightly linked to fairness in federated learning. Fairness here often means ensuring that the learning process does not unduly advantage some clients over others. Without intervention, the global model tends to be biased towards clients with larger or less noisy datasets. Smaller or outlier clients can become "second-class", for whom the global model might perform poorly. Strategies like agnostic FL or q-FFL directly address this by changing the optimization goal to reduce disparity. Another algorithm, Ditto [43], explicitly separates the objective into two parts: a global objective for the shared model and individual local objectives for personalized models, solved in parallel. Ditto showed that by allowing each client to maintain a personalized model through a regularized local training alongside global training, one can improve fairness and robustness as each client gets a personalized model that is less affected by other clients' corrupted data).

It is worth noting that perfect fairness can conflict with overall efficiency or accuracy. For instance, if one client has extremely difficult or noisy data, optimizing for that client's performance could harm the global model for others. Thus, recent work tries to find Pareto-optimal trade-offs or multi-objective optimization for federated fairness. Some approaches adapt the aggregation weights or client sampling frequency based on performance, where clients with higher loss might be sampled more frequently or given higher learning rates to catch up, as a form of affirmative action in training [40].

Ultimately, improving end-user satisfaction in FL is about recognizing that different users define "model quality" differently. A key open question is how to quantitatively measure satisfaction in a federated context which involves metrics for both accuracy and fairness. FL research is moving toward producing a suite of models or a configurable model that can better serve individuals. This trend aligns FL with the broader goal of democratizing model performance so that "no one is left behind" by a collaborative model.

2.4 Efficiency Optimization Techniques

Despite the promise of leveraging distributed data, federated learning faces significant efficiency bottlenecks, primarily in communication. Since FL involves iterative exchange of model updates between clients and server, the process can be dramatically slower than centralized training if not optimized. Communication constraints arise because models, especially modern deep neural networks, consist of millions of parameters, and sending these across network links, which are often wireless or unreliable links, for possibly thousands of rounds is extremely costly. Recent state-of-the-art models such as large language models contain billions of parameters, which would be outright infeasible to train

in a vanilla federated manner over edge devices. Therefore, a major research thrust has been to reduce the communication load in FL while maintaining model accuracy.

2.4.1 Communication Compression

One category of solutions is communication compression. Techniques like quantization and sparsification of model updates can shrink the message size each client sends. For example, instead of sending full 32-bit floating point weights or gradients, a client can send a compressed update: sending only the sign of each gradient component (as in *signSGD* [44]) reduces communication by a factor of 32, and surprisingly, *signSGD* can still converge for non-convex problems with only a modest accuracy loss. Another classic method is quantised SGD (QSGD) [45], which represents gradients with a few bits by probabilistically rounding values – it guarantees unbiased gradient estimates and has theoretical convergence bounds, significantly cutting bandwidth needs. Sparsification approaches have clients send only the largest-magnitude components of the update (top- k sparsification) or updates above a threshold, dropping the rest to zero. This exploits the observation that often gradients are mostly small perturbations with only a few significant directions. By communicating only those, one can achieve significant compression rates (10x–100x) [46]. To make sparsification effective, strategies like error accumulation, which accumulates the dropped components to send later, are used to ensure nothing important is permanently lost. More recent innovations combine quantization and sparsity with learning, such as adaptive compression where the level of compression is tuned during training to maintain just enough signal for progress. Empirical studies show that compression can cut total communication by an order of magnitude or more, often with minor impact on final accuracy, especially when paired with additional local computation to compensate.

Another technique is to reduce the frequency of communication. Local update schemes perform multiple local SGD steps per communication round, as opposed to a fully synchronous SGD that would communicate every single mini-batch. By increasing the number of local steps between communications, one can reduce how often messages are sent. However, too many local steps can cause divergence on non-IID data, as local models drift apart. Building on this, adaptive strategies have been proposed where the server decides if and when to ask clients for an update based on the state of the model. If the global model observes minimal updates, it might delay the next round to avoid redundant communication. Some researchers formulated this as a control problem, determining an optimal communication period to minimise total time [47]. Another approach is to allow asynchronous update aggregation, where faster clients can communicate more frequently than slower ones, which effectively means not every client communicates every round. This reduces per-round communication and can speed up wall-clock time, though the algorithm must account for stale updates from slower clients.

2.4.2 Client Selection and Scheduling

A complementary direction is client selection and scheduling. In large federated networks, it's wasteful and unnecessary to involve every client in every round. Selecting a small but representative subset of clients each round can drastically cut bandwidth and

also mitigate stragglers. The original FedAvg paper used random sampling of clients [3], which is simple and works well when many rounds are possible. Later works consider more intelligent selection approaches, such as prioritizing clients with high-quality or rare data to improve model accuracy faster, or cycling through clients to ensure fairness. Some protocols use multi-armed bandit algorithms to identify which clients yield the best improvements per communication cost [48]. Others enforce proportional fairness that ensures each client to be selected a certain fraction of time relative to its contribution potential or to ensure each has similar influence on the final model. Hierarchical FL is another efficiency strategy, in which local client groups first aggregate models among themselves, and only the aggregated group models are sent to the central server, keeping most communication local, and only compressed higher-level updates traverse the long-distance network [32]. It reduces latency and offloads computation from the central server. Hierarchical methods have shown improved scalability for scenarios like connected cars or IoT networks.

2.4.3 Computation Efficiency Optimisation

Beyond communication, computation efficiency on clients is also considered. Many edge devices are low-power, so heavy training can drain battery or stall the device. Techniques like model pruning and distillation can be used in FL where clients might train a smaller version of the global model, such as a subnetwork, or only train certain layers such as the output layers on device. One example is federated dropout, each client trains with a randomly dropped subset of the model's parameters. This not only reduces computation and memory usage locally but, if orchestrated properly, can also collectively train the full model. Another approach is split learning in a federated context, where each client partially trains a model and sends intermediate activations to the server which completes the forward/backward pass for the remaining layers [49]. This splits the computation, trading client computation offload with increased communication of activations per data point.

A notable recent advance is the use of knowledge distillation for FL efficiency, as introduced in FedKD by [33]. In that scheme, instead of exchanging full model parameters, clients periodically send distilled knowledge to the server. The server aggregates these knowledge vectors and updates a global model accordingly, which is then sent back as a distilled form, such as a small model's predictions, rather than the full model weights. This dramatically lowers communication payload, since only outputs for a limited number of reference examples are shared, not all model parameters. FedKD achieved up to 94.9% reduction in communication cost while obtaining accuracy comparable to standard federated training [33]. The trade-off is the need for a public dataset or synthetic data on which to base distillation, but it demonstrates a promising route for extreme compression.

Overall, efficiency optimization in FL is about striking a balance between communication, computation, and convergence. The past few years have produced a toolkit of methods: quantization, sparsification, client scheduling, local update tuning, and novel paradigms like model splitting and distillation. When combined, these can enable FL in environments previously impractical, such as training a model across millions of phones overnight, or running FL on tiny IoT sensors with limited bandwidth. Applying these techniques often introduces additional hyper-parameters such as compression level, num-

ber of local steps, that need to be tuned for each application. Theoretical research has advanced to where many of these techniques have convergence guarantees under certain assumptions. QSGD has convergence proof for convex problems [45], and FedAvg has been analysed for non-IID settings [31], [5]). Nonetheless, building robust and efficient FL systems require carefully engineered combinations of these methods to avoid poor performance or potential security risks.

2.5 Security and Privacy Vulnerabilities

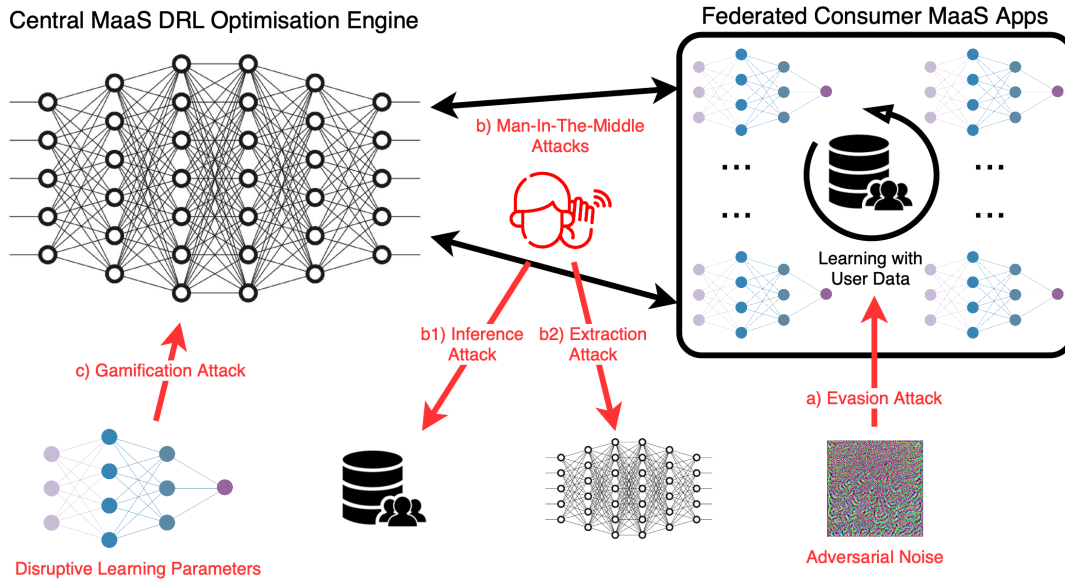


Figure 2.1: Attack vectors against a machine learning MaaS controller: a) evasion data attacks that attempt to cause mis-performance in the machine learning algorithm, b) MitM attacks, b1) inferring personal information by eavesdropping on hyper-parameter or raw data exchanges, b2) model extraction attacks try to infer the overall MaaS controller model, and d) gamification attacks that use a sequence of false data to erode system wide performance.

Federated Learning (FL) was initially proposed as a privacy-preserving solution for decentralised training. However, recent research reveals that FL inherits vulnerabilities similar to conventional machine learning, amplified by its distributed architecture. This section elaborates on key vulnerabilities and mitigation strategies, emphasizing the unique challenges presented by FL and its associated technologies. Fig. 2.1 presents an example of security risks with a Federated AI Mobility as a Service (MaaS) system.

FL exposes several potential vulnerabilities, including data poisoning, inference attacks, model extraction, and communication-based threats. These vulnerabilities threaten the integrity of the model, privacy of users, and the security of the overall system.

2.5.1 Malicious Participation

Malicious participation in federated learning (FL) encompasses a broad spectrum of adversarial behaviours where clients deliberately disrupt or manipulate the collaborative training process. Such threats include both traditional model poisoning and emerging forms of strategic gaming aimed at subverting system incentives or learning dynamics.

One major class of attack is data poisoning, where malicious clients submit corrupted or misleading model updates. These can involve mislabelling data or injecting backdoor triggers into local training sets, ultimately degrading the integrity and performance of the global model [29]. Attacks targeting rare or tail data distributions have further exposed vulnerabilities in robust aggregation schemes, such as those relying on Byzantine-resilient methods, by exploiting their statistical assumptions [50]. Techniques like Krum and trimmed mean have been developed to counteract these threats [51], [52]; however, adaptive adversaries remain capable of evading detection through sophisticated mimicry of benign client updates.

Beyond direct model manipulation, malicious participation also includes gamification attacks—strategies in which clients exploit the incentive or reward structures embedded in FL systems. Rather than merely poisoning data, adversarial clients may manipulate their behaviour to maximize personal gain or disrupt resource allocation. Examples include falsifying reported statistics, misrepresenting training contributions, or strategically submitting updates to trigger favourable outcomes in dynamic resource scheduling, reputation scoring, or reward allocation mechanisms. These behaviours convert the learning process into a competitive game, pitting honest clients seeking global model improvement against attackers optimizing for selfish objectives. When FL leverages deep reinforcement learning (DRL) for participant selection or resource management, adversaries can further exploit system dynamics by falsifying state representations or gaming policy responses [53], [54]. Such manipulations threaten not only model accuracy but also consumer satisfaction, fairness, and long-term system stability.

Mitigation Approaches:

- *Adversarial Training:* Incorporating adversarial scenarios and simulated attacks during training, particularly in federated DRL systems, helps agents learn to anticipate and resist manipulative strategies. Approaches such as adversarial reinforcement learning and robust policy optimization expose models to a diverse set of attack patterns, improving resilience to both model poisoning and incentive gaming [55], [56].
- *Robust Aggregation:* Deploying advanced aggregation algorithms that are resilient to both statistical outliers and adaptively crafted malicious updates is critical. Techniques such as Multi-Krum, Bulyan, and adaptive federated averaging increase the difficulty for adversaries to bias global models, while newer methods integrate anomaly detection or reputation mechanisms to filter suspicious clients [51], [57], [50].

- *Incentive-Aware Defence Design*: Developing incentive mechanisms that are resistant to manipulation is vital for sustainable FL. Game-theoretic defences and mechanism design frameworks can align client interests with system objectives, discouraging reward gaming and Sybil attacks by tying rewards to robust and verifiable performance metrics [58], [37]. Approaches such as reputation-based reward allocation and cryptographically verifiable contributions have shown promise in mitigating exploitation of incentive systems.

2.5.2 Man-in-the-Middle Attacks

Man-in-the-Middle (MitM) attacks present significant threats to federated learning (FL) by intercepting, manipulating, or exploiting communication between clients and aggregators. Within this attack category, adversaries aim to compromise either user privacy or proprietary model confidentiality by mounting targeted inference or extraction attacks. Two prominent variants are Membership Inference Attacks (MIAs) and Model Extraction Attacks (MEAs), which focus on data privacy and model security, respectively [59], [60], [61].

MIAs leverage intercepted model updates to infer whether specific data points or users contributed to training. By analysing subtle differences in gradient patterns or model parameters, adversaries can achieve high inference accuracy even under partial observation, threatening user anonymity in sensitive FL applications [59], [60]. In many cases, MitM attackers enhance their capabilities by manipulating or replaying update messages, further increasing their inference success and rendering traditional privacy guarantees insufficient.

MEAs, on the other hand, exploit the interactive nature of FL, to systematically reconstruct proprietary models. By carefully crafting and sending queries, attackers can extract model weights, replicate the architecture, or approximate model behaviour [62]. Such attacks undermine the economic and strategic value of privately trained models and may open the door to downstream misuse, including intellectual property theft and targeted adversarial attacks.

Mitigation Approaches:

- *Differential Privacy (DP)*: Adds calibrated noise to gradients or model updates, mitigating MIAs by masking the contribution of individual data points without severely impacting model performance [63], [64].
- *Gradient Embedding Perturbation (GEP)*: Projects sensitive gradients into privacy-preserving embedding spaces, reducing the risk of both inference and extraction attacks by limiting exploitable information in intercepted updates [65].
- *Adaptive Misinformation*: Integrates gradient perturbation and query detection, actively misleading unauthorized extraction attempts and increasing the difficulty of successful model replication in real-time [66].
- *Secure Communication Protocols*: Employs encryption and mutual authentication between clients and aggregators to restrict MitM opportunities, adding an essential layer of protection beyond algorithmic defences.

2.5.3 Evasion Attacks

Evasion attacks compromise neural networks by altering input data to evade correct classification, notably impacting FL-based MaaS services reliant on accurate data interpretation for personalised services and autonomous decisions. Recent advances in evasion techniques leverage Generative Adversarial Networks (GANs) to produce sophisticated adversarial examples adaptable to dynamic conditions, such as physical deformations and varying user inputs [67], [68]. Additionally, Large Language Models (LLMs), crucial in FL for decision-making and user interaction, have shown vulnerability to crafted adversarial inputs capable of bypassing standard detection mechanisms [69].

Mitigation Approaches:

- *Adversarial Training*: Incorporates adversarial examples during training to bolster model resilience against similar future attacks [70].
- *Region-based Classification*: Utilises sampling techniques around input data points to diminish the impact of adversarial perturbations [71].
- *Defensive Distillation*: Smooths decision boundaries using distillation methods, increasing resistance to minor perturbations [72].

2.6 Critical Reflections

Federated Learning (FL) has rapidly matured from a conceptual innovation into an extensive research domain addressing key challenges of decentralised model training in real-world environments. This chapter has systematically reviewed five critical contextual factors—client heterogeneity, contribution assessment, end-user satisfaction, efficiency optimisation, and security, each crucially impacting FL’s practical implementation and performance.

The literature demonstrates significant strides in addressing client heterogeneity, particularly through algorithms that mitigate statistical and system heterogeneity, such as FedProx [5], SCAFFOLD [31], and hierarchical FL approaches [32]. Nevertheless, critical research gaps persist, notably in the systematic underutilisation of device-specific capabilities and contexts [25], [7]. Current approaches still predominantly assume simplified or homogeneous client environments, neglecting substantial variability in computational resources, connectivity, and energy availability [8]. As discussed, preliminary context-aware client selection methods [9] show promise but lack comprehensive exploration, particularly in real-world settings where hierarchical and clustered client management could offer scalable, tailored solutions [12].

Contribution assessment and incentive mechanisms have advanced considerably, notably through Shapley value-based methods [36], [32] and robust contribution detection systems designed to mitigate free-riding and malicious behaviour [35]. However, accurately quantifying client contributions in heterogeneous and resource-constrained contexts remains challenging, underscoring the need for sophisticated, context-aware assessment methodologies that consider device-specific constraints and user incentives.

Personalisation and end-user satisfaction research has significantly progressed with methods like local fine-tuning, meta-learning approaches [39], and fairness-driven al-

gorithms [40], [30]). Still, achieving optimal personalisation while maintaining fairness across diverse client bases continues to present complex trade-offs. The integration of human-in-the-loop and configurable personalisation strategies requires deeper investigation, particularly to quantify satisfaction in diverse federated contexts effectively [41], [42], [43].

Efficiency optimisation techniques, including communication compression [45], [44], client scheduling [48], hierarchical aggregation [32], and computation offloading [6], have notably advanced FL toward practical deployment, especially in resource-constrained environments. However, the research gaps highlight the critical need for context-aware efficiency strategies. Techniques successful in centralised settings, such as dynamic quantisation or mixed-precision computations [27], remain inadequately adapted to heterogeneous federated environments. Furthermore, dynamic offloading strategies, while promising, have not been sufficiently explored, particularly in balancing computational efficiency with communication overheads and privacy constraints.

Security considerations, essential given FL’s decentralised nature, have evolved with effective mitigation strategies such as adversarial training [55], [56], robust aggregation [51], [52], differential privacy [63], [64], and secure communication protocols. Despite this progress, efficiency-driven optimisations introduce new vulnerabilities, both software-based, such as the increased susceptibility to adversarial attacks due to model compression and sparsification [29], [28], and hardware-based, such as the risks associated with memory optimisation techniques like DMA. Thus, integrated security measures addressing vulnerabilities arising specifically from efficiency-driven optimisation strategies remain critically under-explored.

An reflection is the complex interplay among contextual factors in FL. Addressing statistical heterogeneity or enhancing efficiency can inadvertently impact fairness, user satisfaction, and security. Achieving an optimal balance among these often conflicting objectives necessitates holistic, multi-objective optimisation frameworks [26], [13]. Emerging research underscores the need for comprehensive methodologies that systematically manage these trade-offs, particularly within heterogeneous, resource-constrained federated environments.

Furthermore, the ongoing transition from theoretical formulations toward practical real-world deployments has uncovered additional complexities, including concept drift, regulatory compliance, and distinctions between cross-silo and cross-device settings [14]. These emergent factors suggest future research directions must incorporate adaptive learning techniques, regulatory-aware system designs, and nuanced cross-setting considerations.

In conclusion, while considerable progress has been made across the contextual dimensions of federated learning, critical gaps remain, particularly concerning context-aware optimisation, efficient utilisation of heterogeneous client capabilities, accurate contribution assessment, and robustness against vulnerabilities introduced by efficiency measures. Addressing these gaps through integrated, multi-faceted approaches is imperative for federated learning to fully realise its potential—enabling secure, efficient, and equitable collaborative intelligence across diverse real-world settings.

Chapter 3

Mixed-Precision Federated Learning via Multi-Precision Over-the-Air Aggregation

3.1 Introduction

Federated Learning (FL) [73] has emerged as a widely adopted distributed learning paradigm that protects privacy by sharing and aggregating model parameters rather than raw data. However, classic FL approaches are often limited by communication bottlenecks and computational heterogeneity among participating edge clients. Over-the-Air Federated Learning (OTA-FL) [74] addresses these challenges by leveraging the electromagnetic channel for direct aggregation of client model updates, thus achieving privacy-preserving and communication-efficient distributed learning in wireless networks. Figure 3.1 illustrates the evolution from traditional FL to OTA-FL and highlights the research frontier: enabling energy-efficient, heterogeneous client aggregation under real-world resource constraints.

A major gap in current OTA-FL research is the assumption of homogeneous computational precision across clients. In practical deployments, edge devices are equipped with varied hardware resources and may exploit Approximate Computing (AxC) to select optimised bit precisions for energy and computational efficiency. This creates a fundamental incompatibility: aggregating model updates of mixed precision over the electromagnetic channel is non-trivial, since the wireless modulation superposition process requires update compatibility at the physical layer.

