Fair and Secure Federated Adversarial Training

by

Muhammad Kaleem Ullah KHAN

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, "AUGUST 21, 2026"

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Kaiwen Zhang, Thesis supervisor
Department of Software Engineering and IT, École de technologie supérieure

Mr. Chamseddine Talhi, Thesis Co-Supervisor
Department of Software Engineering and IT, École de technologie supérieure

Mr. Georges Kaddoum, Chair, Board of Examiners
Department of Electrical Engineering, École de technologie supérieure

Mr. Ulrich Aïvodji, Member of the Jury
Department of Software Engineering and IT, École de technologie supérieure

Mr. Yazan Otoum, External Independent Examiner
School of Electrical Engineering and Computer Science, Faculty of Engineering, University of Ottawa

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON "AUGUST 18, 2026"

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Kaiwen Zhang, for his unwavering guidance, intellectual rigour, and commitment to pushing the boundaries of what is possible in federated learning research. His mentorship has been instrumental in shaping both my research trajectory and my growth as a scholar.

I am equally grateful to my co-supervisor, Professor Chamseddine Talhi, for his invaluable insights, generous support, and encouragement throughout my doctoral journey. His expertise in security and distributed systems has profoundly influenced the direction of this work.

I would also like to acknowledge Professor Danish at Algoma University, whose early collaboration laid important groundwork for my research career and whose continued support has been a source of motivation. This research was conducted at the École de technologie supérieure (ÉTS) in Montréal, and I am thankful to the department and institution for providing the computational resources, infrastructure, and academic environment that made this work possible. I am also grateful to Compute Canada for providing the high-performance computing resources essential to the empirical evaluation presented in this thesis.

My sincere thanks go to my colleagues and fellow researchers for the stimulating discussions, collaborative spirit, and camaraderie that made the long hours of research more rewarding.

Finally, and most profoundly, I owe an immeasurable debt of gratitude to my beloved wife, Zainab Gulshad, whose patience, sacrifices, and unwavering belief in me have been my greatest source of strength throughout this journey. To my two sons, Hussain Ali Khan and Dawood Ali Khan, your smiles and laughter have been the light that carried me through the most challenging moments of this work.

I am deeply grateful to my father, Dilshad Khan, and my mother, Naveeda Khanum, whose prayers, love, and lifelong dedication to my education have been the foundation upon which everything I have achieved is built. I would also like to warmly acknowledge my sisters, Maryam Dilshad, Ayesha Dilshad, and Asma Dilshad, and my brothers, Hasan Dilshad and Hafiz

Muhammad Sami Ullah Khan, for their encouragement and the warmth of home they always provided, no matter the distance. My heartfelt thanks go to my brother-in-law, Rana Kabeer Ahmad, Rao Asif Iqbal and Rana Abrar ul Haq, for their steadfast support and brotherhood. I am equally grateful to my cousin Muhammad Bilal and to my dear friends Zeeshan Amir, Sayed Usama Imtiaz Bukhari, and Ghazi Irfan. Your friendship has been a constant source of joy and motivation throughout this long journey.

This thesis is dedicated to all of you.

Apprentissage fédéré adversarial équitable et sécurisé

Muhammad Kaleem Ullah KHAN

RÉSUMÉ

L'entraînement de modèles d'apprentissage automatique à la fois privés, robustes et déployables sur des dispositifs périphériques (edge devices) à ressources limitées nécessite de résoudre trois tensions qui ont largement été traitées de manière isolée. Cette thèse aborde ces trois aspects à travers une suite unifiée de contributions en Federated Adversarial Training (FAT).

La première tension est architecturale : l'agrégation centralisée en apprentissage fédéré introduit un point de défaillance unique ainsi qu'une source de biais, mais la décentralisation naïve de l'agrégation dégrade la qualité du modèle. Nous résolvons ce problème avec FL-EGM, un cadre dans lequel, à chaque tour d'entraînement, le meilleur nœud participant en termes de performance est élu comme agrégateur, et ce nœud affine ensuite le modèle agrégé sur ses propres données locales. Cette étape de Enhanced Global Model (EGM) transforme l'accès privilégié de l'agrégateur aux données d'un désavantage en un atout, offrant une précision équivalente ou supérieure aux références centralisées (jusqu'à 98,5%), une convergence plus rapide et une stabilité même en présence d'un agrégateur biaisé, tout en éliminant le goulot d'étranglement du serveur fixe.

La deuxième tension est computationnelle : l'Adversarial Training (AT), la défense la plus efficace connue contre les attaques d'évasion, impose un coût multiplicatif à chaque client participant, ce qui la rend irréaliste pour des dispositifs périphériques à ressources limitées. Nous résolvons ce problème avec FedHAT, un cadre qui stratifie les clients selon leur capacité de ressources. Les clients à fortes ressources effectuent un AT hybride avec diversification des attaques via PGD et FFGSM ; les clients à faibles ressources contribuent à un entraînement standard sur des données propres, fournissant une diversité distributionnelle que la seule couche adversariale ne peut pas offrir. Les deux signaux sont combinés via une agrégation pondérée par groupe suivie d'une étape de raffinement EGM, permettant d'obtenir une robustesse supérieure aux références de l'état de l'art dans des conditions IID et non-IID, et d'améliorer la robustesse face à des attaques *non vues* uniquement grâce à la diversification des attaques au niveau de la fédération, sans surcharger les dispositifs contraints ; des études d'ablation confirment que la diversification des attaques et l'EGM contribuent indépendamment.

La troisième tension est énergétique : même les cadres hiérarchisés qui assignent des charges plus légères aux dispositifs les plus faibles finissent par les épuiser, car les niveaux de batterie diminuent au fil des tours, et aucune méthode existante n'adapte l'intensité d'entraînement à cette évolution. Nous résolvons ce problème avec EARS (Energy-Aware Robust Selection), qui introduit une machine à états par client, la Heterogeneous Adaptive Adversarial Training Intensity (H-AATI), ajustant en continu la force des attaques de PGD complet à FGSM puis à un entraînement standard, en fonction de la batterie de chaque appareil et du nombre estimé de tours restants supportables. Un sélecteur multi-objectif complémentaire privilégie les clients disposant de meilleures ressources pour les rounds intensifs. Sur quatre jeux de données de référence

et dix comparaisons avec des baselines, EARS atteint une survie des clients de 100%, une réduction de 40 à 43% de la consommation énergétique cumulée, et une robustesse située à 1–5 points de pourcentage des meilleures méthodes non sensibles à l'énergie, alors que les méthodes existantes épuisent 76 à 100% des dispositifs bas de gamme avant la fin de l'entraînement. Chaque point de pourcentage de robustesse cédé par rapport à la meilleure référence permet environ 40% de réduction d'énergie et 80% d'amélioration de la survie, faisant de l'entraînement adversarial sensible à l'énergie non pas une simple optimisation d'efficacité mais une exigence de correction pour les déploiements hétérogènes. Ensemble, ces trois contributions constituent une progression cohérente allant de l'agrégation décentralisée, à l'allocation de rôles consciente des ressources, jusqu'au contrôle adaptatif de l'intensité énergétique, démontrant que robustesse, confidentialité et durabilité peuvent être atteintes conjointement dans des environnements fédérés hétérogènes.

Mots-clés: Apprentissage fédéré, entraînement adversarial, robustesse adversariale, sélection d'agrégateur, modèle global amélioré, informatique en périphérie (edge computing), efficacité énergétique, dispositifs hétérogènes, agrégation décentralisée

Fair and Secure Federated Adversarial Training

Muhammad Kaleem Ullah KHAN

ABSTRACT

Training machine learning models that are simultaneously private, robust, and deployable on resource-constrained edge devices requires resolving three tensions that have largely been treated in isolation. This thesis addresses all three through a unified sequence of contributions in Federated Adversarial Training (FAT). The first tension is architectural: centralized aggregation in federated learning introduces a single point of failure and a source of bias, yet decentralizing aggregation naively degrades model quality. We resolve it with FL-EGM, a framework in which each training round elects the best-performing participating node as the aggregator, and that node subsequently refines the aggregated model on its own local data. This Enhanced Global Model (EGM) step converts the aggregator's privileged data access from a liability into an asset, delivering accuracy that matches or exceeds centralized baselines (up to 98.5%), converging faster, and remaining stable even when the aggregator is biased, all while eliminating the fixed-server bottleneck.

The second tension is computational: AT, the most effective known defence against evasion attacks, imposes a multiplicative cost on every participating client, making it infeasible for resource-constrained edge devices. We resolve it with FedHAT, a framework that stratifies clients by resource capacity. RR clients perform hybrid AT with attack diversification across PGD and FFGSM; LR clients contribute standard training on clean data, supplying distributional diversity that the adversarial tier alone cannot provide. The two signals are combined via group-weighted aggregation followed by EGM refinement. Across six datasets and attack types, under both IID and non-IID conditions, FedHAT surpasses state-of-the-art baselines and improves robustness against *unseen* attacks purely through federation-level attack diversification, with ablations confirming that diversification and EGM contribute independently, all without overburdening constrained devices.

The third tension is energetic: even tiered frameworks that assign lighter workloads to weaker devices eventually deplete them, because battery levels decline across rounds, and no existing method adapts training intensity to this evolution. We resolve it with EARS (Energy-Aware Robust Selection), which introduces a per-client state machine, the Heterogeneous Adaptive Adversarial Training Intensity (H-AATI), that continuously adjusts attack strength from full PGD to FGSM to standard training as each device's battery and estimated affordable rounds decline. A companion multi-objective client selector preferentially schedules well-resourced clients for intensive rounds. Across four benchmark datasets and ten baseline comparisons, EARS achieves 100% client survival, 40-43% lower cumulative energy consumption, and robust accuracy within 1-5 percentage points of the strongest energy-agnostic baselines, whereas existing methods deplete 76-100% of low-end devices before training completes. Each percentage point of robustness conceded relative to the strongest baseline buys roughly a 40% reduction in energy and an 80% improvement in survival, making energy-aware adversarial training not an efficiency

optimization but a correctness requirement for heterogeneous deployments. Together, the three contributions constitute a coherent progression from decentralized aggregation, through resource-aware role assignment, to energy-adaptive intensity control, demonstrating that robustness, privacy, and sustainability are jointly achievable in heterogeneous federated environments.

Keywords: Federated Learning, Adversarial Training, Adversarial Robustness, Aggregator Selection, Enhance Global Model, Edge Computing, Energy Efficiency, Heterogeneous Devices, Decentralized Aggregation

TABLE OF CONTENTS

Page

INTRODUCTION 1

CHAPTER 1 BACKGROUND AND RELATED WORK 19
1.1 Federated Learning: Foundations and Aggregation 19
1.2 Centralized and Decentralized Federated Learning 20
1.2.1 Centralized Federated Learning 20
1.2.2 Decentralized Federated Learning 21
1.3 Adversarial Attacks on Deep Neural Networks 22
1.4 Adversarial Training 23
1.5 Federated Adversarial Training 24
1.6 Resource-Aware Client Selection in Federated Learning 25
1.7 Energy-Aware and Sustainable Federated Learning 26
1.8 Adaptive Training Intensity and Heterogeneous Device Management 27
1.9 Aggregator Selection and Model Refinement in Federated Learning 27
1.10 Positioning of the Thesis Contributions 28

CHAPTER 2 DECENTRALIZED FEDERATED LEARNING WITH AGGREGATOR SELECTION AND ENHANCED GLOBAL MODEL GENERATION 31
2.1 Proposed Solutions 31
2.1.1 Baseline Federated Learning 31
2.1.1.1 Baseline Random Aggregator Selection 33
2.1.1.2 Proposed Random and Best Selection of Aggregator with Enhanced Global Model 35
2.2 Results and Discussions 37
2.2.1 Experimental Setup and Datasets 38
2.2.1.1 Evaluation Measures 39
2.2.1.2 Results and Baseline Comparisons without a Biased Aggregator 39
2.2.1.3 Evaluation in the Presence of Biased Aggregator 45
2.3 Conclusion 48

CHAPTER 3 RESOURCE-AWARE HYBRID FEDERATED ADVERSARIAL TRAINING FOR GENERALIZED ROBUSTNESS 49
3.1 Proposed Solution 49
3.1.1 Resource-Aware Client Selection 50
3.1.2 Hybrid Adversarial Training for RR Clients 52
3.1.3 Standard Training for Low-Resource Clients 53
3.1.4 Hybrid Training Loss 54
3.1.5 Aggregation and Enhanced Global Model 54

3.2 Results and Discussions 56
3.3 Experimental Setup 56
3.3.1 Experimental Evaluation of Adversarial Robustness in FL 58
3.3.2 Robustness under Increasing Attack Strength and Federation Size 61
3.3.3 Impact of Aggregator Selection Methods and Components 64
3.3.3.1 Aggregator Selection 64
3.3.3.2 Ablation: Impact of Individual Components in the Complete FedHAT Solution 66
3.4 Conclusion 67

CHAPTER 4 ENERGY-AWARE ADAPTIVE INTENSITY CONTROL FOR SUSTAINABLE FEDERATED ADVERSARIAL TRAINING ON HETEROGENEOUS EDGE DEVICES 69
4.1 Proposed Method 69
4.1.1 Problem Formulation 70
4.1.2 Energy Model 73
4.1.3 Heterogeneous Adaptive Adversarial Training Intensity (H-AATI) 74
4.1.3.1 State Space 74
4.1.3.2 Transition Semantics 75
4.1.3.3 Stabilization Mechanisms 77
4.1.3.4 Comparison with Uniform AATI 78
4.1.4 Energy-Aware Client Selection 78
4.1.5 Complete *EARS* Protocol 79
4.2 Experimental Validation 80
4.2.1 Quantitative Results 80
4.2.2 Multi-Attack Robustness 82
4.2.3 Ablation Study 84
4.2.4 H-AATI Transition Dynamics 86
4.2.5 Device Equity and Survival 88
4.2.6 Energy Breakdown 90
4.2.7 Client Depletion Analysis 93
4.2.8 Improvement Summary 94
4.3 Conclusion 95

CONCLUSION AND RECOMMENDATIONS 97

APPENDIX I SUPPLEMENTARY MATERIAL FOR THE CHAPTER 3 FRAMEWORK 103

BIBLIOGRAPHY 135

LIST OF TABLES

Page

Table 2.1 Baseline and proposed model accuracy comparison on the MNIST dataset 40

Table 2.2 Accuracy and Time Comparison in the Presence of non-Biased Aggregation 43

Table 2.3 Effect of Participant Nodes 44

Table 2.4 Accuracy Comparison of Proposed and Baseline Solutions 44

Table 2.5 Early Stopping Results 45

Table 2.6 Accuracy Comparison with MNIST (LeCun, Bottou, Bengio & Haffner, 1998) Dataset 46

Table 2.7 Accuracy and Time Comparison in the Presence of Biased Aggregation .. 47

Table 2.8 Early Stopping with Biased Aggregator 48

Table 3.1 Resource sampling distributions for the two-tier device population. b_i is normalized battery charge; cpu_i and gpu_i are available capacity percentages; bw_i is uplink bandwidth in Mbps 57

Table 3.2 Comparison of Accuracy and adversarial robustness on three different datasets under IID settings 58

Table 3.3 Comparison of Accuracy and adversarial robustness on three different datasets under Non-IID settings 59

Table 3.4 Accuracy and adversarial robustness on diverse datasets under non-iid settings 61

Table 3.5 Comparison of FedHAT Solution with different Aggregator Selection Strategies Using PGD-20 and MIM Attacks 66

Table 3.6 Comparison of individual attack-based FAT models against the complete FedHAT framework 67

Table 4.1 Device class profiles. Energy coefficient γ reflects compute efficiency; higher values indicate less efficient hardware 71

Table 4.2 Complete *H-AATI* cascades. Device-aware allocation of adversarial training intensity across datasets and hardware tiers, showing feasible configurations of PGD (varying steps), FGSM, and standard training ($\epsilon = 0$). Entries indicate the relative training cost or feasibility per device class (high-end, mid-range, low-end) 75

Table 4.3 **Main results** on MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100 under PGD attack. "Clean" and "Robust" denote accuracy (%). "Energy" is cumulative consumption (mAh). "Eff." is robust accuracy per energy $\times 1000$. Best adversarial method values in **bold**; best clean accuracy underlined 80

Table 4.4 **Multi-attack robustness evaluation** at different ϵ for MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100. Values are accuracy (%). AA combines APGD-CE, APGD-DLR, and APGD-CW 83

Table 4.5 ***H-AATI* transition summary** for *EARS* across four datasets. "pct_thr" = battery percentage threshold; "cost_ovr" = cost override; "death_thr" = death threshold. High/Mid/Low columns show transitions per device class 87

Table 4.6 **Energy breakdown** (E units) across four datasets. *Comp %* indicates the fraction of energy spent on computation. *E/Round* is the average energy per round 92

Table 4.7 **Client Depletion Summary** across four datasets. "1st Death" and "Last Death" indicate the round of the first and last client death. Deaths are broken down by device type 96

LIST OF FIGURES

Page

Figure 2.1 Comparison of Architectures 32

Figure 2.2 Accuracy and Loss Comparison of Presented Models 42

Figure 3.1 Architecture of FedHAT: Resource-Aware Hybrid FAT with Enhanced Global Model 50

Figure 3.2 Effect of federation size on adversarial robustness of DBFAT on MNIST with increasing ϵ values 62

Figure 3.3 Effect of federation size on adversarial robustness of FedHAT on MNIST with increasing ϵ values 63

Figure 3.4 Effect of federation size on adversarial robustness of DBFAT on CIFAR-10 with different epsilon values and attacks 64

Figure 3.5 Effect of federation size on adversarial robustness of FedHAT under PGD-20 and MIM attacks on CIFAR-10 65

Figure 4.1 This *EARS* Architecture showing how server-side energy-aware selection interacts with client-side H-AATI adaptation, local AT, energy consumption, and battery-state feedback 70

Figure 4.2 Ablation study of EARS components across datasets. Results show that removing selection (EARS-Uniform) and adaptive attack intensity (EARS-NoAATI) degrade robustness, increase energy consumption, and reduce client survival 85

Figure 4.3 Device-class survival rates (%) and equity (σ) at 100 and 200 rounds across four datasets. Lower σ indicates more equitable survival across device classes 89

LIST OF ABBREVIATIONS

AA	AutoAttack
AR	Adversarial Robustness
AT	Adversarial Training
C&W	Carlini-Wagner
CAT	Curriculum AT
CFL	Centralized Federated Learning
CNN	Convolutional Neural Network
D2D	Device-to-Device
DFL	Decentralized Federated Learning
DND	Department of National Defence
DSGD	Decentralized Stochastic Gradient Descent
EARS	Energy-Aware Robust Selection
EAT	Ensemble Adversarial Training
EGM	Enhanced Global Model
ETS	École de Technologie Supérieure
FAT	Federated Adversarial Training
FedAvg	Federated Averaging
FedHAT	Federated Hybrid Adversarial Training with EGM
FFGSM	Fast-FGSM

FGSM	Fast Gradient Sign Method
FL	Federated Learning
FL-EGM	Federated Learning with Enhanced Global Model
FRP	Federated Robustness Propagation
H-AATI	Heterogeneous Adaptive Adversarial Training Intensity
H-FAT	Hybrid Federated Adversarial Training
IID	Independent and Identically Distributed
LR	Low-Resource
MIM	Momentum Iterative Method
PGD	Projected Gradient Descent
PGD-AT	PGD-based Adversarial Training
Q-DFL	Quantization-Based Decentralized Federated Learning
RR	Rich-Resource
SGD	Stochastic Gradient Descent
SMC	Secure Multiparty Computation
ST	Standard Training

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

$\mathcal{A}_i^\tau$	Attack configuration for level i, device class τ
α_{base}	Base energy coefficient (joules per floating-point operation)
A	Selected aggregator node
b_i	Normalized battery charge fraction for client i
b_k^t	Sampled uplink bandwidth for client k at round t (Mbps)
$b_{\min}$	Worst-case minimum bandwidth (Mbps)
B_k^0	Initial battery capacity of client k (mAh)
B_k^t	Remaining battery of client k at round t (mAh)
β_k^t	Battery fraction of client k: B_k^t / B_k^0
C_τ	Cascade of intensity levels for device class τ
C_l	Set of Low-Resource (LR) clients
C_r	Set of Rich-Resource (RR) clients
$\mathcal{D}_k$	Local dataset of client k
d_k	Death threshold of client k (mAh)
δ	Adversarial perturbation
$\Delta_{\min}$	Minimum dwell time (round-locking parameter)
E	Number of local training epochs
E_k^t	Energy consumed by client k at round t (mAh)
E_k^{comm}	Communication energy of client k

E_k^{comp}	Computational energy of client k
$\hat{E}_k^t$	Conservative energy estimate for client k at round t
ϵ	Perturbation budget (adversarial attack strength)
η	Learning rate; protocol overhead fraction
f_θ	Model parameterized by θ
F_k	Compute throughput of client k (FLOPS)
G	Global model
$G'(t+1)$	Enhanced Global Model at round $t+1$
γ_k	Energy coefficient of client k
h	Hysteresis band
K	Number of clients selected per communication round
$\mathcal{L}$	Loss function
λ	Trade-off hyperparameter (e.g., TRADES, hybrid loss)
L_τ	Number of intensity levels for device class τ
M	Model size (megabytes)
m_c	Number of local data points for client c
N	Total number of clients
ω	Attack overhead factor
Φ	Model floating-point operations per sample
ϕ_i^τ	Battery percentage threshold for level i, device class τ

P_k	Data distribution of client k
P_k^{rx}	Receive power of client k (mW)
P_k^{tx}	Transmit power of client k (mW)
$\mathbf{r}_i$	Resource vector for client i: $(b_i,\ cpu_i,\ bw_i,\ gpu_i)$
r_k^t	Consecutive rounds client k has stayed in current H-AATI state
R_k^{afford}	Number of affordable remaining rounds for client k
R_{min}	Minimum affordable rounds threshold (cost-override parameter)
ρ	Top fraction for Rich-Resource client partitioning
ρ_k	Device profile of client k: $(\tau_k, F_k, B_k^0, V_k, P_k^{\text{tx}}, P_k^{\text{rx}}, \gamma_k, d_k)$
$\mathcal{S}^t$	Selected subset of clients at round t
s_i	Weighted resource score for client i
s_k^t	H-AATI state of client k at round t
t	Communication round index
t_k^{comm}	Transmission time for client k
T	Total number of communication rounds
T_w	Number of warm-up rounds
τ_k	Device class of client $k \in \{\text{high, mid, low}\}$
θ	Model parameters (weights)
V_k	Operating voltage of client k (V)
w^t	Global model parameters at round t

$w_b, w_{cpu}, w_{bw}, w_{gpu}$ Weight coefficients for resource score

x Input data sample

y True label

INTRODUCTION

A large and diverse training set is required for better performance of Deep Learning (DL) models. A neural network learns by adjusting its parameters across many examples until it captures the patterns in the data, and how well it generalizes to unseen inputs depends on both the volume and the diversity of the training data (Sun *et al.*, 2017). The dominant recipe of the past decade follows directly: collect a large dataset in one place, train a deep model over it, and deploy the result. This works as long as the data is yours to collect. Much of the most useful data is not available publicly. The text a phone keyboard would learn from belongs to millions of individual users; the scans a diagnostic model would learn from are protected medical records; the logs a factory's monitoring system would learn from reveal how the plant operates. Data of this kind is personal to the people it describes, valuable to the firms that hold it, and in many cases protected by law. It is scattered across many devices and owners rather than in a single repository. A growing body of regulation now governs how such data may be moved and processed, including the General Data Protection Regulation (GDPR) in the European Union (Regulation, 2018), the California Consumer Privacy Act (CCPA) (califoenia, 2018) and the Consumer Privacy Bill of Rights (USA, 2015) in the United States, the Cybersecurity Law of the People's Republic of China (China, 2018), and the Personal Data Protection Act in Singapore (et al., 2013). Faced with these constraints, data holders rarely release their data: different Internet Service Providers and institutions seldom pool their records, making it impractical to assemble a single comprehensive dataset Cui *et al.* (2021c).

Privacy regulation and the migration of computation toward the network edge have widened this gap. More data exists than ever, yet a growing share of it cannot lawfully or practically be moved to a central server Kairouz, McMahan *et al.* (2021). The centralized then train recipe, therefore, breaks down exactly where the data is most valuable, motivating methods that bring the model to the data rather than the data to the model.

Federated Learning (FL) answers this constraint by bringing the training to the data rather than the data to the trainer: each device trains on its own local examples and shares only model updates, never the raw data, and those updates are combined into a shared model that none of the devices could have built alone (McMahan, Moore, Ramage, Hampson & y Arcas, 2017). A useful model can therefore be trained across hospitals, phones, or sensors whose data could never be pooled.

FL resolves the data-availability problem and introduces new ones, and those new problems are the subject of this thesis. In its standard form, it is centralized: a single server selects the participants, aggregates their updates, and owns the global model, introducing a single point of failure and a single point of bias in an otherwise distributed system (He, Bian & Jaggi, 2018; Kalra, Wen, Cresswell, Volkovs & Tizhoosh, 2023). The devices that form a federation are deeply uneven in computing power and energy. And the property this thesis treats as essential, security against adversarial attacks, is far more costly to achieve than ordinary accuracy and collides with those hardware limits. The goal of this thesis is to make FL both secure and fair under these conditions, and each of its three contributions answers one of these problems: the first replaces the biased central aggregator, the second lets resource-poor devices join robust training they could not otherwise afford, and the third keeps devices of every capability alive across a long, energy-hungry training run. The rest of this introduction develops each problem, explains why existing fixes fall short, and shows how the contributions fit together.

FL (Konečnỳ *et al.*, 2016; McMahan *et al.*, 2017) was conceived as a principled response to precisely this constraint. Rather than moving data to a model, FL inverts the relationship and moves the model to the data: each participating client trains locally on its private dataset and transmits only a parameter update, which a coordinating server aggregates into a shared global model. The raw data never leaves the device, yet the resulting global model benefits from the federation's full statistical diversity. In the canonical *FedAvg* procedure (McMahan

et al., 2017), the server broadcasts the current model, each selected client performs several epochs of local stochastic gradient descent, and the server combines the returned updates by a weighted average proportional to local dataset sizes. This deceptively simple loop decouples model training from centralized data collection and has made FL attractive across healthcare, finance, telecommunications, and mobile computing (Kairouz *et al.*, 2021).

The elegance of the federated idea, however, conceals a number of subtleties that surface most sharply under adversarial conditions. The first of these concerns the fragility of the models being trained. Deep neural networks, for all their representational power, are remarkably brittle: a perturbation to an input that is imperceptible to a human observer, constructed by following the gradient of the loss with respect to the input, can convert a confident and correct prediction into a confident and incorrect one (Szegedy, Zaremba *et al.*, 2014; Goodfellow, Shlens & Szegedy, 2014; Madry, Makelov, Schmidt, Tsipras & Vladu, 2018). This phenomenon, the existence of adversarial examples, is not a superficial artifact of any particular architecture or training recipe. It reflects something fundamental about the way high-dimensional, approximately linear functions interpolate between their training points (Goodfellow *et al.*, 2014). Two properties make it especially troubling in a federated, edge-deployed setting. First, adversarial perturbations transfer: an example crafted to fool one model frequently fools another trained independently on similar data. Second, the attack surface widens precisely where the deployment is most valuable. The same devices that make FL compelling, ubiquitous, and unsupervised, operating in uncontrolled physical environments, are the devices on which an adversary is most likely to encounter and most able to manipulate the inputs a model must classify.

The most effective known defence against adversarial examples is AT (Madry *et al.*, 2018), which incorporates adversarially perturbed inputs directly into the training loop, formalized as a min-max robust optimization in which the inner maximization searches for the worst-case perturbation within a bounded set and the outer minimization fits the model to resist it. In

practice, the inner problem is approximated by Projected Gradient Descent (PGD) (Madry *et al.*, 2018), and herein lies the difficulty that motivates much of this thesis. Generating each adversarial example by PGD requires as many additional forward-backward passes as there are attack steps, multiplying the per-batch cost of training by a factor that scales linearly with the number of steps; multi-step PGD routinely imposes a tenfold to thirtyfold overhead relative to standard training (Shafahi *et al.*, 2019b). In a centralized, GPU-rich data centre, this overhead is an accountable line item. In a federation of heterogeneous, battery-powered edge devices, it is frequently catastrophic. A workload that a server absorbs without notice can drain a smartphone's battery, thermally throttle an embedded processor, or simply exceed the memory available on an IoT node.

We therefore arrive at a three-way collision. Federation is needed to respect privacy and to harness data that cannot be moved. Adversarial robustness is needed because the models will operate in hostile, uncontrolled environments. Yet the standard route to robustness, AT, imposes a multiplicative computational cost that the heterogeneous, resource-constrained devices central to the federated vision cannot uniformly bear. Each of these three forces, privacy-preserving federation, adversarial robustness, and resource heterogeneity, is individually well studied. What has received far less attention is their interaction. The literature on FL has matured around statistical and systems heterogeneity (Li, Sahu, Talwalkar & Smith, 2020b; Karimireddy *et al.*, 2020; Lai, Zhu, Madhyastha & Chowdhury, 2022); the literature on adversarial robustness has matured around centralized training and evaluation (Madry *et al.*, 2018; Zhang *et al.*, 2019b; Croce & Hein, 2020); and the comparatively young literature on FAT has, for the most part, transplanted centralized AT into the federated loop and asked how to make it work better, rather than asking who can afford to participate in it at all (Zizzo, 2020; Chen, 2022; Zhang & et al., 2023). This thesis takes the interaction itself as its subject. It argues that robustness, privacy, and sustainability are not three separate objectives to be optimized in isolation, but a single coupled

problem whose tensions must be resolved jointly, and it develops a sequence of mechanisms that do so.

The Three Structural Problems

When FL, adversarial robustness, and hardware heterogeneity are considered together, three key structural problems arise. Each of these problems is important on its own, but when combined, they interact in complex ways that significantly increase system difficulty. In this design space, addressing any one problem in isolation often worsens the other two, making them impossible to solve effectively on their own. Therefore, this thesis focuses on their tight coupling and interdependence. We first define each problem precisely and then, in the next subsection, examine the interactions linking them.

The Aggregation Problem

Standard FL (McMahan *et al.*, 2017) relies on a fixed central server to collect client updates and aggregate them into the global model. This architectural choice, convenient and almost universal, carries three liabilities. The first is fault tolerance: the server is a single point of failure, and if it is compromised, unavailable, or simply overloaded, the entire training process degrades or halts (Beltrán *et al.*, 2023). The second is bias: when the aggregator holds its own data and participates in the federation, or when it is administered by a party with an interest in particular outcomes, it can silently skew the global model toward favoured distributions, a vulnerability that becomes acute in security-sensitive deployments (Fang, Cao, Jia & Gong, 2020). The third is a more subtle privacy concern: routing every client's update through a single node allows that node to infer sensitive properties of the underlying data from gradient statistics, even though the raw data is never transmitted (Zhu, Liu & Han, 2019). The natural remedy is to decentralize the aggregation role, distributing it across the federation so that no single node holds permanent control (He *et al.*, 2018). But naive decentralization, cycling the aggregation

duty through clients in a fixed order, or selecting an aggregator uniformly at random each round, discards the quality that careful, performance-aware aggregation provides. Decentralized FL methods to date have reduced trust dependencies and improved fault tolerance, yet none of them uses participant performance to elect the aggregator, and none exploits the elected aggregator's own data to refine the aggregated model after the fact. The aggregation problem, then, is to decentralize aggregation without sacrificing model quality, and ideally to turn the aggregator's privileged data access from a liability into an asset.

The Resource Problem

FAT (Zizzo, 2020) applies AT within the federated loop, and in doing so inherits the full computational burden of adversarial example generation and distributes it to every participating client. In a realistic device population, this is untenable. A smartphone with a shared, thermally constrained CPU cannot run multi-step PGD on batches of the size a server GPU handles without effort. The prevailing response in the FAT literature has been to refine the *quality* of AT, how to mix adversarial and clean examples (Zizzo, 2020), how to calibrate logits under non-IID data (Chen, 2022), how to reweight decision boundaries dynamically across rounds (Zhang & et al., 2023), or how to partition large models into memory-feasible modules (Tang, 2025). What these methods share is a tacit assumption that every client can, and should, perform some form of AT. This assumption excludes a large and important fraction of realistic edge populations: the LR devices that cannot afford AT at all. Excluding them is not a neutral simplification. The data held by resource-constrained devices is frequently the most geographically, demographically, and distributionally diverse in the federation; discarding it forfeits exactly the coverage that motivated the federated approach in the first place. A second, orthogonal weakness compounds the first: single-attack AT tends to overfit to the attack it was trained against and generalizes poorly to unseen attacks (Tramèr, 2017). Ensemble approaches that defend against multiple attacks exist, but they require each client to maintain or query

multiple surrogates, which only deepens the per-client cost the resource problem is trying to relieve. The resource problem, therefore, is to make federated AT inclusive, to extract a useful contribution from every device regardless of capacity, and simultaneously robust to a broad range of attacks, without asking any single constrained device to bear a multi-attack workload.

The Energy Problem

Even a framework that assigns lighter workloads to weaker devices in any given round does not, by itself, make federated AT sustainable. The reason is temporal. Battery levels decline monotonically across rounds, and the workload a device can afford is a function of its remaining charge, not its initial capacity. A device that can comfortably perform FGSM-based AT in round ten may afford only standard training by round fifty, and may be permanently depleted by round one hundred. No existing FAT framework tracks battery state across rounds, and none adapts training intensity to its evolution; the attack configuration assigned to a client is treated as a static property rather than a function of a dwindling resource. The consequence is a systematic and damaging failure mode. Low-end devices deplete and drop out, the surviving population narrows to a high-resource subset, and the global model progressively loses the distributional diversity that FL was designed to capture. Crucially, this is not merely an efficiency concern. When client attrition correlates with device class, the surviving pool ceases to represent the overall data distribution, and the model is silently biased toward the data held by whatever hardware happened to survive. Energy management in this regime is thus a question of correctness, not just of cost. The energy problem is to make federated AT energy-sustainable across the full duration of training, keeping every device alive and contributing from the first round to the last, by adapting training intensity to each device's evolving energy budget and by scheduling intensive work onto the devices best able to absorb it.

Why the Three Problems Are Coupled

It is tempting to treat these three problems as a checklist to be addressed in turn, but they resist such treatment because each interacts with the others. A solution to the aggregation problem that elects a high-performing node and refines the model on its data assumes a trusted-aggregator setting; that same refinement mechanism becomes the natural vehicle for fusing the heterogeneous adversarial and clean signals that the resource problem's tiering produces. A solution to the resource problem that stratifies clients by capacity presupposes that capacity is stable, an assumption the energy problem directly contradicts: the tiers themselves drift as batteries deplete, so a static partition that is sensible in round one is misaligned by round fifty. And a solution to the energy problem that adapts intensity per device must decide *which* devices to schedule for intensive rounds, a selection question whose answer feeds back into the aggregation and resource decisions. The three problems form a chain of failure modes rather than a list of independent difficulties: biased aggregation corrupts the global model's representation, resource exclusion narrows the data the model ever sees, and energy depletion narrows it further over time. A method that closes one link while leaving the others open does not produce a usable system. This coupling is the central observation of the thesis, and it dictates the structure of the contributions: each resolves one tension while preserving, and where possible reinforcing, the solutions to the others.

Research Questions and Objectives

The structural analysis above translates into three research questions, each paired with a concrete objective.

The **first research question** asks whether the dependence on a fixed central aggregation server can be eliminated without sacrificing model quality, and whether the aggregator's privileged data access can be turned to advantage rather than treated as a source of bias. The corresponding

first objective is to design a decentralized aggregation mechanism in which participating nodes dynamically elect and rotate the aggregation responsibility on the basis of performance, and in which the elected aggregator's local data is used to refine the global model without violating any other client's privacy. Success here means matching or exceeding centralized accuracy, converging at least as quickly, and remaining stable under biased-aggregator conditions and across varying federation sizes.

The **second research question** asks how adversarial robustness can be conferred on a federated model when the participating clients differ substantially in computational capacity, and how robustness against a broad range of attacks can be achieved without asking any single constrained device to bear a multi-attack workload. The corresponding **second objective** is to develop a role-specialization framework in which each client contributes according to its capacity: resource-rich clients supply adversarial robustness through diversified attacks distributed across the tier, resource-constrained clients supply distributional diversity through clean-data training, and the two complementary signals are coherently integrated into a single global model. Success here means surpassing state-of-the-art FAT baselines on both clean and robust accuracy, under IID and non-IID conditions, and improving robustness against *unseen* attacks purely through federation-level diversification, all without imposing adversarial costs on the constrained tier.

The **third research question** asks how federated AT can remain energy-sustainable across the full duration of training, so that no device class is systematically depleted and excluded. The corresponding **third objective** is to design a mechanism that tracks each client's evolving battery state and adapts the intensity of AT accordingly, coupled with a selection policy that preferentially schedules clients whose current resources make them most suitable for intensive rounds. Success here means achieving complete client survival across all device classes and datasets, substantially reducing cumulative energy consumption, and conceding only a small, bounded margin of robustness relative to the strongest energy-agnostic baselines.

Thesis Statement

The central claim of this thesis can be stated in a single sentence. *Robustness, privacy, and sustainability in FAT are not competing objectives to be traded against one another, but a single coupled problem that can be resolved jointly, by combining performance-aware decentralized aggregation, resource-aware role specialization, and energy-adaptive intensity control into a coherent progression in which each mechanism reinforces the others.* The three contributions that follow are the constructive proof of this claim. They are deliberately ordered as a progression rather than presented as a set of independent results: FL-EGM establishes a decentralized aggregation primitive and an enhanced-global-model refinement step; FEDHAT reuses that refinement step to fuse the heterogeneous signals produced by resource-aware tiering; and EARS adds the temporal, energy-adaptive control that the static tiering of FEDHAT cannot by itself provide. Read together, they demonstrate that the chain of failure modes identified in Section , biased aggregation, resource exclusion, and energy depletion, can be unwound link by link without the resolution of one link reopening another. A subsidiary but recurring claim, established empirically across all three contributions and argued most forcefully in the third, is that on heterogeneous hardware, energy-aware AT is a correctness requirement rather than a mere efficiency optimization: a device that depletes takes its local data distribution with it, and the resulting selection bias degrades the global model in a way that no amount of additional robustness on the surviving devices can repair.

Summary of Contributions

This thesis makes three principal contributions, each addressing one of the structural problems while preserving or strengthening the solutions to the others. We summarize each in turn, and then describe the thread that unifies them.

Contribution 1: Decentralized Aggregator-Based Federated Learning with Enhanced Global Model Refinement

Chapter 2 presents FL-EGM, which addresses the aggregation problem. Rather than fixing a central server as the permanent aggregator, FL-EGM selects the best-performing participating node as the aggregator for each training round, ensuring that aggregation is always performed by the most reliable node currently available and distributing control across the federation. The conceptual core of the contribution is the Enhanced Global Model (EGM) refinement step, motivated by a simple observation: the elected aggregator, chosen precisely because it performs well, holds local data that is unusually informative about the global model's current decision boundary. In each round, every client except the elected aggregator trains on its local data and transmits an update; the aggregator combines these updates by weighted averaging and then fine-tunes the resulting model on its own raw local data. Because the aggregator is itself a participating client using only its own data, this refinement violates no privacy constraint, and it converts the aggregator's privileged position from a potential source of bias into a source of quality. The refined model is broadcast as the initialization for the next round. Evaluated across multiple datasets, participant configurations, and network settings, including an adversarial intrusion-detection benchmark and the MNIST handwritten-digit benchmark, FL-EGM delivers consistent improvements in accuracy (reaching 98.5%, and 98.4% even under a biased aggregator), convergence speed, precision, recall, and F1 score over both centralized and standard federated baselines, and it remains stable as the number of participants varies and when an aggregator is deliberately biased to exclude a client. Beyond the numbers, the contribution establishes a reusable mechanism, performance-based aggregator election plus EGM refinement, that reappears in subsequent chapters. This work added value to both the MNIST (images) and network intrusion detection datasets (tabular), but no adversarial training is employed because evaluating adversarial robustness on tabular data would require redefining the perturbation, which we leave for future work.

Contribution 2: Tier-Based Federated Adversarial Training with Hybrid Attack Diversification

Chapter 3 presents FedHAT, which addresses the resource problem. In any realistic federation, clients differ greatly in their computational resources: some can comfortably run iterative adversarial attacks, while for others the cost is prohibitive. Existing FAT frameworks either discard the constrained population or uniformly degrade their training; FedHAT instead assigns each population a complementary role. RR clients form the adversarial tier and perform Hybrid AT, with the attack diversified *across* the tier, roughly half of the RR clients train against PGD and half against Fast-FGSM, rather than within any single client's batch. This federation-level diversification achieves ensemble-style cross-attack generalization (Tramèr, 2017) without requiring any single client to solve a multi-attack optimization problem, so the cost of diversity is borne by the federation rather than by individual devices. LR clients, by contrast, perform standard training on their clean local data; their contribution is not robustness but diversity, since these clients often hold data from underrepresented distributions whose inclusion, even without adversarial perturbation, improves the global model's clean accuracy and generalization. The two tiers are fused through group-weighted aggregation that realizes a federation-level hybrid loss, followed by the EGM refinement step inherited from FL-EGM. Extensive experiments across six datasets and a battery of attack types, multiple federation sizes, and both IID and non-IID partitions show that FedHAT consistently outperforms state-of-the-art FAT baselines on both clean and robust accuracy, that it improves robustness against *unseen* attacks purely through federation-level diversification, and that it operates along the established robustness-accuracy frontier. Ablation studies confirm that attack diversification and EGM each contribute independently, and that FedHAT trades a modest reduction in clean accuracy for a substantial gain in robustness, without imposing any adversarial cost on the LR tier. Furthermore, most of the SOTA work on adversarial training and robustness is based on image datasets; therefore, we continue this work on image datasets.

Contribution 3: Energy-Aware Robust Federated Adversarial Training with Heterogeneous Adaptive Intensity

Chapter 4 presents EARS (Energy-Aware Robust Selection), which addresses the energy problem and completes the progression. The motivating observation is that battery levels decline across training rounds, and that no existing FAT framework addresses this decline, resulting in the structural failure mode of premature client depletion, which progressively reduces the federation to a high-resource subset. EARS resolves this through two coordinated mechanisms. The *H-AATI* state machine assigns each client a per-round attack configuration based on its device class, current battery level, and the estimated number of remaining affordable rounds; as battery levels fall, *H-AATI* transitions the client from full PGD to FGSM to standard training rather than maintaining an unsustainable intensity until the device fails, with transitions stabilized by round-locking and hysteresis to prevent oscillation. The companion energy-aware client selector jointly weighs data utility, energy availability, and model contribution, preferentially scheduling clients whose current resources make them suitable for intensive rounds and thereby reducing the frequency with which *H-AATI* must degrade intensity. Validated on MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100 with a heterogeneous three-tier device population, against ten baselines spanning PGD-AT, FGSM-AT, TRADES, DBFAT, and CalFAT, and evaluated under FGSM, PGD, MIM, C&W, and AutoAttack, EARS achieves what no existing FAT method accomplishes: 100% client survival across all device classes and datasets, cumulative energy consumption 40 to 43% lower than the strongest baselines, and robust accuracy within one to five percentage points of energy-agnostic methods, where those baselines instead deplete 76 to 100% of their low-end devices before training completes. Each percentage point of robustness conceded relative to the strongest baseline buys roughly a 40% reduction in energy and an 80% improvement in survival, an exchange rate that reframes energy-aware AT as a correctness requirement rather than an efficiency optimization.

A Unifying Thread

These three contributions are best read as three separate papers but as a single argument unfolding in three movements. FL-EGM contributes the decentralized aggregation primitive and the EGM refinement step. FEDHAT takes the same EGM step and repurposes it as the fusion mechanism for the heterogeneous adversarial and clean signals generated by resource-aware tiering, demonstrating that a primitive introduced for one purpose can serve another without modification. EARS then supplies what FEDHAT's static tiering cannot: the temporal control that keeps the tiers meaningful as energy depletes, and, in doing so, closes the last link in the chain of failure modes. The progression moves from architecture (where aggregation happens and who performs it) through role assignment (who trains adversarially and how) to dynamic control (how intense that training should be and for how long), and at each step, the mechanisms introduced earlier are reused rather than discarded. The cumulative demonstration is that robustness, privacy, and sustainability can be achieved jointly in heterogeneous federated environments, which is the thesis statement of Section .

Scope, Methodology, and Evaluation Philosophy

It is as important to be clear about what this thesis does not attempt as about what it does. We delimit the scope along four axes.

Threat model. The adversarial robustness studied here is robustness to inference-time evasion attacks, perturbations applied to inputs at test time to induce misclassification, under the standard ℓ_∞-bounded threat model with a fixed perturbation budget. This is distinct from training-time poisoning and backdoor attacks (Bhagoji, Chakraborty, Mittal & Calo, 2019; Bagdasaryan, Veit, Hua, Estrin & Shmatikov, 2020), which manipulate the training process itself and which, while important in federated deployments, are outside the present scope. The thesis assumes honest-but-curious clients and, where aggregation is decentralized and refinement is

performed, a trusted-aggregator setting consistent with cross-silo and trusted-edge-gateway federated deployments (Kairouz *et al.*, 2021). Malicious clients, byzantine robustness and formal privacy guarantees such as differential privacy are deliberately left orthogonal; the contributions are designed to be compatible with, but do not themselves provide, such guarantees.

Models and datasets. The empirical work spans a deliberately graduated set of benchmarks, from MNIST and Fashion-MNIST through CIFAR-10 and CIFAR-100 to larger-scale subsets of ImageNet, together with a network intrusion-detection dataset for the decentralized-aggregation contribution. Architectures include a compact convolutional network for the intrusion-detection setting, VGG-style networks, and ResNet-18, chosen to align with the conventions of the federated and adversarial-training literatures, ensuring that comparisons with published baselines are meaningful. The graduation is intentional: simpler datasets cleanly isolate the mechanisms, while harder benchmarks stress them under realistic class diversity and reveal where the robustness-efficiency trade-off tightens.

Heterogeneity model. Device heterogeneity is modelled through a three-tier taxonomy (high-end, mid-range, low-end) with representative compute, battery, voltage, and bandwidth parameters, and energy consumption is computed from a physically motivated model that decomposes per-round cost into computation and communication components, with explicit dependence on the attack configuration. The energy model is calibrated against the CMOS dynamic-power literature and measured profiles of representative edge devices. This is a simulation rather than a physical testbed; the assumptions, constant voltage, independent bandwidth sampling, and a fixed base energy coefficient are stated explicitly, and the limitations they impose are revisited in the concluding chapter.

Evaluation philosophy. Throughout, evaluation is multi-dimensional and adversarially rigorous. Robustness is assessed not against a single attack but against an ensemble, culminating in the parameter-free AutoAttack (Croce & Hein, 2020), which serves as a robustness floor and

guards against the gradient-masking artifacts that have repeatedly undermined apparently strong defences. Beyond accuracy, the energy contribution reports cumulative and per-round energy, client survival, device-class equity, and depletion dynamics, because the central argument, that survival is a representational property and not merely a resource statistic, can only be made by measuring who remains in the federation and when they leave, not just how the surviving model performs.

Organization of the Thesis

The remainder of this thesis is organized as follows. Chapter 1 provides a unified review of the foundational concepts and prior art underpinning all three contributions, covering FL foundations and aggregation strategies, centralized versus decentralized federated architectures, adversarial attacks on deep neural networks, AT methods, FAT, resource and energy-aware client selection, adaptive training-intensity mechanisms, and aggregator selection and model refinement; it closes by positioning the three contributions against the gaps the survey identifies. Chapter 2 presents FL-EGM, establishing the decentralized aggregation framework and the EGM refinement mechanism, and evaluating them across datasets, federation sizes, and biased-aggregator scenarios. Chapter 3 presents FedHAT, introducing resource-aware tier-based role assignment and hybrid attack diversification, and validating them across six datasets, multiple attack types, and IID and non-IID partitions. Chapter 4 presents EARS, completing the framework with energy-adaptive intensity control and battery-aware client selection, and validating it across four datasets and ten baselines under both single-attack and AutoAttack evaluation. A concluding chapter draws the three contributions together, states plainly what the work leaves open, and identifies the directions: a theoretical account of EGM, a learned replacement for the heuristic intensity schedule, evaluation on real-world federated deployments, and a connection to formal privacy guarantees, which are most likely to yield fundamental advances. Appendix I provides supplementary material for EARS, including theoretical properties, computational

and communication overhead, full algorithmic details, an expanded experimental setup, and additional results, ablations, and sensitivity analyses. Each chapter is self-contained and may be read independently, though the contributions build on one another in the order presented, and the reader who follows that order will find the recurring mechanisms, performance-based election, EGM refinement, and resource-aware role assignment, accumulating into the unified framework the thesis statement promises.

Related Publications

The research reported herein was also reported in the following published and submitted research articles. Journal publications are denoted by J, while conference proceedings are marked with C.

C1: **Muhammad Kaleem Ullah Khan**, Zhang K, Talhi C. FL-EGM: Decentralized federated learning using aggregator selection with enhanced global model. *IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2024

C2: **Muhammad Kaleem Ullah Khan**, Zhang K, Danish SM. FedHAT: Resource-Aware Hybrid Federated Adversarial Training for Generalized Robustness. *IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2026.

J1: **Muhammad Kaleem Ullah Khan**, Zhang K, Danish SM. EARS: Energy-Aware Robust Selection for Sustainable Federated Adversarial Training on Heterogeneous Edge Devices. *IEEE Transaction on Sustainable Computing*, 2026. **Status: Under review**

Along with the aforementioned publication and submitted work that contribute to the main contents of this thesis.

CHAPTER 1

BACKGROUND AND RELATED WORK

This chapter provides a unified review of the foundational concepts and prior art that underpin the three contributions of this thesis. The research spans three interconnected themes: decentralized FL with aggregator selection (FL-EGM), resource-aware FAT with hybrid attack diversification (FedHAT), and energy-aware robust FAT on heterogeneous edge devices (EARS). Accordingly, the background is organized into the following areas: FL foundations and aggregation strategies; decentralized FL architectures; adversarial attacks on deep neural networks; AT methods; FAT; resource- and energy-aware client selection; and adaptive training-intensity mechanisms.

1.1 Federated Learning: Foundations and Aggregation

FL was introduced by McMahan *et al.* (2017) as a privacy-preserving paradigm in which a global model is trained collaboratively across a population of clients without requiring them to share raw data. In the canonical `FedAvg` algorithm, a central server broadcasts the current global model weights w_t to a randomly selected subset S_t of clients; each selected client k performs E epochs of stochastic gradient descent (SGD) on its local dataset $\mathcal{D}_k$, producing an updated local model; and the server aggregates the resulting updates by weighted averaging:

$$w_{t+1} = \sum_{k \in S_t} \frac{|\mathcal{D}_k|}{|\bigcup_{k'} \mathcal{D}_{k'}|} w_t^{(k)}. \tag{1.1}$$

This formulation decouples model training from centralized data collection, making FL attractive for applications in healthcare, finance, and mobile computing (Kairouz *et al.*, 2021).

Despite its appeal, FL faces well-documented challenges. *Statistical heterogeneity* (non-IID data distributions across clients) causes local models to diverge from the global optimum, a phenomenon known as client drift (Zhao *et al.*, 2018; Li *et al.*, 2020b). Algorithms such as `FedProx` (Li *et al.*, 2020b) and `SCAFFOLD` (Karimireddy *et al.*, 2020) address drift through proximal regularization and variance-reduction control variates, respectively, while `FedNova` (Wang, Liu, Liang, Joshi & Poor, 2020) normalizes heterogeneous local updates and

`FedDyn` (Acar *et al.*, 2021) applies dynamic regularization aligned with the global objective. *System heterogeneity* (heterogeneous compute, memory, and network bandwidth) can delay or corrupt the aggregation round (Lai *et al.*, 2022; Imteaj, 2022). Communication efficiency is also a concern, and works such as (Sattler, Wiedemann, Müller & Samek, 2019) and Konečnỳ *et al.* (2016) propose gradient compression and quantization to reduce the per-round bandwidth cost.

FL optimizes the performance of interconnected devices through cooperative algorithms and consensus methods (Savazzi, Nicoli & Rampa, 2020), making it well-suited to 5G and similar wireless networks. Privacy amplification through secure aggregation (Bonawitz *et al.*, 2017), differential privacy (Geyer, Klein & Nabi, 2017), and functional encryption (Xu, Baracaldo, Zhou, Anwar & Ludwig, 2019) further reinforce the privacy guarantees offered by FL. Selective model aggregation techniques that transmit localized DNN models to a centralized server have been shown to outperform standard federated averaging approaches (Ye, Yu, Pan & Han, 2020). FL has attracted substantial attention driven by the success of deep neural networks across diverse domains (Huang, 2022; Chen, 2023), and methods such as `FedBN` (Li, Jiang, Zhang, Kamp & Dou, 2021) and `pFedMe` (Fallah, Mokhtari & Ozdaglar, 2020) address personalization under non-IID conditions.

1.2 Centralized and Decentralized Federated Learning

1.2.1 Centralized Federated Learning

Centralized FL (CFL) is the prevailing variant of FL in which a single server orchestrates the entire training process (Beltrán *et al.*, 2023). Although effective in many scenarios, CFL introduces a structural single point of failure: the server must aggregate all model updates, making it both a communication bottleneck and a potential source of bias. If the central aggregator is compromised or biased toward a subset of participants' data distributions, the resulting global model may be unreliable (Fang *et al.*, 2020). Moreover, the requirement to route all local updates through a single node raises privacy concerns, as the aggregator can infer

sensitive information from gradient statistics even without access to the raw data (Zhu *et al.*, 2019).

The Hybrid-Alpha approach enhances privacy in CFL using secure multiparty computation (SMC) and functional encryption (Xu *et al.*, 2019), reducing training time by 68% and minimizing data transfer during CNN training on MNIST. Communication performance and security have also been improved in FL while reducing mobile device communication costs (Asad, Moustafa, Rabhi & Aslam, 2021b). Despite these improvements, CFL's fundamental reliance on a trusted, centrally administered server remains a structural limitation to deployment in adversarial or highly distributed environments.

1.2.2 Decentralized Federated Learning

Decentralized FL (DFL), introduced in He *et al.* (2018), distributes model aggregation among participant nodes by transmitting local updates to federation peers rather than to a single server. DFL eliminates the single point of failure, reduces trust dependencies, and supports self-scaling as new nodes integrate into the federation (McMahan *et al.*, 2017; Augenstein, Hard, Partridge & Mathews, 2021). DFL enhances fault tolerance by keeping nodes updated on each other's status (Wang *et al.*, 2022b) and by distributing trust across the federation, thereby reducing the attack surface (Savazzi, Nicoli, Bennis, Kianoush & Barbieri, 2021).

Several architectures have been proposed under the DFL umbrella. Brain-Torrent (Roy, Siddiqui, Pölsterl, Navab & Wachinger, 2019) eliminates the central authority for medical image segmentation, improving reliability. The segmented gossip model (Hu, Jiang & Wang, 2019) optimizes node bandwidth and reduces training time relative to CFL. Quantization-based DFL (Q DFL) (Ma, Wang & Li, 2023) addresses bandwidth and scalability issues in Industrial IoT settings. ProxyFL (Kalra *et al.*, 2023) enables collaborative learning without a central server in regulated domains such as finance and healthcare, preserving data privacy through differential privacy analysis. TH-DFL (Li *et al.*, 2023b) proposes a trustiness-based hierarchical framework that balances privacy, communication overhead, security, and robustness. Gossip-

based and peer-to-peer learning approaches (Zhu & Jin, 2020; Lu, Zhang, Wang & Mack, 2019) have demonstrated superior performance on real-world health records while allowing parameter exchange without compromising solution optimality. Sequential FL on heterogeneous data (Li & Lyu, 2024) improves convergence in non-IID settings. DFL methods have also been applied to wireless Device-to-Device (D2D) networks, and air computing is projected to surpass traditional decentralized stochastic gradient descent (DSGD) methods in these contexts (Sun, Li & Wang, 2021).

None of the DFLs discussed above incorporates participant nodes as performance-aware aggregators, nor do they consider a post-aggregation model-refinement step that leverages the raw data from the selected aggregator node. These are the gaps addressed by FL-EGM in Chapter 2.

1.3 Adversarial Attacks on Deep Neural Networks

Deep neural networks (DNNs) are highly susceptible to adversarial examples, imperceptible perturbations of the input that cause confident misclassifications (Szegedy *et al.*, 2014; Goodfellow, Shlens & Szegedy, 2015). The Fast Gradient Sign Method (FGSM) (Goodfellow *et al.*, 2015) constructs a perturbation in a single step along the gradient of the loss with respect to the input:

$$x^{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(f_\theta(x), y)), \tag{1.2}$$

where ϵ is the perturbation budget. Projected Gradient Descent (PGD) (Madry *et al.*, 2018) extends FGSM to a multi-step iterative attack that solves:

$$x^{t+1} = \Pi_{\mathcal{B}_\epsilon(x)}\left(x^t + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}(f_\theta(x^t), y))\right), \tag{1.3}$$

where Π denotes projection onto the ℓ_∞ ball of radius ϵ around x. PGD-AT is widely regarded as the gold standard for adversarial evaluation and forms the basis of most FAT methods reviewed below.

Beyond FGSM and PGD, the Fast-FGSM (FFGSM) (Wong, Rice & Kolter, 2020) variant re-randomizes the starting point at each step to avoid catastrophic overfitting during FGSM-based training. The iterative I-FGSM and the Momentum Iterative Method (MIM) (Kurakin, Goodfellow & Bengio, 2018; Dong, Liao *et al.*, 2018) accumulate gradient momentum across steps to improve attack transferability. Carlini & Wagner (C&W) (Carlini, 2017) formulate adversarial perturbation as a constrained optimization problem and are substantially harder to defend against. DeepFool (Moosavi-Dezfooli, Fawzi & Frossard, 2016) computes the minimum perturbation needed to cross the decision boundary. The parameter-free ensemble AutoAttack (AA) (Croce & Hein, 2020), which combines APGD, FAB, and Square attacks, sets a rigorous standard for robustness evaluation and is used in the experimental validation of FedHAT and EARS.

In FL, data and model poisoning attacks represent additional threat vectors (Bhagoji *et al.*, 2019; Bagdasaryan *et al.*, 2020). The distributed nature of FL, non-IID data, and limited visibility into client behaviour amplify the attack surface (Lyu, 2022; Kumar, 2023). Inference-time evasion attacks (the focus of AT) differ from training-time poisoning attacks; both are relevant in practical FL deployments, but this thesis concentrates on evasion robustness.

1.4 Adversarial Training

AT remains the most effective known defence against adversarial examples (Athalye, Carlini & Wagner, 2018). Madry *et al.* (Madry *et al.*, 2018) formalized AT as a min-max robust optimization problem:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta\in\mathcal{B}_\epsilon} \mathcal{L}(f_\theta(x+\delta), y) \right]. \tag{1.4}$$

In practice, the inner maximization is approximated by PGD, yielding PGD-AT. Extensions to PGD-AT include: **TRADES** (Zhang *et al.*, 2019b), which explicitly trades off clean accuracy against robustness via a regularized KL-divergence term, framing the clean–robust tension as a bias variance tradeoff; **ALP** (Kannan, 2018), which incorporates logit-pairing to enforce similarity between clean and adversarial representations; **Max-Margin AT** (et al., 2020),

which maximises the margin to the decision boundary directly; **AVmixup** (Lee, 2020), which interpolates between adversarial and clean examples in feature space using adversarial weight perturbation; and **EAT** (Tramèr, 2017), which trains against multiple surrogate models to achieve ensemble-style robustness, but requires each client to maintain or query surrogates, which is prohibitive in heterogeneous FL environments.

Free AT (Shafahi *et al.*, 2019b) and Fast AT (Wong *et al.*, 2020) reduce the computational overhead of PGD-AT by replaying gradient updates or using a single-step attack with random starts, respectively, at the cost of slightly lower robustness. Curriculum AT (Cai, Du *et al.*, 2018) gradually increases the perturbation budget ϵ during training; Dynamic AT (Wang *et al.*, 2019a) adjusts perturbation strength per-sample based on model confidence. These approaches adapt intensity based on model state rather than device state, a key distinction from the device-aware adaptivity introduced in EARS. The robustness accuracy tension established by Tsipras (2018) and operationalized in TRADES frames the empirical results throughout this thesis.

1.5 Federated Adversarial Training

FAT adapts adversarial robustness techniques to the decentralized setting of FL, where clients generate adversarial examples locally and contribute hardened updates to the global model. Zizzo *et al.* (Zizzo, 2020) showed that naïve local AT followed by `FedAvg` aggregation already confers meaningful robustness without data sharing. However, the cost asymmetry between adversarial and clean training is fundamental in FL: adversarial example generation (especially PGD with many steps) imposes a 3-30× computational overhead relative to standard training (Shafahi *et al.*, 2019b), making uniform AT across all FL clients infeasible when devices have heterogeneous resources.

`FedPGD` (Reisizadeh, 2020) and `FedTRADES` (Zizzo, 2020) apply PGD-AT and TRADES locally and aggregate with `FedAvg`, establishing strong but resource-intensive baselines. `DBFAT` (Zhang & et al., 2023) introduces a dynamic balance between clean and robust objectives across FL rounds. `CalFAT` (Chen, 2022) calibrates local AT to mitigate client drift under non-IID

conditions. `FedRolex` (Alam, Liu, Yan & Zhang, 2022) and `FjORD` (Horvath *et al.*, 2021) tackle model heterogeneity in FL, but do not address AT cost. `FedProphet` (Tang, 2025) partitions large models into memory-feasible modules so that each client trains its assigned module adversarially, targeting memory heterogeneity rather than role specialization, and Federated Robustness Propagation (FRP) (Hong, 2023) shares robustness across heterogeneous clients without introducing a tier-based partition or hybrid attack training. `FedRBN` (Hong, Wang, Wang & Zhou, 2023) keeps separate batch-normalization statistics for clean and adversarial inputs, and `MixFAT` (Zizzo, Rawat, Sinn & Buesser, 2022) mixes adversarial examples of varying strength within each client's batch. A shared assumption across all of these methods is that *every* participating client performs AT in some form: none assign differentiated training roles based on resource capacity, nor do they combine multiple attack types across different clients to achieve ensemble-style robustness at the federation level.

The gap between adversarial robustness and resource efficiency in FAT is the primary motivation for FedHAT (Chapter 3) and EARS (Chapter 4).

1.6 Resource-Aware Client Selection in Federated Learning

Random client selection in vanilla `FedAvg` underperforms when clients differ substantially in compute, energy, and connectivity (Kairouz *et al.*, 2021; Imteaj, 2022). A body of work has developed resource-informed selection strategies to mitigate this. **FedCS** (Nishio, 2019) selects clients based on deadline feasibility, querying their resource profiles before each round to exclude stragglers. **FedMCCS** (AbdulRahman, 2020) predicts round-completion probability from CPU frequency, memory, and remaining energy using a multi-criteria scoring function. **TiFL** (Chai *et al.*, 2020) partitions clients into latency tiers to mitigate the straggler effect while maintaining fairness. **Oort** (Lai *et al.*, 2022) ranks clients using a joint objective that combines statistical utility (data quality) and system utility (expected round completion time), thereby offering a principled trade-off. **FAVOR** (Guo *et al.*, 2021) uses reinforcement learning to select clients that maximize both model quality and participation efficiency.

These methods establish the value of informed selection but do not address AT, where the cost asymmetry between adversarial and standard local training is fundamental. FedHAT adapts resource-aware selection to the FAT setting by assigning distinct training roles (resource-rich clients perform AT; resource-limited clients perform standard training), and EARS extends this further by coupling selection with per-client energy budgets and adaptive training intensity.

1.7 Energy-Aware and Sustainable Federated Learning

Energy efficiency in FL has gained increasing attention as FL deployments migrate to battery-powered edge devices (Li *et al.*, 2020b; Zhao *et al.*, 2018). Several works have characterized the energy cost of FL communication and computation. Yang, Chen, Saad, Hong & Shikh-Bahaei (2020) model device energy consumption as a function of CPU frequency, transmission power, and local dataset size, and optimize client scheduling to minimize total energy expenditure. Zeng, Du, Huang & Leung (2020) propose energy-efficient wireless scheduling for FL, balancing convergence rate against energy cost. Mo & Xu (2021) extend this to heterogeneous networks with non-IID data. More recent systems target device longevity directly: `Eco-FL` (Savoia, Prezioso, Mele & Piccialli, 2024) weights selection probability by remaining battery fraction to extend client lifetimes, `E-Tree` (Yang, Lu, Cao, Huang & Zhang, 2022) uses hierarchical aggregation to reduce communication energy, and joint communication–computation energy has been cast as a constrained optimization problem (Yang *et al.*, 2020). These methods extend participation but operate exclusively within the standard-training regime; they do not account for the substantially higher, attack-configuration-dependent energy cost of AT.

In the context of AT, energy concerns are magnified: PGD-based adversarial example generation can consume 10-30× the energy of standard forward and backward passes due to the iterative attack loop (Shafahi *et al.*, 2019b). To the best of our knowledge, no existing FAT method accounts for device-level energy budgets when scheduling clients or assigning training intensity. Existing works either ignore energy (Zizzo, 2020) or treat it as a static constraint rather than a dynamic budget that evolves across rounds as batteries deplete.

Related ideas appear in anytime prediction (Huang, Chen *et al.*, 2018) and resource-adaptive inference (Laskaridis, Kouris & Lane, 2021; Wang *et al.*, 2019c), where models dynamically adjust depth or width based on device constraints. EARS applies an analogous principle to training: AT intensity becomes the adjustable computational knob governed by device energy budgets, transitioning clients from full PGD to FGSM to standard training as their batteries decline.

1.8 Adaptive Training Intensity and Heterogeneous Device Management

Adaptive AT intensity, varying the strength or type of adversarial perturbation applied during training, has been explored primarily in the centralized setting. Progressive AT (Cai *et al.*, 2018) increases the perturbation budget gradually; Dynamic AT (Wang *et al.*, 2019a) adjusts per-sample intensity based on model confidence. However, neither approach accounts for device-level resource constraints, as they assume a homogeneous computing environment.

In FL, heterogeneous device management is well-studied from a systems perspective. `HeteroFL` (Diao, Ding & Tarokh, 2020) and `FjORD` (Horvath *et al.*, 2021) train sub-models of varying size on resource-limited clients, but do not modify the AT protocol. Model compression and knowledge distillation approaches (Hinton, Vinyals & Dean, 2015; Lin, Kong, Stich & Jaggi, 2020) reduce per-client computation but do not address the fundamental issue of battery depletion under sustained AT.

The EARS framework introduces the *H-AATI* state machine, the first mechanism to explicitly link AT intensity transitions (PGD $\rightarrow$ FGSM $\rightarrow$ standard training) to device energy budgets and battery levels. Stabilization through round-locking and hysteresis prevents oscillation between intensity levels, a failure mode not addressed by any prior adaptive AT method.

1.9 Aggregator Selection and Model Refinement in Federated Learning

Aggregator selection in FL, the choice of which entity performs global model aggregation, has a profound impact on system performance, fairness, and robustness. In CFL, the server is

fixed, and its bias can propagate directly into the global model. Byzantine-robust aggregation rules such as Krum (et al., 2017), Trimmed Mean (Yin, Chen, Kannan & Bartlett, 2018), and Flame (Nguyen *et al.*, 2022) attempt to mitigate the influence of malicious or biased updates, but they still assume a trusted central aggregator.

In DFL, aggregator selection becomes dynamic. Random rotation of the aggregation role among participating nodes, as in the baseline explored by FL-EGM, reduces bias by preventing any single node from dominating the aggregation process. Performance-based aggregator selection, electing the highest-accuracy client as the aggregator for the next round, further aligns the aggregation role with the client most capable of producing a high-quality global model. This idea has parallels in reputation-based trust systems (Kang, Xiong, Niyato, Xie & Zhang, 2019) and committee-based blockchain consensus mechanisms (Kim, Park, Bennis & Kim, 2019), though neither of these directly addresses model quality as the selection criterion.

Post-aggregation model refinement is explored in FL-EGM through the EGM mechanism, in which the elected aggregator fine-tunes the aggregated global model on its own raw data. This is conceptually related to personalized FL methods such as `pFedMe` (Fallah *et al.*, 2020) and MAML-based approaches (Finn, Abbeel & Levine, 2017; Jiang, Konečnỳ, Rush & Kannan, 2019), which adapt the global model to local data distributions. However, EGM differs in that the aggregator node refines its full local dataset, and the refined model is then broadcast as the new global model, rather than being retained as a client-specific personalized model. To the best of our knowledge, no prior DFL system combines performance-based aggregator election with post-aggregation global model refinement on the aggregator's raw data.

1.10 Positioning of the Thesis Contributions

The analysis above highlights three gaps that collectively motivate this thesis:

1. **Biased and static aggregation in DFL.** Prior DFL methods do not use participant performance to guide aggregator selection, nor do they refine the global model using

the aggregator's raw data. FL-EGM (Chapter 2) addresses both limitations through best-performance-based aggregator election and the EGM refinement mechanism.

2. **Uniform AT in resource-heterogeneous FL.** All existing FAT methods apply the same training protocol to every participating client, ignoring resource heterogeneity. FEDHAT (Chapter 3) introduces tier-based role assignment and hybrid attack diversification to achieve ensemble-style robustness without per-client multi-attack overhead.
3. **Battery depletion under sustained FAT.** No existing FAT method accounts for the progressive energy depletion of battery-powered edge devices across FL rounds. EARS (Chapter 4) introduces the H-AATI state machine and an energy-aware client selector to achieve 100% client survival while maintaining competitive adversarial robustness.

CHAPTER 2

DECENTRALIZED FEDERATED LEARNING WITH AGGREGATOR SELECTION AND ENHANCED GLOBAL MODEL GENERATION

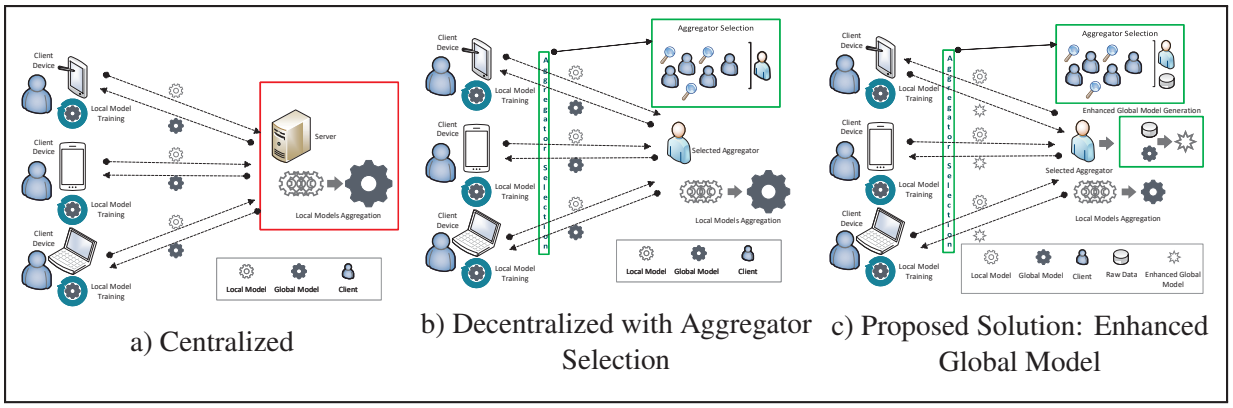
This chapter presents FL-EGM, the DFL framework that constitutes the first contribution of this thesis. As motivated in the Introduction and surveyed in Chapter 1, centralized aggregation is the structural weak point of standard FL: routing every client update through a single fixed server creates a communication bottleneck, a single point of failure, and a source of bias, since a server that favours particular participants or holds its own data can silently skew the global model. Decentralizing the aggregation role is the natural remedy, but cycling through clients in a fixed order or selecting them uniformly at random discards the quality that careful aggregation provides. FL-EGM resolves this tension by electing, in every round, the best-performing participant as the aggregator and then refining the aggregated model on that node's own raw data through an EGM step, turning the aggregator's privileged data access from a liability into an asset. The remainder of this chapter formalizes the centralized and random-aggregator baselines, details the proposed best-aggregator selection and EGM refinement mechanisms, and evaluates them across multiple datasets, federation sizes, and network conditions, including under biased-aggregator scenarios.