Recent advancements in FL have prioritised two main directions: privacy/security enhancement and communication/computation efficiency [75]. Techniques such as SCAF-FOLD [31] and LoSAC [76] improve convergence and reduce communication cost. On the computation side, methods like hardware acceleration [77], network architecture optimisation [78], model slicing [79], and AxC have been explored. Inspiration for our work comes from the flexibility of Field Programmable Gate Arrays (FPGAs) [80], which unlike CPUs and GPUs, can operate efficiently at diverse bit precisions, and are widely adopted in energy-constrained applications across networking, signal processing, and machine learning [81].

This chapter introduces a mixed-precision OTA-FL framework, where clients parti-

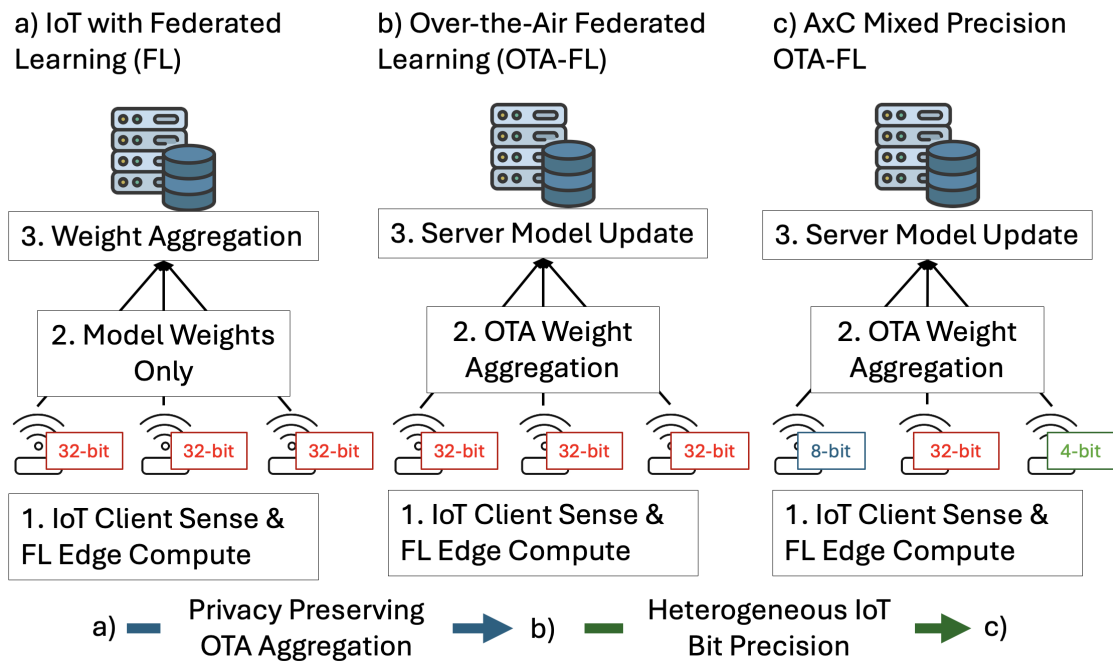


Figure 3.1: End-to-end Federated Learning (FL) system moves from (a) FL, to (b) privacy preserving OTA-FL, to (c) energy efficient AxC OTA-FL. The research challenge from (b) to (c) is to achieve heterogeneous OTA weight aggregation to cater for mixed bit precision IoT edge computation.

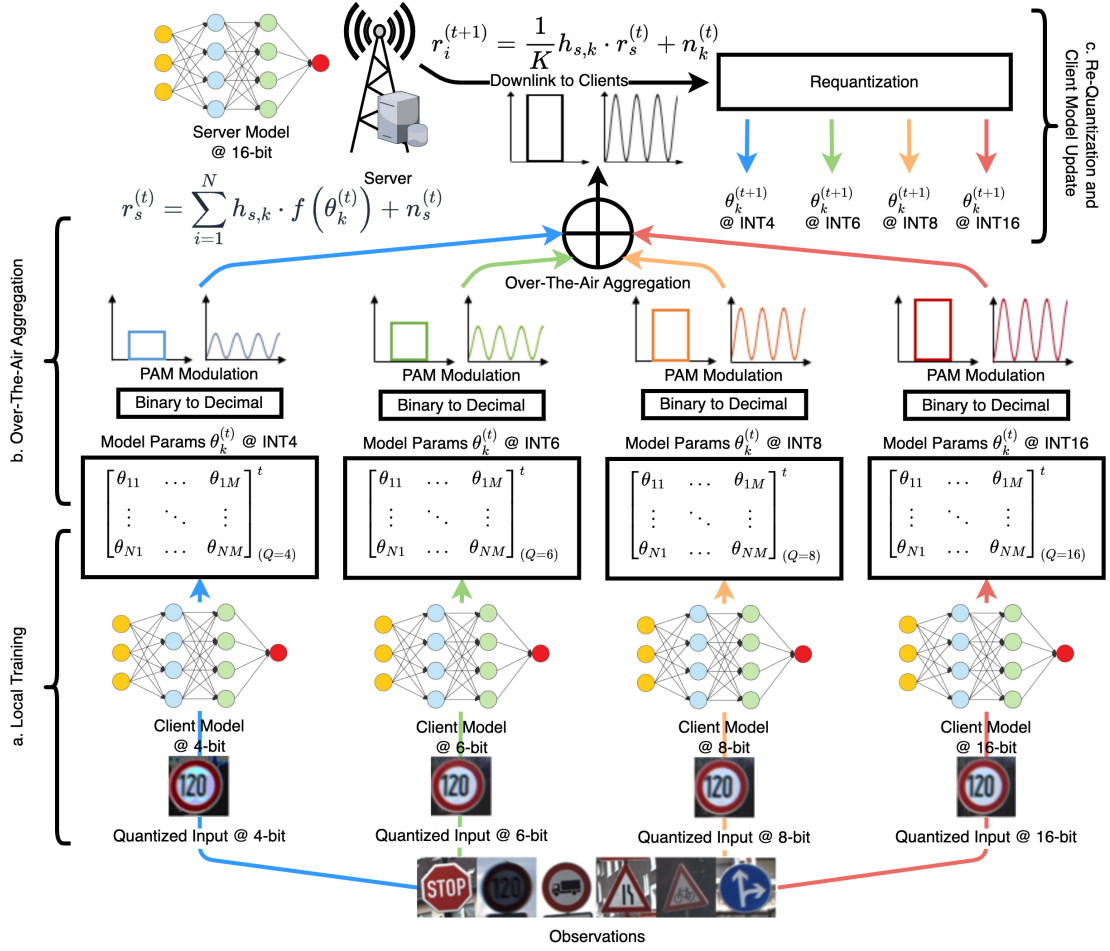


Figure 3.2: Structure of our proposed Approximate Computing (AxC) based OTA-FL framework of multi-precision clients and unified multi-precision modulation scheme. The intelligent transport validation case study here is a multi-precision federated supervised traffic sign recogniser. (a) Clients operate end-to-end at their designated computation precisions with their own data and labels. (b) Multi-precision OTA aggregation process (uplink). (c) Downlink, re-quantization and client model update.

cipate at optimised bit precisions aligned with their local energy and computational capabilities. The primary contributions are as follows:

1. Development of an AxC-based OTA-FL system that supports multi-precision client participation, enhancing both system efficiency and learning performance in heterogeneous environments.
2. Design of a unified multi-precision gradient modulation scheme, which ensures physical-layer compatibility for OTA aggregation and eliminates the need for costly precision conversion.
3. Empirical validation through a real-world intelligent transport application, demonstrating notable server convergence improvements and over 10% accuracy gain for

4-bit ultra-low-precision clients, at the expense of minor accuracy degrade of their higher precision peers, with energy savings exceeding 65% and 13% compared to homogeneous 32-bit and 16-bit client setups, respectively.

Figures 3.1 and 3.2 provide a conceptual overview and system architecture of our approach. By bridging the precision heterogeneity gap, the proposed framework unlocks new possibilities for scalable, energy-efficient federated learning at the wireless edge. The remainder of this chapter details the system setup, technical design, and experimental validation of mixed-precision OTA-FL.

3.2 System Setup

3.2.1 Federated Learning

We consider an over-the-air federated learning system with N clients, at each communication round, K of them are selected to update denoted by $\mathcal{K} = \{1, \dots, K\}$. Each client k has a local dataset $\mathcal{D}_k$. For $\mathcal{T}$ communication rounds, the clients collaboratively refine a unified global model, while preserving the privacy of their local data. This process collects the local trained models of clients, and aggregates these models to update the global model. The refined global model is then distributed back to the edge devices for subsequent predictions and further local training iterations, thereby progressively enhancing the global model's accuracy. The model aggregation phase, central to FL, can be mathematically represented as:

$$\boldsymbol{\theta}^{(t+1)} = \frac{1}{K} \sum_{k=1}^K w_k \cdot \boldsymbol{\theta}_k^{(t)},$$

Equation 1: FedAvg Aggregation

where $\boldsymbol{\theta}^{(t+1)}$ denotes the parameters of the global model after the $(t+1)$ -th training iteration, $\boldsymbol{\theta}_k^{(t)}$ represents the parameters of the local model from the edge device k at the t -th communication round, and w_k signifies the relative contribution (or weight) of the edge device k , typically proportional to its dataset size.

3.2.2 Over-The-Air (OTA) Computation for FL

The principle of Over-The-Air (OTA) computation[82], exploits the natural superposition property inherent to wireless channels. Within the FL framework, this concept finds practical application during the gradient aggregation phase. By utilizing a common uplink bandwidth across all edge devices for gradient transmission, the superposition property of the channel facilitates direct aggregation. Consider a Single-Input Single-Output (SISO) fading channel between the server and an edge device k , characterised by a Rayleigh distributed random variable $h_{s,k} \in \mathbb{C}$, which represents the complex channel gain capturing both amplitude attenuation and phase shift due to multipath fading. The OTA aggregation process can then be modelled as:

$$r_s^{(t)} = \sum_{k=1}^K h_{s,k} \cdot f\left(\theta_k^{(t)}\right) + n_s^{(t)},$$

Equation 2: OTA Aggregation Process

where $f(\cdot)$ embodies the comprehensive process including source coding, constellation design, up-sampling, modulation, and precoding. Here, $r_s^{(t)}$ represents the aggregated signal received by the server in the t -th upload cycle, and $n_s^{(t)}$ denotes the additive noise.

In contrast to traditional federated learning aggregation mechanisms, leveraging OTA computation necessitates careful design of $f(\cdot)$ to address two critical challenges: (i) mitigating the effects of channel fading, and (ii) ensuring accurate linear summation of gradients from multiple edges. This becomes particularly complex when considering edges with disparate computation, storage, and operation precision constraints. A notable challenge arises in aggregating gradients quantised at different levels, exemplified by the non-commutative property of quantised modulations:

$$QAM([\theta_j]_{q_j}) + QAM([\theta_k]_{q_k}) \neq QAM([\theta_j]_{q_j} + [\theta_k]_{q_k}),$$

Equation 3: Non-Commutative Property of Quantised Modulations

where $QAM(\cdot)$ represent the quadrature modulation function and $[\theta_k]_{q_k}$ indicates model parameters of client k is at q_k quantization levels (in terms of bits) utilised for encoding weights and biases of models at different edge devices.

3.2.3 Approximate Computing and Low-Precision ML

Within AxC methods, quantization stands out for its universal applicability, and compatibility with existing FL systems. In the context of edge computing, FPGA accelerators become particularly advantageous, as they can be dynamically reprogrammed to efficiently perform computation at any custom precision level including 4-bit and below, thus significantly reducing computation demand and energy consumption in resource-constrained environments [77].

As quantization can significantly reduce the resource demand for storing, inferring, and training machine learning models [83], [84] at the price of performance degradation, it is essential to demonstrate the trade-offs of low-precision machine learning. Here, the performance degradation of common CNN models in quantization is benchmarked, as shown in Table. 3.1. All these models are trained in 32-bit floating-point format and then quantise to lower bit-levels. The degradation only becomes noticeable when quantised to 8-bit, and retains acceptable before being quantised below 4-bit.

It is also worth noting that such performance can not be stably achieved via conventional end-to-end training at the same lowest precision level due to the limit of gradient dynamic range and cumulative error of low precision, and additionally, when the size of dataset increases, gradients may require a larger dynamic range to fit the data.

Although end-to-end low precision training can be achieved through dedicated design of format, quantization algorithms, and arithmetic implementation [85], a key advantage

Table 3.1: GTSRB Classification Performance of Common Models across Quantization Levels

Model	8-bit	6-bit	4-bit	3-bit	2-bit
densenet_161	96.56%	96.45%	91.55%	39.83%	0.00%
efficient_net_b4	94.77%	94.74%	90.55%	42.24%	0.07%
efficient_net_v2m	95.26%	95.19%	85.75%	7.68%	6.22%
reg_net_x_16gf	96.56%	96.11%	80.84%	0.00%	0.00%
reg_net_y_3_2gf	93.53%	92.75%	72.72%	6.97%	0.00%
resnet_50	94.94%	94.33%	65.21%	0.00%	0.83%
squeeze_net_1_0	87.29%	85.41%	72.95%	39.85%	6.64%

Orange: damaged but usable performance (65 – 85%)

Red: unacceptable performance (< 65%).

of our mixed-precision OTA-FL framework is simplicity for application. Mixed-precision federated learning is leveraged to enable the low precision ML on edge devices and hence achieve superior trade-off of computation and performance, and reduction of energy consumption.

3.3 Mixed-Precision OTA Federated Learning

The process of our Mixed-Precision OTA Federated Learning framework can be described as Algorithm 1¹.

3.3.1 Multi-Precision Over-The-Air Aggregation

Over-The-Air Federated Learning (OTA-FL) encompasses a three-step process during each update round, denoted as the t -th round. The sequence begins with each client k conducting local training and producing a model update $\Delta[\theta_k^{(t)}]_{q_k}$. Subsequently, as shown in Fig. 3.2 (b), the client prepares for transmission by converting binary parameters of its designated precision into decimal equivalents. These decimal values are then modulated onto carrier waves through amplitude modulation, creating signals ready for bandwidth transmission. Our modulation function can be described as follows.

where $M(\cdot)$ represent the modulation function, $[\theta_i]_{q_i}$ indicates model parameters of client k is at q_k quantization levels, and f_c is the channel frequency.

In the upload phase, the client k performs the channel estimation to determine the communication link, denoted as $h_{s,k}$, between the server and itself. This estimated channel information is crucial for implementing transmission beamforming, a technique employed for channel compensation. Such compensation is integral to facilitating efficient OTA aggregation of transmitted data. The procedural intricacies of this approach are elucidated below.

¹The Algorithm can work with any client selection method, such as random selection, loop, connection quality prioritised.

Algorithm 1 Mixed-Precision OTA Federated Learning

```

1: Input: Initial global model  $\theta^{(0)}$ , number of communication rounds  $T$ , number of
   clients  $K$ , number of participants in aggregation stage  $J$ 
2: Output: Trained global model  $\theta^{(T)}$ 
3: for each round  $t = 1$  to  $T$  do
4:   Step 1: Broadcast global model  $\theta^{(t-1)}$ 
5:   Server broadcasts global model  $\theta^{(t-1)}$  to all clients
6:   Step 2: Local training at clients
7:   for each client  $k = 1$  to  $K$  (in parallel) do
8:     Quantise  $\theta^{(t-1)}$  to designated precision level  $q_k$ 
9:     Local training  $[\theta_k^{(t)}]_{q_k} = \text{Training}([\theta^{(t-1)}]_{q_k}, \mathcal{D}_k)$ 
10:    Calculate update  $\Delta[\theta_k^{(t)}]_{q_k} = [\theta_k^{(t)}]_{q_k} - [\theta^{(t-1)}]_{q_k}$ 
11:   end for
12:   Step 3: Over-the-Air aggregation of mixed-precision model updates
13:   Client selection
14:   for each selected client  $j = 1$  to  $J$  (in parallel) do
15:     Convert model update  $\Delta[\theta_j^{(t)}]_{q_j}$  to decimal
16:     Amplitude modulation, channel estimation and uplink
17:   end for
18:   Step 4: Server-side post-processing and downlink
19:   Received signal  $r_s^{(t)} \approx \sum_{j=1}^J \theta_j^{(t)}$ 
20:   Broadcast updated model  $\frac{r_s^{(t)}}{J} \approx \frac{\sum_{j=1}^J \theta_j^{(t)}}{J}$ 
21: end for
22: Return final global model  $\theta^{(T)}$ 

```

$$M([\theta_k]_{q_k}) = [\theta_k]_{q_k} \cdot \cos 2\pi f_c t$$

Equation 4: OTA Modulation Function

3.3.1.1 Channel Estimation at Clients

Channel estimation between the server and each client is done by broadcasting a pre-defined pilot sequences u from the server. Then, client k can estimate the channel between k and the server $h_{s,k}$ as follows.

$$\hat{h}_{s,k} = y_s \cdot \frac{u^*}{|u|^2} \approx h_{s,k},$$

Equation 5: Channel Estimation Function between Client and Server

where $y_s = h_{s,k} \cdot u + n_s$ is the received signals at server with receiving noise n_s . Here, u is the known pilot sequence, and u^* denotes its complex conjugate.

3.3.1.2 Uplink Design

After channel estimation at each client k , the clients modulate model updates to carrier frequency and compensate for channel distortion via its estimated channel. The base-band transmitted signal is designed as follows.

$$f(\theta_k^{(t)}) = \hat{h}_{s,k}^{-1} \cdot \theta_k^{(t)}.$$

Equation 6: Base Band Transmission Function

Hence, aggregation can be done via the natural superposition of the electromagnetic wave, i.e., $r_s^{(t)} \approx \sum_{k=1}^K \theta_k^{(t)}$.

3.3.1.3 Downlink Design

After OTA aggregation, the server broadcasts the updated model, i.e., $r_s^{(t)}/K \approx \sum_{k=1}^K \theta_k^{(t)}/K$ back to clients. The received signal at each client k is:

$$r_k^{(t+1)} = \frac{1}{K} h_{s,k} \cdot r_s^{(t)} + n_k^{(t)}.$$

Equation 7: Client Received Signal

The client i then recovers the aggregated gradient via the estimated channel, as follows.

where $\tilde{\theta}_k^{(t+1)}$ is the updated gradient of client k , and will be local trained for the $(t+1)$ th round.

Furthermore, during the OTA aggregation process, the mixed-precision quantization scheme, which remains inherently opaque to potential adversaries, significantly enhances the system's security against attacks targeting the aggregation process.

$$\tilde{\theta}_k^{(t+1)} = h_{s,k}^{-1} \cdot r_k^{(t+1)} \approx \frac{1}{K} \sum_{k=1}^K \theta_k^{(t)},$$

Equation 8: Aggregated Model Recover Function

Algorithm 2 Quantization Function

```

1: Input: Tensor  $W \in \mathbb{R}^{m \times n}$ , type: "fixed-point" or "floating-point", bit-width  $b$ 
2: Output: Quantised tensor  $Q \in \mathbb{R}^{m \times n}$ 
3: if type is "fixed" then
4:    $w_{\min} = \min(W)$ ,  $w_{\max} = \max(W)$ 
5:    $\text{scale} = \frac{w_{\max} - w_{\min}}{2^b - 1}$ ,  $\text{zero\_point} = -\frac{w_{\min}}{\text{scale}}$ 
6:   for each  $w_{ij}$  in  $W$  do
7:      $q_{ij} = \max(0, \min(2^b - 1, \lfloor \frac{w_{ij}}{\text{scale}} + \text{zero\_point} \rfloor))$ 
8:   end for
9: else if type is "floating-point" then
10:  for each  $w_{ij}$  in  $W$  do
11:    Truncate mantissa and exponent to fit  $b$  bits
12:  end for
13: end if
14: Return  $Q$ 

```

3.3.2 Approximate Computing via Quantization

To ensure broad applicability, a simple and efficient quantization algorithm is employed, as outlined in Algorithm 2. For precision levels of 8-bit and higher, both fixed-point and floating-point formats are supported. However, fixed-point format is preferred for lower precision levels due to the limited dynamic range of floating-point formats under 8-bit representation.

On the client side, the quantization function is systematically applied to every layer of the CNN model, spanning from input to output, and is integrated into both the forward and backward passes. This approach ensures a unified precision level throughout the end-to-end system.

3.3.3 Energy Consumption Estimation

Along with compute savings, as an equally important product of quantization, the energy savings are also measured, which originate from the higher throughput of operations at lower precisions. The energy consumptions of training a ResNet-50 model at multiple precision levels between 32-bit and 4-bit on 9 Xilinx FPGA platforms of varying specifications are estimated. Then the average relative energy saving compared to 32-bit at each precision levels of all these platforms is used to estimate the total savings for our mixed-precision clients.

However, the energy consumption for the same task of FPGA platforms can vary significantly based on the program design of the same hardware. In FPGA focused researches, accurate energy consumption can be obtained through the official power analysis toolkit [86]. Here, I adopted a modest energy consumption estimation for the clients

based on official data sheets [87] of typical FPGA edge platforms with Equation 9 below to showcase the potential of energy savings of our approach.

$$E_{ML} = \frac{D_{ML}}{F_{DSP} \cdot N_{DSP} \cdot N_{MAC}} \cdot E_{Package}$$

Equation 9: FPGA Energy Cost Estimation Function

Where E_{ML} and D_{ML} represent the energy consumption and computation demand by operations per communication round of ML task respectively, F_{DSP} is the frequency of DSP slices, N_{DSP} is the number of onboard DSP slices, and N_{MAC} is the number of multiply-accumulate (MAC) operations each DSP slice can carry out per cycle, and $E_{Package}$ represent typical package energy consumption[88].