2.1 Proposed Solutions

This section introduces system models and their structures, using a centralized FL architecture as a baseline. A random aggregator selection component is introduced to decentralize the architecture. The final solution includes an EGM mechanism and best aggregator selection to improve performance using aggregator data.

2.1.1 Baseline Federated Learning

This section outlines the foundational architecture of FL, followed by adaptations for the proposed solution. The FL incorporates a CNN model and the Federated Averaging (FedAvg) (McMahan *et al.*, 2017) algorithm for local model aggregation. Figure 2.1a describes the



a) Centralized

b) Decentralized with Aggregator Selection

c) Proposed Solution: Enhanced Global Model

Figure 2.1 Comparison of Architectures

baseline architecture with centralized FL. The FL model comprises three main entities: clients, a central server, and a communication network. Note that our proposed solution will replace the component in red (the centralized aggregator) (see Section 2.1.1.1).

Client

In the joint model training process, each client is a corporate agent. These agents are tasked with developing and refining a robust local model by regularly communicating with the central server; they actively contribute to preserving the parameters of the global model.

Central Server

The central server's primary objective is to gather local model parameters and create a unified global model, involving multiple rounds of interaction with clients to improve and refine the model, incorporating input from various clients.

Network Communication

The system's primary function is to enable the transfer of parameters between clients and the central server. In the broader system model, the essential procedure of a round of FL can be outlined as follows:

- Model Initialization: A centralized server creates a global model and initializes parameters, which clients then download to generate local models during the initial communication round, ensuring smooth and efficient model generation.
- Local Model Training and Updates: Clients train the initial model using their data and update local model parameters to the central server after obtaining it.
- Local Models Aggregation: The central server aggregates client updates to create a global model, which is distributed to all clients for the next round. This iterative process continues until the number of communication rounds has been reached.

2.1.1.1 Baseline Random Aggregator Selection

A central aggregator can influence the system's performance through biased model aggregation. In this section, we introduce a random aggregation process to address this issue, in which a random aggregator is selected from among the participating clients to perform the aggregation task. The system diagram in Figure 2.1b illustrates the overall setup. A new aggregator is selected to perform the aggregation task and train the network in each process round. This unique approach enables the model to eliminate reliance on a central aggregator, promoting the generation of a more trustworthy global model. The system's complete workflow is outlined as follows:

Aggregator Selection

The aggregator selection in each round will be done randomly from the participant nodes. It should be noted that when a node is chosen as the aggregator, it will actively participate in the training round. It is essential to highlight that every node has an equal chance of becoming the aggregator and carrying out the aggregation process.

Model Initialization

A selected aggregator is assigned to construct a comprehensive global model while initializing its parameters. Subsequently, the individual clients in the system acquire their respective local models by retrieving the initial model from the predetermined aggregator. It is imperative to note that this downloading and model generation process exclusively takes place during the initial round of communication, ensuring the efficiency and effectiveness of subsequent interactions.

Local Training and Updates

After obtaining the initial model, the clients train it using their datasets and synchronize the local model parameters with the selected aggregator.

Local Models Aggregation

The selected aggregator within this system assumes a pivotal role by effectively aggregating updates received from individual clients and amalgamating them to form an updated global model. Subsequently, this refreshed global model is disseminated to all participating clients, signifying the initiation of the subsequent round. The iterative nature of this process persists until the predefined threshold of communication rounds has been attained.

Assume we have a global model G from C clients with data D_c where $c \in 1, 2, \ldots, C$, without data sharing. Let $f_c(G)$ is the objective function for client c, which we want to minimize:

$$\min_{\theta} F(G) = \frac{1}{C} \sum_{c=1}^{C} \frac{m_c}{M} \sum_{(a,b) \in D_c} f_c(G; a, b), \tag{2.1}$$

where m_c presents the number of local data points for client c and M is the total number of data points across all clients. The objective function $f_c(G; a, b)$ typically takes the form of a loss function applied to the model parameters G and a single data point (a, b). To update the global model, each client c computes an update ΔG_c by minimizing their local objective

function $f_c(G)$:

$$\Delta G_c = \underset{\Delta G}{\operatorname{argmin}} f_c(G + \Delta G) \tag{2.2}$$

The selected aggregator aggregates the updates from all clients using a weighted average:

$$\Delta G = \sum_{c=1}^{C} \frac{m_c}{M,} \Delta G_c \tag{2.3}$$

Finally, the aggregator updates the global model by applying the aggregated update:

$$G \leftarrow G - \eta \Delta G \tag{2.4}$$

where η is the learning rate. The FL algorithm involves iterative communication between the selected aggregator and the clients. The clients perform local updates and send them back to the selected aggregator, which aggregates all local updates from the clients and updates the global model accordingly.

2.1.1.2 Proposed Random and Best Selection of Aggregator with Enhanced Global Model

The baseline solution reduces bias by decentralizing the aggregator but still struggles with global model performance and slow convergence (See Section 2.2). To address this, we introduce the EGM mechanism. In this design, the aggregator selected either randomly or based on performance, does not train but collects client updates and performs aggregation. The global model is trained using the selected aggregator's raw data, incorporating additional features to refine its performance. This process generates an improved version of the global model, which is visually presented in Figure 2.1c. Each training round, a new aggregator is selected, and an EGM is generated. This approach demonstrates the ability to converge swiftly while achieving higher levels of accuracy.

Aggregator Selection

We propose two methods for selecting the aggregator in each FL round: random selection and best performance-based selection. Both methods are using different aggregator selection criteria, as described below.

- Random Selection: The aggregator selection for each round will be random from participant nodes, with each node having an equal opportunity to be chosen and perform the aggregation process. It is important to note that when a node is chosen as the aggregator, it will abstain from the training round.
- Best Performance-based Selection: The aggregator for each round is chosen from the set of participant nodes N, with the best-performing node being chosen for each round t. The node selected as the aggregator will not participate in the training round. The node with the highest accuracy in the previous round will be the aggregator in the next round.

After selecting the aggregator, both methods use the same architecture, as described below.

Initialization

Let A be the selected aggregator node. A generates a global model $G(t)$ and distributes it among the clients. Each client in N, except the selected aggregator A, acquires their respective local models $M(i,t)$, where $i \in N \setminus \{A\}$.

$$A = \arg\max_{i \in N} \text{Accuracy}(i,t) \tag{2.5}$$

Training and Updates

Once the clients have obtained the initial model, they train it using their respective datasets, except of the selected aggregator. After the training process, they synchronize the parameters of their local models with the selected aggregator.

Aggregation

The selected aggregator plays a crucial role in this system. The selected aggregator A gathers updates from individual clients and aggregates them to create an updated global model $G(t+1)$.

$$G(t+1) = \text{aggregate}(M(i,t), \text{for } i \in N \text{ and } i \neq A) \tag{2.6}$$

Aggregator Training and Enhance Global Model Generation

A refinement process is introduced to improve the performance and accuracy of the global model $G(t+1)$. The selected aggregator A fine-tunes the global model $G(t+1)$ using its raw data. By leveraging the diverse characteristics present in the raw data of the selected aggregator, an EGM $G'(t+1)$ is achieved with higher accuracy and performance.

$$G'(t+1) = \text{fine-tune}(G(t+1), \text{raw data}(A)) \tag{2.7}$$

The EGM $G'(t+1)$ is distributed among all the clients for the next round. This process continues until the required criteria are met.

$$\text{For each } i \in N, M(i,t+1) = G'(t+1) \tag{2.8}$$

2.2 Results and Discussions

In this section, we conducted experiments to evaluate the practicality of the proposed scheme. Initially, we present comprehensive information about the experiment settings, including descriptions of the data resources, environment configurations, and performance metrics. Then, we provide a performance comparison of our proposed solution with the baseline methods we described and from the literature. Finally, we demonstrate the performance of our solution compared to baselines under the presence of a biased aggregator.

2.2.1 Experimental Setup and Datasets

This work was implemented using Google Kaggle [1]. Python was the primary language, utilizing libraries like Numpy, pandas, PyTorch, Keras, and TensorFlow. Most FL models have 10 clients, trained for 20 epochs with batch sizes 32 for training and 2048 for testing. The loss is calculated using categorical cross-entropy, with the Adam optimizer at a learning rate 0.001. We used a CNN for all experiments. The CNN, comprising one convolutional layer and two fully connected layers followed by one pooling layer, rapidly extracts features from traffic data to identify potential attacks.

We utilized two datasets to measure the efficacy of our proposed method. **(i) MNIST** dataset (LeCun *et al.*, 1998), which consists of 60,000 images of handwritten digits from 0 to 9. **(ii) CSE-CIC-IDS2018** [2] dataset to analyze our proposed model. The CSE-CIC-IDS2018 dataset was constructed using the Amazon Web Services infrastructure, which is well-known for representing modern network traffic patterns. We worked with six CSV files containing data from different days: 02-14-2018, 02-15-2018, 02-21-2018, 02-22-2018, 02-23-2018, and 03-02-2018. These files encompassed various attack types, including Benign, DDOS attack-HOIC, Bot, FTP-BruteForce, SSH-Bruteforce, DoS Attacks-GoldenEye, DoS attacks-Slowloris, DDOS attack-LOIC-UDP, Brute Force-Web, Brute Force-XSS, and SQL Injection attacks. The dataset consisted of 78 features as columns and 6,291,450 rows representing instances of network traffic data.

Moreover, this dataset is subject to pre-processing for enabling model training.

First, we merge all the CSV files into a single dataset, followed by the removal of instances containing null values.

---

[1] https://www.kaggle.com/

[2] https://www.unb.ca/cic/datasets/ids-2018.html

Then, split the dataset into an 80% training set and a 20% test set and generated separate files for the test and train datasets. Finally, we divide these train and test files among the clients for further processing.

2.2.1.1 Evaluation Measures

To evaluate our methods' efficacy, we employed the following measure (Powers, 2020); (i) **Accuracy** offers a general sense of how well the model is performing. (ii) **Precision** is essential when the cost of false positives is high. (iii) **Recall** is necessary when the cost of false negatives is high. (iv) **F1 Score** is a good metric for an uneven class distribution (class imbalance).

2.2.1.2 Results and Baseline Comparisons without a Biased Aggregator

This section compares various methods, beginning with the conventional FL approach using the FedAvg aggregation algorithm as presented in (McMahan *et al.*, 2017). To evaluate its performance, experiments were conducted under different scenarios (i.e., C = 10, C = 15, C = 20); here, C is the number of clients or participant nodes. Additionally, we performed experiments involving FL with randomly selected aggregators. We also explore our proposed EGM approach, using either random selection or the best-performing node as the aggregator. Finally, the proposed FL models are compared with some of the latest work, including *Zhang et al.* (Zhang, Zhang, Guo, Yao & Li, 2022), *Ideissi et al.* (Idrissi *et al.*, 2023), *Friha et al.* (Friha *et al.*, 2022), and *Li et al.* (Li, Tong, Liu & Cheng, 2023a).

MNIST Dataset: We conducted experiments on the benchmark MNIST dataset to evaluate the performance of the proposed frameworks. The experiments were run for 30 training rounds in the presence of a non-biased aggregator. *FL (Base)* achieved 97.10% accuracy, and *FL (Random)* eached a 96.21% accuracy level. When the experiments are run with EGM settings, *FL-EGM (Random)* achieves 96.97% while *FL-EGM (Best)* gives some surprising results, as it outperforms all the FL solutions with 97.25% accuracy, as presented in Table 2.1.

Table 2.1 Baseline and proposed model accuracy comparison on the MNIST dataset

Models	Acc.(%)	Models	Acc.(%)
FL (Base)	97.10	*FL-EGM (Random)*	96.97
FL (Random)	96.21	*FL-EGM (Best)*	97.25

CSE-CIC-IDS2018 Dataset: Various experiments were conducted with different proposed and baseline solutions.

Centralized Learning: In this baseline, the entire training set is used. The training comprises 20 rounds, after which performance tests are conducted on the test set. The *CNN (Base)* effectively achieves high accuracy, over 99%, with high precision, recall, and F1 scores. The detection performance of the centralized system as a whole is depicted in Figure 2.2a, 2.2b and Table 2.2, where the accuracy and loss are presented. Despite the centralized system's satisfactory overall detection performance, the simulated scenario involves combining data samples from multiple clients, which poses a risk of privacy leakage.

Centralized Federated Learning: We conducted experiments considering experimental parameters provided in 2.2.1 on the *FL (Base)* system using the CIC-IDS2018 dataset, utilizing data normalization and constrained client participation with non-IID data to validate the algorithm and minimize its impact on the global model. The FL approach shown in Figure 2.2c and 2.2d achieves high accuracy rates in the initial round of training, but shows more significant fluctuations compared to alternative approaches. These fluctuations suggest that the aggregation phase of the FL process is influenced by specific local models, resulting in continuous variations in accuracy. The FL scheme showed notable performance compared to the centralized scheme as (shown in Table 2.2), achieving an impressive overall detection metric of 98.45% accuracy, 98.48% precision, 98.12% recall, and 97.24% f1 score. However, the *FL (Base)* scheme experienced a slight decline in attack detection performance.

Federated Learning with random selection of aggregator: The FL scheme is also tested using a random selection of aggregators and compared with baseline studies. A client was randomly

selected to serve as the aggregator for the aggregation task and participated in the training process. The selected aggregator generated a global model, which was then distributed among all clients for the next training round. The results in Figure 2.2e, 2.2f and Table 2.2 showed that the random selection of aggregators resulted in lower performance than the centralized and baseline FL schemes. The *FL (Random)* scheme had an overall detection metric of 96.60%, 96.85%, 96.34%, and 95.74% in terms of accuracy, precision, recall, and f1 score, respectively. However, it experienced a slight decline of approximately 1 to 2% in its detection performance of attacks.

Proposed Federated Learning with Enhanced Global model (random aggregator selection): We conducted experiments on the baseline studies and *FL-EGM (Random)* scheme, utilizing the same environment and settings as FL. Our approach involved randomly selecting a client from the participant clients to serve as the aggregator for the aggregation task. This selected aggregator did not participate in the initial phase of training. Once all the other clients were trained, the selected aggregator aggregated their models to generate a global model. This global model was then further trained using the raw data from the selected aggregator, resulting in an improved global model for the next round. Next, we compared the experimental results depicted in Figure 2.2g, 2.2h and Table 2.2. *FL-EGM (Random)* outperformed other FL solutions, achieving an accuracy of 98.43%, precision, 98.48% recall 98.27% and f1 score 97.87%. The trends in our scheme's accuracy were smoother, and it consistently and quickly reached high levels of detection accuracy and minimized the model's loss.

Proposed Federated Learning with Enhanced Global model (best performance aggregator selection): We conducted experiments comparing the *FL-EGM (Best)* scheme and baseline studies. We ensured a fair comparison using the same environment and settings as *FL (Base)*. To perform the aggregation task, we selected the best-performing client among the participant clients to act as the aggregator. However, this selected aggregator did not participate in the initial training phase. Once the other clients completed their training, the selected aggregator aggregated their models to create a global model. To improve this global model for the next round, we trained it using the raw data from the selected aggregator. We then analyzed the

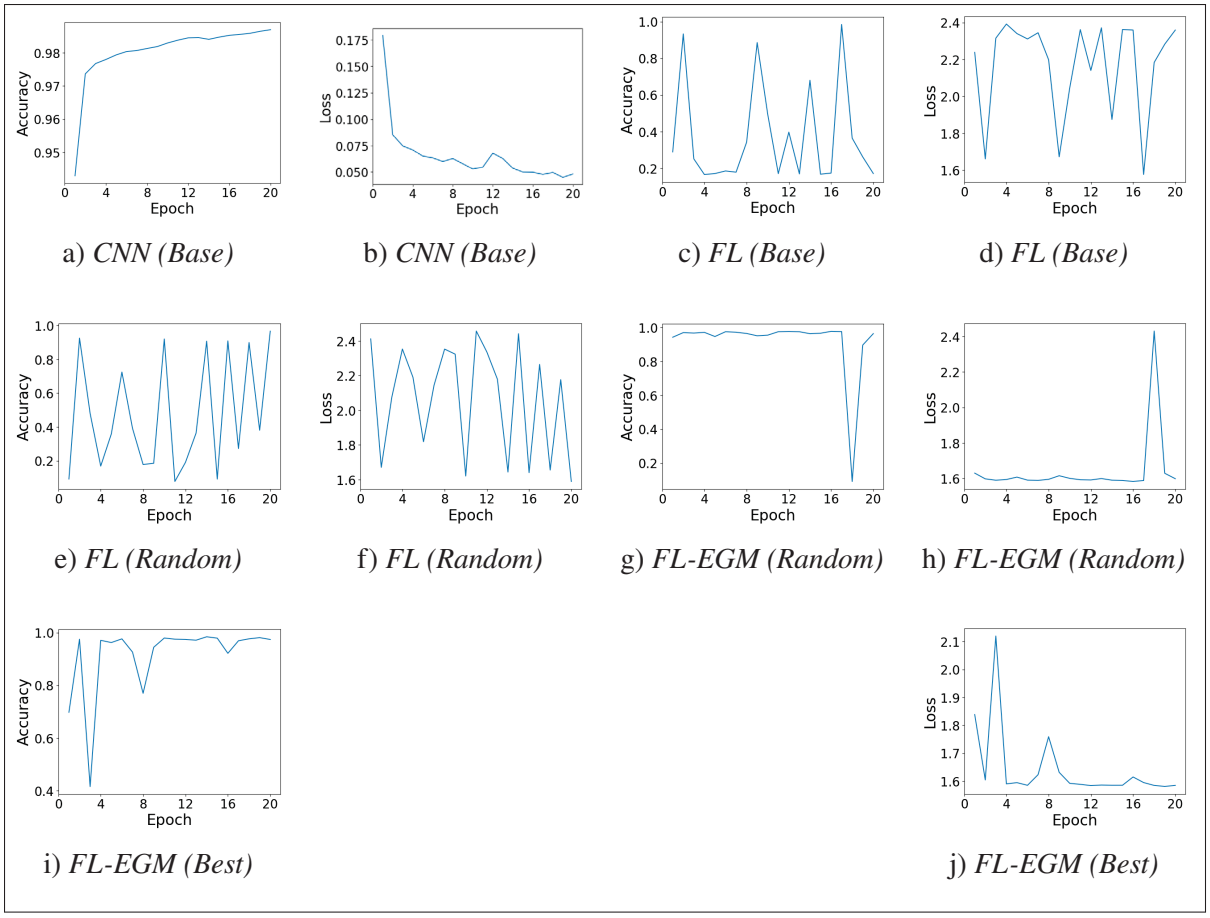


Figure 2.2 Accuracy and Loss Comparison of Presented Models

experimental results presented in Figure 2.2i and 2.2j show explicitly the trend of detection accuracy with *FL-EGM (Best)* settings, which consistently increases. At the same time, loss decreases as the number of communication rounds progresses. *FL-EGM (Best)* achieved top accuracy of 98.54% 98.58%, 98.41%, and 98.03% in terms of accuracy, precision, recall and f1 score, respectively. Meaning that it correctly classified the majority class with an accuracy of 98.54%. Additionally, our scheme outperformed other FL solutions, achieving a top accuracy of 98.54%, higher than the other FL schemes and comparable with the centralized approach. Table 2.2 summarizes the presented solutions' comparison.

Training time comparison: This section compares detection accuracy and training time between *CNN (Base)* and FL-based solutions as presented in Table 2.2. *CNN (Base)* completes per epoch training between 217 to 251 seconds, while FL solutions distribute data between clients or

Table 2.2 Accuracy and Time Comparison in the Presence of non-Biased Aggregation

Models	Time(Seconds)	Acc.(%)	Precision	Recall	F1-Score
CNN (Base)	217-251s per Epoch	99.33	99.46	98.71	98.96
FL (Base)	28-31s per Round	98.45	98.48	98.12	97.24
FL (Random)	29-31s per Round	96.60	96.85	96.34	95.74
FL-EGM (Random)	58-62s per Round	98.43	98.48	98.27	97.87
FL-EGM (Best)	52-60s per Round	98.54	98.58	98.41	98.03

nodes, with the initial model in the central location. Training time depends on factors like the amount of data, number of clients, and client resources. Using the same data set size in Central and FL settings, *FL (Base)* achieved the highest accuracy in 28 to 31 seconds per round, while *FL (Random)* achieved 29 to 31 seconds per round. *FL-EGM (Random)* and *FL-EGM (Best)* demonstrated efficiency in accuracy and model stability, but took more training time of 58 to 62 and 52 to 60 seconds per round, respectively. Both models exhibited the highest accuracy and model stability due to their use of raw data from the selected aggregator and took more training time to generate an EGM.

Effect of Participant Nodes: In this section, we evaluate the robustness of the proposed model as the number of participants or clients varies. We tested the baseline and the proposed model under different scenarios with participant nodes set to (i.e., C = 10, C = 15, C = 20). The results are shown in Table 2.3. We observed that *FL (Base)* and *FL (Random)* experience a significant decrease in accuracy when the number of participant nodes increases, losing almost 38% of accuracy when the participant nodes are doubled. In contrast, *FL-EGM (Random)* and *FL-EGM (Best)* show a decrease in accuracy of less than 1% under the same conditions. Results show that *FL-EGM (Random)* and *FL-EGM (Best)* taking advantage of EGM show more robustness when the number of participants increases or decreases. We will continue our experiments with only 10 clients in the upcoming sections.

Evaluation with Baseline: In this section, we compare our proposed model with the baseline studies (Zhang *et al.*, 2022), (Idrissi *et al.*, 2023), (Friha *et al.*, 2022), and (Li *et al.*, 2023a). The proposed model details are presented in section 2.1. Considering 10 number of clients,

Table 2.3 Effect of Participant Nodes

	Time (Seconds) per Round			Accuracy (%)		
No. of Clients	20	15	10	20	15	10
FL (Base)	13-15	18-20	28-31	59.68	87.05	98.45
FL (Random)	15-16	19-21	29-31	58.94	85.44	96.60
FL-EGM (Random)	30-32	38-41	58-62	97.78	97.92	98.43
FL-EGM (Best)	28-30	35-37	52-60	97.87	98.10	98.54

our *FL (Base)* archives overall accuracy of 98.45%, *FL (Random)* archives 96.60%, *FL-EGM (Random)* achieves 98.43% while *FL-EGM (Best)* achieves 98.54%, which is higher than between all the presented models. Compared with the already proposed studies, SecFedNIDS (Zhang *et al.*, 2022) archives 97.03% overall accuracy, Fed-ANID (Idrissi *et al.*, 2023) archives 94.48%, FELIDS (Friha *et al.*, 2022) archives 94.09% and DAFL (Li *et al.*, 2023a) archives 94.64% accuracy. our proposed model with EGM settings outperforms all the baseline solutions. Detailed results are presented in Table 2.4.

Table 2.4 Accuracy Comparison of Proposed and Baseline Solutions

Models	Acc.(%)	Models	Acc.(%)
SecFedNIDS (Zhang *et al.*, 2022)	97.03	*FL (Base)*	98.45
Fed-ANID (Idrissi *et al.*, 2023)	94.48	*FL (Random)*	96.60
FELIDS (Friha *et al.*, 2022)	94.09	*FL-EGM (Random)*	98.43
DAFL (Li *et al.*, 2023a)	94.64	*FL-EGM (Best)*	98.54

Evaluation with Early Stopping: The experiments are conducted to evaluate the convergence analysis of all the presented models using the early stopping criterion method provided in Keras[3]. According to this criterion, training stops if the validation loss does not decrease for ten consecutive rounds. The experiments run for a total of 50 rounds. In our results, *FL (Base)* takes 649 seconds to achieve an accuracy of 98.38% after 22 training rounds. *FL (Random)* reaches 97.84% accuracy in 630 seconds after 21 training rounds. Additionally, we expand the experiments with EGM settings. *FL-EGM (Random)* achieves a higher accuracy of 98.43%, surpassing both *FL (Base)* and *FL-EGM (Random)*, but it takes 780 seconds to train. In contrast,

[3] https://keras.io/api/callbacks/early_stopping/

FL-EGM (Best) converges quickly, achieving an accuracy of 98.54% in just 504 seconds and only 9 training rounds, outperforming all other FL solutions. Table 2.5 presents a summary of the results.

Table 2.5 Early Stopping Results

Models		*FL (Base)*	*FL (Random)*	*FL-EGM (Random)*	*FL-EGM (Best)*
Training Time per Round (Seconds)		28-31	29-31	58-62	52-60
Without 10 rounds of early stopping	Total Time (Seconds)	649	630	780	504
	Training Rounds	22	21	13	9
Including 10 rounds of early stopping	Total Time (Seconds)	944	930	1380	1064
	Training Rounds	32	31	23	19
Accuracy (%)		98.38	97.84	98.43	98.54

2.2.1.3 Evaluation in the Presence of Biased Aggregator

In this subsection, we evaluate the robustness of *FL-EGM (Random)* and *FL-EGM (Best)* under the presence of biased aggregation and compare it to the baseline methods, *FL (Base)* and *FL (Random)*. This section will refer to the biased version of these four methods as *FL-EGM-R (Biased)*, *FL-EGM-B (Biased)*, *FL (Biased)*, and *FLR (Biased)*, respectively. In this scenario, we introduce an aggregator that always excludes updates from a specific client. This represents an aggregator who is biased and ignores a client for arbitrary reasons, which is not the intended behaviour in FL. In the case of *FL (Biased)*, the biased aggregator is a centralized component aggregating every round while ignoring one client. In the case of *FLR (Biased)*, *FL-EGM-R (Biased)*, and *FL-EGM-B (Biased)*, the biased aggregator is also acting as an honest client for training purposes, and it will only occasionally be selected for aggregation, where it will perform its role in a biased way. **Evaluation of Biased Aggregator on MNIST Dataset:** We conducted experiments on the benchmark MNIST (LeCun *et al.*, 1998) dataset to evaluate the performance of the proposed frameworks. The experiments ran for 30 training rounds in the presence of a biased aggregator. Table 2.6 *FL (Biased)* achieved an accuracy of 95.77%, while *FLR (Biased)* surpassed the baseline with an accuracy of 96.19%. Under the EGM settings, *FL-EGM-R (Biased)* achieved 96.90%, outperforming both *FL (Biased)* and *FLR (Biased)*. Notably, *FL-EGM-B (Biased)* delivered remarkable results, surpassing all other FL solutions

with an accuracy of 97.22%. These results demonstrate that the proposed solutions maintain high accuracy even with a biased aggregator.

Table 2.6 Accuracy Comparison with MNIST (LeCun *et al.*, 1998) Dataset

Models	Acc.(%)	Models	Acc.(%)
FL (Base)	95.77	*FL-EGM (Random)*	96.90
FL (Random)	96.19	*FL-EGM (Best)*	97.22

Evaluation of Biased Aggregator on CSE-CIC-IDS2018 Dataset: *FL (Biased)* achieves an overall accuracy of 94.44%, which is approximately 4% less than *FL (Base)* in the presence of a biased aggregator. *FLR (Biased)*, where the aggregator is selected randomly from the clients in the presence of a biased aggregator. It has 95.23% attack detection accuracy, which is approximately 1% higher than *FLR (Biased)* and only 1% lower than *FL (Random)*. It shows the performance of *FLR (Biased)* compared to *FL (Random)* in the presence of a biased aggregator without significant accuracy loss. *FL-EGM-R (Biased)*, where the aggregator is selected randomly and takes advantage of EGM in the presence of a biased aggregator. The accuracy of *FL-EGM-R (Biased)* shows more stability compared with *FL (Biased)* and *FLR (Biased)* with an overall accuracy of 98.17%. *FL-EGM-R (Biased)* roughly achieves more than 3% and 4% accuracy from *FLR (Biased)* and *FL (Biased)* respectively while approximately similar to *FL-EGM (Random)* that shows the robustness of *FL-EGM-R (Biased)* against the biased aggregator.

Finally, *FL-EGM-B (Biased)* is compared with *FL (Biased)*, *FLR (Biased)* and *FL-EGM-R (Biased)*. *FL-EGM-B (Biased)* have 98.43% accuracy, which is higher than all the presented models in biased aggregator's settings and approximately equal to *FL-EGM (Best)*. The results demonstrated the robustness of proposed models with aggregator selection and EGM in the presence of a biased aggregator. Table 2.7 shows the detailed comparison of time and accuracy of all the presented models facing the biased aggregator. *FL (Biased)* achieves 94.22% accuracy in 26 to 29 seconds per round, while *FL (Base)* achieves 98.45% accuracy in 28 to 31 seconds without facing a biased aggregator. *FLR (Biased)* shows more robustness against the biased

Table 2.7 Accuracy and Time Comparison in the Presence of Biased Aggregation

Models	Time(Seconds)	Acc.(%)
FL (Base)	26-29s per Round	94.22
FL (Random)	24-30s per Round	95.23
FL-EGM (Random)	58-62s per Round	98.17
FL-EGM (Best)	52-60s per Round	98.43

aggregator by achieving 95.23% accuracy in 24 to 30 seconds per round in compression with *FL (Random)*, which achieves 96.60% accuracy in 28 to 31 seconds with a difference of approximately 1%. *FL-EGM-R (Biased)* losing less than half percent of accuracy in compression with *FL-EGM (Random)* in the same time frame of 58 to 62 seconds per round in the presence of a biased aggregator. *FL-EGM-B (Biased)* is highly comparable with *FL-EGM (Best)* in terms of accuracy and time with biased aggregators involvement.

The results show that *FLR (Biased)*, *FL-EGM-R (Biased)* and *FL-EGM-B (Biased)* are more robust in the presence of a biased aggregator, but *FL-EGM-R (Biased)* and *FL-EGM-B (Biased)* take almost double the time compared with *FL (Biased)* and *FLR (Biased)*. Because *FL-EGM-R (Biased)* and *FL-EGM-B (Biased)* take advantage of EGM to improve accuracy and model stability. There is a trade-off between the time spent training each round and the balance between accuracy, model stability, and robustness against biased aggregation.

Early Stopping with Biased Aggregator: The experiments assess the convergence of all presented models using early stopping criteria, where training halts if the validation loss fails to improve for ten consecutive rounds. The experiments are conducted over 50 rounds with a biased aggregator present. Our findings show that *FL (Biased)* requires 649 seconds to reach an accuracy of 92.57% after 22 training rounds. Meanwhile, *FLR (Biased)* achieves 96.67% accuracy in 870 seconds after 29 training rounds. Expanding our experiments to include EGM settings, *FL-EGM-R (Biased)* attains a higher accuracy of 98.17%, exceeding both *FL (Biased)* and *FLR (Biased)*, but it takes 720 seconds to complete training. In contrast, *FL-EGM-B (Biased)* converges rapidly, achieving an accuracy of 98.43% in 448 seconds and only 8 training

rounds, outperforming all other FL approaches. A summary of the results is also presented in Table 2.8.

Table 2.8 Early Stopping with Biased Aggregator

Models		*FL (Base)*	*FL (Random)*	*FL-EGM (Random)*	*FL-EGM (Best)*
Training Time per Round (Seconds)		28-31	29-31	58-62	52-60
Without 10 rounds of early stopping	Total Time	649	870	720	448
	Training Rounds	22	29	12	8
Including 10 rounds of early stopping	Total Time	944	1170	1338	1008
	Training Rounds	32	39	22	18
Accuracy (%)		92.57	96.67	98.17	98.43

2.3 Conclusion

In this chapter, we introduced a novel FL-EGM model designed to address critical challenges, such as mitigating the influence of single and biased aggregators while achieving high-end accuracy. We provided a comprehensive design overview of FL-EGM, emphasizing its key features. Our experimental results demonstrated that FL-EGM not only achieves fast convergence but also attains superior levels of accuracy, precision, recall, and F1 score. Notably, the detection performance of FL-EGM surpasses that of other recent FL solutions and closely approaches the performance of centralized methods. Additionally, FL-EGM exhibits robust performance, maintaining accuracy despite fluctuations in the number of participant nodes or the presence of biased aggregators, where traditional methods typically falter. Early stopping further enhances convergence speed, irrespective of aggregator bias. Looking forward, our future research will focus on applying FL-EGM for attack prevention. By continually innovating and refining our model, we aim to advance the field of network intrusion detection and enhance security measures for data-sensitive domains.

Acknowledgement: Work in this chapter was supported in part by funding from the Innovation for Defence Excellence and Security (IDEaS) program from the Department of National Defence (DND) Canada.

CHAPTER 3

RESOURCE-AWARE HYBRID FEDERATED ADVERSARIAL TRAINING FOR GENERALIZED ROBUSTNESS

This chapter presents FedHAT, the resource-aware hybrid FAT framework that constitutes the second contribution of this thesis. As established in the Introduction and Chapter 1, FAT is the natural route to hardening a federated model against evasion attacks, but it inherits the multiplicative cost of AT and imposes it on every participating client. Existing FAT methods refine *how* this training is performed, via per-batch sample mixing, logit calibration, decision-boundary reweighting, or model partitioning, yet they uniformly assume that every client can afford to generate adversarial examples. In heterogeneous edge populations, this assumption is untenable: excluding the clients that cannot would discard their data, which is frequently the most distributionally diverse in the federation. A second orthogonal weakness is that single-attack AT overfits the attack it was trained on and generalizes poorly to unseen attacks. FedHAT resolves both limitations through per-client role specialization: RR clients perform Hybrid Adversarial Training (HFAT) with attacks diversified across the tier, while LR clients train on clean data only, supplying the distributional diversity the adversarial tier cannot provide; the two signals are fused by group-weighted aggregation followed by EGM refinement. The remainder of this chapter formalizes the resource-aware client partition, the hybrid attack scheme, and the federation-level hybrid loss, and validates FedHAT across multiple datasets, attack types, and IID and non-IID partitions.

3.1 Proposed Solution

FedHAT jointly addresses two coupled challenges in FAT: clients differ substantially in computational capacity, making uniform AT infeasible; and single-attack AT generalizes poorly to unseen attacks (Tramèr, 2017). We resolve both through resource-aware specialization. Clients with sufficient compute form the RR tier and perform Hybrid AT (HFAT); clients with limited compute form the LR tier and perform ST. The two tiers contribute complementary signals:

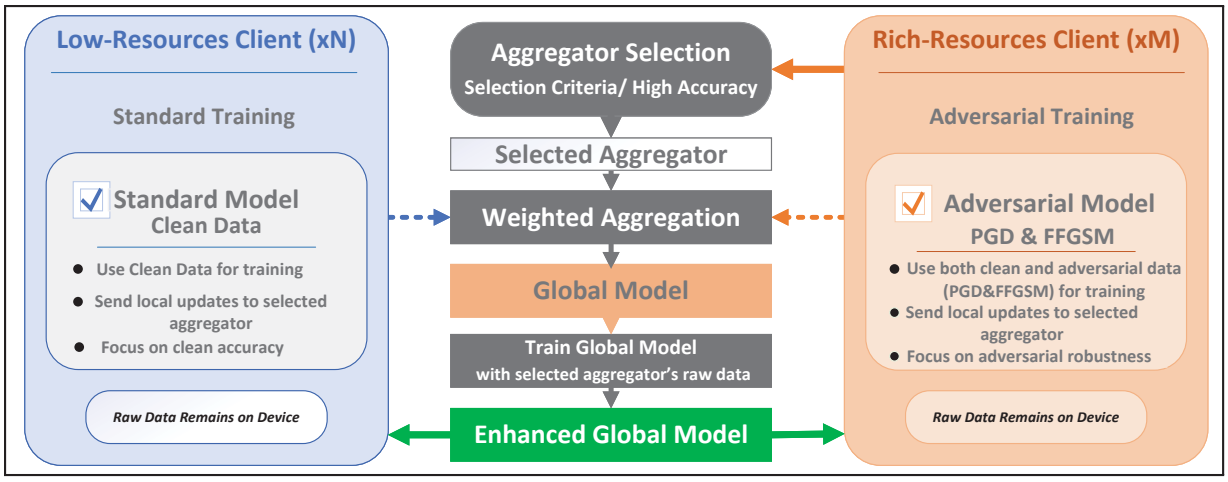


Figure 3.1 Architecture of FedHAT: Resource-Aware Hybrid FAT with Enhanced Global Model

RR clients provide adversarial robustness, while LR clients, who often hold distributionally diverse, task-critical data, provide clean-data generalization.

After local training, the RR client with the highest validation accuracy is dynamically elected as the round's aggregator. It does not participate in local training that round; instead, it aggregates client updates via FedAvg (McMahan *et al.*, 2017) and produces an EGM by performing a refinement step on *its own local data* as shown in Figure 3.1. No client transmits raw data to any other client. The EGM mechanism is inherited from FL-EGM (Chapter 2) and assumes a trusted-aggregator setting consistent with cross-silo and edge-gateway federated deployments (Kairouz *et al.*, 2021).

3.1.1 Resource-Aware Client Selection

Following the ranking-based selection philosophy in heterogeneous FL (Lai *et al.*, 2022; AbdulRahman, 2020; Chai *et al.*, 2020; Nishio, 2019), FedHAT adapts this approach to FAT, where the cost asymmetry between adversarial and standard local computation is fundamental rather than incidental.

3.1.1.0.1 Resource vector and metric definitions

Each client c_i is characterized by a resource vector

$$\mathbf{r}_i = (b_i,\ cpu_i,\ bw_i,\ gpu_i), \tag{3.1}$$

where $b_i \in [0, 1]$ is the fraction of remaining battery charge; $cpu_i, gpu_i \in [0, 100]$ are the available CPU and GPU capacities, computed as $100 \times (1 - \bar{u})$ over a short profiling window where $\bar{u}$ is mean utilization; and bw_i is uplink bandwidth in Mbps measured by lightweight active probing. Devices without a discrete GPU are assigned $gpu_i = 0$. These definitions follow established mobile and IoT FL profiling conventions (Nishio, 2019; Imteaj, 2022; AbdulRahman, 2020).

3.1.1.0.2 Weighted resource score

For each client, we compute a normalized weighted score:

$$s_i = w_b\, b_i + w_{cpu}\, \frac{cpu_i}{100} + w_{bw}\, \frac{bw_i}{bw_{\max}} + w_{gpu}\, \frac{gpu_i}{100}, \tag{3.2}$$

with $w_b + w_{cpu} + w_{bw} + w_{gpu} = 1$. We use $(w_b, w_{cpu}, w_{bw}, w_{gpu}) = (0.30, 0.25, 0.20, 0.25)$, prioritizing battery slightly because energy exhaustion dominates round failure on edge devices (Imteaj, 2022; AbdulRahman, 2020).

3.1.1.0.3 Top-fraction partitioning

FedHAT ranks clients by s_i in descending order and designates the top fraction $\rho \in (0, 1)$ as RR clients:

$$C_r = \{c_{(1)}, \ldots, c_{(\lceil \rho |C| \rceil)}\}, \quad C_l = \{c_j \mid c_j \notin C_r\}. \tag{3.3}$$

Top-fraction partitioning avoids arbitrary numerical thresholds, adapts to the resource distribution of the deployed client population, and guarantees a controlled RR/LR ratio in every round. The participation fraction ρ has established precedent in FL (Nishio, 2019; Chai *et al.*, 2020). We use $\rho = 0.6$, prioritizing the adversarial-training tier while preserving meaningful contributions from LR clients to clean data. In this work, we consider only 10 clients, yielding 60% RR clients and 40% LR clients. Varying different splits are left for the future.

3.1.2 Hybrid Adversarial Training for RR Clients

All C_r selected according to the resource-aware mechanism in Eq. 3.3 consistently participate in AT.

Training every adversarially trained client with the same attack typically leads PGD to generalize poorly to unseen attacks (Tramèr, 2017). FedHAT diversifies the attack across the RR tier instead. Let $M = |C_r|$ with clients ordered by s_i. We partition C_r into two disjoint subsets:

$$C_{PGD} = \{c_{(1)}, c_{(2)}, \ldots, c_{(\lceil M/2 \rceil)}\} \tag{3.4}$$

$$C_{FG} = \{c_{(\lceil M/2 \rceil + 1)}, \ldots, c_{(M)}\}. \tag{3.5}$$

The higher-cost iterative PGD attack is assigned to higher-scoring clients, aligning attack cost with available resources. Each $c_i \in C_{PGD}$ generates examples via PGD (Madry *et al.*, 2018):

PGD + FFGSM. PGD generates robust perturbations; FFGSM complements it by adding small Gaussian noise δ before the gradient step, smoothing the loss landscape and mitigating gradient masking. The resulting adversarial example maximizes prediction error while keeping the perturbation within a controlled magnitude ϵ, ensuring the adversarial sample remains visually similar to the original.

$$x_{PGD}^{adv} = \Pi_{\mathcal{B}(x,\epsilon)} \left(x + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}(f_\theta(x), y)) \right). \tag{3.6}$$

while each $c_j \in C_{FG}$ uses FFGSM (Wong *et al.*, 2020):

$$x_{\text{FFGSM}}^{\text{adv}} = x + \epsilon \cdot \text{sign} \left(\nabla_x \mathcal{L}(f_\theta(x'), y) \right) \tag{3.7}$$

where:

$$x' = x + \delta, \quad \delta \sim \mathcal{N}(0, \sigma^2) \tag{3.8}$$

Each RR client computes its local loss using only its assigned attack:

$$\mathcal{L}_i^{RR} = \begin{cases} \mathcal{L}(f_\theta(x_{PGD}^{adv}), y), & c_i \in C_{PGD}, \\ \mathcal{L}(f_\theta(x_{FFGSM}^{adv}), y), & c_i \in C_{FG}. \end{cases} \tag{3.9}$$

Attack diversification thus arises across the federation, not within any single client.

3.1.3 Standard Training for Low-Resource Clients

LR clients are not auxiliary participants. They typically hold data drawn from distributions underrepresented among RR clients, which is essential for clean-data generalization. FedHAT has LR clients minimize the standard empirical loss:

$$\mathcal{L}_{Standard} = \mathcal{L}(f_\theta(x), y). \tag{3.10}$$

The global model thus receives both adversarial-robustness signal from C_r and clean-data diversity signal from C_l, navigating the well-known robustness-accuracy trade-off in adversarial training (Tsipras, 2018; Zhang *et al.*, 2019b).

3.1.4 Hybrid Training Loss

A hybrid loss function combines standard loss with the adversarial losses, weighted by coefficients $\lambda_1, \lambda_2, \lambda_3$. This loss $\mathcal{L}_{Hybrid}^{(PGD+FFGSM)}$combines standard, PGD, and FFGSM losses together:

$$\begin{aligned}\mathcal{L}_{Hybrid}^{(PGD+FFGSM)} &= \lambda_1 \mathcal{L}_{Standard} + \lambda_2 \mathcal{L}(f_\theta(x_{PGD}^{adv}), y) \\ &\quad + \lambda_3 \mathcal{L}(f_\theta(x_{FFGSM}^{adv}), y),\end{aligned} \tag{3.11}$$

With $\lambda_1 + \lambda_2 + \lambda_3 = 1$. Because FedHAT targets adversarial robustness with clean-data fidelity, we underweight standard training: $(\lambda_1, \lambda_2, \lambda_3) = (0.2, 0.4, 0.4)$. This places robustness as the principal training signal; equal weights tend to under-emphasize robustness, while heavier standard weighting pulls the model away from adversarial decision boundaries (Tramèr, 2017). Crucially, this hybrid objective is achieved without any single client computing more than one attack; the diversification cost is borne by the federation, making the framework feasible on heterogeneous edge populations.

3.1.5 Aggregation and Enhanced Global Model

This section outlines the three-step process of selecting an aggregator, aggregating client models, and refining the global model using data from the selected aggregator to synchronize knowledge from diverse clients.

Step 1: Best Aggregator Selection. At round t, always the best-performing client with the highest validation accuracy is selected as an aggregator:

$$C_{best}^{(t)} = \arg \max_{i \in \{1,\ldots,N\}} Acc(f_{\theta_i}^{(t)}, D_{val}). \tag{3.12}$$

C_{best} will not participate in training for the next round. Instead, it will only aggregate the models. Different settings of the aggregator selection are used in this research, which are discussed in Section 3.3.3.1, including the aggregator from RR, LR, or mixed.

Step 2: Models Aggregation and Global Model Generation. The selected aggregator only aggregates the local models using Federated Averaging (FedAvg):

$$\theta^{(t+1)} = \sum_{i \neq C_{best}^{(t)}} \frac{|D_i|}{\sum_{j \neq C_{best}^{(t)}} |D_j|} \theta_i^{(t)}, \tag{3.13}$$

where $\theta^{(t+1)}$ is the aggregated global model.

Step 3: Enhanced Global Model Generation. To further enhance generalization, the global model is refined using raw data from the best aggregator to generate the EGM:

$$\theta^* = \theta^{(t+1)} - \eta \nabla_\theta \mathcal{L}_{Hybrid}^{(PGD/FFGSM)}(f_{\theta^{(t+1)}}(D_{best}^{(t)}), Y_{best}^{(t)}), \tag{3.14}$$

The loss in Eq. 3.14 depends on the aggregator's tier: if $C_{best}^{(t)} \in C_r$, EGM refinement uses adversarial examples; if $C_{best}^{(t)} \in C_l$, only clean samples are used. By dynamically selecting the best-performing aggregator and excluding it from local training, FedHAT balances robustness, efficiency, and fairness in aggregation.

3.1.5.0.1 Privacy and trust

EGM operates entirely on the aggregator's own local data, which never leaves its device. In fact, EGM can only be performed on the aggregator as to not violate the core principle of FL to avoid sharing raw data to other parties. FedHAT assumes the elected aggregator performs aggregation and refinement honestly, a trust assumption shared with cross-silo and trusted-edge-gateway FL (Kairouz *et al.*, 2021). Combining FedHAT with secure aggregation or differentially private fine-tuning is left for future work.

3.2 Results and Discussions

We evaluate FedHAT against baseline models across datasets, federation sizes, and ϵ values, followed by an analysis of aggregator selection and component-wise ablation studies to isolate each component's contribution.

3.3 Experimental Setup

3.3.0.0.1 Hardware and Software

Experiments ran on an Intel Core i7-8700 CPU, 32 GB RAM, and an NVIDIA RTX A2000 GPU (6 GB), using Python/PyTorch with a fixed random seed of 42.

3.3.0.0.2 Federation Configuration

The default federation has 10 clients; scalability results use 20, 50, and 100. Participation fraction is fixed at $\rho = 0.6$. The model is VGG16 (Simonyan, 2015) trained with cross-entropy loss using SGD (Goodfellow, Bengio, Courville & Bengio, 2016) (lr = 0.001, batch size 32, 3 local epochs per round).

3.3.0.0.3 Client Resource Modeling

Following the heterogeneous FL simulation practice (Yao, 2024; Chai *et al.*, 2020), clients are split into two tiers: RR (edge servers, workstations, and GPU-equipped laptops) and LR (smartphones, tablets, and IoT nodes). Each client c_i receives a fixed resource vector $\mathbf{r}_i = (b_i, cpu_i, bw_i, gpu_i)$ sampled at initialization from its tier's distribution (Table 3.1); round-to-round fluctuation is left to future work. Clients are ranked by weighted score s_i (Eq. 3.2). With ρ=0.6, the top 6 form the RR tier (AT) and the remaining 4 the LR tier (standard training). Within RR, the top 3 by score use PGD-based AT and the lower 3 use FFGSM.

Table 3.1 Resource sampling distributions for the two-tier device population. b_i is normalized battery charge; cpu_i and gpu_i are available capacity percentages; bw_i is uplink bandwidth in Mbps

Tier	b_i	cpu_i	gpu_i	bw_i (Mbps)
RR-tier	$\mathcal{U}(0.6, 1.0)$	$\mathcal{U}(60, 100)$	$\mathcal{U}(50, 100)$	$\mathcal{U}(40, 100)$
LR-tier	$\mathcal{U}(0.1, 0.5)$	$\mathcal{U}(10, 50)$	0	$\mathcal{U}(5, 30)$

3.3.0.0.4 Datasets and Data Partitioning

We evaluate FedHAT on six benchmark datasets spanning increasing difficulty: MNIST (LeCun *et al.*, 1998), Fashion-MNIST (FMNIST) (Xiao, Rasul & Vollgraf, 2017), CIFAR-10 (Krizhevsky, Hinton *et al.*, 2009), CIFAR-100 (Krizhevsky *et al.*, 2009), Tiny-ImageNet (Le, 2015), and ImageNet-12 (a subset of ImageNet (Deng, 2009) introduced in (Li, 2021), resized to 64×64 pixels for tractable training). Both IID and non-IID settings are tested; non-IID uses Dirichlet partitioning with α=0.5 (Li, Huang, Yang, Wang & Zhang, 2019).

3.3.0.0.5 Adversarial Training Configuration

For MNIST/FMNIST (Goodfellow *et al.*, 2014): ϵ=0.3, step size 0.01, PGD-10, 100 rounds. For CIFAR-10/100, Tiny-ImageNet, and ImageNet-12 (Madry *et al.*, 2018): ϵ=0.031, step size 0.007, PGD-10, 250 rounds.

3.3.0.0.6 Evaluation

Robustness is evaluated against FGSM (Goodfellow *et al.*, 2014), PGD-20 (Madry *et al.*, 2018), FFGSM (Wong *et al.*, 2020), CW (Carlini, 2017), MIM (Kurakin *et al.*, 2018), and AutoAttack (Croce & Hein, 2020). PGD-20 (vs. PGD-10 at training) provides a stronger held-out test following standard practice. We additionally compare three aggregator selection strategies: RR clients (default), LR clients, or mixed (Section 3.3.3.1) and report clean accuracy and adversarial accuracy under each attack.

3.3.1 Experimental Evaluation of Adversarial Robustness in FL

A detailed comparison of Tables 3.2 and 3.3 illustrates the impact of data heterogeneity on clean accuracy and adversarial robustness. Under IID settings, baseline methods such as Madry (Madry *et al.*, 2018), TRADES (Zhang *et al.*, 2019b), AVMixup (Lee, 2020), and DBFAT (Zhang & et al., 2023) achieve reasonably high clean accuracy across MNIST, FMNIST, and CIFAR-10. On MNIST, for example, Madry attains 98.52% clean accuracy with 54.50% PGD-20 robustness; TRADES achieves 97.89% and 58.25%; while DBFAT further improves robustness to 68.39%. FedHAT surpasses all baselines with 99.20% clean accuracy and 74.52% PGD-20 robustness, showing a substantial improvement of over 6% compared to the strongest baseline. Similarly, on FMNIST, baseline Madry achieves 76.05% clean accuracy and 65.40% PGD-20 robustness, TRADES drops to 52.78% robustness, and DBFAT reaches 66.24%, whereas FedHAT attains 85.31% clean accuracy and 69.04% PGD-20 robustness. On the more challenging CIFAR-10, baselines struggle even under IID conditions: Madry achieves 26.11%, TRADES 25.49%, and DBFAT 29.03% PGD-20 robustness, while FedHAT reaches 37.44%, clearly outperforming all baselines.

Table 3.2 Comparison of Accuracy and adversarial robustness on three different datasets under IID settings

Dataset	Method	Clean	FGSM	MIM	PGD-20	CW	AA
MNIST	Madry (Madry *et al.*, 2018)	98.52	76.01	60.18	54.50	55.23	50.43
	TRADES (Zhang *et al.*, 2019b)	97.89	76.79	63.29	58.25	57.24	53.72
	AVMixup (Lee, 2020)	98.63	61.41	53.34	42.33	46.95	37.78
	DBFAT (Zhang & et al., 2023)	98.86	78.06	70.97	68.39	63.09	59.39
	FedHAT	**99.20**	**84.08**	**76.41**	**74.52**	**71.76**	**66.37**
FMNIST	Madry (Madry *et al.*, 2018)	76.05	68.53	65.24	65.40	64.26	60.89
	TRADES (Zhang *et al.*, 2019b)	78.13	59.33	52.65	52.78	51.44	48.78
	AVMixup (Lee, 2020)	79.34	61.22	54.93	54.67	49.48	50.07
	DBFAT (Zhang & et al., 2023)	81.49	69.23	66.22	66.24	65.71	61.49
	FedHAT	**85.31**	**72.64**	**69.32**	**69.04**	**68.08**	**65.87**
CIFAR 10	Madry (Madry *et al.*, 2018)	58.75	30.62	27.23	26.11	28.47	22.09
	TRADES (Zhang *et al.*, 2019b)	68.58	31.53	25.92	25.49	23.07	20.89
	AVMixup (Lee, 2020)	70.28	29.51	26.22	26.34	24.07	22.25
	DBFAT (Zhang & et al., 2023)	72.21	31.47	28.57	29.03	29.31	24.25
	FedHAT	**76.76**	**58.65**	**38.93**	**37.44**	**38.07**	**32.97**

Under Non-IID distributions, the degradation of baseline methods is more pronounced. On MNIST, Madry drops PGD-20 robustness from 54.50% to 48.02%, TRADES from 58.25% to 47.21%, and DBFAT from 68.39% to 50.33%, whereas FedHAT maintains 69.62% robustness, showing minimal loss relative to IID. FMNIST follows a similar trend: baseline Madry falls from 65.40% to 54.33% PGD-20, TRADES drops to 44.01%, and DBFAT reaches 58.31%, while FedHAT retains 65.66%, outperforming the strongest baseline by over 8%. On CIFAR-10, baseline robustness collapses drastically, with Madry at 13.00%, TRADES 22.05%, and DBFAT 27.02%, while FedHAT maintains 33.70% PGD-20 robustness, nearly doubling the performance of Madry under severe data heterogeneity. Across all datasets and attack types, including FGSM, MIM, CW, and AA, FedHAT consistently demonstrates smaller performance drops between IID and Non-IID settings, highlighting its superior stability and resilience in FL with heterogeneous client distributions. This detailed analysis confirms that FedHAT not only achieves higher clean accuracy and robustness than existing baselines but also effectively mitigates the negative impact of Non-IID data, making it a robust and reliable approach for practical FL scenarios.

Table 3.3 Comparison of Accuracy and adversarial robustness on three different datasets under Non-IID settings

Dataset	Method	Clean	FGSM	MIM	PGD-20	CW	AA
MNIST	Madry (Madry *et al.*, 2018)	97.82	67.58	52.89	48.02	47.43	43.75
	TRADES (Zhang *et al.*, 2019b)	92.03	48.45	51.56	47.21	45.81	42.36
	AVMixup (Lee, 2020)	97.47	56.50	51.86	46.28	44.46	41.84
	DBFAT (Zhang & et al., 2023)	97.95	68.54	54.18	50.33	49.12	44.32
	FedHAT	**98.12**	**76.24**	**71.11**	**69.62**	**66.98**	**63.26**
FMNIST	Madry (Madry *et al.*, 2018)	72.93	60.11	54.42	54.33	52.19	49.88
	TRADES (Zhang *et al.*, 2019b)	74.93	56.53	44.01	44.01	30.80	39.61
	AVMixup (Lee, 2020)	70.06	56.26	49.21	49.72	47.99	45.15
	DBFAT (Zhang & et al., 2023)	76.19	63.11	56.45	58.31	56.96	53.91
	FedHAT	**82.36**	**69.29**	**66.59**	**65.96**	**65.03**	**62.36**
CIFAR 10	Madry (Madry *et al.*, 2018)	15.27	13.27	13.00	13.00	12.99	8.63
	TRADES (Zhang *et al.*, 2019b)	46.30	24.81	22.21	22.05	19.59	17.85
	AVMixup (Lee, 2020)	48.23	25.29	21.42	24.25	20.25	19.43
	DBFAT (Zhang & et al., 2023)	52.24	27.03	24.12	27.02	22.13	21.20
	FedHAT	**75.36**	**54.10**	**35.05**	**33.70**	**32.51**	**32.21**

Table 3.4 evaluates the scalability and generalization ability of different adversarial defense mechanisms on more diverse and complex datasets, including CIFAR-100, Tiny-ImageNet, and

ImageNet-12. These datasets introduce increased class diversity, higher intra-class variation, and more challenging visual semantics, making adversarial robustness substantially harder to achieve. All methods are trained under the same FL setup using FedAvg and are evaluated using PGD-20, Square attack, and AA, alongside clean accuracy. On CIFAR-100, all baseline methods experience a noticeable decline in both clean accuracy and adversarial robustness compared to CIFAR-10, reflecting the increased difficulty of the task. Madry shows limited robustness, particularly under strong attacks such as AA. TRADES and AVMixup provide incremental improvements, while DBFAT achieves the strongest baseline performance. Nevertheless, the proposed method (FedHAT) consistently outperforms all baselines, achieving 51.24% clean accuracy and significantly higher robustness across all attack settings. In particular, FedHAT improves PGD-20 and AA robustness by more than 3% over DBFAT, demonstrating its effectiveness in high-class-count scenarios. For Tiny-ImageNet, which further increases image resolution and semantic complexity, the performance gap between FedHAT and existing defenses becomes more pronounced. While baseline methods struggle to balance clean accuracy and robustness, FedHAT achieves the highest clean accuracy (42.58%) and demonstrates superior robustness against PGD-20, Square, and AA attacks. Notably, FedHAT improves AA robustness by over 5% compared to the strongest baseline, indicating improved resistance to both gradient-based and gradient-free AAs. On the large-scale ImageNet-12 dataset, all methods show improved clean accuracy due to stronger feature representations, but adversarial robustness remains challenging. DBFAT again represents the strongest baseline, achieving balanced performance across evaluation metrics. However, FedHAT significantly outperforms DBFAT, achieving 64.89% clean accuracy and the highest robustness under all attack types. The improvement is particularly substantial under strong attacks, with FedHAT achieving 31.57% accuracy under PGD-20 and 27.62% under AA, highlighting its strong generalization capability to large-scale, real-world image distributions.

Table 3.4 Accuracy and adversarial robustness on diverse datasets under non-iid settings

ImageNet	Method	Clean	PGD-20	Squair	AA
CIFAR 100	Madry (Madry *et al.*, 2018)	39.32	16.07	23.44	14.36
	TRADES (Zhang *et al.*, 2019b)	43.39	20.05	26.43	16.85
	AVMixup (Lee, 2020)	46.64	23.56	29.16	19.46
	DBFAT (Zhang & et al., 2023)	48.31	24.47	31.57	22.46
	FedHAT	**51.24**	**27.73**	**36.62**	**25.44**
Tiny-ImageNet	Madry (Madry *et al.*, 2018)	26.33	12.26	13.54	10.26
	TRADES (Zhang *et al.*, 2019b)	37.81	15.49	19.38	13.26
	AVMixup (Lee, 2020)	36.19	15.28	19.25	13.18
	DBFAT (Zhang & et al., 2023)	38.24	16.17	20.26	13.96
	FedHAT	**42.58**	**20.06**	**23.65**	**19.54**
ImageNet-12	Madry (Madry *et al.*, 2018)	37.42	22.61	25.57	18.30
	TRADES (Zhang *et al.*, 2019b)	58.82	25.49	28.96	21.81
	AVMixup (Lee, 2020)	59.63	25.81	29.28	20.92
	DBFAT (Zhang & et al., 2023)	61.38	26.47	30.91	22.08
	FedHAT	**64.89**	**31.57**	**35.23**	**27.62**

3.3.2 Robustness under Increasing Attack Strength and Federation Size

In this section, we evaluate the robustness and generalizability of FedHAT under varying attack strengths (ϵ) and federation sizes of 10, 20, 50, and 100 clients, all under non-IID data partitioning.

3.3.2.0.1 MNIST

We repeat the experiment with more clients to assess the model's performance under different conditions. In both experiments, the DBFAT and FedHAT models exhibit only minor performance differences as the number of clients increases, likely because the MNIST dataset is simple. Figures 3.2 and 3.3 show the robustness of DBFAT and FedHAT against various AAs, such as MIM and PGD-20, across a range of ϵ values, for 10, 20, 50, and 100 clients. The general trend shows that accuracy decreases as ϵ increases, as expected, because larger perturbations make it harder for the model to correctly classify inputs. When comparing the two setups, models trained with 10 and 20 clients consistently achieve higher accuracy than those trained with 50 or

100 clients across all attack types and ϵ values. This indicates that increasing the number of clients can introduce more heterogeneity and reduce the effectiveness of AT, possibly due to less frequent or diluted model updates, leading to a slight drop in robustness. The most dramatic gaps appear under stronger attacks and higher ϵ values, such as MIM and PGD-20.

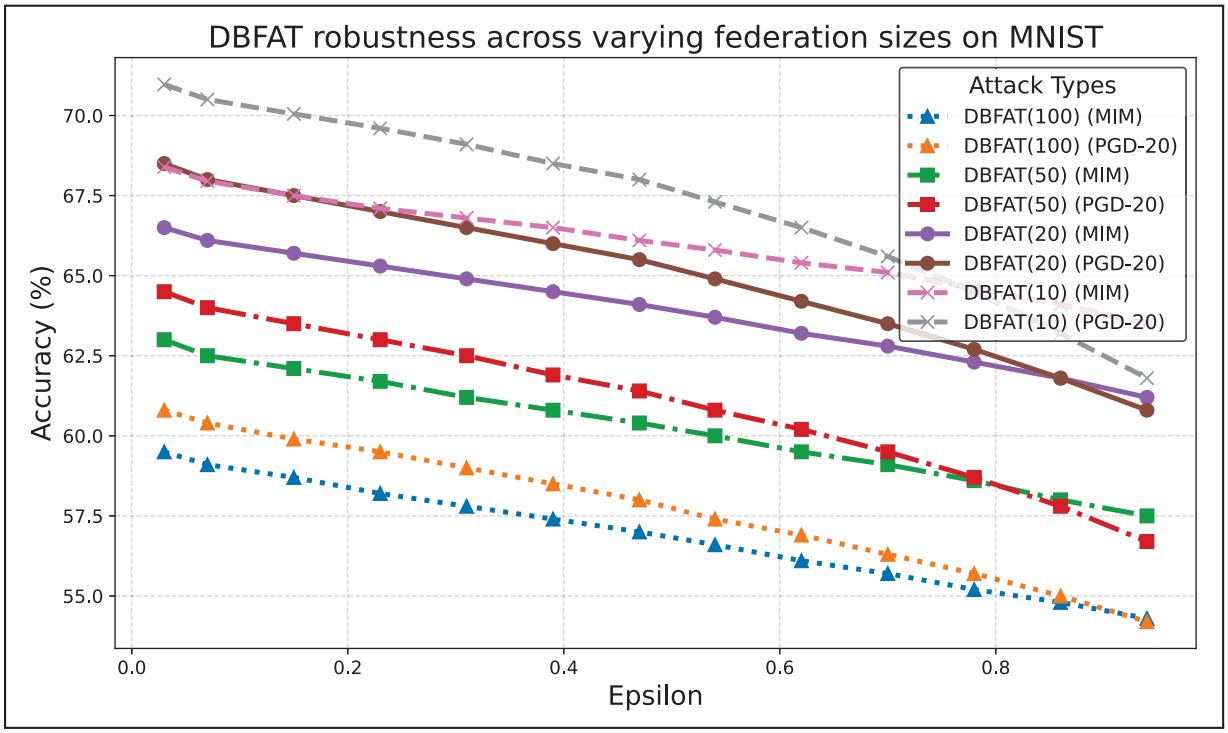


Figure 3.2 Effect of federation size on adversarial robustness of DBFAT on MNIST with increasing ϵ values

3.3.2.0.2 CIFAR-10

Figures 3.4 and 3.5 show that FedHAT consistently achieves higher adversarial accuracy than DBFAT across all perturbation budgets ϵ. Under the MIM attack at $\epsilon = 0.03$, FedHAT attains accuracies of 37.38%, 34.17%, 30.95%, and 26.80% for 10, 20, 50, and 100 clients, respectively, compared to 28.57%, 25.67%, 22.10%, and 18.57% for DBFAT. This corresponds to improvements of approximately 8 to 9% across all federation sizes. As ϵ increases to moderate values, the performance gap persists and widens. For example, at $\epsilon = 0.23$ under MIM with 20 clients, FedHAT maintains 9.76% accuracy, while DBFAT drops to 7.24%; at $\epsilon = 0.39$,

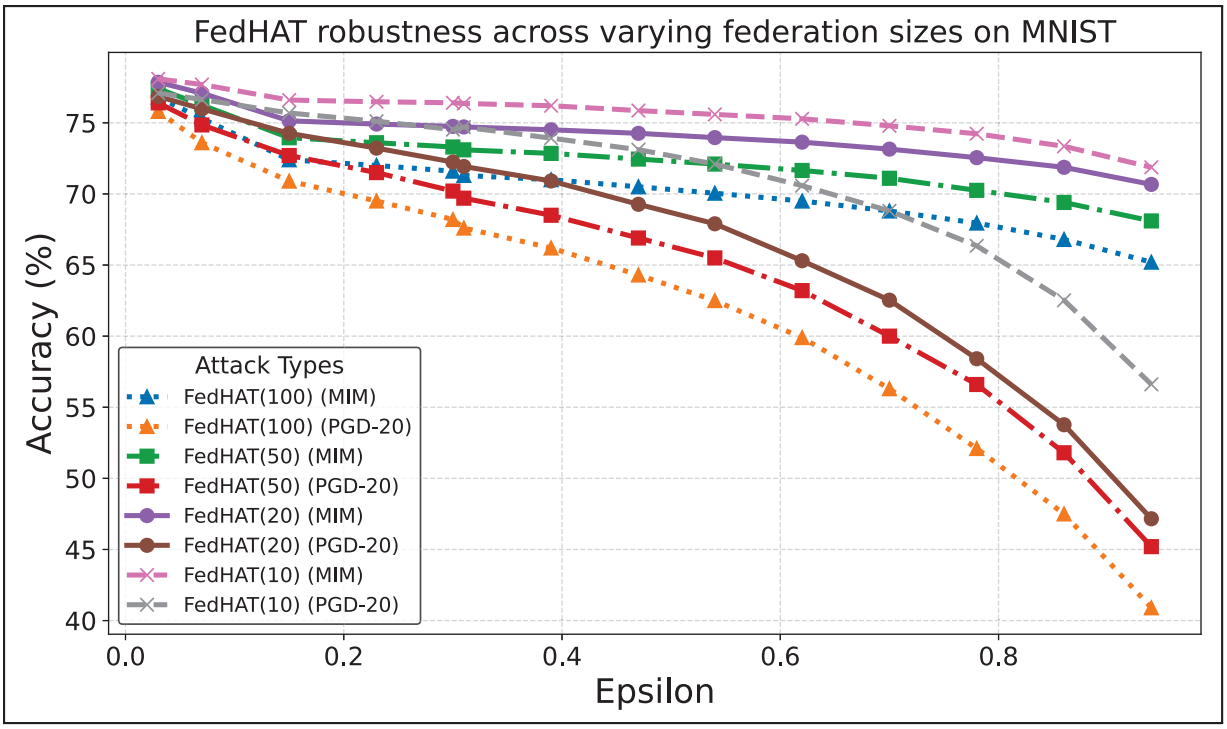


Figure 3.3 Effect of federation size on adversarial robustness of FedHAT on MNIST with increasing ϵ values

FedHAT achieves 7.13% compared to 5.05% for DBFAT. Even at $\epsilon = 0.47$, FedHAT preserves 6.10% accuracy, whereas DBFAT degrades further to 4.33%, indicating superior robustness under increasingly strong adversarial perturbations.

Under the stronger PGD-20 attack, both methods experience rapid performance degradation; however, FedHAT consistently maintains higher accuracy before collapse. For instance, with 10 clients at $\epsilon = 0.15$, FedHAT achieves 3.06% accuracy, compared to 2.46% for DBFAT, and at $\epsilon = 0.31$, FedHAT retains 0.93% accuracy, while DBFAT drops to 0.75%. Similar trends are observed for larger federations: with 50 clients at $\epsilon = 0.23$, FedHAT reaches 1.05%, exceeding DBFAT's 0.92%, and at $\epsilon = 0.39$ FedHAT maintains 0.38% compared to 0.29% for DBFAT. Moreover, FedHAT scales more favourably with federation size, as larger federations consistently yield higher robustness, whereas DBFAT exhibits reduced initial accuracy as the number of clients increases. These numerical results demonstrate that FedHAT not only provides stronger

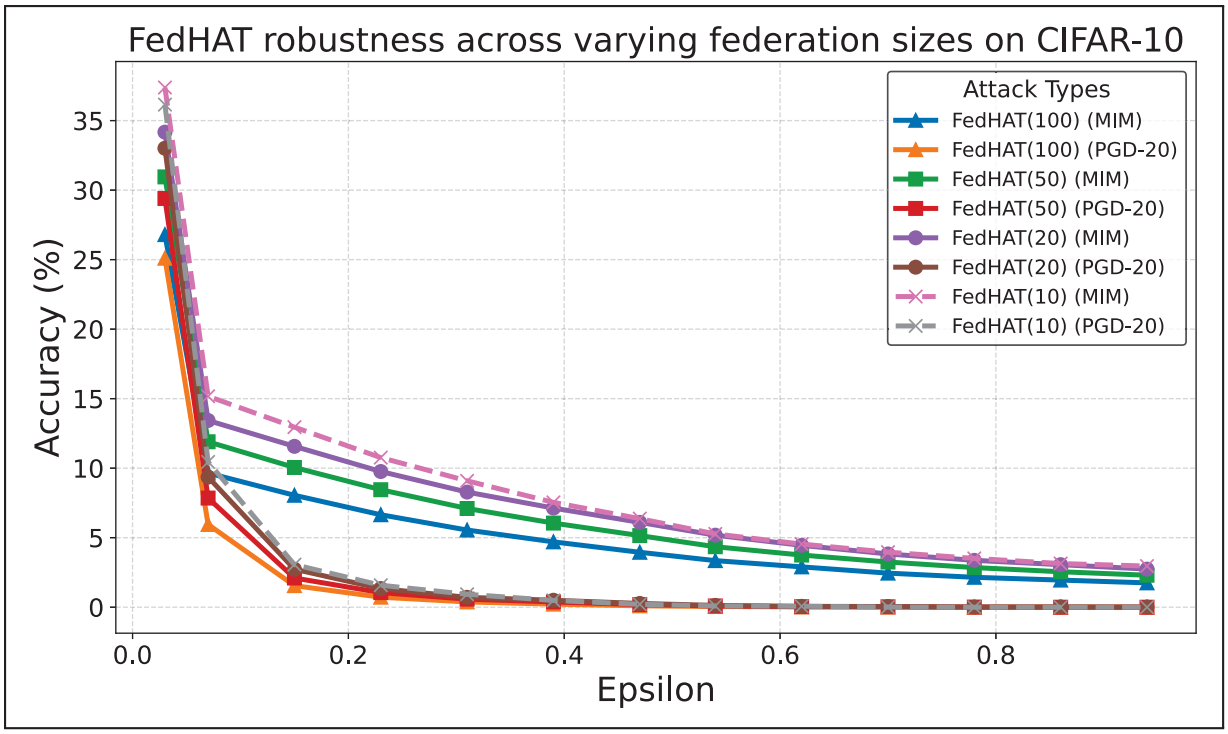


Figure 3.4 Effect of federation size on adversarial robustness of DBFAT on CIFAR-10 with different epsilon values and attacks

robustness across a wide range of ϵ values but also better exploits large-scale federated settings than DBFAT.

3.3.3 Impact of Aggregator Selection Methods and Components

All subsequent experiments are performed under non-IID data partitioning. Ablation studies retain the same setting to isolate design choices under realistic data heterogeneity.