3.4 Experiments

This section details the experiment setup and results addressing the research questions posited earlier, focusing on (i) the efficacy of multi-precision OTA aggregation, and (ii) the formulation of client quantization schemes to optimise performance across both server and client dimensions.

3.4.1 Settings

Our experimental framework emulates an OTA-FL system of 15 clients working at designated quantization levels. The model structure is ResNet-50 with ImageNet [1] pre-trained weights initialization. The FL operates over 100 communication rounds with 5-30dB of emulated Gaussian noise.

3.4.1.1 Data

We utilise the German Traffic Sign Recognition Benchmark (GTSRB) [89] as the dataset for our experiments. This dataset consists of 39,209 training samples and 12,630 testing samples across 43 traffic sign classes, captured in real-world driving environments. It reflects the variability of real-world conditions, including changes in lighting, weather, perspectives, and occlusions. This diversity ensures the dataset’s authenticity and makes it highly suitable for advancing research and development in smart transportation systems. In our experimental setup, each client is assigned an equal subset of the data.

3.4.1.2 Quantization Schemes

We assign quantization levels to the 15 clients by a group of 5. Each scheme consists of 3 precision levels, and each precision level is assigned to 5 clients. Quantization levels are chosen from [32, 24, 16, 12, 8, 6, 4].

3.4.1.3 Performance Metrics

Evaluation metrics for the server include convergence speed, measured by the number of communication rounds the system took to converge, and final performance of the aggregated model. For clients, their performances are assessed after aggregation and re-quantization.

3.4.1.4 Energy Efficiency

The energy savings of the mix-precision clients are compared with homogeneous precision clients at standard 32-bit, 16-bit, 8-bit and 4-bit.

3.4.2 Results

3.4.2.1 Energy Efficiency Estimation

Table 3.2: Estimated Energy Consumption per Sample for ResNet-50 Forward Pass and Relative Savings Compared to 32-bit

	32-bit	16-bit	12-bit	8-bit	6-bit	4-bit
Energy Cost (J)	0.36	0.17	0.16	0.022	0.021	0.0056
Saving (%)	0	52.58	56.15	93.89	94.17	98.45

Based on Eq. 9, the energy consumption of forward passing one sample through a ResNet-50 network, is estimated, across following quantization levels [32, 24, 16, 12, 8, 6, 4] on 9 platforms of different hardware resources such as logic cells and DSP slices. Table 3.2 presents the average energy cost and relative savings to 32-bit of these platforms. Notably, due to under-utilization of hardware, quantizing to 16-bit and 12-bit share very similar degree energy saving, and the same applies to 8-bit and 6-bit. From the table, the diminishing energy saving gain when further quantizing from low precision like 8-bit to ultra low ones like 4-bit can also be observed.

3.4.2.2 Federated Training and Server Performance

The convergence velocity, as depicted in Fig. 3.3, indicates that setups of uniform 4-bit clients, or a mixed-precision schema of [12, 4, 4] bits, exhibit slower and more erratic initial convergence, even when the latter has a better random start in training. In contrast, setups incorporating clients with 16-bit precision or higher demonstrate more rapid and stable convergence, achieving approximately 90% accuracy within 10 communication rounds. Notably, for clients of high resource capacity, 32-bit or 24-bit precision only offers marginal training gains compared to 16-bit precision. The server model performance of all quantization schemes reached 97% top-1 accuracy within a tight 0.3% margin after 100 communication rounds, underscoring the effectiveness of the federated learning framework in achieving high accuracy with mixed-precision clients.

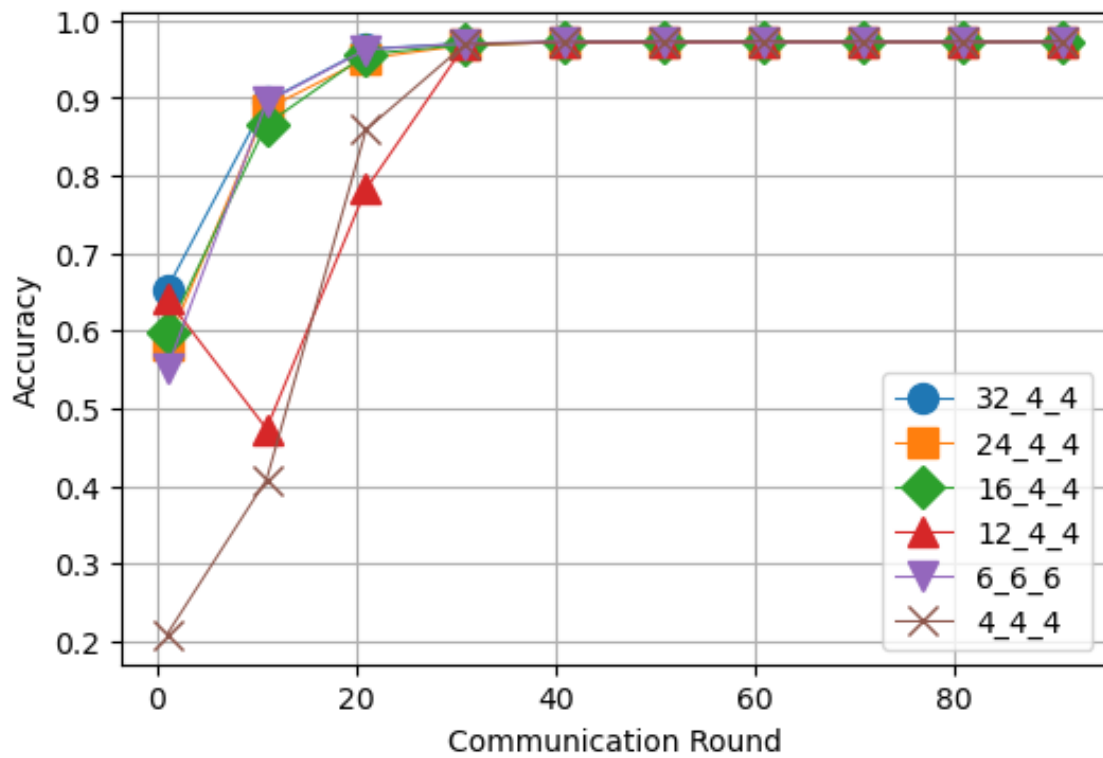


Figure 3.3: Training accuracy in 100 communication rounds, with ImageNet pre-trained weights initialization, scheme $[4, 4, 4]$ (denoted by brown 'X'), and scheme $[12, 4, 4]$ (denoted by red upright triangle) converge significantly slower than other schemes, even when the latter one has a better random start.

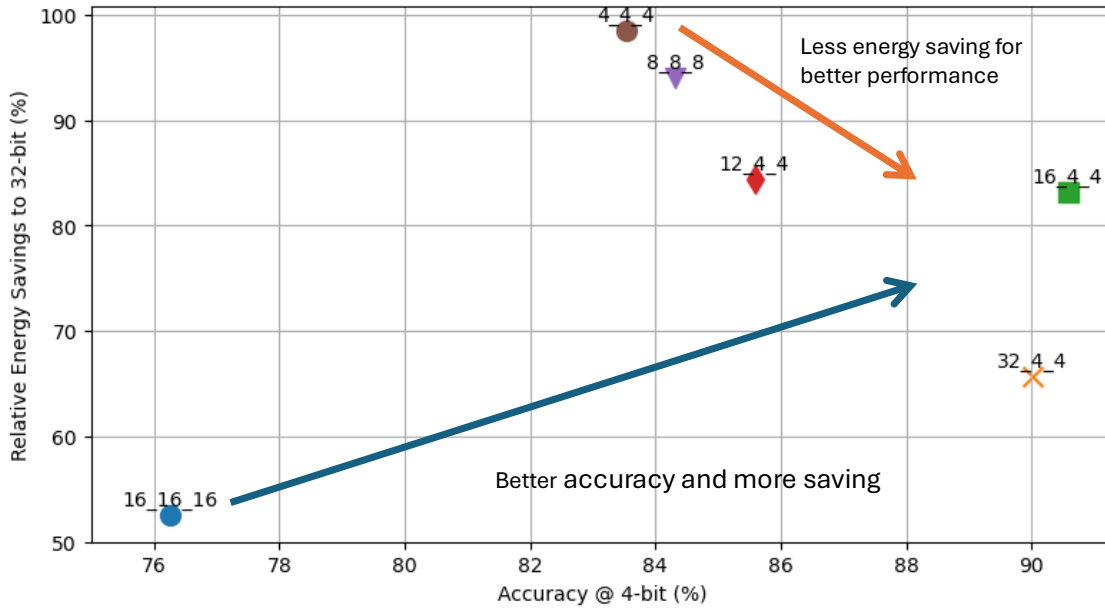


Figure 3.4: Trade-offs between accuracy of model quantised to 4-bit and energy savings in comparison to homogeneous 32-bit and 16-bit clients, schemes near bottom right corner presents superior trade-off towards accuracy.

3.4.2.3 Client Performance

After 100 rounds, the final global model, θ^{100} , is broadcast to clients. Here, the performance of the clients at the lowest precision, 4-bit, are more critical, as higher counterparts have better performance and minor degradation from well converged global model as illustrated in Table 3.1 and Fig. 3.3².

As shown in Fig. 3.4, in comparison to FL systems of homogeneous clients at 32-bit and 16-bit, based on our estimation in Table 3.2, our FL with mixed-precision clients models can save over 65% and 13% of energy consumption respectively, while gaining more than 10% in accuracy on clients at 4-bit. Notably, for those under schemes incorporating 16-bit precision or higher, when re-quantised for 4-bit clients, attain around 5% higher accuracy, and this performance boost for lower precision clients from higher precision counterparts shows diminishing returns beyond 16-bit precision. While comparing to FL of homogeneous clients at 8-bit and 4-bit, our mixed-precision FL can trade mere 10% of energy savings for 5% of accuracy.

Despite all the above benefit, it's worth noting that the more complicated modulation scheme inevitably result in longer training time than conventional OTA-FL, as the overhead quantisation process are covered into the OTA aggregation process, especially considering the channel noise, fading and synchronisation errors in practical deployment. And like all OTA-FL systems, the training process takes longer than conventional FL with equivalent network connection quality.

²The initial drop of the '12-4-4' scheme, marked in red, was the result of lucky random initialised model parameters, but does not affect the over all training trend or the final model performance.

3.5 Conclusion

This chapter introduced a mixed-precision Over-the-Air Federated Learning (OTA-FL) framework grounded in Approximate Computing (AxC), designed to accommodate clients with varying computation precisions in heterogeneous edge environments. Through a unified multi-precision aggregation scheme, the framework delivers substantial improvements in both learning performance and energy efficiency, especially for ultra low precision clients. By demonstrating compatibility with standard FL systems, the work highlights the value of directly leveraging client heterogeneity, revealing significant gains in accuracy and energy savings that traditional homogeneous approaches overlook.

In alignment with the aims and research questions of this thesis, this study addresses several critical aspects of modern federated learning. First, it provides a concrete system-level solution for exploiting client heterogeneity, directly supporting the thesis objective of enhancing efficiency and accuracy in FL through contextual optimisation. The mixed-precision approach presented here serves as a case study for how diverse client contexts—rooted in hardware capability and energy constraints—can be dynamically integrated into federated learning without sacrificing performance or security.

Furthermore, the methodology and results of this chapter respond to the thesis objectives concerning the evaluation of efficiency optimisation techniques and their impact on system robustness. By adopting a cross-disciplinary perspective—encompassing wireless aggregation, precision quantisation, and energy-aware computation—the chapter not only advances the technical frontier of FL, but also sets the foundation for broader inquiry into contextual factors, client profiling, and adversarial vulnerabilities outlined in the thesis objectives.

This work thus contributes to a more comprehensive understanding of how contextual heterogeneity shapes optimisation trade-offs, computational efficiency, and system security in federated learning. Beyond FL, the proposed mixed-precision modulation strategy opens new pathways for resource-efficient, secure distributed computation across wireless networked systems. The findings and methods in this chapter are integral to the thesis' broader ambition of guiding the design and optimisation of next-generation green, robust, and context-aware FL architectures.

Chapter 4

RAG-based User Profiling for Precision Planning in Mixed-precision Over-the-Air Federated Learning

4.1 Introduction

Mixed-precision computing is a widely adopted technique in artificial intelligence, offering a larger trade-off space between accuracy and efficiency. In wireless federated learning, Over-the-Air Federated Learning (OTA-FL) [74] leverages the superposition property of electromagnetic waves for model aggregation, accommodating heterogeneous device capabilities and channel conditions. The Mixed-Precision OTA-FL (MP-OTA-FL) framework [90] further extends this by enabling clients to participate at different computational precision levels, unlocking greater flexibility in balancing performance and energy efficiency, while addressing quantization overheads during OTA aggregation.

A central challenge in MP-OTA-FL lies in determining the optimal quantization (precision) level for each client. This decision is governed by multifaceted factors spanning hardware capabilities, model performance under quantization, client contribution to global model accuracy, and user satisfaction. Notably, these factors fall into two categories: intrinsic technical factors, such as device compute capacity, power states, quantized model performance, and extrinsic contextual factors, such as operational environment, usage patterns, user preferences, as illustrated in Fig. 4.1. The diversity and dynamism of these factors create significant obstacles for systematic collection and evaluation:

- **Comprehensive enumeration of extrinsic factors:** While intrinsic device characteristics can be directly measured, extrinsic factors such as user satisfaction, usage patterns, and operational contexts are dynamic and difficult to enumerate or quantify systematically.
- **Quantifying client contribution:** In federated settings, the contribution of each client to global model accuracy is linked to data quantity, quality, and distribution, which are typically private and thus inaccessible. Existing proxies often require client-specific evaluations, resulting in significant computational and communication overhead.

To address these challenges, this chapter introduces a Retrieval-Augmented Generation (RAG) based user profiling and precision planning framework. By integrating retrieval-augmented large language models (LLMs) with dynamic client profiling, the framework supports more nuanced and adaptive precision selection for each client.

Unlike traditional rule-based or fixed-form approaches, LLMs offer a unique advantage in extracting and reasoning over both structured device information and unstructured, context-rich user input through natural language interaction. This enables the framework to identify latent needs, nuanced preferences, and complex operational contexts that would otherwise be inaccessible or laborious to enumerate. The rapid extendability of LLMs, enabled by their zero-shot and few-shot generalization capabilities, allows the system to accommodate evolving profiling requirements or new user attributes without the need for laborious manual rule updates or retraining. This ensures the profiling process remains adaptable as user expectations, hardware, or regulatory requirements change over time.

The incorporation of a RAG architecture further grounds the generative capabilities of LLMs in curated domain-specific knowledge and historical user data, substantially reducing the risk of hallucination and improving reliability when operating in task-specific settings. This design allows the system to respond with contextually accurate recommendations by retrieving and conditioning on relevant quantization decisions, device states, and prior user feedback, rather than relying solely on model priors.

In addition, the modularity of this approach facilitates compliance with regulatory requirements such as the General Data Protection Regulation (GDPR) [4]. User data and decision records can be stored, processed, and erased in accordance with privacy policies, while the conversational profiling agent can be guided by explicit regulatory prompts or constraints, ensuring transparent data handling and user consent management throughout the process.

Key features of the framework include:

- A hybrid user interface that combines hardware profiling with an LLM-driven conversational agent, leveraging the LLM’s natural language understanding to uncover subtle user needs, preferences, and contextual insights which traditional interfaces may overlook.
- An RAG knowledge database that stores historical quantization decisions, contextual metadata, and user feedback. The RAG mechanism provides a memory-augmented reasoning process, where the LLM generates outputs by referencing precise, up-to-date information, mitigating typical hallucination risks associated with standalone LLMs and enhancing task fidelity.
- A continuous feedback loop where new user data and outcomes are incorporated into the database, enabling ongoing refinement of both user profiling and precision planning.
- Rapid extendability and regulatory compliance, with the framework designed to adapt its profiling logic in response to new requirements, user attributes, or legal constraints, ensuring privacy and transparency are maintained.



Figure 4.1: User satisfaction and client contribution potentials in federated learning vary with contextual factors such as usage patterns and operational environment.

Experiments using a mixed-precision FL voice assistant system demonstrate the proposed framework’s effectiveness, showing improved user satisfaction, energy savings, and global model accuracy. Furthermore, the open-source implementation¹ is provided for the community to use, adapt, and extend.

The remainder of this chapter reviews related work, presents the framework design, describes the experimental setup and results, and concludes with directions for future research.

4.2 Related Works

4.2.1 Mixed-Precision Federated Learning

Federated learning, since its introduction by McMahan et al. [73], has been widely applied in privacy-sensitive distributed computing scenarios such as healthcare, finance, and IoT. Mixed-precision computation has been widely employed in deep learning, improving efficiency in both training and inference [27]. The insight behind such design is that different types of layers in neural networks have different sensitivity to computation precision. Generalizing from this, mixed-precision OTA FL [90], with quantization overheads covered by OTA aggregation, offers a larger trade-off area between precision and performance, especially for those clients operating at the lowest precision levels due to most limited resources.

4.2.2 RAG-LLMs

RAG-LLMs, introduced by Lewis et al. [91], are a class of large language model that generate responses based on retrieved relevant information from external knowledge sources.

¹https://github.com/ntutangyun/user_in_the_loop_quantization_planning

RAG-LLMs have been deployed and achieved impressive performance in various NLP tasks such as user profiling for recommendation systems [92].

4.3 Methodology

4.3.1 Framework Overview

The proposed precision planning framework adopts a full-stack architecture as illustrated in Fig. 4.2. The framework comprises two core components: a user-profiling frontend featuring a chat interface and a backend server hosting an LLM-powered agent.

User Profiling Frontend A LLM agent-driven conversational interface for contextual factor discovery and satisfaction feedback collection. The conversation is tasked by the backend, and the primary tasks are as follows:

- At new device initialization, the user is prompted to provide perspective usage patterns and setup contexts, e.g., device location, intended usage scenarios, and user preferences.
- At the pre-aggregation stage, the user is queried for feedback on the performance of the operation, as well as potential context change, since the last feedback collection.
- In the case of changed hardware specifications, the user is prompted to update contextual factors and preferences.

Quantization Optimization Backend: A knowledge-enhanced processing stack containing:

- A RAG Context-Quantization-Feedback Database that archives precision decision history, including usage patterns, operational contexts, and corresponding user feedback.
- A RAG Hardware-Quantization-Performance Database that archives model performance (e.g., accuracy) with the associated hardware and precision level.
- LLM interview agent that interviews user preferences (e.g., user sensitivity for accuracy, response time and energy consumption) and usage patterns (e.g., noise level, usage frequency and type of interactions).
- Hardware specification extractor that collects device hardware information based on availability and user privacy settings. The parsed hardware specs are then used to query the knowledge database to estimate quantization-performance trade-offs on similar user hardware.
- Context quantization evaluation agent that estimates the potential client contribution and user satisfaction at available precision levels based on interpreted contextual factors and retrieved hardware capabilities.

Federated learning server: The FL server mainly coordinates the following processes:

- Client selection ²: in the default setting, clients are scheduled to participate update and aggregation process regularly. The backend will launch the user profiling and context quantization evaluation process for selected clients.
- Multi-client quantization planning: when all selected clients have completed the profiling and evaluation process, the server will filter the clients with precision levels with similar merits, and choose the optimal precision levels that maximise communication resource utilization in mixed-precision OTA aggregation.
- Mixed-precision OTA aggregation: The FL server aggregates model updates from clients of their current precision levels. After aggregation, the server will send the updated model back to clients along with the optimal precision levels for the next round for them to quantise the received model accordingly.

²Client selection methods are not the focus here, the framework should work with any approach, such as random selection, connection quality prioritised, data volume prioritised.