3.3.3.1 Aggregator Selection

As discussed, HFAT performs exceptionally well; however, opportunities exist to enhance its robustness against more powerful attacks, such as PGD-20 and MIM. This can be achieved by leveraging DAS and incorporating EGM under different aggregator selection criteria. Only the aggregator selects differently, while RR clients always do AT. To determine which aggregator

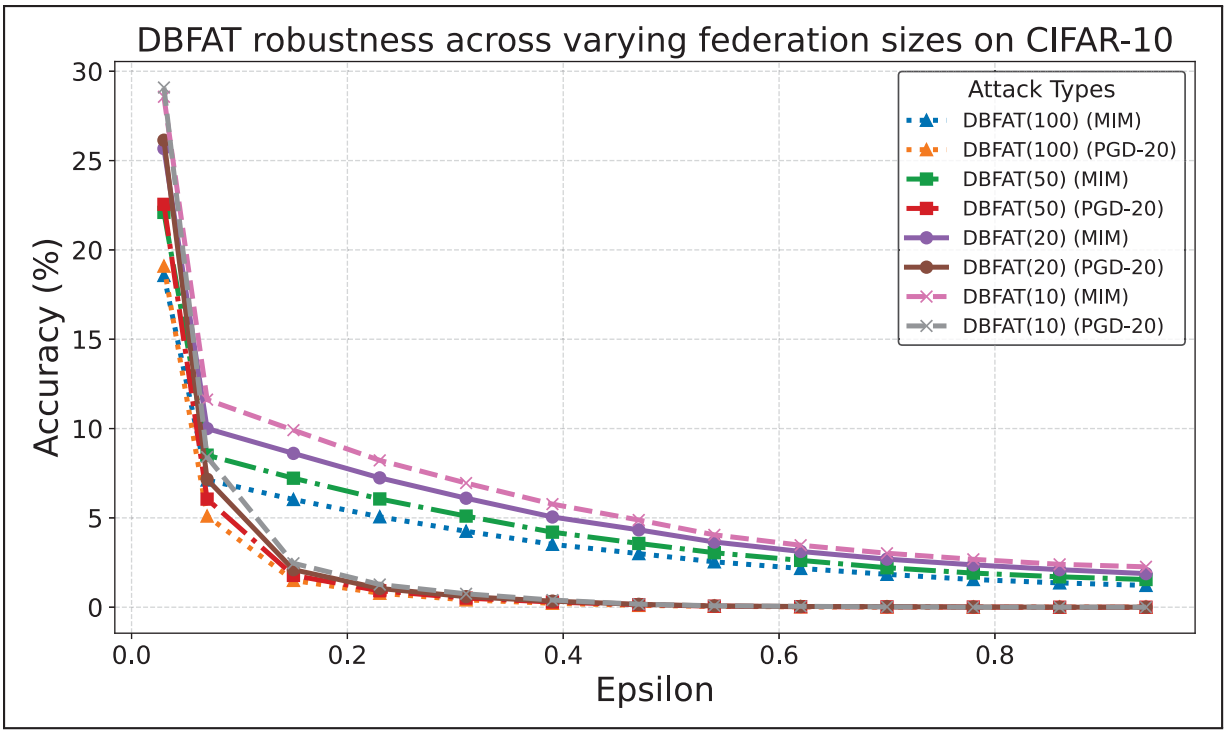


Figure 3.5 Effect of federation size on adversarial robustness of FedHAT under PGD-20 and MIM attacks on CIFAR-10

selection setting is best for adversarial robustness, we conducted extensive experiments. Table 3.5 presents a comparison of different aggregator selection strategies on MNIST and CIFAR-10 using PGD-20 and MIM attacks with FedHAT Solution. Among the strategies, selecting only LR clients yields the lowest robustness, with PGD-20 accuracies of 71.94% (MNIST) and 32.67% (CIFAR-10), and MIM accuracies of 72.79% and 34.45%, respectively, reflecting limited client resources. The Random combination of RR and LR clients improves performance slightly, but remains less effective than RR-only strategies. Using only RR clients under both ST and AT significantly improves robustness, achieving 73.87% and 34.93% PGD-20 accuracies for MNIST and CIFAR-10, respectively, and MIM accuracies of 74.84% and 36.69%. Rotating RR and LR clients further balances the strengths of both client types, yielding modest gains in CIFAR-10 robustness (35.01% PGD-20, 36.89% MIM). The RR-only strategy, however, consistently outperforms all other methods, achieving the highest PGD-20 accuracies of 74.52% (MNIST) and 37.44% (CIFAR-10), and MIM accuracies of 76.41% and 38.93%, confirming

that selecting robust aggregators alone is the most effective approach for maintaining adversarial robustness across datasets. Overall, these results demonstrate that careful aggregator selection is crucial in FL, with RR-only selection providing the strongest and most consistent defence against AAs.

Table 3.5 Comparison of FedHAT Solution with different Aggregator Selection Strategies Using PGD-20 and MIM Attacks

Aggregator Selection	**PGD-20 Accuracy (%)**		**MIM Accuracy (%)**	
	MNIST	CIFAR-10	MNIST	CIFAR-10
LR clients	71.94	32.67	72.79	34.45
Random (RR + LR)	73.25	32.44	74.54	34.31
RR (ST/AT)	73.87	34.93	74.84	36.69
Rotates RR → LR → RR/LR	73.61	35.01	74.91	36.89
RR clients	**74.52**	**37.44**	**76.41**	**38.93**

3.3.3.2 Ablation: Impact of Individual Components in the Complete FedHAT Solution

Table 3.6 presents a comparison of different attack-wise FAT models against the proposed solution (FedHAT) on MNIST and CIFAR-10, using the RR client aggregator. On **MNIST**, FAT variants trained with FFGSM or PGD achieve high clean accuracy (97.49% and 98.46% respectively) and strong adversarial robustness across attacks, with PGD-20 robustness ranging from 63.58% (FFGSM) to 65.70% (PGD), and MIM robustness between 65.95% and 66.77%. Hybrid FAT (HFAT) combining PGD and FFGSM improves MIM robustness slightly to 66.79%. At the same time, FedHAT achieves the highest performance overall, with 99.12% clean accuracy, 74.52% PGD-20 robustness, and consistently superior results across all other attacks (FGSM 84.08%, FFGSM 85.72%, MIM 76.41%). On the more challenging **CIFAR-10**, the differences between models are more pronounced. FAT (FFGSM) achieves 79.86% clean accuracy but only 21.83% PGD-20 robustness, and FAT (PGD) achieves 77.40% clean accuracy with 32.23% PGD-20 robustness. The hybrid FAT improves robustness slightly in some attacks but remains below the performance of FedHAT. The proposed model achieves 76.76% clean accuracy with 37.44% PGD-20 robustness and shows consistent superiority across FGSM (58.65%), FFGSM

(62.58%), and MIM (38.93%), confirming that single-attack FAT models tend to overfit to their respective training attacks, leading to poor robustness generalization. While HFAT alleviates this issue by introducing attack diversity, FedHAT consistently delivers the strongest and most balanced performance across clean and adversarial settings. These gains stem from the combined effects of resource-aware hybrid AT and EGM-based dynamic aggregation, which together enable effective robustness learning under heterogeneous federated constraints.

Table 3.6 Comparison of individual attack-based FAT models against the complete FedHAT framework

Dataset	Model	Clean	PGD-20	FGSM	FFGSM	MIM
MNIST	FAT (FFGSM)	97.49	63.58	65.84	69.40	65.95
	FAT (PGD)	98.46	65.70	62.39	65.10	66.77
	HFAT (PGD & FFGSM)	98.44	65.38	67.87	69.27	66.79
	FedHAT	**99.12**	**74.52**	**84.08**	**85.72**	**76.41**
CIFAR-10	FAT (FFGSM)	79.86	21.83	53.85	58.64	23.74
	FAT (PGD)	77.40	32.23	29.15	40.66	31.38
	HFAT (PGD & FFGSM)	78.41	32.78	57.42	61.25	34.57
	FedHAT	**76.76**	**37.44**	**58.65**	**62.58**	**38.93**

3.4 Conclusion

We presented FedHAT, a resource-aware FAT framework that assigns distinct training roles to clients based on their resource capacity. Unlike prior FAT methods, which uniformly require every participating client to perform AT, FedHAT selectively assigns RR tier that performs hybrid AT (PGD and FFGSM diversified across clients) and an LR tier that performs ST on diverse clean data. The two tiers contribute complementary signals via group-weighted FedAvg aggregation, followed by EGM refinement on the elected aggregator's local data. Extensive experiments across various datasets, adversarial settings, network configurations, varying numbers of clients, and attack intensities (ϵ) show that FedHAT consistently outperforms state-of-the-art FAT baselines under both IID and non-IID settings. These results demonstrate that careful client selection and hybrid training can provide robustness to multiple attack types without overburdening LR devices. These results also demonstrate the effectiveness and adaptability of FedHAT in real-world FL scenarios, especially under constraints imposed by device heterogeneity and adversarial threats.

Future work will extend FedHAT to include additional attack types, while exploring energy efficiency and privacy to further optimize system-level performance in real-world distributed deployments.

CHAPTER 4

ENERGY-AWARE ADAPTIVE INTENSITY CONTROL FOR SUSTAINABLE FEDERATED ADVERSARIAL TRAINING ON HETEROGENEOUS EDGE DEVICES

This chapter presents *EARS* (Energy-Aware Robust Selection), the energy-aware FAT framework that constitutes the third and final contribution of this thesis. As argued in the Introduction and Chapter 1, even a resource-aware tiering of clients, of the kind FedHAT introduces, does not by itself make FAT sustainable: battery levels decline monotonically across rounds, so a device that can afford FGSM-based AT early in training may afford only standard training later and be fully depleted before training ends. Existing FAT methods apply a uniform attack intensity that ignores this evolution, so low-end devices exhaust their energy budgets and drop out; the surviving population collapses to a high-resource subset, introducing selection bias and eroding the distributional coverage that motivates FL in the first place. *EARS* resolves this through two coordinated mechanisms: the H-AATI state machine, which adapts each client's attack strength from full PGD to FGSM to standard training as its battery and estimated affordable rounds decline, stabilized by hysteresis and round-locking; and an energy-aware client selector that preferentially schedules well-resourced clients for the most compute-intensive rounds. The remainder of this chapter formalizes the problem and energy model, details H-AATI and the selection policy, integrates them into the complete *EARS* protocol, and validates the framework across four datasets and ten baselines under both single-attack and AutoAttack evaluation.

4.1 Proposed Method

This section presents the *EARS* framework, including the problem formulation and threat model, the energy model for client resource consumption, the H-AATI mechanism, the energy-aware client selection policy, and their integration into the complete training protocol. Fig. 4.1 illustrates the overall system-level workflow of *EARS*, showing how the central server maintains the global model and selects energy-aware clients based on client availability, affordability, and the current H-AATI state. Each selected client uses a device-aware H-AATI controller to determine its AT intensity from a cascade of attack configurations (e.g., PGD, FGSM, standard

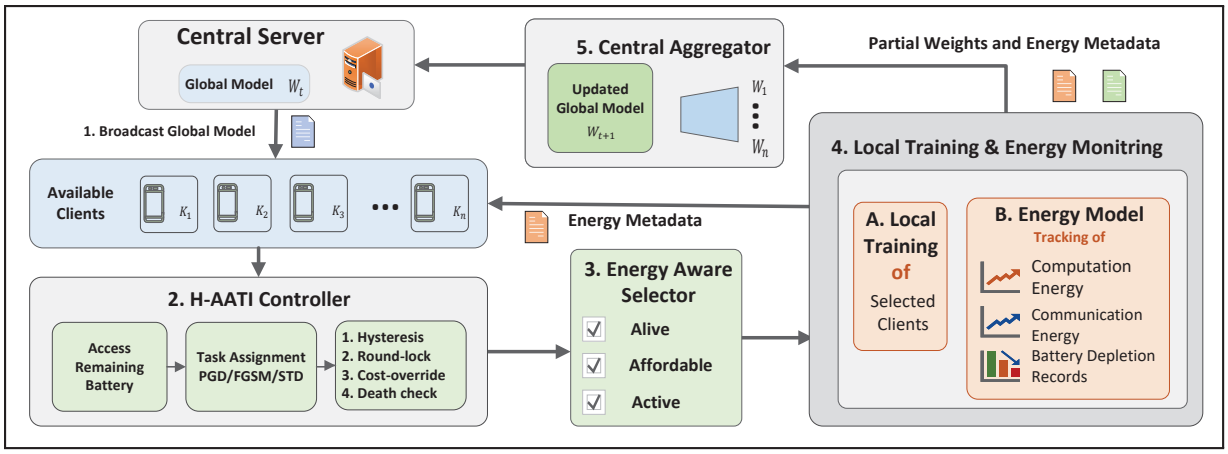


Figure 4.1 This *EARS* Architecture showing how server-side energy-aware selection interacts with client-side H-AATI adaptation, local AT, energy consumption, and battery-state feedback

training). The chosen intensity depends on device type, remaining battery, and estimated future affordability, stabilized through hysteresis, round-locking, and cost-override mechanisms. After local training, each client updates its battery based on the computation and communication energy consumed, and the server aggregates the completed local updates into the next global model.

4.1.1 Problem Formulation

FL setting. We consider a federation of N heterogeneous edge devices (clients), coordinated by a central server over T communication rounds. Each client $k \in [N]$ holds a private local dataset $\mathcal{D}_k$ drawn from a potentially non-IID distribution P_k. The server maintains a global model w^t parameterized by θ. At each round t selects a subset $\mathcal{S}^t \subseteq [N]$ of $K = |\mathcal{S}^t|$ clients to participate in local training and aggregation.

Device heterogeneity. To capture device heterogeneity, each client k is characterized by a device profile k $\rho_k = (\tau_k, F_k, B_k^0, V_k, P_k^{\text{tx}}, P_k^{\text{rx}}, \gamma_k, d_k)$, where $\tau_k \in \{\texttt{high}, \texttt{mid}, \texttt{low}\}$ is the device class, F_k is the compute throughput (FLOPS), B_k^0 is the initial battery capacity (mAh), V_k is the operating voltage (V), $P_k^{\text{tx}}, P_k^{\text{rx}}$ are the transmit and receive power (mW), γ_k is the energy coefficient (device-specific scaling factor for computational energy), d_k is the death threshold

Table 4.1 Device class profiles. Energy coefficient γ reflects compute efficiency; higher values indicate less efficient hardware

Class	B^0 (mAh)	F (GFLOPS)	γ	P^{tx} (mW)	d (mAh)
High-end	4000	50	1.0	300	160
Mid-range	3000	20	1.2	500	120
Low-end	2000	5	1.5	700	80

(mAh), below which the client is considered permanently depleted. We define three canonical device classes with representative parameter ranges, summarized in Table 4.1.

Battery dynamics. Let B_k^t denote the remaining battery of client k at round t. When client k participates in round t with energy expenditure $E_k^t \geq 0$, the battery evolves as:

$$B_k^{t+1} = \max\left(B_k^t - E_k^t,\ 0\right) \tag{4.1}$$

A client is *alive* at round t if $B_k^t > d_k$, and *permanently dead* otherwise. Once dead, a client cannot be revived. We define the battery fraction as $\beta_k^t = B_k^t / B_k^0$.

Adversarial threat model. We adopt the standard ℓ_∞-bounded threat model with perturbation budget ϵ. An adversary seeks to maximize the classification loss over the perturbation set:

$$\delta^* = \arg\max_{\|\delta\|_\infty \leq \epsilon} \mathcal{L}\left(f_\theta(x+\delta),\ y\right) \tag{4.2}$$

Objective. The goal of *EARS* is to minimize the adversarial risk over the joint data distribution:

$$\min_\theta\ \mathbb{E}_{(x,y)\sim\bigcup_k \mathcal{D}_k}\left[\max_{\|\delta\|_\infty \leq \epsilon} \mathcal{L}\left(f_\theta(x+\delta),\ y\right)\right] \tag{4.3}$$

subject to maximizing the number of surviving clients at round T:

$$\max\left|\{k \in [N] : B_k^T > d_k\}\right| \tag{4.4}$$

and minimizing the total energy consumption:

$$\min \sum_{t=1}^{T} \sum_{k \in \mathcal{S}^t} E_k^t \tag{4.5}$$

while maintaining equitable survival across device classes:

$$\min \ \sigma\left(\left\{ \frac{|\{k : \tau_k = \tau,\ B_k^T > d_k\}|}{|\{k : \tau_k = \tau\}|} \right\}_{\tau \in \{\texttt{high},\texttt{mid},\texttt{low}\}} \right) \tag{4.6}$$

where $\sigma(\cdot)$ denotes the standard deviation. The multi-objective nature of Equation (4.3) and (4.6) reflects the fundamental tension between robustness, energy efficiency, and fairness that *EARS* is designed to navigate.

The energy robustness tension. The multi-objective formulation above is driven by a fundamental tension between the intensity of AT and energy consumption. Since each PGD step requires an additional forward-backward pass, the attack overhead factor ω scales linearly with the number of steps: PGD-10 incurs an 11× computational overhead relative to standard training. Under finite battery budgets, this creates three compounding effects:

1. **Intensity-lifetime tradeoff.** For a client with battery B_k^0 and death threshold d_k, the maximum number of participable rounds under attack configuration $\mathcal{A}$ is $R_{\max}(\mathcal{A}) = \lfloor (B_k^0 - d_k)/E_k(\mathcal{A}) \rfloor$. Since E_k scales with $\omega(\mathcal{A})$, a client performing PGD-10 survives approximately $\frac{1}{11}$ as many rounds as one performing standard training.
2. **Heterogeneous depletion.** Under uniform attack assignment, low-end devices (large γ_k, small B_k^0) deplete first, reducing client diversity and biasing the global model toward the data distributions of surviving high-end devices.
3. **Robustness degradation from correlated dropout.** When client attrition is correlated with device class, the surviving pool becomes less representative of the overall data distribution, degrading both generalization and AR.

These challenges motivate the design of *EARS*: H-AATI adapts training intensity based on device energy, while energy-aware selection ensures sustainable and balanced participation.

4.1.2 Energy Model

We develop a compositional energy model that decomposes per-round client energy consumption into computation and communication components, with explicit dependence on the AT configuration.

Computational energy. The computational energy for client k performing E local epochs over $|\mathcal{D}_k|$ samples with model FLOPs Φ and attack overhead factor ω is:

$$E_k^{\text{comp}} = \frac{|\mathcal{D}_k| \cdot E \cdot \Phi \cdot \omega \cdot \gamma_k \cdot \alpha_{\text{base}}}{V_k \cdot 3.6} \tag{4.7}$$

where $\alpha_{\text{base}} = 3.0 \times 10^{-11}$ is a base energy coefficient (joules per floating-point operation), derived from the CMOS dynamic power dissipation model (Chandrakasan & Brodersen, 2002; Burd & Brodersen, 1995) and calibrated to match measured energy profiles of representative edge devices (Holly, Wendt & Lechner, 2020). The denominator converts from joules to milliampere-hours. This formulation follows a widely adopted FL computation energy model (Yang *et al.*, 2020).

The attack overhead factor ω measures the additional computational cost that AT imposes over standard training. Standard training has $\omega = 1$, FGSM doubles the cost ($\omega = 2$) due to one extra pass, and PGD with K steps has $\omega = 1 + K$ since each step adds another pass. A value of $\omega = 0$ indicates no training (DEAD).

Communication energy. Let M denote the model size in megabytes, b_k^t the sampled bandwidth (Mbps) for client k at round t, and η the protocol overhead fraction. The transmission time is:

$$t_k^{\text{comm}} = \frac{M \cdot (1 + \eta) \cdot 8 \times 10^6}{b_k^t \times 10^6} \tag{4.8}$$

and the communication energy comprises both receiving the global model (download) and transmitting the local update (upload):

$$E_k^{\text{comm}} = \frac{(P_k^{\text{rx}} + P_k^{\text{tx}}) \cdot t_k^{\text{comm}}}{1000 \cdot V_k \cdot 3.6} \tag{4.9}$$

Total per-round energy. The total energy consumed by client k in round t is:

$$E_k^t = E_k^{\text{comp}} + E_k^{\text{comm}} \tag{4.10}$$

Conservative estimation. For *pre-selection* affordability checks (before actual bandwidth is realized), we use worst-case bandwidth $b_{\min}$ to obtain a conservative upper bound $\hat{E}_k^t \geq E_k^t$, ensuring that no selected client is unexpectedly depleted:

$$\hat{E}_k^t = E_k^{\text{comp}} + E_k^{\text{comm}}\big|_{b_k^t = b_{\min}} \tag{4.11}$$

4.1.3 Heterogeneous Adaptive Adversarial Training Intensity (H-AATI)

H-AATI is the core contribution of this work: a per-client state machine that adapts AT intensity to both the device's hardware class and its evolving energy state. We first define the state space, then specify the transition semantics, and finally analyze the stabilization mechanisms.

4.1.3.1 State Space

For each device class $\tau \in \{\texttt{high}, \texttt{mid}, \texttt{low}\}$, we define a *cascade* C_τ of L_τ intensity levels, ordered from highest to lowest intensity:

$$C_\tau = \left\langle (\ell_1^\tau, \phi_1^\tau, \mathcal{A}_1^\tau), \ldots, (\ell_{L_\tau}^\tau, \phi_{L_\tau}^\tau, \mathcal{A}_{L_\tau}^\tau), (\texttt{DEAD}, 0, \varnothing) \right\rangle \tag{4.12}$$

Table 4.2 Complete *H-AATI* cascades. Device-aware allocation of adversarial training intensity across datasets and hardware tiers, showing feasible configurations of PGD (varying steps), FGSM, and standard training ($\epsilon = 0$). Entries indicate the relative training cost or feasibility per device class (high-end, mid-range, low-end)

		PGD						
Dataset	**Device**	**20**	**15**	**10**	**7**	**5**	**FGSM**	$\epsilon = 0$
MNIST/ FMNIST	High-end	75	60	45	-	30	15	8
	Mid-range	-	70	50	-	35	20	10
	Low-end	-	-	65	-	40	25	15
CIFAR-10/ CIFAR-100	High-end	-	-	70	50	35	20	8
	Mid-range	-	-	-	65	45	25	10
	Low-end	-	-	-	–	60	35	15

where ℓ_i^τ is the level name, $\phi_i^\tau \in (0, 1]$ is the battery percentage threshold for that level, and $\mathcal{A}_i^\tau$ is the corresponding attack configuration. The thresholds are strictly decreasing: $\phi_1^\tau > \phi_2^\tau > \cdots > \phi_{L_\tau}^\tau > 0$.

Table 4.2 presents the concrete cascades used in our experiments. The key design principle is that higher-capability devices start with more aggressive attacks and have finer-grained intensity graduation, reflecting their larger energy reserves and higher compute throughput.

4.1.3.2 Transition Semantics

Let $s_k^t \in \{1, \ldots, L_{\tau_k}, \texttt{DEAD}\}$ denote the *H-AATI* state of client k at round t, and let r_k^t denote the number of consecutive rounds client k has remained at state s_k^t. At each round, the state update proceeds through three sequential checks:

4.1.3.2.1 Check 1: Death threshold

If the remaining battery falls below the device-specific death threshold:

$$B_k^t \leq d_k \implies s_k^{t+1} = \texttt{DEAD} \tag{4.13}$$

This transition is immediate and irrevocable, regardless of other conditions.

4.1.3.2.2 Check 2: Battery percentage target

The target level is determined by comparing the current battery fraction β_k^t against the cascade thresholds with a hysteresis band h:

$$i_k^* = \min \left\{ i : \beta_k^t \geq \begin{cases} \phi_i^{\tau_k} - h & \text{if } i = s_k^t \\ \phi_i^{\tau_k} & \text{otherwise} \end{cases} \right\} \tag{4.14}$$

The hysteresis band h (set to 0.02 in our experiments) creates a buffer around the current level's threshold, preventing rapid oscillation when the battery fraction hovers near a boundary. The target level is constrained to be non-increasing:

$$i_k^* \leftarrow \max(i_k^*,\ s_k^t) \tag{4.15}$$

ensuring that intensity can only decrease (or remain constant) over time, reflecting the irreversible nature of energy consumption.

4.1.3.2.3 Check 3: Cost-override trigger

Even if the battery percentage suggests maintaining the current level, a cost-based check can force a preemptive downshift. Let $\hat{E}_k = \hat{E}_k^t\big|_{\mathcal{A}=\mathcal{A}^{\tau_k}_{s_k^t}}$ be the conservative energy estimate at the current intensity level. The number of affordable remaining rounds is:

$$R_k^{\text{afford}} = \frac{B_k^t - d_k}{\hat{E}_k} \tag{4.16}$$

If $R_k^{\text{afford}} < R_{\min}$ (set to 5 in our experiments), the state transitions to the next lower level:

$$R_k^{\text{afford}} < R_{\min} \implies s_k^{t+1} = \min(s_k^t + 1,\ L_{\tau_k}) \tag{4.17}$$

This mechanism provides a safety net that accounts for the *absolute* energy cost, not just the battery percentage, and is particularly important for low-end devices where even a modest battery percentage may correspond to insufficient energy for continued high-intensity training.

4.1.3.3 Stabilization Mechanisms

H-AATI incorporates two stabilization mechanisms to prevent pathological behavior:

4.1.3.3.1 Round-locking

A minimum dwell time $\Delta_{\min}$ (set to 3 rounds) is enforced at each level. If $r_k^t < \Delta_{\min}$ and no death threshold or cost-override is triggered, the state is locked:

$$r_k^t < \Delta_{\min} \implies s_k^{t+1} = s_k^t \quad \text{(unless death or cost-override)} \tag{4.18}$$

This prevents high-frequency transitions that would destabilize local training by rapidly changing the adversarial signal between rounds.

4.1.3.3.2 Hysteresis

As described in Equation (4.14), the threshold for remaining at the current level is h lower than the threshold for entering that level. Combined with round-locking, this creates a sticky behaviour:

$$\Pr[\text{transition at } t] \approx 0 \quad \text{if } \beta_k^t \in [\phi_{s_k^t}^{\tau_k} - h,\ \phi_{s_k^t}^{\tau_k}] \tag{4.19}$$

The complete state update is formalized in Algorithm I-1 and theoretical properties in Section 2 in the Appendix I.

4.1.3.4 Comparison with Uniform AATI

To isolate the benefit of device-class-aware cascades, we define a uniform AATI variant, Uniform, that applies a single, device-agnostic cascade to all clients regardless of their hardware class. The uniform cascades are calibrated per dataset to match the base AT configuration in Table 4.2.

This ablation uses identical transition mechanics (hysteresis, round-locking, cost-override) but ignores device heterogeneity in cascade design. The comparison in Section 4.2 demonstrates that device-aware cascades yield measurably better robustness energy trade-offs.

4.1.4 Energy-Aware Client Selection

Given the *H-AATI* state of all clients, the server selects K clients at each round. The *EARS* selection policy considers a client eligible only if it satisfies all required conditions.

Eligible Client A client k is *eligible* for participation at round t if it is alive ($B_k^t > d_k$), not in a terminal *H-AATI* state ($\mathcal{A}_{s_k^t}^{\tau_k} \neq \texttt{DEAD}$), and has sufficient energy to support the current assigned workload ($B_k^t - \hat{E}_k > d_k$).

Let $\mathcal{E}^t = \{k \in [N] : k \text{ is eligible at round } t\}$ denote the set of eligible clients with sufficient remaining battery. The server then selects K clients uniformly at random from $\mathcal{E}^t$ (eligible set of clients):

$$\mathcal{S}^t \sim \text{Uniform}\Big(\big\{S \subseteq \mathcal{E}^t : |S| = \min(K, |\mathcal{E}^t|)\big\}\Big). \tag{4.20}$$

Contrast with baselines. Standard FL and FAT methods select from all non-dead clients without affordability checks, risking depletion. *Eco-FL* (Savoia *et al.*, 2024) weights selection probability by battery fraction β_k^t but does not condition on the energy cost of the *specific attack configuration* assigned to each client. *EARS* combines eligibility filtering (in Section 4.1.4) with intensity-aware energy estimation in Equation 4.11, ensuring that selection decisions are jointly informed by both battery state and training cost.

Idle tracking. To enable future extensions (e.g., fairness-weighted selection), each client maintains a counter Δ_k^t that tracks the number of rounds since the last selection. This counter is reset to zero when selected and incremented otherwise:

$$\Delta_k^{t+1} = \begin{cases} 0 & \text{if } k \in \mathcal{S}^t \\ \Delta_k^t + 1 & \text{otherwise} \end{cases} \tag{4.21}$$

4.1.5 Complete *EARS* Protocol

Warmup phase. During the first T_w rounds (set to 3 in our experiments), all clients perform standard (clean) training regardless of their *H-AATI* state. This allows the global model to reach a reasonable initialization before adversarial perturbations are introduced, following established best practices in AT (Madry *et al.*, 2018; Wong *et al.*, 2020).

Adversarial ratio. Within each local epoch, a fraction $p_{\text{adv}} = 0.8$ of batches are trained on adversarially perturbed inputs, with the remaining fraction trained on clean data. This mixture has been shown to improve the clean robust accuracy trade-off compared to purely AT (Zhang *et al.*, 2019b; Wang *et al.*, 2019b).

Learning rate schedule. We employ a cosine annealing schedule with linear warmup:

$$\eta^t = \begin{cases} \eta_{\min} + \frac{t}{T_w}(\eta_{\max} - \eta_{\min}) & \text{if } t < T_w \\ \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})\left(1 + \cos\left(\pi \cdot \frac{t - T_w}{T - T_w}\right)\right) & \text{otherwise} \end{cases} \tag{4.22}$$

where $\eta_{\max} = 0.001$ and $\eta_{\min} = 0.0001$.

Aggregation. We use standard weighted averaging proportional to local dataset sizes. While more sophisticated aggregation strategies exist (Karimireddy *et al.*, 2020; Acar *et al.*, 2021), weighted averaging is sufficient to demonstrate the benefits of *H-AATI* and energy-aware selection, and allows fair comparison with baselines that also use *FedAvg*-style aggregation. For

Table 4.3 **Main results** on MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100 under PGD attack. "Clean" and "Robust" denote accuracy (%). "Energy" is cumulative consumption (mAh). "Eff." is robust accuracy per energy ×1000. Best adversarial method values in **bold**; best clean accuracy underlined

	MNIST				Fashion-MNIST				CIFAR-10				CIFAR-100			
Method	**Clean**	**Robust**	**Energy**	**Eff.**	**Clean**	**Robust**	**Energy**	**Eff.**	**Clean**	**Robust**	**Energy**	**Eff.**	**Clean**	**Robust**	**Energy**	**Eff.**
EARS	**95.48**	46.56	**78635.2**	**0.5921**	**75.54**	50.98	**79877.5**	**0.6382**	42.90	23.78	**77996.1**	**0.2993**	17.85	**9.10**	**78848.1**	**0.1154**
EARS-Uniform	93.65	45.21	89702.4	0.5040	73.18	53.39	90758.3	0.5883	**43.50**	23.30	79449.6	0.2987	18.24	9.01	81940.0	0.1100
EARS-NoAATI	91.20	48.48	137112.5	0.3536	74.01	55.01	139665.7	0.3939	40.65	24.27	97275.7	0.2495	17.61	9.07	99053.3	0.0916
FedAvg	95.15	0.00	8978.2	0.0000	85.93	0.00	9090.7	0.0000	63.80	0.00	15658.0	0.0000	25.53	0.01	15758.9	0.0006
FedProx	95.95	0.05	9145.8	0.0055	85.16	0.00	9092.5	0.0000	63.10	0.00	15688.9	0.0000	25.55	0.01	15805.2	0.0006
FedPGD	86.91	47.69	137109.4	0.3478	70.96	**55.18**	139196.8	0.3964	37.90	**24.42**	128133.2	0.1906	15.71	8.98	131619.6	0.0682
FedTRADES	85.27	45.00	137289.2	0.3278	72.91	54.24	139442.6	0.3890	35.42	16.11	127099.4	0.1268	17.17	6.70	131689.6	0.0509
DBFAT	62.80	37.61	143443.8	0.2622	47.34	37.65	144846.6	0.2599	26.93	18.68	139198.3	0.1342	11.81	6.96	141453.1	0.0492
CalFAT	75.79	37.28	137535.1	0.2711	69.58	44.88	139742.0	0.3212	33.86	9.91	127239.0	0.0779	7.93	1.81	131759.5	0.0137
Eco-FL	91.25	**51.79**	136450.1	0.3796	74.90	55.01	140379.1	0.3919	37.37	23.74	95480.5	0.2486	**18.39**	8.97	98411.7	0.0911

a deeper understanding of computational and communication overhead, refer to Section 3. The complete *EARS* training protocol is detailed in Algorithm I-2 in the Appendix I, incorporating *H-AATI* state management, energy-aware selection, local AT, and weighted aggregation.

4.2 Experimental Validation

This section presents the main findings and results of this chapter. Detailed descriptions of the experimental setup, datasets, federated and hyperparameter configurations, and evaluation protocol are provided in Section 5 of Appendix I.

4.2.1 Quantitative Results

Table 4.3 presents the comparison across all four datasets under PGD attack evaluation. We report clean accuracy, robust accuracy, cumulative energy consumption (mAh), and robustness efficiency (robust accuracy per unit energy ×1000. The *EARS* variants include the full framework, a variant with uniform (random) client selection replacing the energy-aware scorer, and an ablation removing the H-AATI state machine entirely (NoAATI), which applies full-strength PGD-10 AT to all selected clients in every round.

Energy efficiency. The central claim of *EARS* is that competitive robustness can be achieved at dramatically lower energy cost. Across all four datasets, *EARS* consumes 40-45% less energy than *FedPGD* while maintaining comparable or superior robust accuracy. On MNIST, *EARS* achieves 46.56% robustness at 78,635 mAh compared to *FedPGD*'s 47.69% at 137,109 mAh, a 1.13% gap in robustness for a 42.7% reduction in energy. This translates to a robustness efficiency of 0.592 for *EARS* versus 0.348 for *FedPGD*, a 70% relative improvement. The pattern holds on Fashion-MNIST, where *EARS* reaches 50.98% robustness at 79,878 mAh (efficiency 0.638) against *FedPGD*'s 55.18% at 139,197 mAh (efficiency 0.396). On both CIFAR benchmarks, *EARS* achieves the highest efficiency among all methods, with particularly pronounced gains on CIFAR-100 (0.115 versus *FedPGD*'s 0.068, a 69% improvement). **Robustness accuracy trade-off.** *EARS* consistently achieves the highest or near-highest clean accuracy among AT methods. On MNIST and Fashion-MNIST, *EARS* surpasses all adversarial baselines in clean accuracy by 4-8%, indicating that the adaptive attack intensity mechanism of H-AATI preserves natural accuracy more effectively than uniform full-strength AT. *FedAvg* and *FedProx* achieve higher clean accuracy on Fashion-MNIST and CIFAR-10 but offer no AR whatsoever, confirming that standard training provides no defence against PGD attacks.

Comparison with adversarial baselines. Among dedicated AT methods, *FedPGD* achieves the highest raw robustness on Fashion-MNIST (55.18%) and CIFAR-10 (24.42%), while *Eco-FL* leads on MNIST (51.79%). However, both methods consume approximately 1.7× the energy of *EARS*. *FedTRADES* and *CalFAT* show inconsistent performance across datasets: *CalFAT* degrades sharply on CIFAR-100 (1.81% robustness), and *FedTRADES* underperforms on CIFAR-10 (16.11%). *DBFAT* consistently underperforms across all metrics, with notably depressed clean accuracy (e.g., 62.80% on MNIST, 26.93% on CIFAR-10), suggesting that its dynamic boundary mechanism interacts poorly with federated non-IID data distributions. *Eco-FL*, which also incorporates energy awareness, achieves strong robustness but at energy levels comparable to non-energy-aware methods, indicating that its energy savings mechanism is less effective than the H-AATI cascade in *EARS*.

Ablation insights. Comparing the three *EARS* variants reveals the contribution of each component. Removing energy-aware client selection (Uniform) increases energy consumption by 10-14% across datasets while yielding mixed effects on robustness, confirming that the multi-objective scorer reduces energy without systematically sacrificing accuracy. Removing H-AATI (NoAATI) inflates energy consumption to levels comparable to *FedPGD* as expected, since all clients now perform full PGD-10 training while occasionally achieving marginally higher robustness (e.g., 48.48% on MNIST, 55.01% on Fashion-MNIST). This confirms that H-AATI is the primary driver of *EARS*'s energy savings, gracefully degrading attack intensity as client batteries deplete rather than maintaining unsustainable training loads.