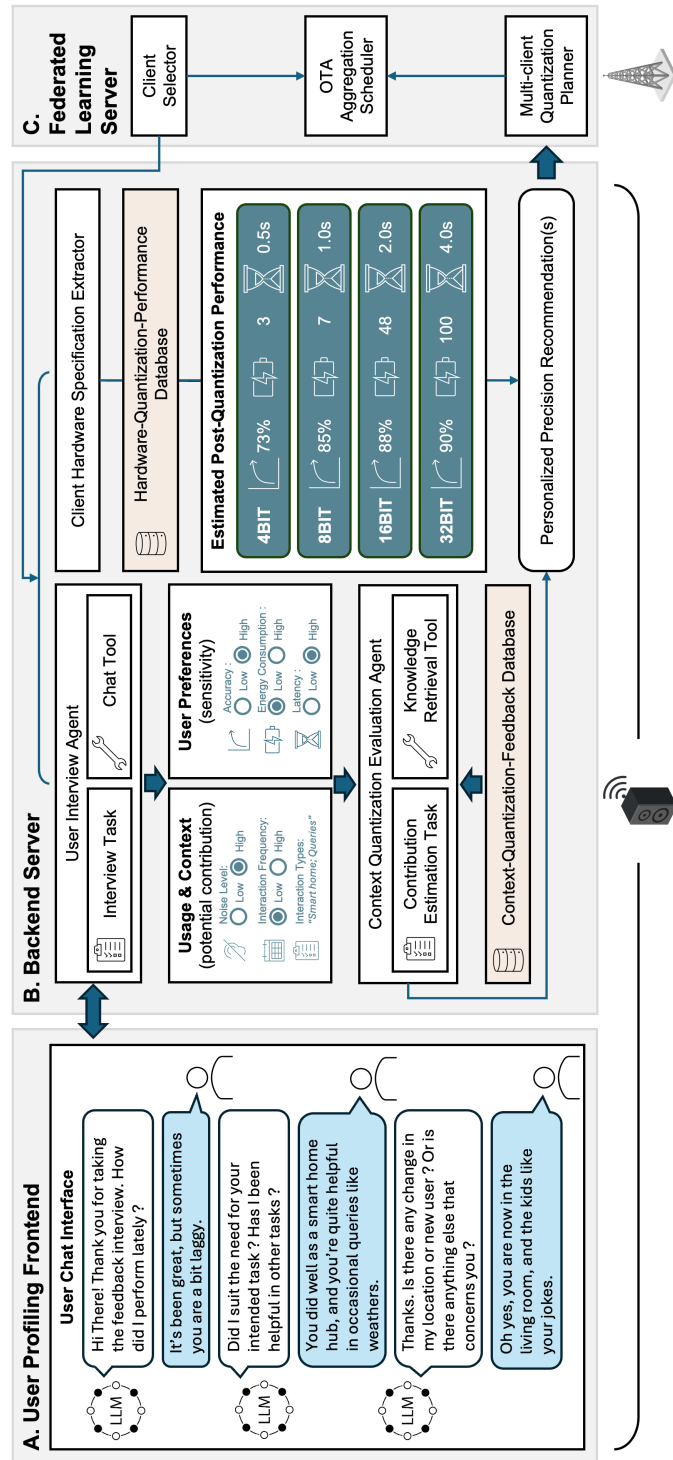


Figure 4.2: User-in-the-loop Quantization Planning Framework Overview. The aim is to collect the user’s feedback on the T round and select the optimal quantization level for the $T + 1$ round for the federated learning process.

Table 4.1: Examples of Contextual Factors and Inferable Factors

Contextual Factor	Inferable Factor	Examples
Device location	Input noise level	Bedroom → Low noise; Living room → High noise
Interaction time	Input noise level, data quantity	Daytime → High noise, High quantity; Nighttime → Low noise, Low quantity
Interaction frequency	Data quantity	High frequency → High quantity
Task Type	Data distribution	Smart home hub → Short requests

Table 4.2: Word Accuracy of the Global Model after 100 Communication Rounds by Classes with Different Strategies.

FL Strategy	Entertainment	Smart Home	General Query	Personal Request
a. Sample Equal (Baseline)	71.2	68.5	78.6	69.4
b. Class Equal	68.7 (-2.5%)	77.3 (+8.8%)	76.4 (-2.2%)	74.8 (+5.4%)
c. Majority Centric	75.5 (+4.3%)	66.2 (-2.3%)	81.9 (+3.3%)	67.8 (+1.6%)

Compared to the default strategy, with b) class equal strategy, biased towards minority classes, our framework trades 2% accuracy of the majorities for 5% of that of the minorities; while with c) majority centric strategy, our framework extended the accuracies of majority classes by 4% with 3% lower accuracies for minority classes.

4.3.2 RAG-based User Profiling

The RAG-based user profiling process collects and infers user preferences on performance and operational contexts through a user-friendly conversational interface.

4.3.2.1 Contextual Factors

The user preference for performance consists of three metrics: accuracy, energy consumption, and latency. These metrics are quantised by retrieving similar user cases from the knowledge database, based on the user's current feedback, operational contexts, and the estimated performance of their devices at the current precision levels. In comparison to conventional form-based feedback collection, RAG-LLM can analyse the user's sensitivity in these metrics through wording nuances in their feedback, prioritise primary user concerns in performance, and hence, facilitate accurate adjustments to meet their expectations. In addition, the RAG-LLM can analyse and link these sensitivities to operational contexts, as the same user could have different expectations and sensitivities in different scenarios.

Apart from supporting performance feedback, operational contexts are also indicators of the potential contribution of the client to the global model, see Table 4.1 for examples of such contextual factors and their potential effects. Factors such as data quality, quantity and distribution, which can be inferred from these contextual factors, are essential for the client contribution estimation. FL service providers can use these inferred factors to estimate potential client contributions at different precision levels, and hence, select the optimal precision level for each client based on their learning strategies.

4.3.2.2 RAG Database and LLM Integration

To support the LLM agents, we build a Context-Quantization-Feedback database, which stores the feedback from users of different contextual factors on performance at different quantization levels. When user feedback and contexts are collected via the chat interface, the LLM agent will retrieve similar user cases from the database, and estimate the user satisfaction, preference and client contribution at different precision levels based on the retrieved cases.

4.3.2.3 User Profiling Pipeline

The user profiling pipeline consists of the following steps:

1. **Hardware specification extraction:** The backend extracts the hardware specification of the user device, including processor specs, RAM size, and power states.
2. **Hardware quantization performance trade-off retrieval:** The backend queries the knowledge database for the quantization-performance trade-off on the same or similar hardware.
3. **User interview feedback collection:** The agent prompts the user to provide feedback on the current performance and potential context changes since the last feedback collection, see Fig. 4.2-A for a chat example.

4. **Contextual factor inference:** The LLM agent infers user preferences and contexts from past conversations.
5. **User preference and contextual factor retrieval:** The agent retrieves similar user cases from the knowledge database with inferred user preferences and contexts.
6. **User satisfaction and client contribution estimation:** The agent estimates the potential client contribution and user satisfaction at available precision levels based on retrieved contextual factors and hardware capability.

4.3.3 Context-Quantization Evaluation

We define a reward-penalty model for determining the optimal quantization level for each client in a federated learning (FL) setting. The model considers multiple factors, each with an associated user-defined sensitivity weight. Assume:

- $\mathcal{F}$: Set of factors (e.g., accuracy, energy cost, latency).
- q : Quantization level assigned to a client.
- w_f : Sensitivity weight of factor $f \in \mathcal{F}$, where $\sum_{f \in \mathcal{F}} w_f = 1$.
- $R_f(q)$: Reward obtained from operating at quantization level q for factor f (e.g., improved accuracy).
- $P_f(q)$: Penalty incurred by operating at quantization level q for factor f (e.g., energy consumption).
- C_q : Contribution multiplier for potential client contribution operating at quantization level q .

Initialization: $R_f(q)$, $P_f(q)$ and C_q are initialised with experiments on smaller FL simulations of the contextual factors. $R_f(q)$, $P_f(q)$ are set by normalizing raw factor measurements across the quantization grid, and C_q is initialised with conventional methods, which are effective for smaller federations.

Then, the total reward and total penalty for quantization level q are computed as the following weighted sums:

$$R_{\text{Total}}(q) = C_q \cdot \sum_{f \in \mathcal{F}} w_f \cdot R_f(q)$$

Equation 10: Total Reward Calculation for Context-Quantization Evaluation

$$P_{\text{Total}}(q) = \sum_{f \in \mathcal{F}} w_f \cdot P_f(q)$$

Equation 11: Total Penalty Calculation for Context-Quantization Evaluation

The optimization goal is to select the quantization level q that maximises the Satisfaction Score defined as the difference between total reward and total penalty:

$$\text{Satisfaction Score}(q) = R_{\text{Total}}(q) - P_{\text{Total}}(q)$$

Equation 12: Satisfaction Score Calculation for Context-Quantization Evaluation

$$q^* = \arg \max_q (\text{Satisfaction Score}(q))$$

Equation 13: Optimal Quantization Level

Table 4.3: Smart Voice Assistant Data Distribution

Category	Entertainment	Smart Home	General Query	Personal Request
Percentage	32.7%	16.0%	31.9%	19.4%

4.4 Experiments

4.4.1 Experimental Setup

We validate our proposed RAG-based precision planning framework on a federated smart voice assistant system with the Automatic Speech Recognition (AER) task. The federation consists of 100 simulated clients with diverse hardware capabilities and Gaussian distributed sensitivity to performance factors including accuracy, energy savings, and latency. We define the following experimental settings:

Dataset and Model: The model structure is DeepSpeech2 [93], and the federated model is trained for 100 communication rounds. For client datasets, we filter the Common Voice dataset [94] with keywords related to the four main uses of smart voice assistants, *Entertainment*, *Smart Home*, *General Query* and *Personal Request*. We define these categories and their distribution, see Table 4.3, based on the usage statistics from the PWC research report [95].

Metrics and Comparison: To showcase the advantage of our RAG-based user profiling precision planning framework, we compared it on the same federated learning system but with a unified standard precision planner, i.e., divide users in tiers by their hardware capabilities and assign the same precision level to each tier regardless of user preference and operational contexts. We measure the following metrics:

- **User Satisfaction Score:** the user satisfaction score defined in Equation 12.
- **Relative Energy Cost:** we do not directly measure energy costs, instead, we measure the relative energy cost compared to the highest available precision level, and therefore the relative energy cost is a percentage below 100%.
- **Final Global Model Accuracy:** The final word accuracy of the global model after 100 communication rounds.

4.4.2 Results

4.4.2.1 User Satisfaction versus Energy Cost

Our RAG-based user profiling precision planning framework generates personalised standards based on user preference and their operational contexts, resulting in a more accurate satisfaction estimate compared to the FL system that plans precision levels with unified standards, and the average satisfaction score is 10% higher (0.66 compared to 0.60) while saving about 20% energy. We also tested that when energy savings is the top priority of the mixed-precision FL system, our framework can trade 22% average satisfaction score (0.47 compared to 0.60) for a total of 28% energy savings.

Profiling users to calibrate their computation precision according to their sensitivities to accuracy, energy cost, enables their client to operate more efficiently, reducing energy consumption while maintaining the aggregate satisfaction level, but this targeted allocation amplifies heterogeneity in individual scores, resulting in the higher variance in both personalised standards and energy saving prioritised standards.

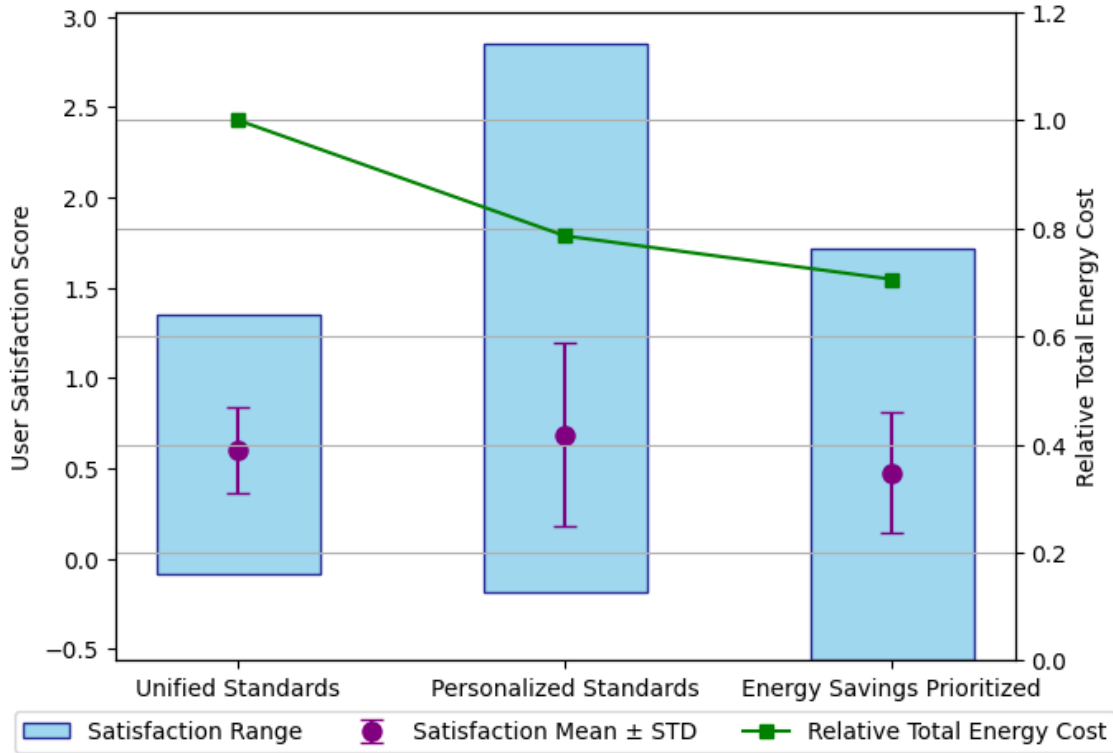


Figure 4.3: Distribution of User Satisfaction Scores and Relative Energy Cost. Compared to planning precision levels with unified standards, personalised standards can achieve 10% higher average satisfaction score, and 20% of energy cost. When prioritise the federated system towards energy savings, 22% satisfaction score can be traded for a total of 28% energy saving.

4.4.2.2 Global Model Performance

Estimation of potential client contribution to the global model depends on the training strategy. We experimented with three different strategies with our framework: a) default FedAvg [73] i.e. treat every sample equally; b) class equal strategy, attempts higher precision levels to samples of minority classes; c) majority centric strategy, attempts higher precision levels to samples of majority classes. Our RAG-based framework can estimate data distribution via contextual factors without breaching user privacy. Refer to actual data distribution in Table 4.3, see Table 4.2 compared to accuracies of FedAvg [73], our framework improved the accuracies of minority classes (smart home and personal request) and majority classes (entertainment and general query) with the corresponding biased strategies.

4.5 Conclusion

This chapter presented a Retrieval-Augmented Generation (RAG) based user profiling framework for precision planning in Mixed-Precision Over-the-Air Federated Learning (MP-OTA-FL) systems. By leveraging RAG-powered Large Language Models (LLMs), the proposed framework enables dynamic and context-aware client profiling, facilitating adaptive quantization-level selection that accounts for heterogeneous device capabilities, operational contexts, and user preferences.

Aligned with the aims and objectives of this thesis, the framework directly addresses several core research questions. It systematically incorporates contextual factors into federated learning, harnesses client heterogeneity for improved efficiency and model accuracy, and provides a scalable approach to integrating individual contributions and user satisfaction into the FL process. The adoption of LLMs, further strengthened with RAG for task-specific reliability, brings notable advantages in extensibility, rapid adaptation to new profiling requirements, and regulatory compliance, as well as minimizing hallucination risks.

Experimental results demonstrate that this approach delivers substantial gains in user satisfaction, energy efficiency, and global model performance compared to traditional unified quantization strategies. The open-source implementation is intended to catalyse further research and community-led innovation in human-centred, context-aware federated learning.

Through these contributions, this chapter advances the thesis objective of bridging efficiency optimisation, user-centric adaptation, and robust contextual integration within heterogeneous federated learning environments.

Chapter 5

Rowhammer Attack via PPO Generated Physical Interference on Federated Learning Clients

5.1 Introduction

Federated Learning (FL) has emerged as a cornerstone in decentralised machine learning, enabling collaborative model training across distributed client devices while preserving user privacy by avoiding raw data sharing [3]. With the adoption of FL in large-scale applications such as automatic speech recognition (ASR), new performance optimisation techniques have been integrated, including sparse gradient updates, page-locked (pinned) memory, huge pages, and Remote Direct Memory Access (RDMA). While these optimisations dramatically increase training efficiency, they also expose FL systems to hardware-level vulnerabilities, most notably the Rowhammer attack [96].

The Rowhammer vulnerability arises from the electrical interference induced by rapidly and repeatedly activating adjacent rows within Dynamic Random Access Memory (DRAM), potentially causing bit-flips in memory cells. Despite modern countermeasures such as Target Row Refresh (TRR), empirical studies show that recent DRAM modules remain susceptible to sophisticated Rowhammer exploits [97], [98]. Historically, Rowhammer attacks have assumed a degree of system-level access or the ability to execute code on the victim hardware [99], [100], [101]. However, an emerging class of threats exploits physical adversarial perturbations, carefully crafted electromagnetic or acoustic signals, to influence the sensor inputs of edge devices, bypassing conventional access control mechanisms [102], [103].

Motivated by this intersection of hardware vulnerabilities and physical-domain attacks, this chapter investigates a new, stealthy Rowhammer attack vector targeting FL deployments. Our approach employs a reinforcement learning-based Proximal Policy Optimization (PPO) agent that generates adversarial physical waveforms. These waveforms target the sensors of FL client devices, such as the microphone of smart speakers or phones, prompting them to produce clustered and synchronised model updates. When transmitted to the FL server, these updates exploit memory access patterns created by efficiency optimisations such as RDMA and pinned memory, increasing the likelihood of Rowhammer-induced bit-flips on server-side DRAM. Notably, this threat vector does not

require explicit control over victim hardware or direct digital intrusion to issue memory related commands, thereby significantly expanding the FL attack surface.

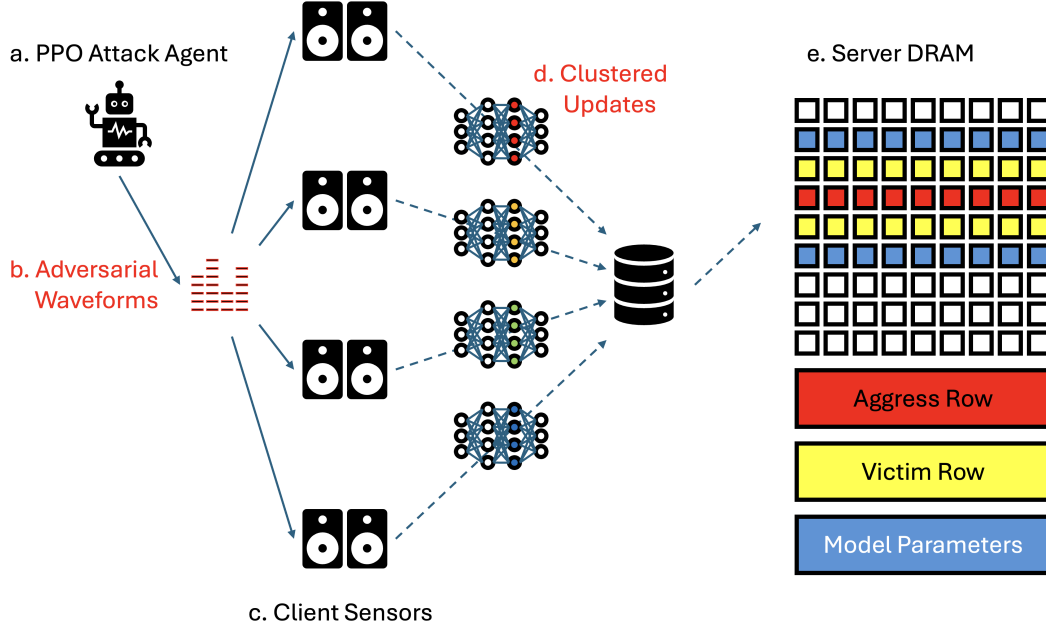


Figure 5.1: Framework of our proposed threat vector. a) A PPO agent generates b) adversarial waveforms that interfere with the inputs of c) client sensors in FL, causing clients to send d) clustered updates, which can trigger Rowhammer attacks on e) server DRAM.

In this chapter, we present:

1. **A novel Rowhammer attack methodology**: Introducing a physical-domain-driven attack vector targeting FL servers, which undercovers in the aggregation process, bypassing traditional requirements for access to issue memory commands.
2. **A PPO-based adversarial waveform generation framework**: A two-stage reinforcement learning pipeline designed to optimise physical perturbations for maximally clustered and sustained client updates, enabling targeted bit-flip induction on remote DRAM.
3. **Comprehensive vulnerability and effectiveness analysis**: Experimental evaluations using large-scale FL scenarios with modern ASR models, revealing the impact of efficiency optimisations on system susceptibility.
4. **Practical mitigation strategies**: Discussion of potential defences, including adaptive memory allocation, perturbation detection, and memory access restrictions.

The remainder of this chapter is organised as follows. Section II presents essential background on DRAM architecture, Rowhammer vulnerabilities, federated learning security, and physical adversarial attacks. Section III formalises the threat model and describes the PPO-based attack framework. Section IV details the experimental setup and

evaluation methodology. Section V presents and analyses empirical results. Section VI discusses mitigation strategies, and Section VII concludes with future research directions.

5.2 Background and Related Work

5.2.1 DRAM Architecture and Rowhammer Attack

Modern DRAM modules are organised by different levels. The DRAM module can be classified into single-rank, dual-rank, and multiple-rank according to the number of ranks. Each rank includes multiple chips that share the same data bus. Further, every chip contains multiple banks, which are organised into bank groups. It is necessary to identify the bank group number when accessing one specific bank. A DRAM bank consists of an array of memory cells, where all cells in a horizontal direction are collectively referred to a DRAM row, while those in a vertical direction are referred to a DRAM column. Each cell represents one bit, according to different memory strategies, a fully charged cell representing a value of "1" is called true-cell, whereas the opposite is referred to as an anti-cell. The specific DRAM organization has shown in Fig. 5.2. When data requires to be read from or written to DRAM, the memory controller (MC) first will issue an **ACT** command to open one row. The entire row of data, including the accessed data, will be driven by the wordline and retrieved into the sense amplifier (or row buffer). Then, bitlines will process specific columns in the row buffer to complete the exact read or write operations. Finally, the MC will issue a **PRE** command to close the DRAM row.

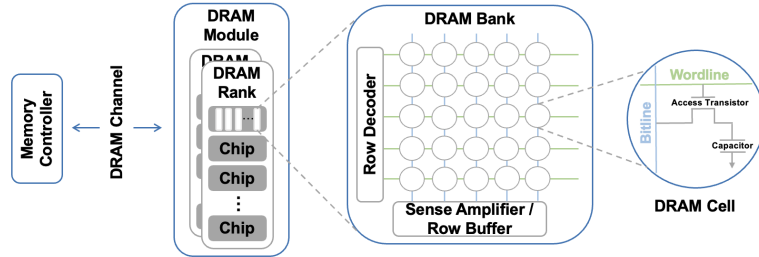


Figure 5.2: Hierarchical organization of a Dynamic Random-Access Memory (DRAM) system, depicting the structural composition from the memory controller and channel architecture through modules, ranks, and chips, down to individual bank-level components (row decoder, sense amplifier/row buffer) and the underlying DRAM cell structure (wordline, access transistor, capacitor). The schematic illustrates the multi-tiered interconnectivity and functional partitioning critical to DRAM operation, spanning system-level interfaces to transistor-level charge storage mechanisms.

For DRAM access, whether it is a read or write operation, the MC issues both an **ACT** command and a **PRE** command for accessed row. The fundamental cause of the Rowhammer attack lies in the frequent issuance of **ACT** commands. When a victim row's adjacent rows (aggressor rows) are repeatedly activated, the voltage of wordline driving the aggressor row rapid and frequent changes. It then induces electromagnetic coupling with the memory cells in the victim row, altering their charge state. It should be

noted that even if only a single aggressor row is used for the Rowhammer attack, it still requires the coordinated activation of two rows within the same bank. This is because if only one aggressor row is accessed repeatedly, the MC will not precharge this row before completing all operations, making the attack ineffective.

Furthermore, the reason why the Rowhammer attack requires rapid and frequent access to the aggressor row lies in the DRAM refresh mechanism. DRAM performs periodic refresh operations to maintain the charge state of all memory cells. Commonly, this refresh period is 64 ms or 128 ms, but under high-temperature conditions, it may be reduced to 32 ms. In each refresh period, the MC typically issues 8192 **REF** commands, which sequentially refresh all rows on the DRAM module. Taking a 64 ms refresh period as an example, the average interval between each REF command is 7.8 μ s. If the number of activations on the aggressor row fails to effectively cause charge leakage in the victim row before its next refresh, the attacker must accumulate more activations again in next refresh period to execute Rowhammer attack.

The factors influencing the Rowhammer attack includes manufacturing, data pattern, and access pattern. Typically, there are some inherently vulnerable memory cells in each DRAM chips are more prone to charge leakage compared to normal cells, regardless other factors. The data pattern refers to the scenario where the Rowhammer attack is more likely to succeed when the charge states of memory cell in the aggressor row and victim row are different [104]. Although adversary attempting to tailor the data in aggressor row based on the victim data may be affected by anti-cells, existing research has proved that the proportion of anti-cells with the same chip is very small [96]. The access pattern includes single-sided, double-sided, and many-rows Rowhammer attack [97], [105]. The double-sided Rowhammer, compared to a single-sided, involves two aggressor rows targeting the same victim row. This type of attack is generally more effective than single-sided Rowhammer. However, it requires more detailed memory mapping preparation. On the other hand, the many-rows Rowhammer is often used to bypass the TRR protection mechanism.

The key points and progress details of executing and exploiting a Rowhammer attack are as follows:

1. **Selecting Rowhammer access patterns:** The specific DRAM modules of target system guides the choice of Rowhammer attack access patterns, primarily depending on whether the victim chips have TRR enabled. For early DDR3, both simple single-sided and more effective double-sided Rowhammer are ideal access patterns. However, for modern DDR4 chips, where TRR is widely implemented, many-rows Rowhammer and its variants must be used instead [97], [106]. This is because TRR protection mechanism rely on its internal sampler [107]. By using multiple dummy rows, attackers can successfully overload the size of sampler, preventing the actual aggressor rows from being affected by extra refresh operations. It is worth noting that error-correcting code (ECC) has been proven ineffective against this attack [108], as efficient access patterns such as double-sided Rowhammer can also overwhelm ECC mechanisms [109].
2. **Determining Address Mapping:** Since Rowhammer attack is triggered by adjacent rows of victim data, determining the addresses of aggressor rows is crucial. However, the virtual memory address used in application do not correspond to actual DRAM address. The process of locating the correct address typically involves

two mapping stages [110]. The first is application-level virtual address to physical address. In operating system, the memory management unit (MMU) in the CPU, maps contiguous virtual addresses to non-contiguous physical addresses in a non-linear manner. In some operating systems, such as Ubuntu, the mapping can be retrieved via `/proc/pid/pagemap`. The next stage involves physical address to DRAM address mapping, which varies by CPU architecture and DRAM manufactory. This mapping is undocumented and common approaches to reverse-engineering it relies on side-channel analysis [98], [111], [112], [113]. A key point to mention is that in a double-sided Rowhammer scenario, a common strategy for finding usable contiguous physical addresses is to use huge pages.

3. **Avoiding Cache Effects:** Frequently accessed data will be stored in the cache to avoid CPU complex operations. Thus, Rowhammer attack requires to use specialised CPU instructions to bypass cache to achieve high-frequency activations of aggressor rows in real attack scenarios. Currently, the main instructions used to achieve this purpose include `clflush` [114], [96], `clflushopt` [115], and `invd` or `wbinvd`[110]. In a carefully crafted attack sequence, incorporating the aforementioned instructions can help the attacker bypass the cache. Sometimes, this also requires the use of memory barriers such as `mfence`, `lfence`, and `sfence`, but they may not achieve the optimal activation rate. Listing 5.1 illustrates a Rowhammer attack sequence utilising the `clflushopt` instruction and memory barrier. Additionally, there are some attacks based on cache line conflicts to bypass [108], [116].
4. **Ensuring Effective Activation Rate and Count:** After determine all above conditions, attackers should ensure enough activations to induce bit-flips. There are two key factors to ensure the successful execution of the attack. One is maintaining an effective activation rate. Even though the standard tRC (Row Cycle Time) is only 46 ns, the average interval between two activation instructions should be around 220 ns, considering the influence of other instructions. The other is accumulating a sufficient number of activations to ensure that the aggressor rows are activated enough times to induce bit flips. Additionally, when TRR are present, the dummy rows used to bypass TRR must also be accounted for in the total activation count. The total number of activations should not exceed the maximum allowable activations within a single refresh period (64 ms) to avoid DRAM refresh interventions.

```

1      loop:
2          mov (X), %eax
3          mov (Y), %ebx
4          clflushopt (X)
5          clflushopt (Y)
6          mfence
7          jmp loop

```

Listing 5.1: Pseudo code of Rowhammer attack utilising the `clflushopt` instruction

5.2.2 Federated Learning

Federated Learning (FL) is a distributed and decentralised machine learning paradigm that enables collaborative model training across multiple devices while retaining the raw data

locally [3]. Unlike conventional centralised training schemes, FL enhances data privacy by transmitting only model updates—such as parameters or gradients—to a central server for aggregation. This decentralised framework is particularly advantageous for use cases where data confidentiality and regulatory compliance are paramount (e.g., healthcare, finance, or mobile devices). Moreover, the scalability of FL allows diverse and expansive datasets to be leveraged by increasing the number of participating clients, thereby improving generalization across heterogeneous data distributions. The typical FL process involves the following steps:

1. **Local Training:** Each client trains a local model using its own private dataset.
2. **Model Aggregation:** In each communication round, a designated subset of clients transmits model updates (e.g., weights or gradients) to the central server. These updates are then aggregated into a global model using algorithms such as FedAvg [3].
3. **Redistribution:** The aggregated global model is redistributed to the participating clients, which then continue local training in the next round.

5.2.2.1 Security Threats in Federated Learning

Federated learning (FL) systems are vulnerable to a variety of security threats that arise from both adversarial machine learning (ML) techniques and conventional communication/network-based attacks. The former category includes model poisoning, backdoor, and inference attacks, while the latter mainly consists of eavesdropping and distributed denial-of-service (DDoS) attacks. In this paper, we focus on an attack vector more closely aligned with adversarial ML methodologies.

- **Poisoning Attacks:** Poisoning attacks are a specialised form of injection attack that targets ML models. Within FL, adversaries exploit malicious or compromised clients to manipulate the training process, which can degrade the global model’s performance or introduce concealed backdoors. Specifically, *data poisoning* involves injecting or modifying training samples (often through mislabeled or corrupted data), whereas *model poisoning* targets model updates (e.g., gradients or weights) to undermine accuracy or embed hidden behaviors.
- **Backdoor Attacks:** Backdoor attacks implant triggers in an ML model during training, enabling the adversary to covertly control the model’s predictions. By submitting local updates with deliberately engineered triggers or patterns, malicious clients can force the global model to behave normally for most inputs, yet generate attacker-controlled outputs for specific trigger inputs. These attacks are especially dangerous because they usually go undetected until the trigger is activated.
- **Inference Attacks:** Inference attacks exploit partial information shared in FL systems to reveal or deduce private data from participating clients. By analyzing the distributed global model, adversaries may ascertain membership (i.e., whether a particular sample was part of the training set) or other sensitive attributes. This highlights the necessity for more robust privacy-preserving techniques in FL.

5.2.2.2 Efficiency Optimization

Federated Learning (FL) systems, especially those deployed in high-performance computing (HPC) facilities and data centers, often incorporate efficiency-oriented techniques such as sparse gradient updates, page-locked memory, and Remote Direct Memory Access (RDMA) for accelerated communication and reduced overhead.

- **Page-lock memory**, also referred to as pinned-memory, marks a region of memory as 'pinned', keeping the section of memory resident in RAM at all times, hence improves the efficiency of program by avoiding delays caused relocation or page-out of memory. In modern GPU based deep learning scenario, pinned-memory locks a section of DRAM to enable asynchronous transfers between DRAM and VRAM on GPUs, which significantly improves efficiency in data loading and training throughput.
- **Huge Pages** are a memory management technique that allows the operating system to use larger page sizes than the default 4KB, reducing the number of page table entries, which can improve translation lookaside buffer (TLB) hit rates and reduce the overhead of memory management. This can lead to performance improvements in applications that have large memory footprints and high memory access rates, such as high-performance computing (HPC) applications. For AI workloads, variables such as model parameters, data are often too large for default page size, hence huge pages can significantly improve the performance of AI workloads.
- **DMA and RDMA** Direct Memory Access (DMA) is a feature of computer systems that allows certain hardware subsystems to access main system memory (DRAM) independently of the central processing unit (CPU). Remote Direct Memory Access (RDMA) is a specialised form of DMA that enables high-speed data transfers between networked computers by allowing data to be transferred directly from the DRAM of one computer to the DRAM of another without involving the CPU. RDMA is commonly used in high-performance computing (HPC) and data center environments to accelerate data transfers and reduce latency. By bypassing the CPU and operating system, RDMA can achieve latencies as low as 1-2 microseconds and bandwidths exceeding 200 Gbps, making it ideal for applications that require high-speed, low-latency data transfers, such as machine learning training and distributed computing. In FL systems, RDMA can significantly reduce communication overhead and improve scalability by enabling efficient aggregation and distribution of model parameters between central servers and distributed clients. By minimizing synchronization delays and network bottlenecks, RDMA can enhance the overall training performance and accelerate convergence in FL systems, particularly in large-scale deployments involving hundreds or thousands of clients.

5.2.3 Physical Adversarial Attacks

Deep learning has revolutionised a wide range of applications, extending from computer vision to audio processing, enabling technologies such as autonomous driving, surveillance, biometric authentication, robotics, and voice-controlled systems. However, the reliability and robustness of these systems have been called into question due to adversarial

attacks. These attacks involve purposefully crafted perturbations—often imperceptible to human observers, that can mislead or corrupt a model’s predictions. Originally explored in the digital domain, adversarial examples have since transitioned into the physical world, maintaining their deceptive properties even after real-world transformations like printing, photographing, or acoustic playback [117], [118], [119].

In the context of computer vision, early works by Szegedy et al. [117] and Goodfellow et al. [118] unveiled the vulnerability of neural networks to subtle pixel-level perturbations. Kurakin et al. [119] and Eykholt et al. [102] extended these findings into the physical domain, revealing that adversarial examples remain effective when transformed into physical objects or printed images. Consequently, research has demonstrated that adversarial patches [120], malicious alterations to wearable items, and tampered traffic signs [121] can consistently deceive image classification and object detection models. Such attacks often operate under the radar of human vision and can be engineered to withstand environmental noise and viewpoint changes [122], [123], [124].

The concept of adversarial attacks has also proliferated into the audio domain. Speech recognition and voice-activated systems are increasingly critical in applications ranging from smart home devices to security systems. Adversarial perturbations can be embedded into audio signals to cause automatic speech recognition (ASR) models to misinterpret commands or produce incorrect transcriptions. These perturbations can be made so subtle that they remain inaudible to humans. For instance, Carlini and Wagner [103] demonstrated targeted adversarial examples against speech-to-text systems, while Yuan et al. [125] and Qin et al. [126] introduced methods for embedding covert commands within naturalistic audio clips. As with visual adversaries, these audio-based attacks persist in the physical domain, maintaining their effectiveness when played through speakers and recorded by microphones—thus bypassing human scrutiny and threatening the security of voice-controlled systems.

The subtlety of these adversarial perturbations in both vision and audio modalities, and their ability to survive real-world conditions, pose a significant threat to the integrity, safety, and trustworthiness of deep learning applications. While human perception often remains untouched, machine learning models suffer degraded performance, leading to dangerous misclassifications or unauthorised actions. As a result, research efforts increasingly focus on developing robust defence mechanisms, detection strategies, and theoretical understandings of how adversarial examples can transfer seamlessly from the digital to the physical domain [121]. Addressing these challenges is crucial for safeguarding the future of AI-driven systems and ensuring their reliability in real-world conditions.

5.3 Attack Vector Formulation

5.3.1 Inspiration

In computation intensive scenarios like modern AI systems, optimizations meant to enhance efficiency can inadvertently introduce vulnerabilities exploitable by Rowhammer-style attacks. We identify three specific properties induced by optimizations commonly found in federated learning (FL) systems, particularly suited for exploitation:

1. **Stability in Physical Address:** In FL workflows, training data and model paramet-

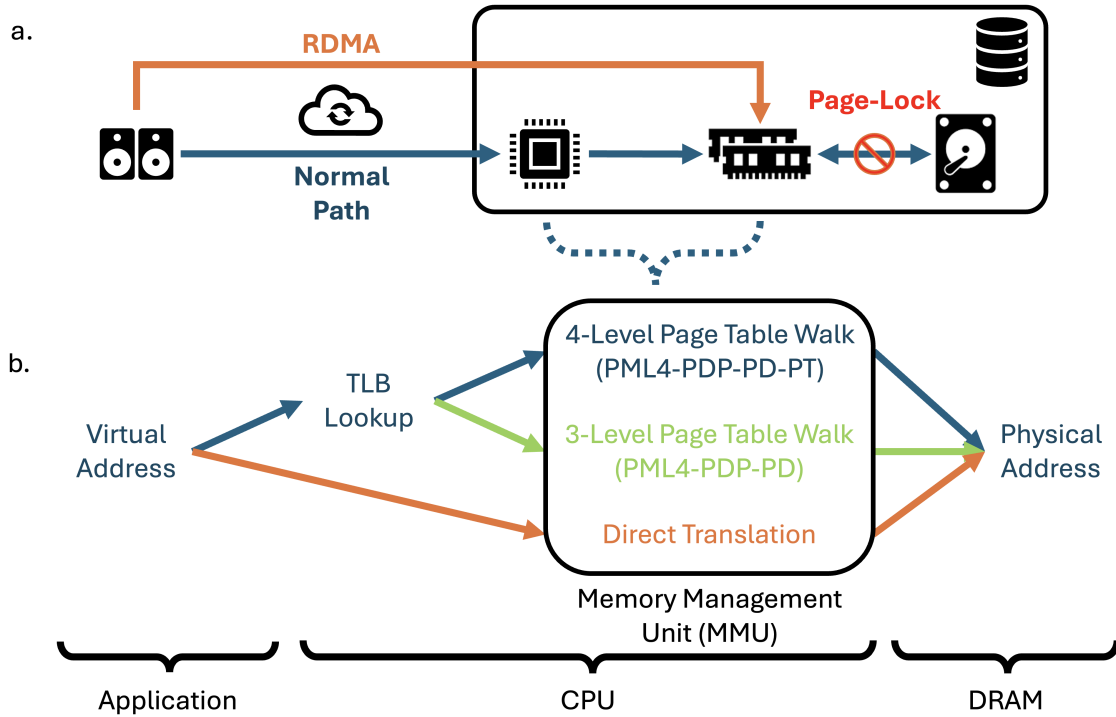


Figure 5.3: DRAM access pathways and processes of FL. a) DRAM access pathways. For normal access, client updates go through network to the CPU of server before getting to the DRAM of the server, while RDMA allows clients to bypass the CPU; b) DRAM access processes. Normally, when mapping virtual addresses of applications to physical addresses on DRAM, the CPU lookup TLB, and walk through the page table if TLB miss. Using **Huge Page** reduces a level in page table walk, and **DMA/RDMA** bypass both TLB lookup and page table walk. **Page-lock** prevents variables from offloading from DRAM to disk.

ers are assigned to locked huge pages with DMA enabled to maximise data transfer efficiency between CPU and GPU. By allocating the buffer as a huge page, fewer TLB entries are needed, reducing address-translation overhead, while page-locking guarantees that the memory cannot be paged out or moved. DMA then enables the device to read or write directly to this pinned, contiguous memory region, leveraging consistent physical addresses (or IOMMU translations) without interruption. When employing these three techniques simultaneously, the operating system ensures that the allocated memory region remains both physically contiguous and non-swappable, effectively 'locking in' a stable translation to a fixed physical address, especially when considering a short period of time such as the 64ms window for typical rowhammer attacks. Therefore, potential attacker can lock the target row via the FL workflows, e.g., locking a physical DRAM row via specific continuous parameters of the global model.

2. **Minimal Cache Interference:** FL systems that employ DMA or RDMA and non-temporal memory accesses to optimise distributed parameter exchange enable more frequent and predictable direct DRAM accesses. RDMA's zero-copy semantics let attackers initiate high-frequency remote memory reads/writes that bypass CPU caches on a targeted node's DRAM. Additionally, sparse updates—commonly used in FL to reduce bandwidth—focus access patterns on a subset of model parameters. By continuously hammering these frequently accessed parameters or their associated pinned-memory regions, attackers can ensure sustained direct DRAM access while minimizing cache interference. Non-temporal instructions, often integrated for performance reasons, further reduce cache pollution, promoting repeated DRAM row activations.
3. **Sustained High Rate of Row Activations:** Sustaining the high access rate needed for Rowhammer is eased by FL's performance-driven environment. The use of pinned memory, RDMA, and HPC optimizations (such as NUMA-aware memory placement and hardware-accelerated data transfers) can significantly reduce latency and jitter in memory accesses. By exploiting these properties, attackers can maintain tighter loops and higher hammering frequencies. Sparse gradient updates, which limit data movement to a known set of parameters, also streamline repeated row activations by ensuring that a small, known set of memory pages remain hot targets, further simplifying and speeding up hammering sequences. Specifically, the gather-scatter mechanism, commonly used to efficiently aggregate sparse gradient updates by collecting and redistributing only the most significant gradients (e.g., the top 0.1%), further minimises data transfer overhead. This concentration on a small subset of parameters not only enhances communication efficiency but also maintains consistent memory access patterns, reinforcing the repeated activation of targeted memory rows required for effective Rowhammer exploitation.

5.3.2 Evaluation Metrics

5.3.2.1 Notations

To evaluate and validate the proposed attack strategy, we map the repeated rate of sparse client updates to the number of Rowhammer operations required for DRAM. We define

the following terms:

- $H_{\max}(p)$: The maximum number of sparse updates processed within a single DRAM refresh period for an update fraction p that leads to update size $S_{\text{update}}(p)$.
- p : The fraction of model parameters updated in each sparse update.
- $T_{\text{flip}}(r)$: The Rowhammer threshold, measured as the number of updates required to induce a bit-flip in row r .
- $\overline{T_{\text{flip}}}$: The average Rowhammer threshold across all rows.

5.3.2.2 Reinforcement Learning Performance Metrics

We define two complementary metrics to quantify the clustering and consistency of parameter updates induced by the RL agent:

1. **Repeated Update Rate (RUR)** measures the persistence of updates across consecutive episodes:

$$\text{RUR} = \frac{\sum_{t=1}^{T-1} |U_t \cap U_{t+1}|}{\sum_{t=1}^{T-1} |U_t|},$$

Equation 14: Repeated Update Rate (RUR)

where U_t denotes the set of parameter indices, the location of the parameter in the model tensor, updated during episode t . Higher values indicate more consistent targeting of the same parameters.