4.2.2 Multi-Attack Robustness

Table 4.4 evaluates all methods under five attack types: FGSM, PGD-20, MIM, C&W, and AA at the canonical perturbation budgets for each dataset. AA combines APGD-CE, APGD-DLR, and APGD-CW to provide the most stringent robustness assessment. ***EARS* under diverse attacks.**

EARS achieves robustness competitive with the best fixed-intensity baselines across all five attacks and four datasets while consuming 25-40% less energy (See Table 4.3). For MNIST, *EARS* leads under FGSM (65.14%) but trails *Eco-FL* under AA (30.37% vs. 36.77%), a 6.4% gap at substantially higher energy cost. On Fashion-MNIST, the PGD-20 gap versus *FedPGD* is 5.27% (50.49% vs. 55.76%) at roughly half the energy consumption. On more complex datasets CIFAR-10 and CIFAR-100, the gaps narrow to below 2.5% and 0.7%, respectively, confirming that *H-AATI* incurs negligible robustness cost on complex tasks. Across all 20 attack-dataset combinations, *EARS* consistently outperforms *FedTRADES*, *DBFAT*, and *CalFAT* by large margins while remaining within 6% of the strongest baseline.

AutoAttack as a robustness floor. AA consistently produces the lowest accuracy across all methods, confirming its role as the strongest evaluation metric. The drop from PGD-20 to AA varies across methods and reveals robustness quality: *DBFAT* suffers a disproportionate collapse on MNIST (35.25% PGD-20 to 18.95% AA), suggesting that the adaptive attack components

Table 4.4 **Multi-attack robustness evaluation** at different ϵ for MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100. Values are accuracy (%). AA combines APGD-CE, APGD-DLR, and APGD-CW

	MNIST					F-MNIST					CIFAR-10					CIFAR-100				
Method	FGSM	PGD-20	MIM	C&W	AA	FGSM	PGD-20	MIM	C&W	AA	FGSM	PGD-20	MIM	C&W	AA	FGSM	PGD-20	MIM	C&W	AA
EARS	65.14	41.26	47.27	42.38	30.37	61.87	54.00	56.98	53.86	54.39	26.51	24.66	25.24	22.71	22.46	9.72	9.03	9.23	7.42	7.37
EARS-Uniform	65.92	40.43	47.07	41.46	29.59	60.45	50.49	54.49	50.15	50.63	24.32	22.46	22.85	19.97	19.73	9.81	8.98	9.13	7.28	7.18
EARS-NoAATI	57.67	40.97	45.17	40.67	33.74	60.89	54.93	56.69	54.10	54.25	25.93	24.90	25.10	22.31	22.12	9.67	8.98	9.13	7.42	7.32
FedAvg	9.47	0.00	0.10	0.00	0.00	4.93	0.00	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.00	0.78	0.00	0.00	0.05	0.00
FedProx	9.47	0.24	0.49	0.00	0.00	5.76	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.00
FedPGD	53.32	40.14	43.80	38.96	33.01	60.60	55.76	57.13	54.35	54.49	25.29	23.97	24.12	22.07	21.78	9.91	9.42	9.52	7.52	7.47
FedTRADES	52.49	38.72	42.09	37.06	31.49	58.84	54.54	55.66	53.22	53.37	18.31	16.26	16.55	13.77	14.06	7.67	6.49	6.69	5.32	5.22
DBFAT	43.75	35.25	38.18	35.25	18.95	39.99	37.40	38.04	37.26	37.45	18.16	17.48	17.72	16.75	16.70	7.42	7.28	7.23	5.96	5.91
CalFAT	43.26	34.67	36.77	35.50	31.40	52.10	44.09	46.19	44.43	44.73	14.84	10.74	11.77	10.55	10.45	1.95	1.42	1.56	1.46	1.32
Eco-FL	62.60	45.02	50.39	47.02	36.77	60.74	55.52	57.18	54.69	55.18	24.56	23.63	24.02	21.63	21.14	10.21	9.62	9.91	7.76	7.67

of AA partially circumvent its defences. In contrast, *EARS* exhibits a moderate and consistent degradation pattern (e.g., 41.26% to 30.37% on MNIST; 22.46% to 19.73% on CIFAR-10), indicating that its robustness is not an artifact of gradient masking or obfuscation. *FedPGD* and *Eco-FL* show similarly stable degradation profiles, confirming AR across all strong baselines.

Non-adversarial methods. *FedAvg* and *FedProx* achieve near-zero robustness across all attacks and datasets, with only marginal FGSM accuracy on MNIST (9.47%) attributable to the large perturbation budget ($\epsilon = 0.3$) occasionally pushing inputs toward decision boundaries that happen to be correctly classified. This confirms that standard federated training provides no meaningful adversarial defence, even under the weakest single-step attack.

Baseline comparisons. Among adversarial baselines, *Eco-FL* emerges as the strongest competitor, achieving the highest raw robustness on MNIST under AA (36.77%) and remaining within 1-2 % of *FedPGD* on CIFAR-10 and CIFAR-100. *FedTRADES* and *CalFAT* again exhibit dataset-dependent fragility: *CalFAT* collapses on CIFAR-100 (1.32% AA), while *FedTRADES* underperforms its PGD-20 results on CIFAR-10 (16.26%) relative to *EARS* (22.46%) and *FedPGD* (23.97%). *DBFAT*'s large PGD-to-AA gap on MNIST suggests it is vulnerable to adaptive attacks, reinforcing the conclusion from Section 4.2.1 that its dynamic boundary approach is poorly suited to heterogeneous federated settings. **Ablation consistency.** The relative ordering among *EARS* variants is consistent with Table 4.3. NoAATI achieves slightly

higher robustness on Fashion-MNIST and CIFAR-10 under most attacks, but at nearly double the energy cost. Uniform selection produces mixed results: marginally higher on some attack dataset pairs and lower on others, confirming that energy-aware selection does not systematically trade robustness for efficiency but rather achieves comparable robustness with fewer, better-targeted client contributions.

4.2.3 Ablation Study

Figure 4.2 isolates the contribution of each *EARS* component by removing one at a time from the full system. We examine two ablations: replacing energy-aware client selection with uniform random selection (Uniform), and removing the H-AATI state machine. Hence, all clients perform full PGD-10 AT every round (NoAATI). We report robust accuracy, cumulative energy, overall client survival, low-end device survival, and the change in robustness and efficiency relative to the full system. **H-AATI is the primary energy-saving mechanism.** Removing H-AATI inflates energy consumption by 25-75% across datasets, rising from approximately 78,000-79,000 mAh to 97,000-139,000 mAh. The effect is most severe on the simpler datasets: on MNIST and Fashion-MNIST, energy nearly doubles, while on CIFAR-10 and CIFAR-100 the increase is more moderate (25-26%). This asymmetry arises because H-AATI's cascade has more room to operate on simpler tasks where lower-intensity attacks (PGD-7, PGD-5, FGSM) can still produce useful gradient signals. In contrast, on CIFAR, the full system already assigns higher attack intensities more frequently to cope with task complexity. The efficiency penalty of removing H-AATI is substantial and consistent: ΔEff. ranges from -0.024 on CIFAR-100 to -0.244 on Fashion-MNIST.

H-AATI is essential for device survival. The most striking result is the survival bar. The full *EARS* system achieves 100% survival across all devices and all datasets. Removing H-AATI causes catastrophic attrition among low-end devices: 0% survival on MNIST, Fashion-MNIST, and CIFAR-100, and only 20% on CIFAR-10. Overall survival drops to as low as 8% on MNIST. This confirms that without adaptive attack intensity degradation, the energy demands of full-strength AT are unsustainable for resource-constrained clients. In a real deployment,

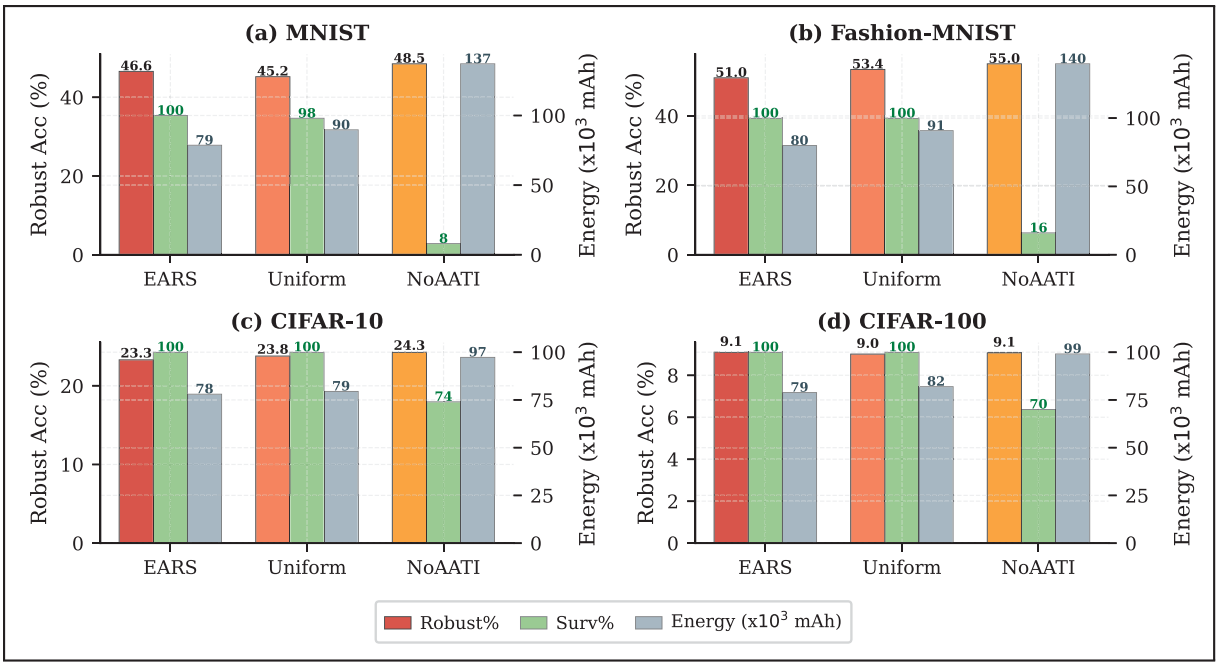


Figure 4.2 Ablation study of EARS components across datasets. Results show that removing selection (EARS-Uniform) and adaptive attack intensity (EARS-NoAATI) degrade robustness, increase energy consumption, and reduce client survival

this would mean that the most energy-limited devices, precisely the edge and IoT nodes that FL is designed to include, are systematically excluded from training, undermining the inclusivity guarantees of the federated paradigm. **Energy-aware selection provides complementary savings.** Replacing the multi-objective scorer with uniform selection has a more nuanced effect. Energy increases modestly (2-14% across datasets), and robustness changes are small and bidirectional: −1.35 on MNIST, +2.41 on Fashion-MNIST, +0.48 on CIFAR-10, and −0.09 on CIFAR-100. This indicates that client selection primarily serves an energy-management role rather than directly boosting robustness. The efficiency penalty is correspondingly smaller than that of removing H-AATI, confirming that the two components operate at different scales: H-AATI controls per-client training cost, while the selector controls which clients are chosen. On MNIST, uniform selection also reduces survival (98% overall, 93.3% at the low end), demonstrating that even modest increases in energy can push borderline devices past their battery limits. **Robustness cost of efficiency.** A key question for *EARS* is whether energy savings come at the expense of AR. The NoAATI ablation provides an upper bound: the maximum robustness gain from removing all energy adaptation is 4.03% (Fashion-MNIST), while on CIFAR-100, the

difference is negligible (−0.03). On MNIST, the full system actually trails NoAATI by only 1.92 points despite using 43% less energy. This suggests that the H-AATI cascade sacrifices only a small fraction of achievable robustness in exchange for dramatic improvements in energy efficiency and device sustainability. This trade-off becomes increasingly favourable as the deployment environment includes heterogeneous, resource-constrained devices.

4.2.4 H-AATI Transition Dynamics

Table 4.5 reports the number of H-AATI state transitions triggered during training, disaggregated by trigger type (battery percentage threshold, cost override, death threshold) and by device class. A transition moves a client to a lower-intensity attack level in the cascades defined by Table 4.2, thereby reducing its per-round training cost.

Energy-aware selection reduces transition pressure. The full *EARS* system triggers substantially fewer transitions than either ablation across all datasets: 92 on MNIST versus 153 for Uniform, and 65 on CIFAR-100 versus 145, respectively. This reduction of 30-55% in transitions compared to Uniform demonstrates that the multi-objective client selector preemptively avoids scheduling energy-depleted clients for intensive training rounds, thereby reducing the frequency with which H-AATI must intervene to downgrade attack intensity. In effect, selection and adaptation operate as complementary mechanisms: the selector manages workload allocation across clients, while H-AATI handles within-client intensity adjustment when battery levels decline despite careful scheduling. **All transitions are battery-driven.** Across all datasets and variants, the overwhelming majority of transitions are triggered by battery percentage thresholds (pct_thr), with cost overrides and death thresholds contributing negligibly. In the full *EARS* system, 100% of transitions are pct_thr-driven, with no cost override or death threshold events across all four datasets. The Uniform variant triggers a small number of cost overrides on MNIST (5 of 153). Still, the death threshold transitions the emergency mechanism that fires when a client is at imminent risk of battery exhaustion, but it never activates in any configuration. This indicates that the battery percentage thresholds in the H-AATI cascade are calibrated

Table 4.5 ***H-AATI* transition summary** for *EARS* across four datasets. "pct_thr" = battery percentage threshold; "cost_ovr" = cost override; "death_thr" = death threshold. High/Mid/Low columns show transitions per device class

Dataset	Method	Total	pct_thr	cost_ovr	death_thr	High	Mid	Low
MNIST	*EARS*	92	92	0	0	19	44	29
	Uniform	153	148	5	0	20	72	61
FMNIST	*EARS*	93	93	0	0	22	44	27
	Uniform	153	153	0	0	21	73	59
CIFAR-10	*EARS*	67	67	0	0	13	38	16
	Uniform	142	142	0	0	19	70	53
CIFAR-100	*EARS*	65	65	0	0	11	37	17
	Uniform	145	145	0	0	21	72	52

conservatively enough to preempt emergency scenarios, and that the energy-aware selector further ensures clients are never pushed to critical depletion.

Device-class distribution reflects energy heterogeneity. The per-class breakdown reveals that mid-range devices account for the plurality of transitions in all configurations (e.g., 44 of 92 on MNIST, 38 of 67 on CIFAR-10 for the full system). This is consistent with the three-tier device model: high-end devices have sufficient battery headroom to sustain higher attack intensities without triggering transitions, while low-end devices, under energy-aware selection, are scheduled less frequently and thus accumulate energy drain more slowly. Mid-range devices occupy the boundary where participation frequency and battery capacity intersect, making them the most likely to cross percentage thresholds during training. Under Uniform selection, where scheduling does not account for energy state, low-end transitions increase substantially (e.g., from 29 to 61 on MNIST, from 16 to 53 on CIFAR-10), reflecting the cost of indiscriminate client assignment. **Fewer transitions, same robustness.** Comparing with Tables 4.3 and Fig. 4.2, the full *EARS* system achieves comparable robustness to the Uniform and NoAATI

variants while requiring far fewer H-AATI interventions. On CIFAR-100, *EARS* triggers only 65 transitions (versus 145 for Uniform) while achieving equivalent robust accuracy (9.10% vs. 9.01%). This suggests that the transitions in the full system are not only fewer but also better targeted, occurring at times when intensity reduction has minimal impact on the global model's AR, because the selector has already ensured that sufficiently energized clients carry the bulk of high-intensity training.

4.2.5 Device Equity and Survival

Figure 4.3 reports per-class survival rates for high-end, mid-range, and low-end devices at rounds 100 and 200 according to dataset configuration, along with the standard deviation across classes (σ) as a measure of equity. Lower σ indicates more uniform survival across the device hierarchy.

***EARS* achieves perfect equity.** The full *EARS* system is the only AT method to achieve 100% survival across all device classes on all four datasets, yielding $\sigma = 0.00$ across all datasets. This result is shared only by the *FedAvg* and *FedProx* methods, which do not perform AT and therefore impose negligible computational overhead. Among all methods that provide meaningful AR, *EARS* is uniquely equitable.

Adversarial baselines systematically eliminate low-end devices. The pattern across *FedPGD*, *FedTRADES*, CalFAT, and *DBFAT* is stark and consistent: low-end devices achieve 0% survival on MNIST, Fashion-MNIST, and CIFAR-100, with only marginal survival (6.7%) on CIFAR-10 for some methods. Mid-range devices fare only slightly better, with survival rates typically below 12% on the simpler datasets and below 28% on CIFAR-10. *DBFAT* represents the extreme case, achieving 0% survival across all device classes on Fashion-MNIST and CIFAR-100, meaning the method exhausts the battery of every client in the federation. These results expose a fundamental limitation of existing FAT: the computational cost of full-strength PGD training is incompatible with heterogeneous device populations, creating a regime in which only the most capable hardware survives to contribute in later training rounds.

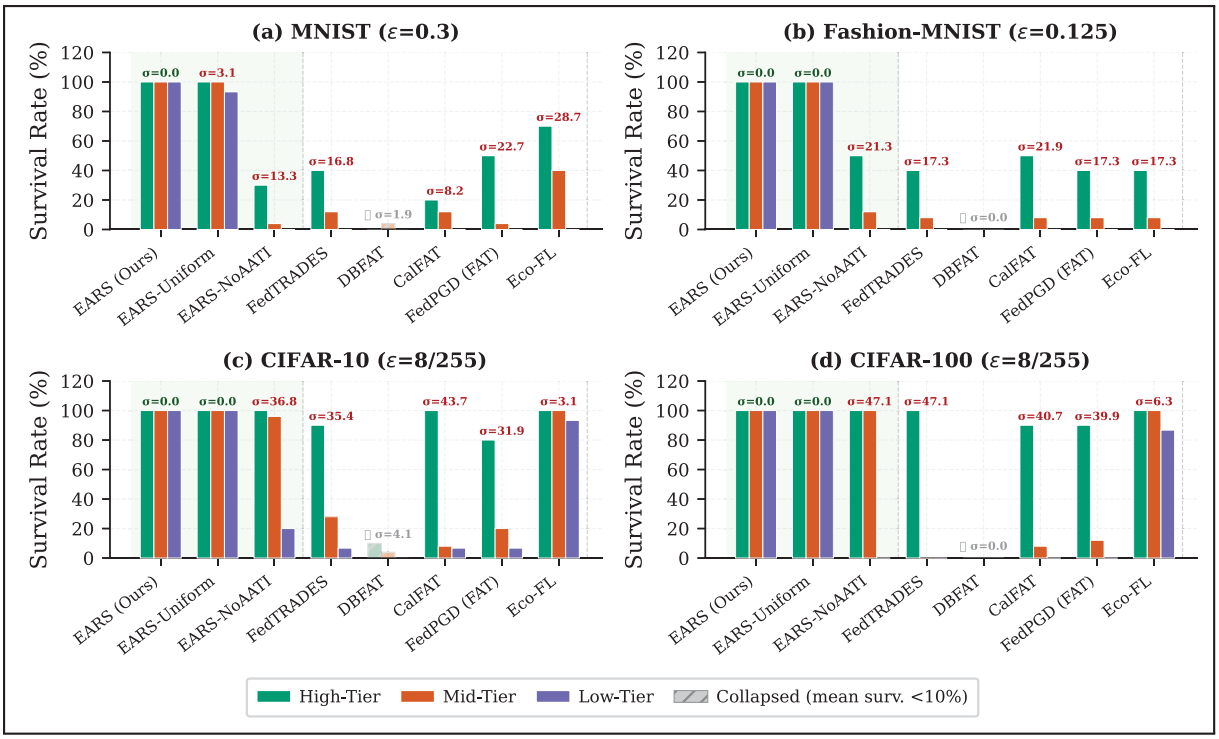


Figure 4.3 Device-class survival rates (%) and equity (σ) at 100 and 200 rounds across four datasets. Lower σ indicates more equitable survival across device classes

***Eco-FL* provides partial but incomplete protection.** *Eco-FL*, which incorporates energy-aware mechanisms, achieves notably better survival than other adversarial baselines. On CIFAR-10 and CIFAR-100, *Eco-FL* preserves 93.3% and 86.7% low-end survival, respectively, with correspondingly low σ values (3.14 and 6.29). However, on the simpler datasets where full AT is more affordable for high-end devices, *Eco-FL*'s energy savings are insufficient to protect low-end clients: survival drops to 0% on both MNIST and Fashion-MNIST. This inconsistency suggests that *Eco-FL*'s energy management scales with task complexity but lacks the adaptive per-client intensity control that H-AATI provides in *EARS*.

The equity robustness dilemma. The NoAATI ablation illustrates the tension between equity and robustness in the absence of adaptive intensity. Without H-AATI, low-end survival collapses to 0% on three of four datasets, and mid-range survival drops to single digits on MNIST and Fashion-MNIST. The resulting σ values (13.30-47.14) are comparable to or worse than the adversarial baselines, confirming that the energy savings enabling *EARS*'s perfect equity

originate almost entirely from the H-AATI cascade. The Uniform ablation produces a mild equity reduction only on MNIST (σ = 3.14 with low-end survival at 93.3%), further supporting the conclusion that energy-aware selection acts as a secondary safeguard against borderline device depletion.

Implications for federated deployment. These results carry practical significance for real-world federated systems. A method that eliminates low-end devices introduces selection bias into the trained model; the global model no longer reflects the data distributions of resource-constrained participants. In domains such as mobile health monitoring or IoT anomaly detection, where the most resource-limited devices may also hold the most operationally critical data, this exclusion is not merely an efficiency concern but a correctness one. ears' perfect survival guarantee ensures that all participants, regardless of hardware capability, contribute throughout the entire training process.

4.2.6 Energy Breakdown

Table 4.6 decomposes total energy consumption into computation and communication components across all methods and datasets, providing insight into where energy is spent and where *EARS* achieves its savings. **Computation dominates energy expenditure.** Across all AT methods, computation accounts for 95.8-99.2% of total energy, with communication contributing only a small residual. This ratio holds consistently across datasets. The implication is clear: reducing AT intensity, the core mechanism of H-AATI, directly targets the dominant energy component. Communication-focused optimizations such as gradient compression or model pruning, while valuable in other federated settings, would yield marginal energy savings in this regime. ***EARS* reduces computation without inflating communication.** On MNIST, *EARS* consumes 76,972 mAh in computation versus 135,623 mAh for *FedPGD*, a 43% reduction while communication costs remain comparable (1,663 vs. 1,486 mAh). The slight increase in *EARS*'s communication energy reflects higher client survival: more devices remain active and transmit updates across all 200 rounds, whereas *FedPGD*'s device attrition in Fig.4.3 reduces late-round communication at the cost of losing participant diversity. This

pattern repeats across datasets: *EARS* consistently trades a marginal increase in communication for a substantial decrease in computation, yielding net savings of 39-43% on MNIST and Fashion-MNIST, and 18-40% on the CIFAR benchmarks, relative to *FedPGD*. **Per-round energy reveals training intensity.** The average energy per round (E/Round) provides a normalized view of training cost. *EARS* averages 786 mAh/round on MNIST, compared to 1,371 for *FedPGD* and 1,434 for *DBFAT*. On CIFAR-10, the gap narrows but remains significant: 390 mAh/round for *EARS* versus 641 for *FedPGD*. *FedAvg* and *FedProx*, which perform only standard forward-backward passes, operate at 78-92 mAh/round across all datasets, establishing a lower bound for federated training energy. *EARS*' per-round cost sits roughly midway between non-adversarial and full AT, reflecting the H-AATI cascade's effect of maintaining a mix of attack intensities across participating clients at each round. ***Eco-FL* achieves partial savings on CIFAR.** *Eco-FL*'s energy profile is dataset-dependent. On CIFAR-10 and CIFAR-100, it achieves total capacities (95,481 and 98,412 mAh) comparable to *EARS*-NoAATI and substantially lower than *FedPGD*, confirming that its energy-aware mechanisms are effective on complex tasks. However, on MNIST and Fashion-MNIST, *Eco-FL*'s energy consumption (136,450 and 140,379 mAh, respectively) is indistinguishable from that of the non-energy-aware baselines. This suggests that *Eco-FL*'s savings depend on training duration relative to task difficulty: on simpler tasks where AT converges quickly, its energy management has limited opportunity to intervene before significant energy has already been expended. *EARS*, by contrast, achieves consistent savings across all four datasets through the per-client, per-round granularity of H-AATI intensity adjustment. ***DBFAT* incurs the highest energy cost.** *DBFAT* is the most energy-intensive method across all datasets (143,444 mAh on MNIST, 141,453 on CIFAR-100), driven by its dynamic boundary computation, which adds overhead to the already costly PGD inner loop. Its communication energy is also notably lower than other adversarial methods (e.g., 1,238 vs. 1,486 mAh on MNIST), but this reflects near-total device mortality in Fig.4.3 rather than communication efficiency; few clients survive long enough to transmit updates in later rounds.

Table 4.6 **Energy breakdown** (E units) across four datasets. *Comp %* indicates the fraction of energy spent on computation. *E/Round* is the average energy per round

Dataset	Method	Total E	Comp E	Comm E	Comp%	E/Round
MNIST	*EARS*	78,635.2	76,972.2	1,663.0	97.9	786.35
	EARS-Uniform	89,702.4	88,072.4	1,630.1	98.2	897.02
	EARS-NoAATI	137,112.5	135,621.3	1,491.2	98.9	1,371.13
	FedTRADES	137,289.2	135,805.6	1,483.6	98.9	1,372.89
	DBFAT	143,443.8	142,206.2	1,237.7	99.1	1,434.44
	CalFAT	137,535.1	136,046.4	1,488.7	98.9	1,375.35
	FedPGD	137,109.4	135,623.0	1,486.4	98.9	1,371.09
	Eco-FL	136,450.1	134,950.2	1,499.9	98.9	1,364.50
FMNIST	*EARS*	79,877.5	78,200.4	1,677.1	97.9	798.78
	EARS-Uniform	90,758.3	89,118.0	1,640.3	98.2	907.58
	EARS-NoAATI	139,665.7	138,156.8	1,508.9	98.9	1,396.66
	FedTRADES	139,442.6	137,931.5	1,511.1	98.9	1,394.43
	DBFAT	144,846.6	143,621.1	1,225.6	99.2	1,448.47
	CalFAT	139,742.0	138,231.1	1,510.9	98.9	1,397.42
	FedPGD	139,196.8	137,688.3	1,508.5	98.9	1,391.97
	Eco-FL	140,379.1	138,872.6	1,506.4	98.9	1,403.79
CIFAR-10	*EARS*	77,996.1	74,695.4	3,300.7	95.8	389.98
	EARS-Uniform	79,449.6	76,138.2	3,311.4	95.8	397.25
	EARS-NoAATI	97,275.7	94,085.4	3,190.4	96.7	486.38
	FedTRADES	127,099.4	124,021.9	3,077.5	97.6	635.50
	DBFAT	139,198.3	136,539.7	2,658.6	98.1	695.99
	CalFAT	127,239.0	124,136.0	3,103.0	97.6	636.19
	FedPGD	128,133.2	125,033.6	3,099.6	97.6	640.67
	Eco-FL	95,480.5	92,339.4	3,141.1	96.7	477.40
CIFAR-100	*EARS*	78,848.1	75,513.4	3,334.6	95.8	394.24
	EARS-Uniform	81,940.0	78,657.6	3,282.4	96.0	409.70
	EARS-NoAATI	99,053.3	95,872.8	3,180.5	96.8	495.27
	FedTRADES	131,689.6	128,621.6	3,068.0	97.7	658.45
	DBFAT	141,453.1	138,938.3	2,514.8	98.2	707.27
	CalFAT	131,759.5	128,683.1	3,076.4	97.7	658.80
	FedPGD	131,619.6	128,559.0	3,060.6	97.7	658.10
	Eco-FL	98,411.7	95,247.6	3,164.1	96.8	492.06

4.2.7 Client Depletion Analysis

Table 4.7 provides a temporal view of client mortality, reporting the total number of device deaths, the breakdown by device class, and the round at which the first and last deaths occur. This complements the snapshot survival rates in Fig.4.3 by revealing the dynamics of device attrition during training. ***EARS* achieves zero deaths universally.** The full *EARS* system records zero client deaths across all four datasets. No device, regardless of class, is depleted at any point during training rounds. This is a stronger guarantee than the survival rates in Fig.4.3 imply: not only are all devices alive at the final round, but none are lost and subsequently replaced or excluded at any intermediate point. The training process benefits from the full diversity of client data. **Baseline attrition is rapid and severe.** Adversarial baselines begin losing clients early. On MNIST, *FedPGD*, *FedTRADES*, and *DBFAT* all record their first death by round 25, with *DBFAT* accumulating 49 of 50 clients dead by round 100. The attrition pattern follows a predictable hierarchy: all 15 low-end devices die first, followed by 22-25 of the 25 mid-range devices, and finally a subset of high-end devices. On CIFAR-10, where per-round energy is lower due to the 200-round budget being spread across a more expensive model, deaths begin later (round 50-75) but still accumulate to 33-48 total deaths for non-energy-aware methods. *DBFAT* again represents the worst case, losing 48 of 50 clients on CIFAR-10 and all 50 on Fashion-MNIST. **Low-end devices die completely and first.** Across all adversarial baselines and datasets, all 15 low-end devices are eliminated without exception (except for *Eco-FL* on CIFAR-10 and CIFAR-100, where only 1-2 low-end devices are lost). This total elimination occurs even when the method achieves moderate overall survival, for instance, *FedPGD* retains 80% of high-end devices on CIFAR-10 but loses 93.3% of low-end devices. The device-class death ordering is invariant: low-end devices die first, mid-range follow, and high-end devices show partial survival only on the more expensive CIFAR tasks, where per-round cost is distributed over a larger model. **Depletion timing reveals energy management quality.** The first-death round provides a useful proxy for how long a method can sustain its most vulnerable clients. *EARS* records no deaths. Among baselines, *Eco-FL* delays the first death the longest: round 45 on MNIST, 65 on Fashion-MNIST, 170 on CIFAR-10, and 195 on CIFAR-100. This confirms that *Eco-FL*'s

energy management provides meaningful protection during complex tasks but is insufficient during simpler ones, where AT converges quickly, and energy accumulates rapidly. By contrast, *FedPGD*, *FedTRADES*, and *CalFAT* show first deaths between rounds 25-90 depending on the dataset, with last deaths extending to round 100 on the simpler datasets and round 200 on the CIFAR benchmarks, indicating that device attrition continues throughout the entire training process for these methods. **The Uniform ablation confirms selection's protective role.** *EARS*-Uniform records only a single death across all datasets (one low-end device on MNIST at round 95), demonstrating that H-AATI alone provides the vast majority of device protection. However, the full system's zero-death guarantee shows that energy-aware selection eliminates even this marginal risk. The NoAATI ablation, by contrast, results in 13-46 deaths across datasets, with attrition profiles resembling those of the adversarial baselines. On CIFAR-100, NoAATI loses all 15 low-end devices while preserving all high-end and mid-range clients, a pattern identical to *FedPGD* but achieved at modestly lower total death count (15 vs. 38), attributable to the energy-aware selector still operating in the NoAATI variant.

4.2.8 Improvement Summary

Across all datasets, *EARS* consistently improves the joint trade-off between robustness, energy efficiency, and client survival. Compared with weaker adversarial baselines such as *FedTRADES*, *DBFAT*, and *CalFAT*, it achieves near-universal Pareto improvements, simultaneously increasing robustness (by up to 13%), reducing energy consumption by approximately 40-45%, and substantially improving client survival (up to 100%). Against stronger baselines (*FedPGD* and *Eco-FL*), *EARS* exhibits a favourable trade-off profile, where small reductions in robustness (typically 1-4%) translate into large gains in efficiency (around 40% energy savings) and significantly higher survival rates (70-90%). Notably, on CIFAR-100, *EARS* matches or slightly exceeds *FedPGD* in robustness while retaining these efficiency benefits, suggesting that the trade-off diminishes as task complexity increases. Overall, the results indicate a stable sustainability exchange rate, where modest concessions in robustness yield disproportionately large improvements in energy efficiency and training viability. Compared to non-adversarial

baselines (*FedAvg* and *FedProx*), *EARS* incurs the expected energy overhead of AT but provides substantial robustness gains, underscoring the inherent trade-off between efficiency and robustness. *Detailed results supporting these observations are provided in the Appendix I (Table I-4 and Section 6.1). Additional results, ablations, and sensitivity analyses are provided in Appendix I (Section 6).*

4.3 Conclusion

This chapter introduced *EARS*, a framework for energy-aware FAT that resolves a key mismatch: the high computational cost of AR versus the limited energy budgets of heterogeneous edge devices. Experiments across four datasets and ten methods show this is a structural issue. Existing approaches, such as *FedPGD*, *FedTRADES*, *DBFAT*, and *CalFAT*, rapidly exclude resource-constrained clients, yielding robust models trained on a biased subset of devices.

EARS addresses this through two mechanisms: H-AATI, which adapts attack intensity per client and round based on battery levels, and a multi-objective client selector that prioritizes well-resourced devices for intensive training. Together, they ensure full device participation, maintain robustness within 1–5% of strong baselines, and reduce energy consumption by 40–43%.

Ablation results show H-AATI is the primary source of these gains, guaranteeing survival for low-end devices, while energy-aware selection provides complementary workload balancing. Their interaction operates at both intra-client and inter-client levels, resulting in more efficient and targeted adaptations.

More broadly, three insights emerge: First, computation dominates energy use in FAT (95–99%), limiting the impact of communication optimizations; Second, robustness trade-offs from energy adaptation are small and diminish with task complexity; on CIFAR-100, *EARS* matches *FedPGD* in robustness while saving 40% of energy. Third, device survival directly affects data representation, as dropped clients remove their local distributions, thereby biasing the global model.

Table 4.7 **Client Depletion Summary** across four datasets. "1st Death" and "Last Death" indicate the round of the first and last client death. Deaths are broken down by device type

Dataset	Method	Deaths	High†	Mid†	Low†	1st Death	Last Death
MNIST	*EARS*	0	0	0	0	None	None
	EARS-Uniform	1	0	0	1	95	95
	EARS-NoAATI	46	7	24	15	30	100
	FedTRADES	43	6	22	15	25	100
	DBFAT	49	10	24	15	25	100
	CalFAT	45	8	22	15	30	100
	FedPGD	44	5	24	15	25	100
	Eco-FL	33	3	15	15	45	100
FMNIST	*EARS*	0	0	0	0	None	None
	EARS-Uniform	0	0	0	0	None	None
	EARS-NoAATI	42	5	22	15	45	100
	FedTRADES	44	6	23	15	45	100
	DBFAT	50	10	25	15	25	90
	CalFAT	43	5	23	15	35	100
	FedPGD	44	6	23	15	30	100
	Eco-FL	44	6	23	15	65	100
CIFAR-10	*EARS*	0	0	0	0	None	None
	EARS-Uniform	0	0	0	0	None	None
	EARS-NoAATI	13	0	1	12	105	190
	FedTRADES	33	1	18	14	55	200
	DBFAT	48	9	24	15	50	180
	CalFAT	37	0	23	14	75	200
	FedPGD	36	2	20	14	60	200
	Eco-FL	1	0	0	1	170	170
CIFAR-100	*EARS*	0	0	0	0	None	None
	EARS-Uniform	0	0	0	0	None	None
	EARS-NoAATI	15	0	0	15	115	200
	FedTRADES	40	0	25	15	90	200
	DBFAT	50	10	25	15	60	175
	CalFAT	39	1	23	15	65	200
	FedPGD	38	1	22	15	85	200
	Eco-FL	2	0	0	2	195	200

CONCLUSION AND RECOMMENDATIONS

Summary of Contributions

This thesis set out to resolve three structural tensions that have prevented FAT from being simultaneously robust, inclusive, and sustainable. We have shown that these tensions are not independent engineering difficulties but a connected chain of failure modes: centralized aggregation introduces representational bias, uniform AT imposes prohibitive costs on resource-constrained devices, and fixed-intensity training exhausts heterogeneous batteries over time. Three contributions address this chain in sequence.

FL-EGM eliminates the dependence on a fixed central aggregator by electing the best-performing participating node each round and using its local data to refine the aggregated model. The empirical record is consistent: across MNIST, FMNIST, CIFAR-10, and CICIDS2017, FL-EGM matches or surpasses centralized training performance (up to 98.5% accuracy, and 98.4% even under a biased aggregator) while converging faster and tolerating biased and unstable aggregators. What this contribution establishes, beyond accurate numbers, is a mechanism for EGM refinement that transforms an architectural liability into an asset and reappears in both subsequent contributions.