2. **Cluster Density (CD)** assesses the spatial concentration of updates within the global parameter space. Let $k = |U_t|$ be the number of updates in an episode and $m = \lceil 0.9k \rceil$. Sort the update indices as $i_{(1)} \leq i_{(2)} \leq \dots \leq i_{(k)}$ and define

$$L = \min_{1 \leq j \leq k-m+1} (i_{(j+m-1)} - i_{(j)} + 1), \quad \text{CD} = \frac{L}{|\theta|}.$$

Equation 15: Cluster Density (CD)

Lower CD indicates that a large fraction of updates lie within a narrow index span, conducive to inducing Rowhammer effects.

5.3.2.3 Bit-Flip Validation

Given the DRAM bandwidth BW , the refresh period $\Delta t_{\text{refresh}}$, and a fraction p that determines the size $S_{\text{update}}(p)$, the maximum number of updates is computed as where $S_{\text{update}}(p)$ is the size of a sparse update when a fraction p of parameters is transmitted.

To assess the practical feasibility of inducing Rowhammer conditions, we estimate the number of expected activations on a single targeted DRAM row (or closely spaced group of rows) based on empirically measured update clustering behaviors.

$$H_{\max}(p) = \frac{BW \times \Delta t_{\text{refresh}}}{S_{\text{update}}(p)},$$

Equation 16: Maximum number of updates

$$\mathbb{E}_{\text{Activation}} = \text{RUR} \times H_{\max}(p),$$

Equation 17: Expectation of activations

The expected number of row activations induced by the repeated updates, denoted as $\mathbb{E}_{\text{Activation}}$, can be approximated as: where RUR is the empirically measured repeated update rate

Finally, given a sparse update fraction p , bit-flips are considered successfully triggered when $\mathbb{E}_{\text{Activation}}$ is higher than $\overline{T}_{\text{flip}}$.

5.3.3 Target System

The target system is a self-supervised federated voice recognition system, to emulate large scale voice assistant pre-training systems, which are widely deployed on various platforms including smart home devices, mobile phones, and other IoT devices, and involving large quantity of clients.

5.3.3.1 Model Structure

We train the agent with target FL ASR systems of the following five model structures, the size details of these models are presented in 5.1.

5.3.3.2 Dataset

For dataset, the FL takes the Common Voice 17.0 dataset [94] as the training data, which is a large-scale multilingual voice dataset. We only use the English subset for simplicity. The subset contains 1.1M voice clips with a total duration of more than 3k hours. The data is randomly split into equal client partitions, where the size of the partition is determined by the number of clients.

5.3.3.3 Server Setups

On the server side, global model and aggregation are performed using page-locked memory, and the communication between the server and clients are optimised using RDMA.

5.3.3.4 Clients Setups

On the client side, each client emulates different physical environments via applying random Gaussian noise in addition to the adversarial noise generated by the RL agent, after which the mixed waveforms are resampled to 16kHz to match the Wav2Vec2 model.

Also, the clients are set to send sparse updates, i.e., at each communication round, only a controllable portion of the most significant gradient updates are sent to the server.

5.3.4 RL Agent Design

We propose a two-stage PPO-based agent to generate adversarial waveforms that consistently induce parameter updates clustered within targeted indices of an ASR model’s parameter vector, denoted by $\theta \in \mathbb{R}^M$. The two-stage design initially provides explicit guidance for parameter targeting (Stage 1), followed by generalised continued training under limited supervision (Stage 2).

5.3.4.1 Observation and Action Spaces

In **Stage 1**, observations explicitly encode parameter update locations to facilitate initial learning:

- **Observation:** $\mathcal{O}_1 = \{x_{\text{clean}}, \mathbf{u}_t\}$, with binary vector $\mathbf{u}_t \in \{0, 1\}^M$ indicating recently updated parameters, where x_{clean} is the unperturbed input waveform.
- **Action:** $\mathcal{A}_1 = \{\delta \in \mathbb{R}^D \mid \|\delta\|_\infty \leq \varepsilon\}$, constrained adversarial perturbations.

Rationale: Explicit signals in $\mathbf{u}_t$ facilitate clear mapping from adversarial perturbations to model updates, aligning with curriculum-learning principles [127].

In **Stage 2**, explicit update signals are removed:

- **Observation:** $\mathcal{O}_2 = \{x_{\text{clean}}\}$, relying solely on waveform input to generalise learned perturbation strategies.
- **Action:** Maintained as $\mathcal{A}_2 = \mathcal{A}_1$.

Design Choice: Removing explicit update feedback prevents dependence on instantaneous signals, promoting robust generalization across diverse inputs.

5.3.4.2 Reward Formulation

Both stages share a unified reward structure composed of three components balancing localization, clustering specificity, and imperceptibility constraints:

$$R_t = \alpha \text{EMD}(\mathbf{u}_t, \mathbf{u}_{t-1}) + \beta \frac{\|\mathbf{u}_t[i:j]\|_0}{\|\mathbf{u}_t\|_0} - \gamma \mathcal{L}_{\text{perc}}$$

Equation 18: Reward function of PPO agent

Update Proximity: Earth Mover’s Distance (EMD) penalises large shifts between consecutive update indices, promoting stable parameter clustering.

Target Specificity: Ratio term emphasises updates within targeted parameter indices $[i:j]$.

Perceptibility Penalty:

This perceptual constraint simultaneously limits frequency-domain artifacts and temporal waveform distortions [103].

$$\mathcal{L}_{\text{perc}} = \lambda_1 \|\text{STFT}(x') - \text{STFT}(x)\|_F + \lambda_2 \text{rms}(\delta)$$

Equation 19: Perceptibility Penalty

5.3.4.3 Policy Network Architecture

The agent employs a dual-head policy network with shared feature extraction and stage-specific heads:

$$\pi(a_t|o_t) = \begin{cases} \pi_{\psi_1}(E_\phi(x, \mathbf{u}_t)), & \text{Stage 1} \\ \pi_{\psi_2}(E_\phi(x)), & \text{Stage 2} \end{cases}$$

Shared Encoder (E_ϕ): Combines parallel multi-resolution 1D convolutional layers (4kHz, 2kHz, 1kHz) followed by bidirectional LSTM units, capturing both fine-grained acoustic features and long-range temporal dependencies.

Stage-Specific Heads: Stage 1 directly integrates explicit parameter updates, while Stage 2 employs attention mechanisms over encoder outputs to implicitly capture persistent perturbation-update relationships.

5.3.4.4 Training Protocol

We adopt a phased curriculum-learning strategy:

1. **Stage 1 (50k episodes):** Explicit parameter-update signals provided, guiding direct perturbation-update associations.
2. **Transition Phase (10k episodes):** Exponentially annealed interpolation between explicit and implicit observations, smoothly transitioning towards Stage 2.
3. **Stage 2 (20k episodes):** Fully implicit observation environment to foster robust adversarial generalization.

Transition criterion between stages is adaptive, triggered when target clustering ratio exceeds 85% consistently for 10 episodes. We include stage-specific value functions and orthogonal regularization ($\beta = 0.001$) on shared encoder parameters to mitigate training instability from parameter-sharing (approx. 78%) between policy heads.

Validation of Two-Stage Design: Empirical analyses suggest that bypassing Stage 1—even with larger network capacities—impairs learning efficiency due to absence of explicit guidance in early training, thus validating the necessity of our staged approach.

5.4 Experiments

5.4.1 Experimental Setups

This work utilises the DRAM Bender open-source platform to test the Rowhammer attack threshold [128]. The platform is built on the Alveo U200 Accelerated Card [129] and is designed to emulate a memory controller using an FPGA, allowing us to precisely issue Rowhammer-related instructions. The specific memory module tested is the

MTA18ASF2G72PZ-2G3B1-16GB [130], which complies with the standard DDR4 specification. Additionally, it has data rate of 2400 MT/s and bit-width of 72 bits. After excluding 8 bits of ECC, the effective bit-width for transmission is 64 bits: According to equation (5.4.1),

$$BW = \frac{\text{Data Rate} \times \text{Bit-Width}}{8 \text{ bits/byte}}$$

Equation 20: DRAM bandwidth

we can first get the bandwidth of tested module is 18.75 GB/s. Therefore, the data throughput for each 64ms refresh period will be approximately 1.2 GB. Furthermore, the default **tRC** is 46.16 ns.

In an operating system-level attack environment, the optimal **ACT** rate based on the **clflushopt** instruction sequence is 159 ACTs/tREFI, meaning that the actual average **tRC** used for frequent activations is 49 ns, which is close to the theoretical ACT limit. Under this condition, the attacker can issue up to 1,306K activations per refresh window. Moreover, all other timing parameters of tested module follow the default configuration.

Table 5.1: Common ASR models with their sizes and computation precision.

Model	Params(M)	Individual Tensors	Precision	$H_{\max}(0.0005)$
Conformer-CTC-S [131]	8.7	480	INT8	296K
QuartzNet 5x5 [132]	6.7	130	INT8	384K
Citrinet 256 [133]	10.3	635	INT8	250K
Squeezeformer-XS [134]	9.0	480	INT8	286K
Eff. Conformer-CTC [135]	13.2	520	INT8	196K

Table 5.2: Rowhammer Attack Thresholds under Different Patterns

Access Pattern	Data Pattern (Victim)	Data Pattern (Aggressor)	Threshold (ACTs)
Single-sided	0x00000000	0xFFFFFFFF	240K
Single-sided	0xFFFFFFFF	0x00000000	185K
Single-sided	0x55555555	0x55555555	260K
Single-sided	0xAAAAAAAA	0xAAAAAAAA	265K
Double-sided	0x00000000	0xFFFFFFFF	140K
Double-sided	0xFFFFFFFF	0x00000000	115K
Double-sided	0x55555555	0x55555555	160K
Double-sided	0xAAAAAAAA	0xAAAAAAAA	165K

5.4.2 Rowhammer Test Bench

Considering that FL may use different memory replacement strategies and that data diversity plays a role, this work tested different Rowhammer attack thresholds, based on two access patterns and four data patterns. Compare to single-sided access pattern, the double-sided requires less activations. However, it is also more difficult to utilise as it requires two adjacent rows for victim data. The data pattern will be the data layout between victim rows and aggressor rows. The tests not only reveal patterns that are more likely to induce bit flips, such as $0 \rightarrow 1$ or $1 \rightarrow 0$, but also demonstrate relatively more challenging patterns, such as $0 \rightarrow 0$ or $1 \rightarrow 1$. The obtained attack thresholds are derived from the average values across all banks of the tested DRAM module. These activation thresholds ensure that the attack induces bit-flips in at least 95% of the DRAM rows. The inability to obtain thresholds for all rows is likely due to the presence of exceptionally resilient cells within the module, which results in extreme deviations in the average threshold. These thresholds are specifically presented in Table 5.2. All threshold values fall within the theoretical limit of 1,309K activations and reserve enough dummy rows activations to bypass TRR. It can be observed that double-sided Rowhammer generally requires fewer activations than single-sided Rowhammer. In the worst case, 265K activations are sufficient to induce bit flips.

5.4.3 PPO Agent Training

We empirically evaluate our proposed PPO-based adversarial framework to assess its effectiveness in consistently inducing clustered and repetitive parameter updates in federated Automatic Speech Recognition (ASR) systems under varying sparsity settings, which are suitable for Rowhammer exploitation.

5.4.3.1 Experimental Setup

We selected five widely-used ASR architectures of varying complexity, parameter counts, and architectural patterns for evaluation, see in Table 5.1 the five models and their parameter statistics.

For each model, two levels of parameter-update sparsity were considered:

- **Medium Sparsity (0.5% updates):** Represents scenarios where moderate communication efficiency is required without severely limiting model convergence speed.
- **High Sparsity (0.1% updates):** Reflects extreme communication efficiency conditions, simulating highly bandwidth-constrained or large-scale deployments that are particularly susceptible to targeted memory perturbations.

The PPO agent was trained to generate adversarial waveforms constrained by a maximum perturbation limit ($\|\delta\|_\infty \leq 0.1$), ensuring the adversarial audio samples remain stealthy and practically imperceptible. The training dataset employed was the English subset of Common Voice 17.0 [94]. Hyperparameters were empirically optimised as $\alpha = 1.0$, $\beta = 0.8$, and $\gamma = 0.6$, with perceptibility penalty weights set to $\lambda_1 = \lambda_2 = 0.5$.

5.4.3.2 Baseline Methods

To provide clear insights into the effectiveness of our two-stage PPO approach, we compared it against two baselines:

- **Single-Stage PPO (PPO-SS):** PPO trained directly without the staged curriculum learning proposed in our framework.
- **Unperturbed:** leave the FL clients operating without any external influences.

5.4.3.3 Evaluation Metrics

To evaluate our PPO agent, we measure their performance with the following metrics, previously defined in 5.3.4.

- **Repeated Update Rate (RUR)**
- **Cluster Density (CD)**

5.4.3.4 Results and Analysis

Tables 5.3 and 5.4 present comprehensive evaluations of our PPO-TS method against baseline approaches across different model architectures and sparsity levels.

Table 5.3: Repeated Tensor Overlap Rate (RUR, %) across all models and sparsity levels.

Model	Method	RUR (0.1 %)	RUR (0.05 %)
Conformer-CTC-S	PPO-TS	73.7	68.3
	PPO-SS	65.1	53.9
	Unperturbed	12.5	9.3
QuartzNet 5x5	PPO-TS	75.6	71.1
	PPO-SS	64.8	57.2
	Unperturbed	11.8	8.7
Citrinet 256	PPO-TS	71.9	66.4
	PPO-SS	57.4	52.9
	Unperturbed	13.9	10.2
Squeezeformer-XS	PPO-TS	72.5	68.1
	PPO-SS	61.4	54.4
	Unperturbed	11.2	8.1
Eff. Conformer-CTC	PPO-TS	71.1	66.1
	PPO-SS	59.3	51.7
	Unperturbed	13.2	9.6

Repeated Update Rate (RUR): As summarised in Table 5.3, our PPO-TS approach consistently outperformed both the PPO-SS and Unperturbed baselines. At high sparsity (0.05%), PPO-TS achieved RUR ranging from 66.1% (Eff. Conformer-CTC) to 71.1%

(QuartzNet 5x5), significantly surpassing PPO-SS (approximately 51%–57%). This substantial improvement underscores the two stage design to learn robust, long-term strategies for generating perturbations that repeatedly affect the same parameters.

Table 5.4: Cluster Density (CD, %) across all models and sparsity levels.

Model	Method	CD (%)	
		0.1%	0.05%
Conformer-CTC-S	PPO-TS	1.2	0.6
	PPO-SS	3.7	1.8
	Unperturbed	19.2	15.4
QuartzNet 5x5	PPO-TS	1.5	0.8
	PPO-SS	4.1	3.2
	Unperturbed	18.6	14.8
Citrinet 256	PPO-TS	1.0	0.7
	PPO-SS	3.4	1.9
	Unperturbed	17.9	13.9
Squeezeformer-XS	PPO-TS	1.6	0.9
	PPO-SS	3.8	2.4
	Unperturbed	19.5	15.8
Eff. Conformer-CTC	PPO-TS	1.1	0.5
	PPO-SS	3.6	2.3
	Unperturbed	18.2	14.3

Cluster Density (CD): As detailed in Table 5.4, PPO-TS resulted in significantly lower CD values compared to baselines across all tested configurations. At the high sparsity level, PPO-TS achieved CD values as low as 1.0%–1.6%, marking approximately a 2–3 $\times$ reduction compared to PPO-SS (around 3.4%–4.1%). Such extreme clustering effectively concentrates updates into tightly bound memory regions, dramatically increasing the probability of repeatedly activating the same DRAM rows.

These results demonstrate the efficacy of our approach in systematically inducing concentrated and stable update patterns necessary for facilitating targeted Rowhammer attacks in realistic FL settings.

5.4.4 Bit-Flip Validation

In this subsection, we jointly analyze the effectiveness of our proposed PPO-based adversarial attack framework in inducing practical Rowhammer conditions, considering both the theoretical Rowhammer thresholds obtained in Section IV-B and the empirically measured clustering results from PPO agent training presented in Section IV-C. Specifically, we quantify the practical feasibility of triggering Rowhammer bit-flips by correlating the observed Repeated Update Rate (RUR) with the number of sparse gradient updates performed in federated learning scenarios.

This straightforward estimation facilitates practical reasoning about the potential for successful Rowhammer attacks based on experimentally observed adversarial clustering.

5.4.4.1 Estimated Row Activation Counts

Table 5.5 summarises the calculated expected row activations for all evaluated ASR models at the high sparsity setting (0.1%), comparing our two-stage PPO method (**PPO-TS**) against single-stage PPO (**PPO-SS**) and Unperturbed baseline methods.

Table 5.5: Estimated Row Activations Based on Empirical Clustering (Top 0.05% Sparsity Level)

Model	Method	$H_{\max}(0.0005)$	RUR (%)	$\mathbb{E}_{\text{Activations}}$
Conformer-CTC-S	PPO-TS	296,000	68.3	202,168
	PPO-SS	296,000	53.9	159,544
	Unperturbed	296,000	9.3	27,528
QuartzNet 5x5	PPO-TS	384,000	71.1	273,024
	PPO-SS	384,000	57.2	219,648
	Unperturbed	384,000	8.7	33,408
Citrinet 256	PPO-TS	250,000	66.4	166,000
	PPO-SS	250,000	52.9	132,250
	Unperturbed	250,000	10.2	25,500
Squeezeformer-XS	PPO-TS	286,000	68.1	194,766
	PPO-SS	286,000	54.4	155,584
	Unperturbed	286,000	8.1	23,166
Eff. Conformer-CTC	PPO-TS	196,000	66.1	129,556
	PPO-SS	196,000	56.9	111,524
	Unperturbed	196,000	9.6	18,816

5.4.4.2 Estimated Bit Flips

Refer to Table 5.2, averaging bit flip thresholds over different single sided aggressor patterns, bit flips are expected when repeated activation exceeds 240K, with more vulnerable cells flips at 200K.

$$\overline{T_{\text{flip}}} = 240,000,$$

$$T_{\text{flip}_{\min}} = 200,000$$

Our agent’s attack on QuartzNet 5x5 with sparsity of 0.05% can be deemed successful in inducing Rowhammer bit flips, and there is a high possibility of success in inducing Rowhammer bit flips through attacking Conformer-CTC-S.

5.5 Discussion

In this section, we discuss the practical implications, limitations, security considerations, and mitigation strategies related to our proposed PPO-driven Rowhammer attack against federated learning (FL) systems.

5.5.1 Practical Implications and Feasibility

Our experiments provide empirical evidence that PPO-generated physical-domain perturbations can effectively induce clustered and repetitive parameter updates critical for successful Rowhammer exploitation in federated learning scenarios. Particularly at higher sparsity levels in large-scale FL deployments, our PPO-TS approach significantly outperforms baseline methods by achieving high repetition rates of over 70% and highly localised cluster densities. Such performance substantially increases the likelihood of repeatedly activating specific DRAM rows, thus enhancing the practical feasibility of successful Rowhammer bit-flips.

However, practical real-world deployment may introduce complexities such as partial system knowledge, environmental variability, or limited precision in physical perturbation generation. Despite these uncertainties, our joint evaluation clearly demonstrates that the activation counts achieved by our PPO-based attacks are realistically within the threshold ranges required to trigger Rowhammer attacks, particularly under favorable DRAM configurations or minor system optimizations (e.g., higher memory bandwidth or extended refresh intervals). Therefore, our results underscore a significant and realistic security risk to federated learning systems utilizing common performance optimization practices.

5.5.2 Limitations and Open Challenges

We acknowledge two primary limitations affecting the practical applicability of the proposed PPO-driven Rowhammer attack:

- 1. Opacity of Realistic FL Setups:** Real-world federated systems typically obscure internal details such as memory management policies, parameter layouts, and system optimizations from external attackers. Without explicit feedback or partial system knowledge, the current reinforcement learning framework might experience significantly reduced efficiency or effectiveness. Future research could address this limitation by developing RL agents with highly sparse rewards or inferable rewards, or transferable adversarial perturbation techniques.
- 2. Lack of Control over Specific Victim Rows:** The proposed indirect Rowhammer attack inherently lacks precise control over the specific DRAM rows affected by induced bit-flips, making outcomes unpredictable and potentially transient. This uncertainty reduces the reliability of achieving specific attack objectives (e.g., targeted data corruption or privilege escalation). Future research should explore advanced side-channel techniques or hardware-software co-design approaches to improve the precision and predictability of the Rowhammer effects induced by physical-domain perturbations.

5.5.3 Security Considerations for Federated Learning Systems

Our findings highlight significant security risks arising from commonly used performance optimization techniques in federated learning systems, including sparse updates, RDMA, huge pages, and pinned memory. While these optimizations greatly enhance computational and communication efficiency, they also inadvertently create predictable and sustained DRAM activation patterns exploitable by hardware-level adversaries. Consequently, FL practitioners must carefully reconsider optimization-related trade-offs, explicitly evaluating hardware security vulnerabilities when designing and deploying federated learning systems, especially in sensitive or critical infrastructure scenarios.