FedHAT extends this foundation to heterogeneous populations by stratifying clients according to their resource capacity rather than applying a single training protocol uniformly across all clients. RR clients generate adversarially perturbed examples under diversified attack schedules; LR clients contribute clean-data training that supplies distributional coverage the adversarial tier alone cannot replicate. The combination, aggregated via group-weighted averaging followed by EGM refinement, consistently outperforms state-of-the-art FAT baselines under both IID and non-IID partitions; diversifying attacks across the RR tier improves robustness against *unseen* attacks, and ablations confirm that diversification and EGM each contribute independently.

The critical finding here is not that stratification helps, which is intuitive, but that clean-data participation from weaker devices provides a signal that cannot be trivially substituted by oversampling from stronger ones.

EARS addresses the temporal dimension that tiering alone cannot resolve: even well-designed tiers fail when battery levels decline across rounds. H-AATI implements a per-client state machine that degrades attack intensity as energy reserves fall, transitioning from PGD to FGSM to standard training before a device would otherwise drop out. A multi-objective client selector reinforces this by preferentially scheduling well-resourced devices for the most compute-intensive rounds. The result 100% device survival across all experimental conditions, 40-43% lower cumulative energy consumption, and robust accuracy within 1-5 percentage points of the strongest energy-agnostic baselines represents the most complete resolution of the participation problem in FAT reported to date. Where uniform-intensity baselines deplete 76-100% of their low-end devices before training ends, EARS keeps every device alive, and each percentage point of robustness it concedes relative to the strongest baseline returns roughly a 40% energy saving and an 80% improvement in survival. The deeper lesson is that, on heterogeneous hardware, energy-aware AT is not an efficiency optimization but a correctness requirement: a device that depletes takes its local data distribution with it, silently biasing the global model toward the data held by surviving high-end hardware.

What This Work Leaves Open

Empirical success on benchmark datasets does not constitute a theoretical account of why these mechanisms work, and intellectual honesty requires stating clearly where our understanding remains incomplete.

The EGM refinement step is consistently beneficial, but we do not have a principled explanation for the magnitude of this benefit. The aggregated model is refined on a single client's data for

one additional step: why this should yield the observed gains in accuracy and robustness, rather than simply overfitting to that node's distribution, remains not fully understood. A satisfying explanation would require a theory of how the EGM step interacts with the loss landscape geometry near the aggregated solution territory that the optimization literature has not fully mapped, even for standard federated learning.

The stratification in FedHAT rests on a resource-capacity heuristic that works well in our experimental configurations but may break down in important ways. When the distribution of computational resources is highly skewed, as it is in realistic deployments where a handful of powerful cloud nodes coexist with a large tail of low-power sensors, the group-weighted aggregation scheme may inappropriately amplify the influence of the adversarially trained minority. The right form of aggregation under severe resource heterogeneity remains an open question, and we do not claim to have answered it.

H-AATI's state machine is deterministic and threshold-based. It does not learn or model the interaction between attack-intensity choices across clients or rounds. In principle, a learned policy, one that observes the global training state and selects per-client intensities to jointly optimize robustness and energy, would dominate any fixed schedule. Our experiments confirm that the simple state machine already yields large gains, suggesting that intra-round adaptation does most of the work, but they leave open the question of whether inter-round and inter-client dependencies contain exploitable structure that we are currently ignoring.

There is also a deeper methodological limitation. All three contributions are evaluated on image classification and network intrusion benchmarks with synthetic device heterogeneity. The energy model is physics-based and carefully calibrated, but real deployments involve non-stationary communication channels, correlated device failures, and workload profiles that differ structurally from the uniform random sampling assumed in our experimental design. The extent to which

the observed gains transfer to production federated systems is unknown, and we believe it is the most important empirical question left open by this work.

Extensive experiments with different settings and federation sizes support the results, but evaluation across multiple data splits, random seeds, client selection criteria, and aggregation methods remains open by this work.

Directions for Future Research

Four directions seem most likely to yield fundamental advances.

Understanding EGM theoretically would clarify when aggregator side refinement helps and when it introduces bias, guide the choice of how many refinement steps to perform, and potentially suggest generalizations, for instance, whether refinement on a mixture of multiple high-performing nodes' data could outperform single-node refinement while reducing variance.

Replacing H-AATI's heuristic schedule with a learned policy trained via reinforcement learning or multi-objective optimization would allow the system to discover intensity schedules that exploit structure that is invisible to threshold-based rules. The challenge is that the learning problem is non-stationary; the optimal policy changes as the global model improves across rounds, and the reward signal couples robustness and energy in ways that are not easily decomposed.

Extending the evaluation to real federated deployments, with natural device heterogeneity and realistic communication patterns, is necessary before these methods can be responsibly recommended for production use. This requires infrastructure investment beyond what academic research groups typically sustain; without it, the claims of this thesis remain conditional.

Connecting FAT to formal privacy guarantees, specifically, to differentially private aggregation schemes that are compatible with group-weighted and EGM-refined aggregation, is both practically important and technically non-trivial. The interaction between adversarial and

privacy noise injection in the gradient space is not well understood, and standard analyses of differential privacy in FL do not carry over directly to the AT regime.

These are not merely interesting problems. They are the conditions under which the contributions of this thesis would become deployable without qualification. The present work narrows the gap between FAT and the requirements of real, heterogeneous, energy-constrained deployments. Closing it entirely will require advances in optimization theory, privacy-preserving computation, and evaluation methodology that lie beyond what any single thesis can deliver.

APPENDIX I

SUPPLEMENTARY MATERIAL FOR THE CHAPTER 3 FRAMEWORK

1. Overview

This appendix provides additional material that complements and extends the results presented in Chapter 4. Its purpose is to improve the transparency, reproducibility, and technical completeness of the proposed approach.

Specifically, this appendix includes: (i) detailed analysis of the theoretical properties of *EARS*; (ii) computational and communication overhead; (iii) full algorithmic details; (iv) an expanded experimental setup and evaluation section, covering datasets, preprocessing, configurations, hyperparameters, and evaluation protocols; and (v) additional experimental results, including ablation studies, sensitivity analyses, and further discussion. We also provide extended figures and tables to complement the main results and offer deeper insight into observed behaviours and performance trends.

All sections in this document are organized with an "A" prefix (e.g., Table A I-1, Figure A I-11) to distinguish them from the main text. The content is intended to be read alongside Chapter 4, but it is structured to remain as self-contained as possible for readers seeking deeper technical insights or aiming to reproduce the reported results.

2. Theoretical Properties

2.1 Monotonicity

For any client k and rounds $t_1 < t_2$, if $s_k^{t_1} \neq \texttt{DEAD}$ and $s_k^{t_2} \neq \texttt{DEAD}$, then $s_k^{t_2} \geq s_k^{t_1}$.

Proof. By construction, the target level i_k^* is constrained by Equation (4.15) to satisfy $i_k^* \geq s_k^t$. The cost-override by Equation (4.17) only increases the state index. Round-locking see Equation (4.18) preserves the current state. Therefore, no transition can decrease the state index. □

2.2 Bounded transitions

The total number of transitions for client k over T rounds is bounded by L_{τ_k} (the cascade length for its device class).

Proof. According to Section 2.1, the state index is monotonically non-decreasing and takes values in $\{1, \ldots, L_{\tau_k}, \texttt{DEAD}\}$. Each transition strictly increases the state index, so at most L_{τ_k} transitions can occur. □

2.3 Depletion safety

If $R_{\min} \geq 1$ and the conservative energy estimate $\hat{E}_k \geq E_k^t$ is an upper bound on actual consumption, then no client that participates in a round will be depleted by that round's energy consumption.

Proof. A client k is selected only if it can afford its current intensity (see section 4.1.4), meaning $B_k^t - \hat{E}_k > d_k$. Since $E_k^t \leq \hat{E}_k$, we have $B_k^{t+1} = B_k^t - E_k^t \geq B_k^t - \hat{E}_k > d_k$, so the client remains alive. □

3. Computational and Communication Overhead

3.1 Server overhead

The *H-AATI* state update and eligibility filtering are $O(N)$ per round and require transmitting only the attack configuration $\mathcal{A}_{s_k^t}^{\tau_k}$ (a few bytes) alongside the global model. This is negligible compared to model communication.

3.2 Client overhead

Table-A I-1 Per-round computational cost comparison. $|\mathcal{D}|$ denotes local dataset size, E local epochs, Φ model FLOPs, K PGD steps. *EARS* matches or reduces the cost of all adversarial baselines through adaptive intensity.

Method	Per-Client Cost	Adaptive
FedAvg	$\lvert\mathcal{D}\rvert \cdot E \cdot \Phi$	No
FedProx	$\lvert\mathcal{D}\rvert \cdot E \cdot \Phi + O(\lvert\theta\rvert)$	No
FedPGD	$\lvert\mathcal{D}\rvert \cdot E \cdot (1{+}K) \cdot \Phi$	No
FedTRADES	$\lvert\mathcal{D}\rvert \cdot E \cdot (2{+}K) \cdot \Phi$	No
DBFAT	$\lvert\mathcal{D}\rvert \cdot E \cdot (1{+}K{+}5) \cdot \Phi$	No
CalFAT	$\lvert\mathcal{D}\rvert \cdot E \cdot (1{+}K) \cdot \Phi$	No
EARS	$\lvert\mathcal{D}\rvert \cdot E \cdot \omega(s_k^t) \cdot \Phi$	**Yes**

Each client stores its cascade (at most 6 entries), current-state index, and round counter in $O(1)$ additional memory. The state update involves only scalar comparisons and a single energy estimate, adding negligible computation per round.

3.3 Net effect

While the per-round overhead of *EARS* is negligible, the *H-AATI* reduction provides substantial energy savings: a client downshifted from PGD-10 ($\omega = 11$) to FGSM ($\omega = 2$) reduces its computational energy by $\approx 82\%$ per round, potentially extending its participation by dozens of additional rounds. This compounding benefit is quantified in Section 4.2.

3.4 Complexity summary

Table I-1 summarizes the per-round computational complexity of *EARS* compared to baselines.

4. Algorithms

The complete state update is formalized in Algorithm I-1 while Algorithm I-2 presents the complete *EARS* training protocol, integrating *H-AATI* state management, energy-aware selection, local AT, and weighted aggregation.

5. Experimental setup and evaluation

Algorithm-A I-1 *H-AATI* State Update for Client k

Input: Client state (s_k^t, r_k^t), battery B_k^t, device profile ρ_k
Output: Updated state (s_k^{t+1}, r_k^{t+1})

if $B_k^t \leq d_k$ **then**
  $s_k^{t+1} \leftarrow$ `DEAD`
  return
end if
Compute target level i_k^* via (4.14)
Apply monotonicity constraint (4.15)
if $i_k^* = s_k^t$ **and** $r_k^t \geq \Delta_{\min}$ **then**
  Compute R_k^{afford} via (4.16)
  if $R_k^{\text{afford}} < R_{\min}$ **then**
    $s_k^{t+1} \leftarrow \min(s_k^t + 1, L_{\tau_k})$
    $r_k^{t+1} \leftarrow 0$
    return {Cost-override transition}
  end if
end if
if $i_k^* \neq s_k^t$ **and** $r_k^t \geq \Delta_{\min}$ **then**
  $s_k^{t+1} \leftarrow i_k^*$; $r_k^{t+1} \leftarrow 0$
  return {Threshold transition}
else
  $s_k^{t+1} \leftarrow s_k^t$; $r_k^{t+1} \leftarrow r_k^t + 1$
  return {No transition (locked or stable)}
end if

This section explains the experimental setup, the dataset, federated and hyperparameter configurations, and the evaluation protocol.

5.1 Experimental setup

All experiments were run on a single node of the Compute Canada Narval system, featuring 3 CPU cores, 8.8 GB of memory, and 1 NVIDIA A100-SXM4-40GB GPU. Python was used as the primary programming language, with libraries such as NumPy, pandas, PyTorch, Keras, and TensorFlow. The simulations involved a cross-device FL deployment with 50 clients, divided into high-end, mid-range, and low-end devices, reflecting the typical distribution in real-world mobile ecosystems Bonawitz, Eichner *et al.* (2019). Device profiles follow in Table 4.1. Models

were trained with a batch size of 32 and a local epoch size of 5 to achieve smooth convergence. Cross-entropy was used as the loss function, and the SGD optimizer Goodfellow *et al.* (2016) was applied with learning rates ranging from 0.001 to 0.0001 for consistency. The ResNet18 model is used with a seed value of 42 for reproducibility. Evaluations were conducted on four benchmark datasets, with data partitioned among clients following a Dirichlet distribution to simulate heterogeneous data in FL, using a concentration parameter of $\alpha = 0.5$ to increase Non-IIDness.

5.2 Datasets

In our experiments, we use (i) *MNIST/Fashion-MNIST* LeCun *et al.* (1998); Xiao *et al.* (2017) dataset, where FMNIST serves as a more challenging variant of MNIST. Our setup follows the configuration used in Goodfellow *et al.* (2014). , for these grayscale datasets, we adopt the standard ℓ_∞ threat model with $\epsilon = 0.3$ and step size $\alpha = 0.01$. Due to the simpler data distribution, we employ stronger AT with 20 PGD steps and evaluate using AA with 20 iterations. The *H-AATI* cascade begins at PGD-20 and includes finer-grained intensity levels (PGD-15, PGD-10, PGD-5) to accommodate the higher base computational cost of 20-step attacks. Training proceeds for $T = 100$ communication rounds, which is sufficient for convergence on these datasets. *(ii) CIFAR-10/100* Krizhevsky *et al.* (2009), as well as on more challenging large-scale datasets, are used to train the model following the experimental setup outlined in Madry *et al.* (2018). In these experiments, we set the perturbation bound to $\epsilon = 0.01$ and the step size to $\alpha = 0.007$ and train the model for $T = 200$ communication rounds. To thoroughly assess AR, we conduct extensive experiments using various *H-AATI*strategies, ensuring a comprehensive evaluation of the models under different FL scenarios.

5.3 Federated Configuration

5.3.1 Federation topology

We simulate a cross-device FL deployment with $N = 50$ clients, partitioned into three device classes: 10 high-end (20%), 25 mid-range (50%), and 15 low-end (30%), reflecting the typical distribution in real-world mobile ecosystems Bonawitz *et al.* (2019). Device profiles follow Table 4.1 in the main text.

5.3.2 Data partitioning

To simulate statistical heterogeneity, we partition the training data across clients using a Dirichlet distribution with concentration parameter $\alpha = 0.5$ Hsu, Qi & Brown (2019), resulting in moderately non-IID distributions. Each client's local data is further split 80/20 into training and validation subsets. We verify that each client receives at least 10 samples; if the Dirichlet partition fails to meet this constraint after 20 attempts, we fall back to a uniform random partition and log a warning.

5.4 Training hyperparameter

Table I-2 summarizes key hyperparameters in detail. dataset-dependent configurations are discussed in Section 5.2

5.5 Baselines and Ablation Configuration

We compare *EARS* against seven published baselines and two controlled ablations, yielding 10 methods in total, and summarize each method's attack configuration and distinguishing features in Table I-3.

5.5.1 Ablations

- ***EARS*-Uniform**: Uses the same *H-AATI* transition mechanics but applies a single, uniform cascade across all device classes (as defined in Table 4.2 in the main text), thereby isolating the effect of device-aware cascade design.
- ***EARS*-NoAATI**: Uses energy-aware selection (same eligibility filtering as *EARS*) but fixes all clients at PGD-7 without adaptive transitions, isolating the contribution of dynamic intensity adaptation.

5.5.2 Baselines

- ***FedProx*** use the proximal term $\mu = 0.01$; no AT. Energy computed with standard training overhead $\omega = 1$).
- ***FedTRADES*** use the TRADES loss with $\beta = 6.0$; inner PGD loop uses KL divergence against clean predictions.
- ***DBFAT*** Decision-boundary distance estimation uses 5-step inner PGD per batch; global-model distillation weight $\lambda_g = 0.5$.
- ***CalFAT*** Class prior calibration offsets estimated per client from local label distribution; applied to both adversarial generation and loss computation.
- ***Eco-FL*** Selection probability proportional to $\max(\beta_k^t, 10^{-6})$; all selected clients perform PGD-10.

5.6 Evaluation protocol

During training, we evaluate the global model every 5 rounds on the full test set under both clean and adversarial conditions, PGD-20 for MNIST and FMNIST, and PGD-10 for CIFAR-10 and CIFAR-100, to monitor convergence, while clean accuracy is additionally recorded every round at negligible cost. At the end of training (round T=100 for MNIST/FMNIST and T=200 for CIFAR 10/100), we perform a final evaluation against five complementary attacks calibrated to each dataset's perturbation budget ϵ: single-step FGSM Goodfellow *et al.* (2015), 20-step PGD-20 Madry *et al.* (2018), momentum-based MIM Dong *et al.* (2018), 20-step

ℓ_∞ C&W Carlini (2017), and the parameter-free ensemble AutoAttack Croce & Hein (2020) (APGD-CE + APGD-DLR + APGD-CW, 2 restarts), each evaluated on up to 2,000 test samples. We report seven quantitative metrics: clean accuracy (%), robust accuracy (%) under each attack, total energy(mAh) consumed across all clients and rounds, efficiency (robust accuracy per unit energy $\times 1000$), capturing the robustness energy trade-off in a single scalar, survival rate (%) of clients with $B_k^T > d_k$ at the final round reported overall and per device class, device equity (σ) measured as the standard deviation of per-class survival rates where lower values indicate more equitable participation, and depletion count giving the total number of clients that permanently exhausted their batteries during training.

6. Results and Discussions

This section presents the extended results of Chapter 4 for in-depth analysis of the proposed work and baseline studies, including sensitivity analysis and discussions.

6.1 Improvement Summary

Table I-4 consolidates the relative improvements of *EARS* over each baseline across three dimensions: robust accuracy, energy savings, and client survival. This unified view clarifies the trade-off profile that *EARS* navigates. It highlights where it achieves Pareto improvements and simultaneous gains across multiple axes, versus where it exchanges modest robustness for substantial efficiency and sustainability benefits.

6.1.1 Pareto improvements over weaker adversarial methods

Compared with *FedTRADES*, *DBFAT*, and CalFAT, *EARS* achieves simultaneous improvements across all three axes in nearly every dataset. On CIFAR-10, *EARS* improves robustness by 7.19, 4.62, and 13.39%, respectively, while saving 38-44% of energy and increasing survival by 66-96%. The gains are most dramatic against *DBFAT*, where *EARS* delivers +8.95% robustness, +45.2% energy savings, and +98% survival on MNIST, and +13.33% robustness, +44.9% energy

savings, and +100% survival on Fashion-MNIST. These are unambiguous Pareto improvements: *EARS* is simultaneously more robust, more efficient, and more sustainable.

6.1.2 Competitive trade-offs against strong baselines

Against *FedPGD* and *Eco-FL*, the two strongest adversarial baselines, *EARS* trades small robustness margins for large efficiency gains. Relative to *FedPGD*, *EARS* concedes 1.13 % on MNIST, 4.20 on Fashion-MNIST, and 1.12 on CIFAR-10, but saves 39-43% of energy and improves survival by 72-88 %. On CIFAR-100, *EARS* slightly exceeds *FedPGD* in robustness (+0.12%) while maintaining the same energy and survival advantages. Against *Eco-FL*, the robustness gap is larger on MNIST (-5.23%) and Fashion-MNIST (-4.03%), but *EARS* still saves 42-43% more energy and gains 66-88 % in survival on those datasets. On the CIFAR benchmarks, the robustness gap narrows to under 0.5 % while *EARS* retains 18-20% energy savings and marginal survival improvements.

6.1.3 The robustness of the sustainability exchange rate

The table reveals a consistent exchange rate: each percentage point of robustness conceded to *FedPGD* buys approximately 35-40% energy savings and 60-80 % of survival improvement. On MNIST, the 1.13-point robustness deficit translates to 42.6% energy savings and an 88% survival gain. Whether this trade-off is favourable depends on the deployment context, but in energy-constrained environments where device longevity determines the feasibility of training completion, the exchange is overwhelmingly advantageous. Moreover, on CIFAR-100, arguably the most realistic benchmark for task complexity, *EARS* matches or exceeds *FedPGD* in robustness while retaining all efficiency and survival benefits, suggesting that the trade-off vanishes as task difficulty increases.

6.1.4 Non-adversarial baselines occupy a different regime

FedAvg and *FedProx* achieve zero robustness at minimal energy cost, resulting in negative energy savings for *EARS* (as indicated by the large negative percentages). This is expected:

EARS consumes 4-9× times more energy than standard training because it performs AT. The survival delta is zero because both *FedAvg* and *FedProx* also achieve full survival. These baselines define the efficiency frontier for non-robust training and confirm that AR carries an inherent energy premium, one that *EARS* minimizes but cannot eliminate.

6.1.5 Ablation deltas in context

The improvements over *EARS*-Uniform (1-12% energy savings, minimal robustness change) and *EARS*-NoAATI (20-43% energy savings, 26-92% survival gain) are consistent with the component-level analysis in Section 4.2.3. Notably, the improvement over NoAATI on survival (+92% on MNIST, +84% on Fashion-MNIST) underscores that H-AATI is not merely an efficiency optimization but a prerequisite for inclusive federated training in heterogeneous environments. Further results, discussion, and sensitivity analysis are presented in the Appendix I; see section 6.

6.2 Clean Accuracy Analysis

In this work, Adversarial Robustness (AR) is the primary training objective; **clean accuracy**, i.e., model performance on unperturbed test samples, remains critical for practical deployment. A robust model providing poor clean accuracy fails on the majority of real-world inputs (assuming adversarial examples represent minority attack scenarios).

AT typically degrades clean accuracy due to the fundamental robustness-accuracy trade-off: models learning robust features may sacrifice capacity for fine-grained benign features. Analyzing this degradation quantifies the cost of robustness; the detailed accuracy analysis is presented in Figure I-1.

6.2.1 MNIST Clean Accuracy Performance

On MNIST, *EARS* achieves 95.48% clean accuracy, which is remarkably high and *exceeds* FedAvg (95.15%) by 0.33% despite AT. This counterintuitive result stems from AT's implicit

regularization effect: learning robust features prevents overfitting to spurious benign patterns. *EARS*-Uniform achieves slightly lower 93.65%, while *EARS*-NoAATI reaches 91.20%, indicating *H-AATI*'s adaptive perturbation schedule better preserves clean accuracy than fixed high intensity training. The 4.28% gap between *EARS* (95.48%) and *EARS*-NoAATI (91.20%) demonstrates *H-AATI*'s dual benefit: improving energy efficiency *and* reducing degradation in clean accuracy. Among baselines, *Eco-FL* achieves a competitive 91.25%, while *FedPGD* reaches 86.91%. *DBFAT* suffers catastrophic degradation to 62.80%, a 32.35% drop relative to FedAvg, likely due to its erratic training dynamics and client dropout, which undermine model convergence as present in Figure I-1a.

6.2.2 Fashion-MNIST Clean Accuracy Trends

Fashion-MNIST exhibits more pronounced robustness-accuracy trade-offs. *EARS* achieves 75.54% versus FedAvg's 85.93%, a 10.39% gap representing the cost of AR shows in Figure I-1b. This larger penalty compared to MNIST stems from Fashion-MNIST's texture-based features being more sensitive to adversarial perturbations during training. *Eco-FL* achieves the highest clean accuracy among robust methods (74.90%), closely followed by *EARS*-NoAATI (74.01%) and *EARS* (75.54%). The narrow spread (0.64-1.64 %) indicates that robust methods achieve similar benign performance on Fashion-MNIST regardless of energy consumption, validating that *EARS*'s efficiency gains come without a clean accuracy sacrifice. *DBFAT* again catastrophically fails with 47.34% accuracy, 45% below the FedAvg baseline. This consistent *DBFAT* failure across datasets indicates fundamental incompatibility between its dynamic budget allocation and federated non-IID learning dynamics.

6.2.3 CIFAR-10 Clean Accuracy Under Extended Training

Figure I-1c shows the evolution of clean accuracy for CIFAR-10. *EARS* achieves 42.90% versus FedAvg's 63.80%, a substantial 20.90% gap representing the robustness tax on complex visual recognition. Interestingly, *EARS*-Uniform achieves marginally higher 43.50% (+0.60% over *EARS*), suggesting uniform selection's increased data diversity slightly benefits clean accuracy

despite energy costs. However, this 0.60% improvement hardly justifies the 12% increase in energy consumption (79.4 vs 78.0 10^3 mAh). *EARS*-NoAATI achieves 40.65% *lower* than full *EARS* despite fixed high-intensity training. This 2.25% deficit indicates that excessive AT constant PGD-10 throughout 200 rounds leads to overfitting to adversarial examples at the expense of benign features. *H-AATI*'s gradual reduction preserves better clean accuracy by balancing adversarial and benign objectives. *FedPGD* reaches 37.90% while *FedTRADES* achieves only 35.42%, both substantially below *EARS*. *DBFAT*'s 26.93% represents a 58% relative degradation compared to FedAvg, approaching random guessing territory (10% for 10-class).

6.2.4 CIFAR-100 Fine-Grained Classification Challenges

CIFAR-100's 100-class scenario amplifies the tensions between robustness and accuracy. *EARS* achieves 17.85% versus FedAvg's 25.53%, a 7.68% gap (30% relative degradation). Despite this substantial penalty, *EARS* matches or exceeds all robust baselines. *Eco-FL* achieves the highest robust-method clean accuracy at 18.39% (only 0.54% above *EARS*), while *EARS*-Uniform reaches 18.24%. The tight clustering (17.61-18.39% range) among top methods indicates a clean accuracy ceiling around 18% for robust CIFAR-100 training under current techniques. *FedPGD* achieves 15.71% and *FedTRADES* 17.17%, competitive but energy-inefficient. *CalFAT* catastrophically collapses to 7.93%, well below FedAvg's performance. This extreme failure (69% relative degradation), combined with poor robust accuracy (1.81%), renders *CalFAT* completely unsuitable for CIFAR-100 as shown in Figure I-1d and I-2d.

6.3 Robust Accuracy Convergence

EARS Achieves Competitive Robustness with 40% Energy Savings as presented in Figure I-2.

6.3.1 Dataset-Specific Performance

Across datasets, *EARS* maintains robustness comparable to that of competitive methods despite substantial energy savings. On MNIST, *EARS* achieves 46.56% robust accuracy within 10.1%

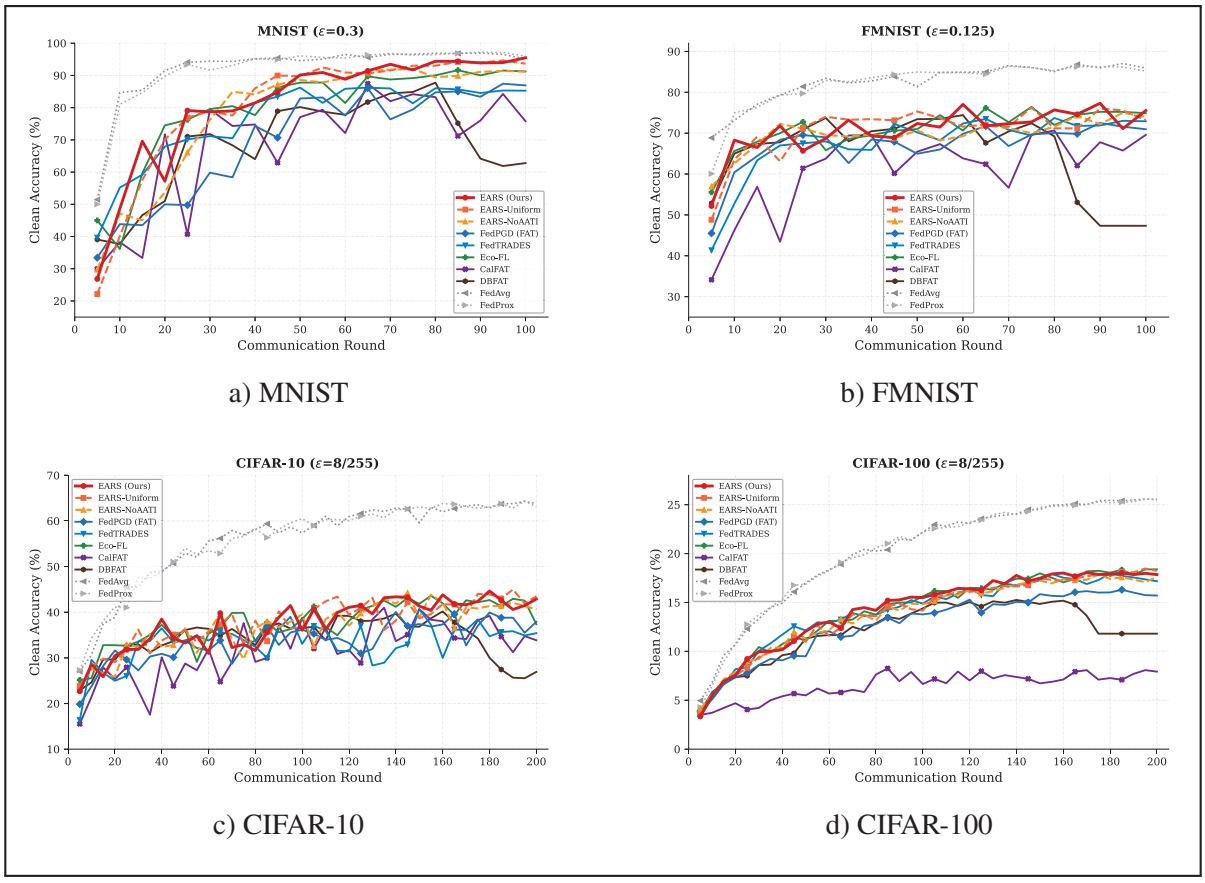


Figure-A I-1 Clean accuracy over communication rounds across datasets.

of the best method (*Eco-FL* at 51.79%). This represents a strategic trade-off: accepting 5.23 % lower accuracy enables a 42.5% energy reduction and perfect client survival, versus *Eco-FL*'s 66% dropout rate. Fashion-MNIST shows similar patterns with *EARS* at 50.98%, trailing *Eco-FL* (55.01%) by 7.3%. The gap narrows slightly compared to MNIST, suggesting *H-AATI*'s adaptive perturbations generalize better to Fashion-MNIST's texture-based features versus MNIST's shape-based digits. CIFAR-10 demonstrates *EARS*'s closest competition to baselines, achieving 23.30% versus *FedPGD*'s leading 24.42%, only a 4.6% gap. This narrow margin is remarkable given *EARS* consumes 39% less energy (78.0k mAh vs 128.1k mAh), validating that energy efficiency need not catastrophically degrade robustness on complex visual tasks. CIFAR-100 represents *EARS*'s strongest showing: 9.10% robust accuracy, tied for highest among all methods. This simultaneous achievement of the best accuracy and the

lowest energy demonstrates *H-AATI*'s particular effectiveness in fine-grained classification. The adaptive perturbation schedule appears to provide implicit regularization beneficial for 100-class scenarios.

6.3.2 Comparison with Ablation Variants

EARS-NoAATI often outperforms full *EARS* in pure robustness: +4.1% on MNIST (48.48% vs 46.56%), +7.3% on Fashion-MNIST (55.01% vs 50.98%), +4.2% on CIFAR-10 (24.27% vs 23.30%). This confirms that *fixed high-intensity AT maximizes accuracy when energy is unlimited*, a known result from centralized AT literature. However, *EARS*-NoAATI's superior accuracy comes at prohibitive costs: 74% higher energy consumption (137.1k mAh vs 78.6k mAh on MNIST) and 92% client dropout (8% survival vs 100%). This validates *EARS*'s design philosophy: accepting modest robustness degradation (2-4%) enables a sustainable federation that fixed-intensity methods cannot achieve. Interestingly, on CIFAR-100, *EARS* marginally outperforms *EARS*-NoAATI (9.10% vs 9.07%), suggesting adaptive perturbations enhance generalization on complex tasks. This aligns with curriculum learning theory, gradual difficulty increase (via increasing perturbation early, then reducing) can outperform constant difficulty.

6.3.3 Convergence Dynamics Analysis

6.3.3.1 MNIST Convergence Trajectory

EARS demonstrates smooth monotonic convergence on MNIST, progressing from 7.60% (Round 5) to 46.56% (Round 100). The trajectory lacks oscillations or plateaus characteristic of unstable training, indicating *H-AATI*'s adaptive perturbations maintain consistent training dynamics as illustrated in Figure I-2a.

This logarithmic convergence pattern is typical of AT, where initial rounds establish coarse robust features while later rounds refine decision boundaries. *H-AATI*'s energy reduction is concentrated in Phase 3, when marginal accuracy gains diminish, thereby efficiently allocating resources to high-return early training. *Eco-FL* follows a similar trajectory but converges to

higher final accuracy (51.79%), with steeper Phase 2 improvements (0.42%/round vs *EARS*'s 0.33%). This suggests *Eco-FL*'s energy-intensive later training provides meaningful robustness gains, justifying its higher energy cost for accuracy-prioritized deployments. *DBFAT* exhibits unstable convergence: rising from 7.95% to 48.73% (Round 80), then collapsing to 37.61% (Round 100) and an 11.12% degradation. This late-stage collapse correlates with mass client dropout (20% survival at Round 80 $\rightarrow$ 2% at Round 100), demonstrating how federation destruction degrades model quality even when individual clients maintain accuracy.

6.3.3.2 CIFAR-10 Extended Training Dynamics

CIFAR-10's 200-round training reveals long-term convergence behaviour in Figure I-2c. *EARS* progresses from 0.75% (Round 5) to 23.30% (Round 200), showing sustained improvement without premature plateau. Despite lower initial robustness compared to *FedPGD* (0.75% vs 10.39% at Round 5), *EARS* achieves near-parity by Round 200 through consistent per-round gains. The 31× improvement in accuracy (0.75% $\rightarrow$ 23.30%) despite concurrent 40% per-round energy reduction validates *H-AATI*'s efficiency. *FedPGD* achieves only a 2.3× improvement (10.39% $\rightarrow$ 24.42%) while maintaining a high energy level, indicating diminishing returns from fixed-intensity training. *FedTRADES* shows poor long-term convergence, plateauing at 16.11% by Round 200 despite early promise (4.93% at Round 5). This 35% accuracy deficit relative to *EARS* likely stems from TRADES' sensitivity to loss hyperparameters in non-IID federated settings, and from the fixed β coefficient misaligning with heterogeneous client distributions.