5.5.4 Potential Mitigation Strategies

Based on our comprehensive analysis and findings, we propose several practical defensive measures to mitigate the identified threat:

- **Adversarial Input Detection:** Deploying anomaly detection systems at the sensor or client levels could proactively detect and mitigate adversarial acoustic or electromagnetic signals before they propagate into clustered model updates.
- **Systematic Analysis of Optimization Technique Deployment:** Analyse and assess the effect of deploying optimization techniques in combinations, selectively using such techniques to lower attack feasibility.
- **Hardware-Level Countermeasures:** Employing DRAM modules with enhanced Rowhammer resilience (e.g., advanced error correction or hardware-level activation anomaly detection) offers robust defence against direct exploitation attempts.

5.5.5 Broader Impact and Future Directions

Our study demonstrates a compelling need for interdisciplinary approaches combining hardware security, adversarial machine learning, and federated learning systems design. By integrating hardware-level vulnerabilities into FL threat modeling, we advocate a holistic view that proactively identifies, assesses, and mitigates emerging threats.

Future research directions include:

- Extending our approach to diverse model architectures, and real-world applications through stronger agents that can process side channel information and learn with highly sparse rewards in real-world applications to enhance the attack.
- Exploring hardware-software co-design approaches that mitigate Rowhammer risks without severely compromising computational efficiency.

By addressing these open challenges and continuing to bridge security gaps between hardware vulnerabilities and distributed learning systems, we aim to contribute toward more secure, dependable, and resilient federated learning deployments.

5.6 Conclusion

This chapter introduced a novel Rowhammer attack vector against federated learning (FL) systems, systematically integrating and expanding on the thesis aim of understanding how contextual and efficiency-oriented factors in heterogeneous FL environments shape security and optimisation trade-offs. Leveraging a Proximal Policy Optimization (PPO)-based reinforcement learning agent, we demonstrated that physical-domain adversarial perturbations, delivered to FL client sensors, can indirectly trigger clustered and repetitive model updates on FL servers. These clustered updates, when combined with common FL efficiency optimisations such as sparse gradients, pinned memory, and RDMA, create memory access patterns conducive to inducing Rowhammer bit-flips in server DRAM.

Our empirical evaluation across multiple automatic speech recognition (ASR) architectures confirmed the feasibility and impact of the proposed attack under realistic FL deployment conditions. The PPO-based adversarial framework consistently achieved higher repeated update rates and more concentrated update clustering than conventional approaches, illustrating that hardware-level vulnerabilities are increasingly relevant as FL systems pursue aggressive optimisation strategies.

By examining the interplay between software and hardware optimisation, our work provides a concrete example of how efficiency techniques designed for scalability and performance may inadvertently weaken system robustness and expand the attack surface.

To mitigate the identified risks, we outlined a set of practical defence strategies, including randomised memory allocation, input perturbation detection, and more judicious use of memory and communication optimisations. These findings emphasise the necessity for interdisciplinary security assessment that spans algorithmic, systems, and hardware domains, ensuring that federated learning frameworks remain both efficient and resilient in the face of emerging threats.

Collectively, this work reinforces the imperative for holistic threat modelling and proactive defence in FL research, advocating for a continuous, context-aware balance between optimisation and security as foundational for dependable and secure federated intelligence.

Chapter 6

Discussion

This chapter synthesises the findings presented in previous chapters, highlighting how each contribution collectively addresses the aims and objectives articulated in Chapter 1. It critically evaluates the significance of these findings within the broader context of Federated Learning (FL), elucidates novel insights, acknowledges limitations, and examines practical implications to inform future research and application strategies.

6.1 Overview of Thesis Contributions

This thesis contributes to the advancement of Federated Learning by addressing specific research gaps identified in Chapter 1, the effective utilisation of client heterogeneity, development of context-aware efficiency strategies, and exploration of vulnerabilities arising from optimisation methods. The main contributions are:

1. A Mixed-Precision Over-the-Air Federated Learning (MP-OTA-FL) framework [90] tailored to the computational heterogeneity of FL clients.
2. A Retrieval-Augmented Generation (RAG)-based user profiling method [136] enhancing precision allocation and client satisfaction within MP-OTA-FL.
3. Identification and systematic assessment of novel hardware vulnerabilities introduced by optimisation strategies, exemplified through a reinforcement learning (RL)-based Rowhammer attack [137].

Each contribution is discussed in detail in the subsequent sections.

6.2 Leveraging Client Heterogeneity in FL

Addressing the underutilisation of client heterogeneity, the proposed MP-OTA-FL framework significantly improves efficiency and client participation by dynamically adjusting computational precision. Unlike traditional homogeneous strategies, our framework effectively exploits device diversity, reducing energy consumption (up to 65%) and computational overhead, notably enhancing resource-limited client performance.

This work expands upon existing OTA-FL methods [74], [138] by introducing quantisation modulation schemes suitable for heterogeneous federated environments. Notably,

low-precision (4-bit) clients demonstrated performance improvements of over 10% compared to uniform precision baselines, validating mixed-precision strategies within distributed, client-specific contexts.

Nevertheless, scalability challenges persist, especially when extending precision modulation to a diverse and large-scale client population, necessitating future exploration into hierarchical or clustered precision allocation approaches.

6.3 Context-Aware Optimisation via User Profiling

The thesis further advances efficiency strategies through a novel RAG-based profiling method introduced in Chapter 4. This method systematically incorporates contextual client data—including hardware capabilities, user behaviours, and satisfaction metrics—to dynamically optimise computational precision. Contrasting previous static or heuristic-based methods [9], [13], this context-aware approach significantly improves user satisfaction, energy efficiency, and model performance.

The integration of conversational LLMs and RAG databases enabled continual adaptive optimisation with lower risk of hallucination. However, deploying such sophisticated tools introduces computational overhead and potential privacy risks. Future research should investigate decentralised or federated implementations, refining privacy-preserving mechanisms to enhance practical scalability.

6.4 Emergent Vulnerabilities from Efficiency Optimisation

Chapter 5 highlights critical vulnerabilities introduced by optimisation methods, particularly focusing on hardware-based risks exemplified by the RL-driven Rowhammer attack. The analysis reveals how sparse gradient optimisation techniques and hardware-level memory optimisations inadvertently facilitate predictable and repeated memory access patterns, significantly elevating the risk of Rowhammer vulnerabilities.

This research shifts conventional FL threat models, emphasising the necessity of comprehensive security measures that encompass both hardware and software layers. Although the demonstrated attack succeeded under controlled laboratory conditions, further empirical evaluations under varied environmental and practical deployment scenarios such as server configurations employing ECC or TRR mechanisms, are essential to fully quantify real-world threats and establish robust mitigation strategies.

6.5 Integration and Real-world Implications

Collectively, the findings presented in this thesis deliver a cohesive contribution towards developing robust and practical FL systems capable of operating effectively under client heterogeneity. By integrating efficiency optimisation with adaptive precision strategies and considering related security risks, this research provides actionable insights and methodologies for deploying FL solutions across diverse industries, including healthcare, autonomous transport, and consumer IoT ecosystems.

From a practical perspective, the MP-OTA-FL and RAG-based profiling frameworks can significantly enhance deployment feasibility by customising computations according to user preferences and hardware constraints, thus potentially improving long-term user participation and satisfaction. Concurrently, recognising and addressing hardware-level vulnerabilities early in system design phases can substantially strengthen overall system resilience, contributing towards trustworthy and secure FL infrastructures.

6.6 Limitations

The contributions of this thesis, while robust, come with several inherent limitations:

- **Generalisation and Scalability:** While the MP-OTA-FL framework demonstrated clear benefits, its scalability to large amount of heterogeneous clients in practical large-scale deployments needs further empirical validation and possibly new optimisation algorithms.
- **Complexity and Overhead of LLMs:** The computational overhead and potential privacy concerns introduced by the RAG-based user profiling framework, although addressed in part, need further evaluation in realistic, large-scale federated settings.
- **Environmental and Practical Constraints:** The hardware vulnerability attack vector needs further real-world evaluation under varied environmental and operational conditions to confirm the robustness and viability of practical exploitations and effective mitigations.

Addressing these limitations through targeted empirical and theoretical exploration will be critical for transitioning these methodologies from controlled experiments to widespread industrial applications.

6.7 Summary

In summary, this thesis advances Federated Learning by systematically addressing key gaps related to client heterogeneity, contextual computational efficiency, and optimisation-driven vulnerabilities. Through integrated frameworks, it highlights actionable pathways towards secure, efficient, and personalised FL deployments. The contributions provide foundational insights for both academia and industry, outlining clear avenues for future research aimed at enhancing FL’s practical effectiveness, resilience, and real-world applicability.

Chapter 7

Conclusion and Further Work

This chapter summarises the key outcomes of the research undertaken in this thesis, reflecting upon how the aims and objectives stated in Chapter 1 were successfully addressed. It explicitly highlights the main contributions, discusses their significance within the wider context of Federated Learning (FL), and proposes avenues for future research based on identified limitations and emerging opportunities.

7.1 Summary of Research Outcomes

The primary aim of this thesis was to develop adaptive Federated Learning frameworks capable of optimising computational efficiency, effectively managing client heterogeneity, and addressing vulnerabilities introduced by optimisation techniques. This aim was successfully realised through the achievement of three specific research objectives:

1. **Adaptive Mixed-Precision Federated Learning Framework:** A novel Mixed-Precision Over-the-Air Federated Learning (MP-OTA-FL) framework was proposed, explicitly accommodating client heterogeneity by dynamically adapting computational precision based on client hardware capabilities. The developed quantisation modulation scheme significantly enhanced energy efficiency (achieving up to 65% reduction compared to uniform 32-bit precision) and computational performance, particularly benefiting resource-limited clients.
2. **Context-Aware Precision Optimisation via User Profiling:** A Retrieval-Augmented Generation (RAG)-based user profiling methodology was developed to optimise precision allocation dynamically. By effectively leveraging client-specific contextual data, this method significantly improved global model accuracy, energy efficiency, and overall user satisfaction in heterogeneous and realistic FL scenarios.
3. **Analysis of Hardware Vulnerabilities Introduced by Optimisations:** The research demonstrated a previously unexplored threat vector exploiting hardware-level vulnerabilities via reinforcement learning-driven Rowhammer attacks. Specifically, it was shown that common efficiency optimisation strategies, such as sparse updates and memory optimisation techniques, inadvertently increased susceptibility to sophisticated hardware-level adversarial attacks.

Together, these contributions provide a comprehensive framework addressing critical research gaps, harmonising efficiency optimisation with robust security practices.

7.2 Contributions to Knowledge

The contributions of this thesis to the broader FL literature are as follows:

- Demonstration of the practical effectiveness of mixed-precision OTA-FL strategies, showcasing significant improvements in computational efficiency and client resource management over traditional homogeneous precision approaches.
- Development and empirical validation of an innovative RAG-based, context-aware optimisation strategy, substantially enhancing personalised client experiences and precision planning in heterogeneous FL deployments.
- Identification and systematic analysis of novel hardware-based security vulnerabilities arising from optimisation techniques, significantly expanding existing FL threat models through practical reinforcement learning-driven Rowhammer scenarios.

These findings are crucial for guiding future research and informing real-world deployment strategies, particularly in resource-constrained and heterogeneous client environments.

7.3 Practical Implications

The methodologies and insights presented carry significant implications for real-world FL system deployments:

- **Optimised Deployment Strategies for FL:** The frameworks for mixed-precision and context-aware optimisation offer practical methods for implementing personalised and efficient FL solutions, particularly beneficial for large-scale deployments across IoT, healthcare, and mobile ecosystems.
- **Integrated Security and System Design:** The demonstrated hardware-level vulnerabilities emphasise the need for comprehensive, integrated security strategies spanning both software and hardware layers, significantly strengthening resilience against advanced adversarial threats.

7.4 Further Work

Despite the significant advancements made, several areas remain for future exploration to enhance robustness, scalability, and applicability:

1. **Scalability and Generalisation of MP-OTA-FL:** Future research should empirically evaluate mixed-precision modulation scalability across extremely heterogeneous and large-scale client populations, potentially integrating hierarchical or clustered aggregation methods to enhance generalisation.
2. **Refinement and Decentralisation of RAG-based Profiling:** Further studies should focus on federated or decentralised implementations of RAG-based user profiling to reduce computational overhead and privacy concerns. Investigating alternative language model architectures may also improve resource efficiency and real-time responsiveness.
3. **Empirical Validation of Hardware Vulnerabilities:** Rigorous real-world testing under varied operational conditions is essential to validate the practical impact of the identified hardware vulnerabilities. Additional research should aim to develop robust mitigation strategies effective across diverse environmental conditions.

7.5 Closing Remarks

This thesis has made substantial progress towards robust, context-aware, and secure Federated Learning systems tailored to heterogeneous environments. By addressing efficiency, client satisfaction, and security challenges comprehensively, the work has laid important foundations for future research and practical implementations. The findings presented provide meaningful guidance for both academia and industry, highlighting clear opportunities to further advance the resilience, efficiency, and trustworthiness of Federated Learning technologies.

References

- [1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, ‘Imagenet: A large-scale hierarchical image database,’ in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.
- [2] T. Brown, B. Mann, N. Ryder *et al.*, ‘Language models are few-shot learners,’ 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. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.
- [3] 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.
- [4] N. Truong, K. Sun, S. Wang, F. Guitton and Y. Guo, ‘Privacy preservation in federated learning: An insightful survey from the gdpr perspective,’ *Computers & Security*, vol. 110, p. 102402, 2021.
- [5] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar and V. Smith, ‘Federated optimization in heterogeneous networks,’ *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.
- [6] W. Y. B. Lim, Z. Xiong, C. Miao *et al.*, ‘Hierarchical incentive mechanism design for federated machine learning in mobile networks,’ *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9575–9588, 2020.
- [7] K. Bonawitz, H. Eichner, W. Grieskamp *et al.*, ‘Towards federated learning at scale: System design,’ *Proceedings of machine learning and systems*, vol. 1, pp. 374–388, 2019.
- [8] L. Wang, S. Xu, X. Wang and Q. Zhu, ‘Addressing class imbalance in federated learning,’ in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 10165–10173.
- [9] T. Nishio and R. Yonetani, ‘Client selection for federated learning with heterogeneous resources in mobile edge,’ in *ICC 2019-2019 IEEE international conference on communications (ICC)*, IEEE, 2019, pp. 1–7.
- [10] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin and V. Chandra, ‘Federated learning with non-iid data,’ *arXiv preprint arXiv:1806.00582*, 2018.

- [11] V. Kulkarni, M. Kulkarni and A. Pant, ‘Survey of personalization techniques for federated learning,’ in *2020 fourth world conference on smart trends in systems, security and sustainability (WorldS4)*, IEEE, 2020, pp. 794–797.
- [12] C. Briggs, Z. Fan and P. Andras, ‘Federated learning with hierarchical clustering of local updates to improve training on non-iid data,’ in *2020 international joint conference on neural networks (IJCNN)*, IEEE, 2020, pp. 1–9.
- [13] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen and C. S. Hong, ‘Federated learning over wireless networks: Optimization model design and analysis,’ in *IEEE INFOCOM 2019-IEEE conference on computer communications*, IEEE, 2019, pp. 1387–1395.
- [14] P. Kairouz, H. B. McMahan, B. Avent *et al.*, ‘Advances and open problems in federated learning,’ *Foundations and trends® in machine learning*, vol. 14, no. 1–2, pp. 1–210, 2021.
- [15] K. Bonawitz, V. Ivanov, B. Kreuter *et al.*, ‘Practical secure aggregation for privacy-preserving machine learning,’ in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.
- [16] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie and R. Pedarsani, ‘Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization,’ in *International conference on artificial intelligence and statistics*, PMLR, 2020, pp. 2021–2031.
- [17] Z. Zhu, Y. Shi, J. Luo *et al.*, ‘Fedlp: Layer-wise pruning mechanism for communication-computation efficient federated learning,’ in *ICC 2023-IEEE International Conference on Communications*, IEEE, 2023, pp. 1250–1255.
- [18] H. Wang, Z. Liu, K. Hoshino, T. Zhang, J. P. Walters and S. Crago, *Fedpai: Achieving extreme sparsity in federated learning via pruning at initialization*, 2025. arXiv: 2504.00308 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2504.00308>.
- [19] Y. Ji and L. Chen, ‘Fedqnn: A computation–communication-efficient federated learning framework for iot with low-bitwidth neural network quantization,’ *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2494–2507, 2022.
- [20] Z. Li, Y. Li, B. Lin, Z. Jin and W. Zhang, ‘Low precision local training is enough for federated learning,’ in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave *et al.*, Eds., vol. 37, Curran Associates, Inc., 2024, pp. 90 160–90 197. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/a421e88bb7db67cd0401475a0ed4a17d-Paper-Conference.pdf.
- [21] H. Chen and H. Vikalo, ‘Mixed-precision quantization for federated learning on resource-constrained heterogeneous devices,’ in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 6138–6148. DOI: 10.1109/CVPR52733.2024.00587.
- [22] X. Mo and J. Xu, ‘Energy-efficient federated edge learning with joint communication and computation design,’ *Journal of Communications and Information Networks*, vol. 6, no. 2, pp. 110–124, 2021. DOI: 10.23919/JCIN.2021.9475121.

- [23] G. Wilkins, S. Di, J. C. Calhoun *et al.*, ‘Fedsz: Leveraging error-bounded lossy compression for federated learning communications,’ in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, 2024, pp. 577–588. DOI: 10.1109/ICDCS60910.2024.00060.
- [24] Z. Xu, Z. Zhang and Y. Cheng, *Efficient federated learning in the era of llms with message quantization and streaming*, NVIDIA Developer Blog, Apr. 2025. [Online]. Available: <https://developer.nvidia.com/blog/efficient-federated-learning-in-the-era-of-llms-with-message-quantization-and-streaming/>.
- [25] T. Li, A. K. Sahu, A. Talwalkar and V. Smith, ‘Federated learning: Challenges, methods, and future directions,’ *IEEE signal processing magazine*, vol. 37, no. 3, pp. 50–60, 2020.
- [26] F. Sattler, S. Wiedemann, K.-R. Müller and W. Samek, ‘Robust and communication-efficient federated learning from non-iid data,’ *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3400–3413, 2019.
- [27] P. Micikevicius *et al.*, ‘Mixed precision training,’ *arXiv preprint arXiv:1710.03740*, 2017.
- [28] Y. Chen, Y. Ning, M. Slawski and H. Rangwala, ‘Asynchronous online federated learning for edge devices with non-iid data,’ in *2020 IEEE International Conference on Big Data (Big Data)*, IEEE, 2020, pp. 15–24.
- [29] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin and V. Shmatikov, ‘How to backdoor federated learning,’ in *International conference on artificial intelligence and statistics*, PMLR, 2020, pp. 2938–2948.
- [30] T. Li, M. Sanjabi, A. Beirami and V. Smith, ‘Fair resource allocation in federated learning,’ in *International Conference on Learning Representations (ICLR)*, 2020.
- [31] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich and A. T. Suresh, ‘Scaffold: Stochastic controlled averaging for federated learning,’ in *International conference on machine learning*, PMLR, 2020.
- [32] Z. Liu, Y. Chen, H. Yu, Y. Liu and L. Cui, ‘Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning,’ *ACM Transactions on intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–21, 2022.
- [33] 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.
- [34] M. Ye, X. Fang, B. Du, P. C. Yuen and D. Tao, ‘Heterogeneous federated learning: State-of-the-art and research challenges,’ *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–44, 2023.
- [35] B. Ghosh, D. Basu, F. Huazhu *et al.*, *Don’t forget what i did?: Assessing client contributions in federated learning*, 2024. arXiv: 2403 . 07151 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2403.07151>.
- [36] A. Ghorbani and J. Zou, ‘Data shapley: Equitable valuation of data for machine learning,’ in *International conference on machine learning*, PMLR, 2019, pp. 2242–2251.

- [37] J. Kang, Z. Xiong, D. Niyato, S. Xie and J. Zhang, ‘Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory,’ *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, 2019.
- [38] H. Yu, Z. Liu, Y. Liu *et al.*, ‘A fairness-aware incentive scheme for federated learning,’ in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020, pp. 393–399.
- [39] A. Fallah, A. Mokhtari and A. Ozdaglar, ‘Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach,’ *Advances in neural information processing systems*, vol. 33, pp. 3557–3568, 2020.
- [40] M. Mohri, G. Sivek and A. T. Suresh, ‘Agnostic federated learning,’ in *International conference on machine learning*, PMLR, 2019, pp. 4615–4625.
- [41] M. G. Arivazhagan, V. Aggarwal, A. K. Singh and S. Choudhary, ‘Federated learning with personalization layers,’ *arXiv preprint arXiv:1912.00818*, 2019.
- [42] F. Sattler, K.-R. Müller and W. Samek, ‘Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints,’ *IEEE transactions on neural networks and learning systems*, vol. 32, no. 8, pp. 3710–3722, 2020.
- [43] T. Li, S. Hu, A. Beirami and V. Smith, ‘Ditto: Fair and robust federated learning through personalization,’ in *International conference on machine learning*, PMLR, 2021, pp. 6357–6368.
- [44] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli and A. Anandkumar, ‘Signsgd: Compressed optimisation for non-convex problems,’ in *International conference on machine learning*, PMLR, 2018, pp. 560–569.
- [45] D. Alistarh, D. Grubic, J. Li, R. Tomioka and M. Vojnovic, ‘Qsgd: Communication-efficient sgd via gradient quantization and encoding,’ in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017.
- [46] 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)*, IEEE, 2019, pp. 1–8.
- [47] H. Wang, Z. Kaplan, D. Niu and B. Li, ‘Optimizing federated learning on non-iid data with reinforcement learning,’ in *IEEE INFOCOM 2020-IEEE conference on computer communications*, IEEE, 2020, pp. 1698–1707.
- [48] Y. J. Cho, J. Wang and G. Joshi, ‘Client selection in federated learning: Convergence analysis and power-of-choice selection strategies,’ *arXiv preprint arXiv:2010.01243*, 2020.
- [49] O. Gupta and R. Raskar, ‘Distributed learning of deep neural network over multiple agents,’ *Journal of Network and Computer Applications*, vol. 116, pp. 1–8, 2018.

- [50] H. Wang, K. Sreenivasan, S. Rajput *et al.*, ‘Attack of the tails: Yes, you really can backdoor federated learning,’ *Advances in neural information processing systems*, vol. 33, pp. 16 070–16 084, 2020.
- [51] P. Blanchard, E. M. El Mhamdi, R. Guerraoui and J. Stainer, ‘Machine learning with adversaries: Byzantine tolerant gradient descent,’ *Advances in neural information processing systems*, vol. 30, 2017.
- [52] M. Fang, X. Cao, J. Jia and N. Gong, ‘Local model poisoning attacks to {byzantine-robust} federated learning,’ in *29th USENIX security symposium (USENIX Security 20)*, 2020, pp. 1605–1622.
- [53] K.-F. Chu and W. Guo, ‘Passenger spoofing attack for artificial intelligence-based Mobility-as-a-Service,’ in *Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems*, IEEE, 2023, pp. 4874–4880. DOI: 10.1109/ITSC57777.2023.10422567.
- [54] K.-F. Chu and W. Guo, ‘Multi-agent reinforcement learning-based passenger spoofing attack on Mobility-as-a-Service,’ *IEEE Transactions on Dependable and Secure Computing*, 2024. DOI: 10.1109/TDSC.2024.3379283.
- [55] V. Behzadan and A. Munir, ‘Vulnerability of deep reinforcement learning to policy induction attacks,’ in *Machine Learning and Data Mining in Pattern Recognition: 13th International Conference, MLDM 2017, New York, NY, USA, July 15–20, 2017, Proceedings*, Springer, 2017, pp. 262–275. DOI: 10.1007/978-3-319-62416-7_19.
- [56] A. Madry, A. Makelov, L. Schmidt, D. Tsipras and A. Vladu, ‘Towards deep learning models resistant to adversarial attacks,’ *arXiv preprint arXiv:1706.06083*, 2017.
- [57] 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*, J. Dy and A. Krause, Eds., ser. Proceedings of Machine Learning Research, vol. 80, PMLR, Jul. 2018, pp. 3521–3530. [Online]. Available: <https://proceedings.mlr.press/v80/mhamdi18a.html>.
- [58] L. Lyu, H. Yu, X. Ma *et al.*, ‘Privacy and robustness in federated learning: Attacks and defenses,’ *IEEE transactions on neural networks and learning systems*, vol. 35, no. 7, pp. 8726–8746, 2022.
- [59] R. Shokri, M. Stronati, C. Song and V. Shmatikov, ‘Membership inference attacks against machine learning models,’ in *Proceedings of the 2017 IEEE Symposium on Security and Privacy*, IEEE, 2017, pp. 3–18. DOI: 10.1109/SP.2017.41.
- [60] M. Nasr, R. Shokri and A. Houmansadr, ‘Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning,’ in *Proceedings of the 2019 IEEE Symposium on Security and Privacy*, IEEE, 2019, pp. 739–753. DOI: 10.1109/SP.2019.00065.
- [61] H. Hu, Z. Salcic, L. Sun, G. Dobbie and X. Zhang, ‘Source inference attacks in federated learning,’ in *Proceedings of the 2021 IEEE International Conference on Data Mining*, IEEE, 2021, pp. 1102–1107. DOI: 10.1109/ICDM51629.2021.00129.

- [62] T. Orekondy, B. Schiele and M. Fritz, ‘Knockoff nets: Stealing functionality of black-box models,’ in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, 2019, pp. 4949–4958. DOI: 10.1109/CVPR.2019.00509.
- [63] M. Abadi, A. Chu, I. Goodfellow *et al.*, ‘Deep learning with differential privacy,’ in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.
- [64] M. Nasr, R. Shokri and A. Houmansadr, ‘Machine learning with membership privacy using adversarial regularization,’ in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2018, pp. 634–646. DOI: 10.1145/3243734.3243855.
- [65] D. Yu, H. Zhang, W. Chen and T.-Y. Liu, ‘Do not let privacy overbill utility: Gradient embedding perturbation for private learning,’ in *Proceedings of the 2021 International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=7aog0j_VY00.
- [66] S. Kariyappa and M. K. Qureshi, ‘Defending against model stealing attacks with adaptive misinformation,’ in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, 2020, pp. 767–775. DOI: 10.1109/CVPR42600.2020.00085.
- [67] A. Liu, X. Liu, J. Fan *et al.*, ‘Perceptual-sensitive GAN for generating adversarial patches,’ *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 1028–1035, 2019. DOI: 10.1609/aaai.v33i01.33011028.
- [68] H. Yakura, Y. Akimoto and J. Sakuma, ‘Generate (non-software) bugs to fool classifiers,’ *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 1070–1078, 2020. DOI: 10.1609/aaai.v34i01.5457.
- [69] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter and M. Fredrikson, *Universal and transferable adversarial attacks on aligned language models*, arXiv:2307.15043 [cs.CL], 2023. DOI: 10.48550/arXiv.2307.15043.
- [70] I. J. Goodfellow, J. Shlens and C. Szegedy, ‘Explaining and harnessing adversarial examples,’ in *Proceedings of the 2015 3rd International Conference on Learning Representations*, 2015. DOI: 10.48550/arXiv.1412.6572.
- [71] Y. Cao, X. Niu and Y. Dou, ‘Region-based convolutional neural networks for object detection in very high resolution remote sensing images,’ in *Proceedings of the 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, IEEE, 2016, pp. 548–554. DOI: 10.1109/FSKD.2016.7603232.
- [72] N. Papernot, P. McDaniel, X. Wu, S. Jha and A. Swami, ‘Distillation as a defense to adversarial perturbations against deep neural networks,’ in *2016 IEEE symposium on security and privacy (SP)*, IEEE, 2016, pp. 582–597.
- [73] 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*, vol. 54, 2017, pp. 1273–1282.

- [74] K. Yang, T. Jiang, Y. Shi and Z. Ding, ‘Federated learning via over-the-air computation,’ *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2022–2035, 2020. DOI: 10.1109/TWC.2019.2961673.
- [75] H. Chen, H. Wang, Q. Long, D. Jin and Y. Li, ‘Advancements in federated learning: Models, methods, and privacy,’ *ACM Comput. Surv.*, vol. 57, no. 2, Nov. 2024, ISSN: 0360-0300. DOI: 10.1145/3664650.
- [76] H. Chen, H. Wang, Q. Yao, Y. Li, D. Jin and Q. Yang, ‘Losac: An efficient local stochastic average control method for federated optimization,’ *ACM Transactions on Knowledge Discovery from Data*, vol. 17, no. 4, pp. 1–28, 2023.
- [77] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao and J. Cong, ‘Optimizing fpga-based accelerator design for deep convolutional neural networks,’ in *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, Monterey, California, USA, 2015, pp. 161–170, ISBN: 9781450333153. DOI: 10.1145/2684746.2689060.
- [78] T. Elsken, J. H. Metzen and F. Hutter, ‘Neural architecture search: A survey,’ *Journal of Machine Learning Research*, vol. 20, no. 55, 2019.
- [79] R. Lee, J. Fernandez-Marques, S. X. Hu *et al.*, ‘Recurrent early exits for federated learning with heterogeneous clients,’ *arXiv preprint arXiv:2405.14791*, 2024.
- [80] G. Flegar, F. Scheidegger, V. Novaković *et al.*, ‘Floatx: A c++ library for customized floating-point arithmetic,’ *ACM Transactions on Mathematical Software*, vol. 45, no. 4, Dec. 2019, ISSN: 0098-3500. DOI: 10.1145/3368086.
- [81] S. Mittal, ‘A survey of techniques for architecting and managing asymmetric multicore processors,’ *ACM Computing Surveys (CSUR)*, vol. 48, no. 3, pp. 1–38, 2016.
- [82] B. Nazer and M. Gastpar, ‘Computation over multiple-access channels,’ *IEEE Transactions on information theory*, vol. 53, no. 10, 2007.
- [83] S. Gupta, A. Agrawal, K. Gopalakrishnan and P. Narayanan, ‘Deep learning with limited numerical precision,’ in *International conference on machine learning*, PMLR, 2015, pp. 1737–1746.
- [84] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv and Y. Bengio, ‘Binarized neural networks,’ in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon and R. Garnett, Eds., vol. 29, Curran Associates, Inc., 2016.
- [85] X. Sun, N. Wang, C.-Y. Chen *et al.*, ‘Ultra-low precision 4-bit training of deep neural networks,’ in *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [86] AMD, *Xilinx power estimator (xpe)*, 2024. [Online]. Available: <https://www.xilinx.com/products/technology/power/xpe.html>.
- [87] AMD, *Virtex ultrascale+ fpga data sheet (ds923)*, 2021. [Online]. Available: <https://docs.amd.com/v/u/en-US/ds923-virtex-ultrascale-plus>.

- [88] R. B. Abdelhamid, G. Kuwazawa and Y. Yamaguchi, ‘Quantitative study of floating-point precision on modern fpgas,’ in *Proceedings of the 13th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies*, 2023, pp. 49–58.
- [89] J. Stallkamp, M. Schlipsing, J. Salmen and C. Igel, ‘Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition,’ *Neural Networks*, no. 0, 2012, ISSN: 0893-6080. DOI: 10.1016/j.neunet.2012.02.016.
- [90] J. Yuan, Z. Wei and W. Guo, ‘Mixed-precision federated learning via multi-precision over-the-air aggregation,’ in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, 2025, pp. 1–7. DOI: 10.1109/WCNC61545.2025.10978511.
- [91] P. Lewis *et al.*, ‘Retrieval-augmented generation for knowledge-intensive nlp tasks,’ *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.
- [92] Y. Deldjoo *et al.*, ‘A review of modern recommender systems using generative models (gen-recsys),’ in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD ’24, Barcelona, Spain: Association for Computing Machinery, 2024, pp. 6448–6458, ISBN: 9798400704901. DOI: 10.1145/3637528.3671474.
- [93] D. Amodei *et al.*, ‘Deep speech 2 : End-to-end speech recognition in english and mandarin,’ in *Proceedings of The 33rd International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., ser. Proceedings of Machine Learning Research, vol. 48, New York, New York, USA: PMLR, Jun. 2016, pp. 173–182.
- [94] R. Ardila, M. Branson, K. Davis *et al.*, ‘Common voice: A massively-multilingual speech corpus,’ in *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, 2020, pp. 4211–4215.
- [95] PwC, *The impact of voice assistants on consumer behavior*, Accessed: 2025-03-14, 2025. [Online]. Available: <https://www.pwc.com/us/en/services/consulting/library/consumer-intelligence-series/voice-assistants.html>.
- [96] Y. Kim, R. Daly, J. Kim *et al.*, ‘Flipping bits in memory without accessing them: An experimental study of dram disturbance errors,’ *ACM SIGARCH Computer Architecture News*, vol. 42, no. 3, pp. 361–372, 2014.
- [97] P. Frigo, E. Vannacc, H. Hassan *et al.*, ‘Trtrespass: Exploiting the many sides of target row refresh,’ in *2020 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2020, pp. 747–762.
- [98] P. Jattke, M. Wipfli, F. Solt, M. Marazzi, M. Bölcskei and K. Razavi, ‘Zenhammer: Rowhammer attacks on amd zen-based platforms,’ in *33rd USENIX Security Symposium (USENIX Security 2024)*, 2024.
- [99] M. Seaborn and T. Dullien, ‘Exploiting the dram rowhammer bug to gain kernel privileges,’ *Black Hat*, vol. 15, no. 71, p. 2, 2015.

- [100] D. Gruss, C. Maurice and S. Mangard, ‘Rowhammer. js: A remote software-induced fault attack in javascript,’ in *Detection of Intrusions and Malware, and Vulnerability Assessment: 13th International Conference, DIMVA 2016, San Sebastián, Spain, July 7-8, 2016, Proceedings 13*, Springer, 2016, pp. 300–321.
- [101] A. Tatar, R. K. Konoth, E. Athanasopoulos, C. Giuffrida, H. Bos and K. Razavi, ‘Throwhammer: Rowhammer attacks over the network and defenses,’ in *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, 2018, pp. 213–226.
- [102] K. Eykholt, I. Evtimov, E. Fernandes *et al.*, ‘Robust physical-world attacks on deep learning visual classification,’ in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1625–1634.
- [103] N. Carlini and D. Wagner, ‘Audio adversarial examples: Targeted attacks on speech-to-text,’ in *2018 IEEE security and privacy workshops (SPW)*, IEEE, 2018, pp. 1–7.
- [104] O. Mutlu and J. S. Kim, ‘Rowhammer: A retrospective,’ *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 8, pp. 1555–1571, 2019.
- [105] H. Hassan, Y. C. Tugrul, J. S. Kim, V. Van der Veen, K. Razavi and O. Mutlu, ‘Uncovering in-dram rowhammer protection mechanisms: A new methodology, custom rowhammer patterns, and implications,’ in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, 2021, pp. 1198–1213.
- [106] F. de Ridder, P. Frigo, E. Vannacci, H. Bos, C. Giuffrida and K. Razavi, ‘{Smash}: Synchronized many-sided rowhammer attacks from {javascript},’ in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 1001–1018.
- [107] Y. Jiang, H. Zhu, H. Shan, X. Guo, X. Zhang and Y. Jin, ‘Trroscope: Understanding target row refresh mechanism for modern ddr protection,’ in *2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, IEEE, 2021, pp. 239–247.
- [108] Z. B. Aweke, S. F. Yitbarek, R. Qiao *et al.*, ‘Anvil: Software-based protection against next-generation rowhammer attacks,’ *ACM SIGPLAN Notices*, vol. 51, no. 4, pp. 743–755, 2016.
- [109] L. Cojocar, K. Razavi, C. Giuffrida and H. Bos, ‘Exploiting correcting codes: On the effectiveness of ecc memory against rowhammer attacks,’ in *2019 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2019, pp. 55–71.
- [110] L. Cojocar, J. Kim, M. Patel *et al.*, ‘Are we susceptible to rowhammer? an end-to-end methodology for cloud providers,’ in *2020 IEEE symposium on security and privacy (SP)*, IEEE, 2020, pp. 712–728.
- [111] Y. Xiao, X. Zhang, Y. Zhang and R. Teodorescu, ‘One bit flips, one cloud flops: {cross-vm} row hammer attacks and privilege escalation,’ in *25th USENIX security symposium (USENIX Security 16)*, 2016, pp. 19–35.
- [112] I. Kang, W. Wang, J. Kim *et al.*, ‘{Sledgehammer}: Amplifying rowhammer via bank-level parallelism,’ in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 1597–1614.

- [113] P. Pessl, D. Gruss, C. Maurice, M. Schwarz and S. Mangard, ‘{Drama}: Exploiting {dram} addressing for {cross-cpu} attacks,’ in *25th USENIX security symposium (USENIX security 16)*, 2016, pp. 565–581.
- [114] R. Qiao and M. Seaborn, ‘A new approach for rowhammer attacks,’ in *2016 IEEE international symposium on hardware oriented security and trust (HOST)*, IEEE, 2016, pp. 161–166.
- [115] Y. Jang, J. Lee, S. Lee and T. Kim, ‘Sgx-bomb: Locking down the processor via rowhammer attack,’ in *Proceedings of the 2nd Workshop on System Software for Trusted Execution*, 2017, pp. 1–6.
- [116] M. Lipp, M. Schwarz, L. Raab *et al.*, ‘Nethammer: Inducing rowhammer faults through network requests,’ in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, IEEE, 2020, pp. 710–719.
- [117] C. Szegedy, ‘Intriguing properties of neural networks,’ *arXiv preprint arXiv:1312.6199*, 2013.
- [118] I. J. Goodfellow, J. Shlens and C. Szegedy, ‘Explaining and harnessing adversarial examples,’ *arXiv preprint arXiv:1412.6572*, 2014.
- [119] A. Kurakin, I. J. Goodfellow and S. Bengio, ‘Adversarial examples in the physical world,’ in *Artificial intelligence safety and security*, Chapman and Hall/CRC, 2018, pp. 99–112.
- [120] T. B. Brown, D. Mané, A. Roy, M. Abadi and J. Gilmer, ‘Adversarial patch,’ *arXiv preprint arXiv:1712.09665*, 2017.
- [121] I. Evtimov, K. Eykholt, E. Fernandes *et al.*, ‘Robust physical-world attacks on machine learning models,’ *arXiv preprint arXiv:1707.08945*, vol. 2, no. 3, p. 4, 2017.
- [122] M. Sharif, S. Bhagavatula, L. Bauer and M. K. Reiter, ‘Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition,’ in *Proceedings of the 2016 acm sigsac conference on computer and communications security*, 2016, pp. 1528–1540.
- [123] A. Athalye, L. Engstrom, A. Ilyas and K. Kwok, ‘Synthesizing robust adversarial examples,’ in *International conference on machine learning*, PMLR, 2018, pp. 284–293.
- [124] R. Duan, X. Ma, Y. Wang, J. Bailey, A. K. Qin and Y. Yang, ‘Adversarial camouflage: Hiding physical-world attacks with natural styles,’ in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1000–1008.
- [125] X. Yuan, Y. Chen, Y. Zhao *et al.*, ‘{Commandersong}: A systematic approach for practical adversarial voice recognition,’ in *27th USENIX security symposium (USENIX security 18)*, 2018, pp. 49–64.
- [126] M. Zhou, Z. Qin, X. Lin, S. Hu, Q. Wang and K. Ren, ‘Hidden voice commands: Attacks and defenses on the vcs of autonomous driving cars,’ *IEEE Wireless Communications*, vol. 26, no. 5, pp. 128–133, 2019.

- [127] Y. Bengio, J. Louradour, R. Collobert and J. Weston, ‘Curriculum learning,’ in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML ’09, Montreal, Quebec, Canada: Association for Computing Machinery, 2009, pp. 41–48, ISBN: 9781605585161. DOI: 10 . 1145 / 1553374 . 1553380. [Online]. Available: <https://doi.org/10.1145/1553374.1553380>.
- [128] A. Olgun, H. Hassan, A. G. Yağlıkçı *et al.*, ‘Dram bender: An extensible and versatile fpga-based infrastructure to easily test state-of-the-art dram chips,’ *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 12, pp. 5098–5112, 2023.
- [129] *Alveo u200 and u250 data center accelerator cards data sheet (ds962)*, Accessed: 28-Feb-2025. [Online]. Available: <https://docs.amd.com/r/en-US/ds962-u200-u250>.
- [130] *Mta18asf2g72pz-2g3b1 datasheet*, Accessed: 28-Feb-2025. [Online]. Available: <https://www.mouser.co.uk/datasheet/2/671/asf18c2gx72pz-3079314.pdf>.
- [131] A. Gulati, J. Qin, C.-C. Chiu *et al.*, ‘Conformer: Convolution-augmented transformer for speech recognition,’ *arXiv preprint arXiv:2005.08100*, 2020.
- [132] S. Krizan, S. Beliaev, B. Ginsburg *et al.*, ‘Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions,’ in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 6124–6128.
- [133] S. Majumdar, J. Balam, O. Hrinchuk, V. Lavrukhin, V. Noroozi and B. Ginsburg, ‘CitriNet: Closing the gap between non-autoregressive and autoregressive end-to-end models for automatic speech recognition,’ *arXiv preprint arXiv:2104.01721*, 2021.
- [134] S. Kim, A. Gholami, A. Shaw *et al.*, ‘Squeezeformer: An efficient transformer for automatic speech recognition,’ *Advances in Neural Information Processing Systems*, vol. 35, pp. 9361–9373, 2022.
- [135] M. Burchi and V. Vielzeuf, ‘Efficient conformer: Progressive downsampling and grouped attention for automatic speech recognition,’ in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2021, pp. 8–15.
- [136] J. Yuan, Y. Tang and W. Guo, *Rag-based user profiling for precision planning in mixed-precision over-the-air federated learning*, 2025. arXiv: 2503 . 15569 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2503.15569>.
- [137] J. Yuan, Y. Hao, W. Guo, Y. Wu and C. Gu, *Remote rowhammer attack using adversarial observations on federated learning clients*, 2025. arXiv: 2505.06335 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2505.06335>.
- [138] T. Sery, N. Shlezinger, K. Cohen and Y. C. Eldar, ‘Over-the-air federated learning from heterogeneous data,’ *IEEE Transactions on Signal Processing*, vol. 69, pp. 3796–3811, 2021.