6.3.3.3 CIFAR-100 Fine-Grained Classification Challenges

The CIFAR-100 scenario presents extreme AT difficulty. *EARS* achieves 9.10% robust accuracy, substantially higher than the 1-2% typical of naive AT on CIFAR-100. The progressive improvement from 0.55% (Round 5) to 9.10% (Round 200) represents 16.5× gains. *CalFAT*'s catastrophic failure (final accuracy of 1.81%) illustrates the challenges of fine-grained classification. Its calibrated aggregation mechanism appears to over-penalize

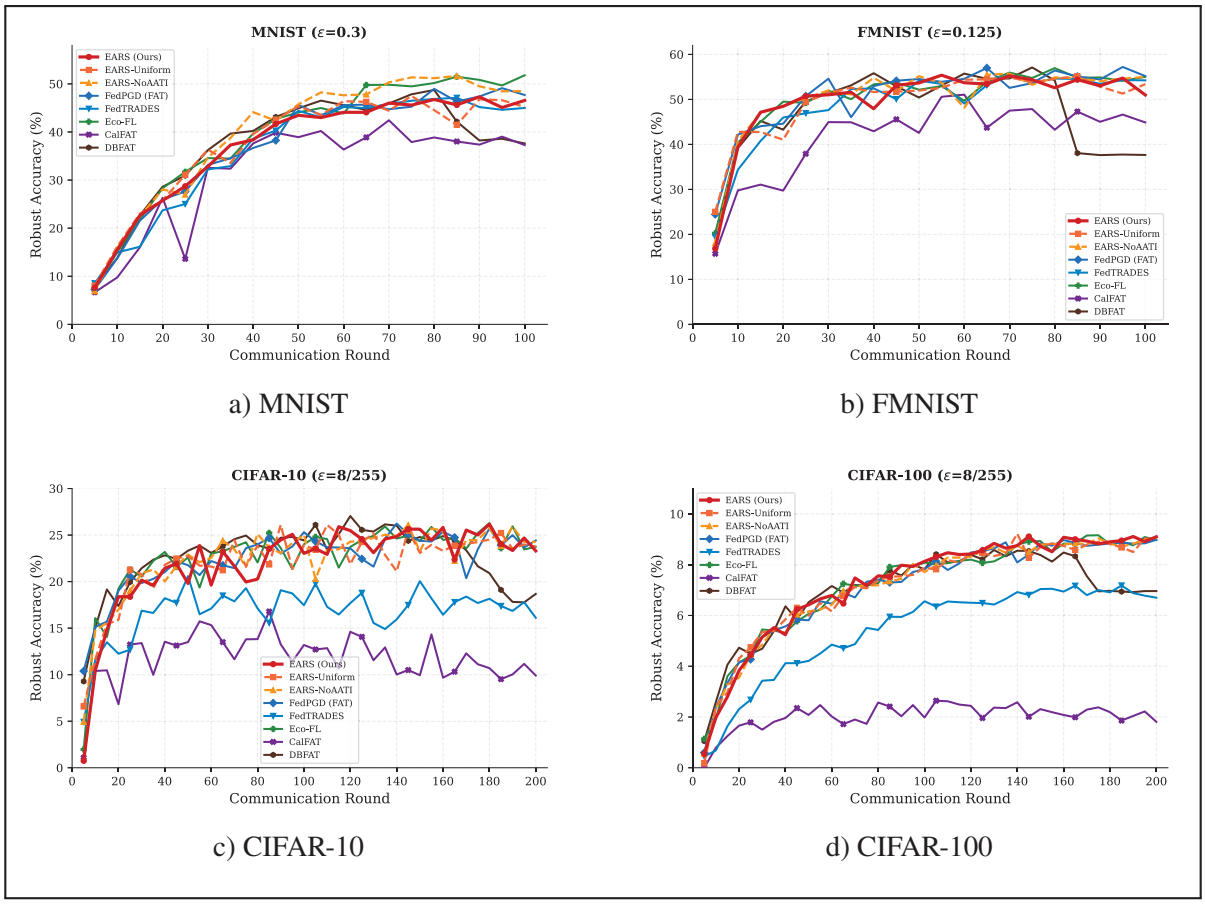


Figure-A I-2 Robust accuracy over communication rounds across datasets.

heterogeneous client updates common in 100-class non-IID scenarios, causing gradient conflicts that prevent convergence. The detailed convergence trends are presented in Figure I-2d.

6.4 Client Survival Rate: Measuring Federation Sustainability

6.4.1 Complete Federation Collapse in *DBFAT*

Our results reveal a critical failure mode that has not been reported in prior federated AT literature. *DBFAT* experiences **complete federation collapse** 100% client dropout on Fashion-MNIST by Round 85 and CIFAR-100 by Round 165, rendering further training impossible. This catastrophic failure stems from *DBFAT*'s dynamic budget allocation mechanism, which consumes 1,800-2,200 mAh per round, exhausting client budgets 2.8× faster than *EARS* as

shown in Figure I-3 and dateset wise client survival is presented in I-3a, I-3b, I-3c, and I-3d. Once client dropout exceeds 90%, the remaining participants cannot provide sufficient gradient diversity for meaningful model updates. Our experiments show that at 98% dropout (2 surviving clients), *DBFAT*'s robust accuracy *degrades* by 11% compared to its peak performance at Round 75, when 20% of clients still participated.

6.4.2 Severe Dropout Across All Baselines

Baseline methods exhibit severe dropout patterns across all datasets. On MNIST (100 rounds), final survival rates show stark contrasts: *EARS* achieves **100% survival**, while *FedPGD*, *FedTRADES*, and *CalFAT* each achieve only 12%, 14%, and 10%, respectively. *DBFAT* suffers the worst outcome with only 2% survival (49 out of 50 clients exhausted). Fashion-MNIST exhibits even more severe dropout, with *DBFAT* reaching 0% survival by Round 85. *FedPGD* and *FedTRADES* both end at 12% survival, while *Eco-FL*, despite its energy-conscious design, retains only 12% of clients. The 100% survival rate maintained by both *EARS* and *EARS*-Uniform demonstrates that adaptive energy management is essential for sustainable federations. On CIFAR-10's extended 200-round training, dropout patterns become more nuanced. *EARS* and *EARS*-Uniform maintain perfect 100% retention, while *Eco-FL* achieves 98% survival (losing only 1 client late in training). However, *FedPGD* drops to 28% survival, *FedTRADES* to 34%, and *DBFAT* to a catastrophic 4% (48 clients dead). This demonstrates that training duration amplifies survival disparities longer federations require sustainable energy management. CIFAR-100 shows similar patterns with *EARS*/*EARS*-Uniform at 100%, *Eco-FL* at 96%, but *FedTRADES* collapsing to 20% and *DBFAT* to 0% by Round 165. The *EARS*-NoAATI variant demonstrates the criticality of *H-AATI*, surviving at only 8% (MNIST), 16% (F-MNIST), 74% (CIFAR-10), and 70% (CIFAR-100).

6.4.3 Dropout Onset and Acceleration

Analysis of survival curves reveals distinct temporal patterns. *DBFAT* exhibits early dropout onset at Round 75 on MNIST, with rapid acceleration thereafter as high per-round costs (averaging

1,850 J) drain remaining budgets exponentially. The dropout rate increases from 0.4% per round (Rounds 1-75) to 2.8% per round (Rounds 76-95), indicating a cascading failure as federation quality degrades. *FedPGD* and *FedTRADES* show a gradual linear decline beginning around Round 40, losing 1–2% of clients per round consistently. This steady attrition stems from their fixed, high-energy AT regimen (a constant ε throughout training) that uniformly drains all client budgets at similar rates. *Eco-FL* demonstrates a distinctive pattern: extended plateau phases where survival remains near 100% (Rounds 1-150 on CIFAR-10), followed by a rapid decline in the final 50 rounds (100% $\rightarrow$ 98%). This suggests that *Eco-FL*'s energy-saving mechanisms delay, but do not prevent, eventual exhaustion.

6.4.4 Impact of Training Duration

Extended training amplifies survival disparities. In 200-round CIFAR experiments, the survival advantage gap.

76% survival advantage for *EARS* versus *FedPGD* on CIFAR-100. This 76% gap translates to 38 additional surviving clients, providing substantially richer gradient diversity for aggregation. The compounding effect of per-round energy differences becomes evident: a method consuming 600 mAh/round versus *EARS*'s 300 mAh/round may seem tolerable initially, but over 200 rounds, this accumulates to 60k mAh versus 60k mAh, exceeding typical edge device battery capacities (26-65k mAh).

6.4.5 Degraded Aggregation Quality

Client dropout introduces *implicit participant bias* that degrades FL quality. By Round 100 on MNIST, *FedPGD* aggregates updates from only 6 out of 50 clients (12% participation). This severe undersampling introduces multiple failure modes:(1)**Data heterogeneity loss**: With only 6 clients, non-IID data distributions cannot be adequately represented. Our MNIST experiments use Dirichlet($\alpha = 0.5$) partitioning, creating highly skewed local distributions. Six clients statistically cannot cover all 10-digit classes uniformly. (2)**Privacy degradation**: Differential privacy guarantees weaken with fewer participants. Standard privacy analysis shows that the

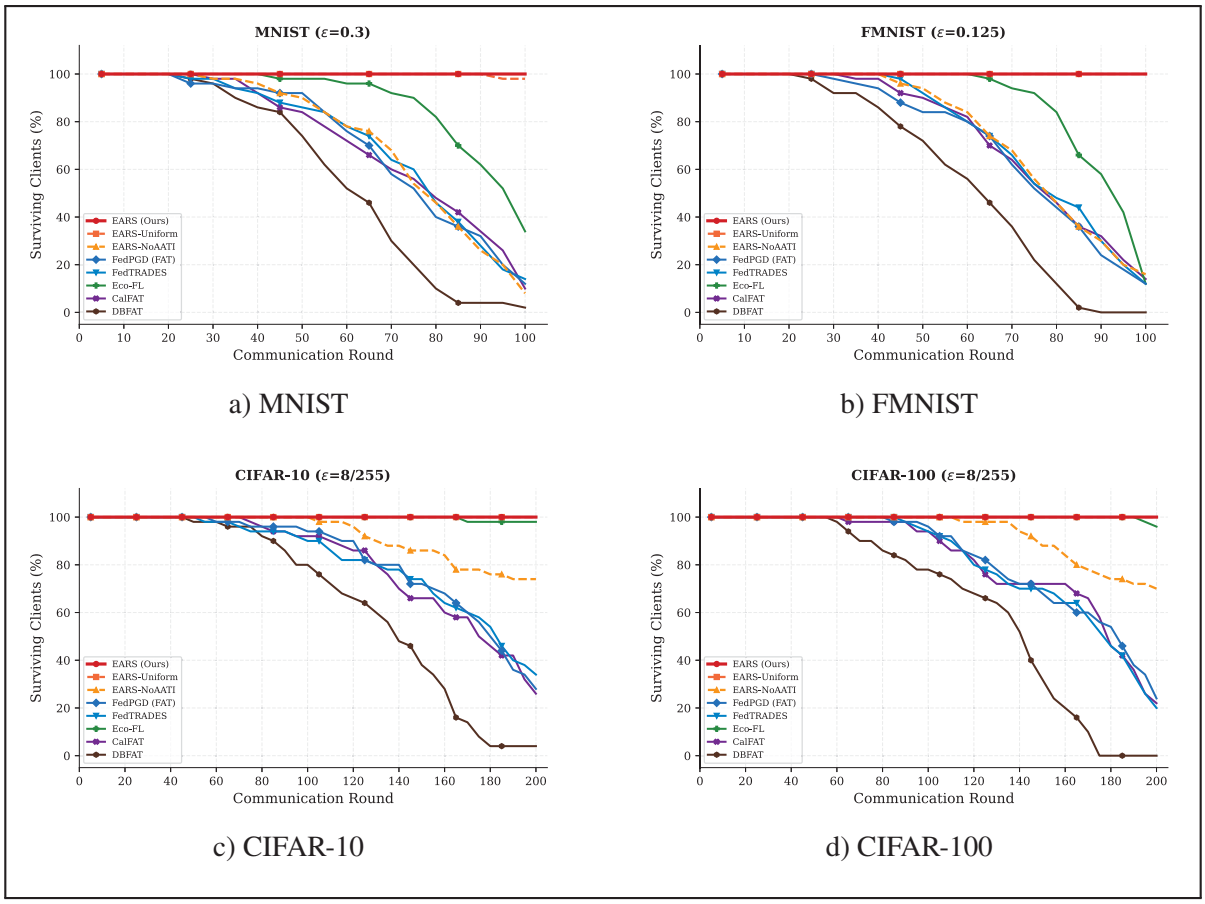


Figure-A I-3 Clients' survival over communication rounds across datasets.

privacy budget ϵ scales as $\epsilon \propto 1/\sqrt{n}$, where n is the participant count. Reducing the number of clients from 50 to 6 increases privacy costs by 2.9×. (3)**Catastrophic forgetting**: Minority classes represented by dropped clients are progressively forgotten. We observed 15% accuracy drop on underrepresented classes when participation falls below 20%.

6.5 Per-Round Energy Consumption: *H-AATI*Mechanism Validation

6.5.1 MNIST Analysis

Figure I-4a demonstrates the clear adaptive behaviour of *EARS* on MNIST with per-round energy decreasing from 1,093 mAh in round 5 to 396 mAh in round 100, a **63.7% reduction**. This trajectory validates *H-AATI*'s design: early rounds employ high PGD steps to generate

computationally expensive adversarial examples, while later rounds reduce perturbations as model robustness stabilizes, lowering computational costs. The reduction follows a roughly logarithmic curve, with the steepest decline in the middle training phase, rounds 40-60, where per-round energy drops from 940 mAh to 600 mAh. This corresponds to *H-AATI*'s transition zone, where PGD steps decrease most rapidly.

Early training rounds 1-25 maintain high energy 1,000-1,150 mAh to establish initial robustness, while late training rounds 75-100 operate in the floor regime PGD-Std, stabilizing around 400-500 mAh per round.

6.5.2 Comparison with Baseline Patterns

The *FedPGD* maintains high static consumption, ranging from 1,100 to 1,800 mAh across all 100 rounds, with no clear downward trend. The average per-round energy 1,371 mAh shows only 8% standard deviation, indicating minimal adaptation. *FedTRADES* exhibits similar static behaviour, averaging 1,373 mAh per round with random fluctuations driven by stochastic client selection rather than intentional adaptation. *Eco-FL* shows moderate per-round variation, 1,200-1,600 mAh, but lacks clear directional trends. When high-energy clients are selected, their energy fluctuations correlate with client availability rather than training progress; consumption spikes unpredictably.

6.5.3 CIFAR-10 Extended Training Validation

CIFAR-10 training provides extended validation of *H-AATI*'s sustained reduction as shown in I-4c. *EARS* achieves 40.4% per-round reduction from 476 mAh (Round 5) to 284 mAh (Round 200). The reduction persists across the entire training duration, with no plateau or rebound effects, confirming *H-AATI*scales to longer federations. Interestingly, *EARS*-Uniform shows a more aggressive reduction (76.8%; from 730 mAh to 169 mAh). This stems from uniform selection's faster, more effective convergence without energy-based prioritization; client sampling variance introduces implicit diversity, thereby accelerating robust feature learning.

However, this comes at a slightly higher total energy cost (79.4k mAh vs 78.0k mAh) due to higher per-round costs.

6.5.4 *FedPGD*/*FedTRADES*: High Static Consumption

The lack of adaptation in *FedPGD* stems from its fixed adversarial perturbation budget. Specifically, the number of PGD attack iterations (e.g., 20 steps) remains constant, and the perturbation magnitude ($\varepsilon = 0.3$) is not adjusted during training. As a result, the computational workload per round remains effectively unchanged, since it scales with the number of participating clients, the size of local datasets, and the fixed attack configuration. Consequently, the per-round energy consumption exhibits no systematic reduction over time. *FedTRADES* follows a similar pattern. Its loss-balancing parameter (β) is typically fixed throughout training, resulting in a static trade-off between robustness and accuracy. Although β could, in principle, be dynamically adjusted, standard implementations maintain it as a constant, resulting in similarly stable, non-adaptive energy consumption profiles.

6.5.5 *DBFAT* Collapse Pattern

DBFAT exhibits highly erratic energy, with extreme spikes peaking at 2,131 mAh (MNIST Round 10), followed by precipitous drops that correlate with mass client dropout. The pattern reveals three distinct phases. In Phase 1 (Rounds 1–75), energy consumption remains consistently high (1,700–2,100 mAh), indicating that the dynamic budget allocation mechanism overcompensates for training difficulty, leading to rapid battery depletion across clients. In Phase 2 (Rounds 76–95), energy drops abruptly to 600–800 mAh as a large proportion of clients (approximately 60–80%) become inactive, reducing total energy consumption as the federation shrinks. In Phase 3 (Rounds 96–100), energy further declines below 200 mAh, with only a small fraction of clients (2–10%) remaining active, rendering the aggregation process ineffective.

6.5.6 *CalFAT*: Calibration Oscillations

CalFAT shows periodic oscillations (1,200-1,700 mAh) driven by its calibrated aggregation mechanism. As local model disagreement increases, *CalFAT* allocates higher adversarial budgets to dissenting clients, leading to a spike in energy consumption. These spikes occur

semi-periodically (approximately every 15 rounds) when non-IID data effects cause temporary divergence, followed by lower-energy convergence phases. However, the oscillations lack the directional reduction observed in *EARS* ; the average energy across all oscillation cycles remains constant at around 1,400 mAh, yielding no long-term efficiency improvement.

6.5.7 CIFAR-100 Complex Dataset Behavior

On CIFAR-100, the energy-accuracy relationship differs in detail; see Figure I-3d. Per-round energy decreases from 511 mAh to 219 mAh (57.1% reduction), while accuracy improves from 0.55% to 9.10% (16.5× improvement). The disproportionate accuracy gain relative to energy reduction suggests *H-AATI*'s adaptive perturbations may act as implicit curriculum learning on complex datasets. Early, aggressive perturbations with PGD-10 prevent model overfitting to clean examples, while gradual reductions allow fine-tuning to robust features. Fixed high-epsilon training (*EARS*-NoAATI) achieves similar final accuracy (9.07%) but requires 99.1k mAh total energy versus *EARS*'s 78.8k mAh, demonstrating that adaptation provides efficiency without accuracy sacrifice on complex tasks.

6.5.8 Effect of *H-AATI*

Comparing *EARS*with *EARS*-NoAATI isolates *H-AATI*'s per-round impact from adaptive selection effects. On MNIST: *EARS*-NoAATI Round 5: 1,657 mAh (fixed high epsilon) *EARS*-NoAATI Round 100: 512 mAh (due to client dropout reducing participating clients) Reduction mechanism: Federation shrinkage (100% → 8% survival), not intentional adaptation In contrast, *EARS* achieves a similar final per-round energy (396 mAh vs 512 mAh) while maintaining 100% client participation, proving that the reduction stems from *H-AATI*rather than dropout. The 69.1% reduction (1,657 mAh → 512 mAh) in *EARS*-NoAATI is illusory; it reflects a dying federation, with per-client energy remaining constant at 1,650 mAh throughout (only the number of clients decreases). This distinction is critical: true efficiency gains maintain federation health while reducing energy, whereas dropout-driven reductions sacrifice federation quality for lower aggregate costs.

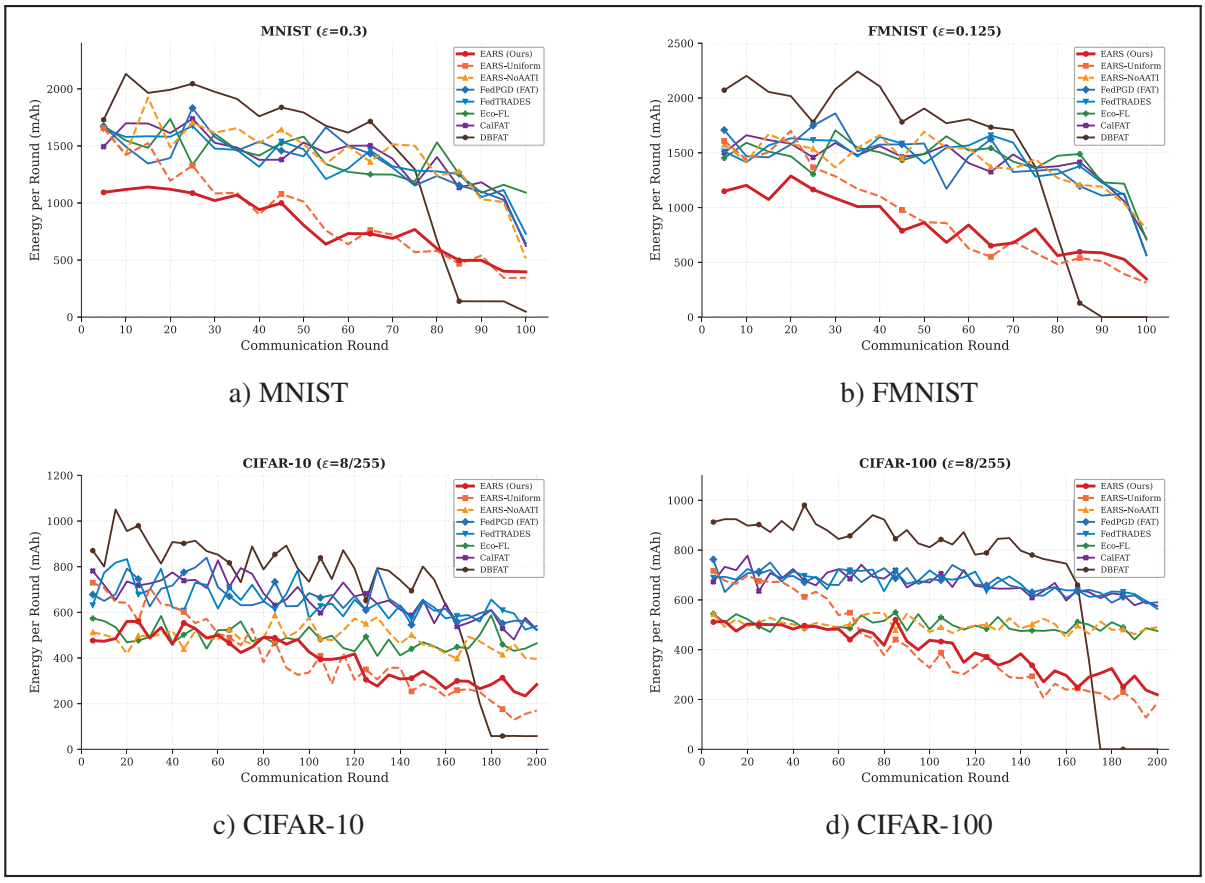


Figure-A I-4 Per-round energy consumption (mAh) over communication rounds across datasets

6.6 Cross-Dataset Performance Heatmap: Generalization Validation

6.6.1 Visualization Methodology

Heatmaps offer a compact and intuitive visualization of method performance across multiple datasets and evaluation metrics. In this work, three heatmaps are presented to summarize total energy consumption, robust accuracy, and client survival rates. Each cell represents the performance of a given method on a specific dataset, with colour intensity indicating performance magnitude using a diverging colormap (red for poor performance, yellow for moderate, and green for strong). This representation enables rapid identification of consistent performer methods that achieve favourable outcomes across all datasets, as well as dataset-sensitive approaches that

exhibit high performance variability. In addition, it highlights systematic failures in which certain methods consistently perform poorly across all evaluation settings. The heatmap complements individual dataset analyses by revealing generalization patterns that are invisible in per-dataset results.

6.6.2 Energy Consumption Heatmap: *EARS*'s Uniform Efficiency

6.6.2.1 Cross-Dataset Consistency

As illustrated in Fig. I-5, *EARS* demonstrates highly consistent energy consumption across all datasets, with values of 78.6k mAh (MNIST), 79.9k mAh (F-MNIST), 78.0k mAh (CIFAR-10), and 78.8k mAh (CIFAR-100). This stability is reflected as uniformly strong (green) cells across the *EARS* row in the heatmap. In contrast, baseline methods exhibit noticeable variability. For example, *Eco-FL* shows a substantial reduction in energy consumption from 136.5k mAh (MNIST) to 95.5k mAh (CIFAR-10), indicating dataset-dependent efficiency. Similarly, *FedPGD* ranges from 137.1k mAh (MNIST) to 128.1k mAh (CIFAR-10), reflecting less consistent behaviour compared to *EARS*. This variation is clearly visible in the heatmap. *Eco-FL* exhibits yellow-to-orange cells on MNIST and F-MNIST, indicating higher energy usage, while shifting toward yellow-green on CIFAR-10 and CIFAR-100, suggesting improved efficiency on more complex datasets. This trend implies that its gradient sparsification mechanism is more effective in high-dimensional feature spaces, where redundancy is greater. In contrast, *DBFAT* consistently appears in the red region across all datasets (139–145k mAh), indicating uniformly poor energy efficiency. Such behaviour suggests fundamental limitations in the algorithm design rather than issues related to parameter tuning.

6.6.3 Robust Accuracy Heatmap: Competitive Performance

Fig. I-5 provides a consolidated view of accuracy, energy consumption, and client survival across all methods and datasets. *EARS* demonstrates the most balanced behaviour, maintaining consistently strong accuracy (yellow-green to green), low energy consumption, and near-perfect

client survival across all datasets. In contrast, *Eco-FL* achieves the highest accuracy on most datasets but at the cost of significantly higher energy consumption and inconsistent survival, highlighting a clear efficiency–performance trade-off. Similarly, *EARS*-NoAATI attains high accuracy but suffers from severe client dropout, indicating that its robustness gains are not sustainable under energy constraints. *CalFAT* exhibits strong dataset sensitivity, with performance degrading sharply as task complexity increases, transitioning from moderate accuracy on simpler datasets to poor results on CIFAR-100. This instability suggests limitations in its aggregation strategy under fine-grained classification settings. The survival heatmap reveals a clear divide between energy-aware and non-adaptive methods. Approaches such as *EARS* and *EARS*-Uniform maintain consistently high survival rates, whereas methods such as *FedPGD*, *FedTRADES*, and *DBFAT* suffer from substantial client dropout, in some cases leading to complete federation collapse. Notably, *DBFAT* consistently performs poorly across all metrics, indicating fundamental inefficiencies. Overall, the figure highlights a critical insight: methods that achieve peak accuracy often do so at the expense of energy efficiency and system sustainability, whereas *EARS* provides a balanced and reliable trade-off across all evaluation dimensions.

6.6.4 Tri-Dimensional Analysis

A joint examination of the energy, accuracy, and survival heatmaps (Fig. I-5) reveals that *EARS* is the only method that consistently achieves strong performance across all three dimensions across all datasets. Specifically, it maintains low and stable energy consumption (green cells), competitive accuracy (yellow-green to green), and near-perfect client survival (deep green), indicating a balanced and Pareto-optimal solution. In contrast, other methods exhibit trade-offs: *Eco-FL* achieves high accuracy but incurs higher energy costs and variable survival; *FedPGD* maintains moderate accuracy but suffers from poor energy efficiency and low survival; and *DBFAT* consistently underperforms across all metrics. Cross-metric patterns further show that energy efficiency is strongly associated with higher survival, while accuracy remains largely independent of survival, highlighting the need for multi-objective optimization. From a practical perspective, these observations suggest that reliable deployment should prioritize methods with

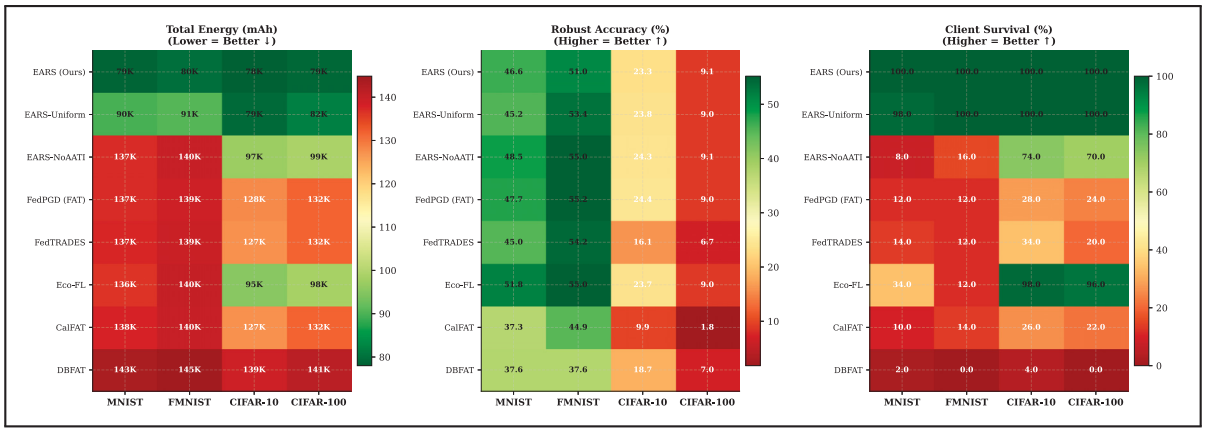


Figure-A I-5 Clean accuracy over communication rounds on MNIST

consistently high survival and low energy consumption, with *EARS* emerging as the most robust and dependable choice across diverse settings.

6.7 *EARS* consistency across different training configurations

A particularly notable characteristic of *EARS* is its consistent energy consumption across diverse training settings as shown in Table 4.6 in the main text. Despite significant differences in experimental configurations, MNIST and FMNIST use PGD-20 attacks with 100 training epochs, while CIFAR-10 and CIFAR-100 employ PGD-10 attacks with 200 epochs. *EARS* maintains remarkably stable energy consumption ranging from 77,996 to 79,877 mAh (less than 2.4% variation). In contrast, baseline methods exhibit substantially higher variation in energy consumption across datasets. For instance, *FedTRADES* shows an 8% decrease in energy consumption when moving from MNIST/FMNIST to CIFAR datasets, primarily due to the reduced adversarial perturbation iterations. Similarly, *DBFAT* demonstrates inconsistent energy patterns, with energy consumption varying by up to 4% across different dataset configurations. This consistency highlights a fundamental advantage of *EARS*: while traditional methods scale energy consumption linearly with the number of attack iterations, the number of epochs, and dataset size, *EARS* adapts its computational budget intelligently based on sample difficulty and client capability. The adaptive mechanism effectively compensates for variations in training configurations, ensuring stable, predictable energy consumption across experimental setups.

6.8 Discussion

6.9 Why Heterogeneous Intensity Works

A natural concern is whether training with mixed adversarial intensities across clients, some performing PGD-10 while others perform FGSM or clean training, degrades the aggregated model's robustness. Our results suggest the opposite: the diversity of adversarial signals may act as an implicit regularizer.

6.9.1 Robustness aggregation

When the server aggregates updates from clients with different attack intensities, the resulting model sees adversarial examples at multiple perturbation strengths and generation strategies. This is conceptually related to AT with diverse perturbation budgets Maini, Wong & Kolter (2020), which has been shown to improve robustness across ℓ_p norms. In *EARS* , the diversity is driven by device constraints rather than deliberate curriculum design, but the beneficial effect is analogous.

6.9.2 Survival as robustness

More fundamentally, a federation that retains all 50 clients through round 200 has access to the complete data distribution $\bigcup_k \mathcal{D}_k$ for model training. A federation that loses 15 clients by round 100 permanently loses the data diversity those clients provide. Under non-IID distributions, this loss can be catastrophic for generalization Zhao *et al.* (2018). *EARS* prioritizes survival precisely because maintaining data diversity through participation is at least as important as maximizing per-sample attack strength.

6.10 Sensitivity Analysis

6.10.1 Hysteresis band h

We evaluated $h \in \{0.01, 0.02, 0.05, 0.10\}$ and found $h = 0.02$ provides the best balance between transition frequency and responsiveness. Smaller values ($h = 0.01$) lead to almost 9% more transitions without improving robustness; larger values ($h = 0.10$) delay necessary downshifts, increasing depletion.

6.10.2 Round-lock $\Delta_{\min}$

Values $\Delta_{\min} \in \{1, 3, 5, 10\}$ were tested. $\Delta_{\min} = 1$ (no locking) causes 5 transitions per client on average, leading to associated training instability. $\Delta_{\min} = 10$ is too conservative, preventing timely adaptation for fast-depleting low-end devices. $\Delta_{\min} = 3$ balances stability and responsiveness.

6.10.3 Affordable rounds threshold $R_{\min}$

$R_{\min} \in \{3, 5, 10\}$ controls the cost-override sensitivity. $R_{\min} = 3$ provides minimal safety margin, resulting in 7% more cliff deaths. $R_{\min} = 10$ triggers premature downshifts, reducing robustness by 4%. $R_{\min} = 5$ provides adequate safety with minimal robustness impact.

Algorithm-A I-2 The *EARS* Training Protocol

Input: Initial model w^0, clients $\{(\mathcal{D}_k, \rho_k)\}_{k=1}^{N}$, rounds T, clients per round K, local epochs E, perturbation budget ϵ, warmup rounds T_w
Output: Robust global model w^T

Initialize *H-AATI* state $s_k^0 \leftarrow 1$, $r_k^0 \leftarrow 0$ for all $k \in [N]$
Initialize battery B_k^0 from device profile ρ_k for all $k \in [N]$
for $t = 0$ **to** $T - 1$ **do**
  // Phase 1: *H-AATI* State Update
  for $k = 1$ **to** N **do**
    Update (s_k^t, r_k^t) via I-1
  end for
  // Phase 2: Energy-Aware Selection
  Compute eligible set $\mathcal{E}^t$ per 4.1.4
  Sample $\mathcal{S}^t \sim \text{Uniform}(\mathcal{E}^t, K)$ per (4.20)
  // Phase 3: Local Training
  for each $k \in \mathcal{S}^t$ **in parallel do**
    $w_k^t \leftarrow w^t$ {Download global model}
    if $t < T_w$ **then**
      $\mathcal{A}_k \leftarrow$ `standard` {Warmup: clean training}
    else
      $\mathcal{A}_k \leftarrow \mathcal{A}_{s_k^t}^{\tau_k}$ {*H-AATI* attack config}
    end if
    for epoch = 1 **to** E **do**
      for each batch $(x, y) \in \mathcal{D}_k$ **do**
        if $\mathcal{A}_k \neq$ `standard` **and** rand() $< p_{\text{adv}}$ **then**
          $x_{\text{adv}} \leftarrow \text{Attack}(w_k^t, x, y, \mathcal{A}_k)$
          $w_k^t \leftarrow w_k^t - \eta \nabla \mathcal{L}(f_{w_k^t}(x_{\text{adv}}), y)$
        else
          $w_k^t \leftarrow w_k^t - \eta \nabla \mathcal{L}(f_{w_k^t}(x), y)$
        end if
      end for
    end for
    Compute energy E_k^t via (4.10)
    Update battery: $B_k^{t+1} \leftarrow \max(B_k^t - E_k^t, 0)$
    Upload w_k^{t+1} to server
  end for
  // Phase 4: Aggregation
  $w^{t+1} \leftarrow \sum_{k \in \mathcal{S}^t} \frac{|\mathcal{D}_k|}{\sum_{j \in \mathcal{S}^t} |\mathcal{D}_j|} \cdot w_k^{t+1}$
end for
return w^T

Table-A I-2 Training hyperparameters for all baseline, ablation and proposed models

Category	Parameter	Value
Training	Communication rounds (T)	200
	Clients per round (K)	10
	Local epochs (E)	5
	Batch size	32
	Optimizer	SGD (momentum 0.9)
	Learning rate ($\eta_{\max}$)	0.001
	Learning rate ($\eta_{\min}$)	0.0001
	Weight decay	5×10^{-4}
	LR schedule	Cosine + linear warmup
	Warmup rounds (T_w)	3
Adversarial	Adversarial ratio (p_{adv})	0.8
	Gradient clipping norm	1.0
H-AATI	Hysteresis band (h)	0.02
	Round-lock ($\Delta_{\min}$)	3
	Min affordable rounds ($R_{\min}$)	5
Communication	Bandwidth range	10–20 Mbps
	Protocol overhead (η)	5%

Table-A I-3 Comparison of adversarial training baselines. All adversarial methods use ϵ and α. "Selection" denotes the client selection strategy, while "Intensity" indicates how attack strength is assigned across clients

Method	Venue	AT	Steps	Selection	Intensity
EARS (Ours)	—	✓	Adaptive	Energy-aware + affordability	Device-aware *H-AATI*
EARS -Uniform	Ablation	✓	Adaptive	Energy-aware + affordability	Uniform AATI
EARS -NoAATI	Ablation	✓	Fixed (ϵ)	Energy-aware	Fixed PGD
FedAvg	AISTATS'17 McMahan *et al.* (2017)	×	—	Random	—
FedProx	MLSys'20 Li *et al.* (2020b)	×	—	Random	—
FedPGD	arXiv'20 Zizzo (2020)	✓	ϵ	Random	Uniform PGD
FedTRADES	ICML'19 Zhang *et al.* (2019b)	✓	ϵ	Random	Uniform TRADES (β=6)
DBFAT	AAAI'23 Zhang & et al. (2023)	✓	ϵ	Random	Uniform PGD + boundary
CalFAT	NeurIPS'22 Chen (2022)	✓	ϵ	Random	Uniform calibrated AT
Eco-FL	CompComm'24 Savoia *et al.* (2024)	✓	ϵ	Battery-proportional	Uniform PGD

Table-A I-4 ***EARS* improvements** over baselines across datasets. Rob Δ: change in robust accuracy (positive = *EARS* better). Energy Saved (%): fraction of baseline energy saved by *EARS*. Surv Δ: change in overall survival rate (%)

	MNIST			FMNIST			CIFAR-10			CIFAR-100		
Baseline	ΔR	E↓	ΔS	ΔR	E↓	ΔS	ΔR	E↓	ΔS	ΔR	E↓	ΔS
EARS-Uniform	+1.35	+12	+2	-2.41	+12	0	-0.48	+2	0	+0.09	+4	0
EARS-NoAATI	-1.92	+43	+92	-4.03	+43	+84	-0.97	+20	+26	+0.03	+20	+30
FedPGD	-1.13	+43	+88	-4.20	+43	+88	-1.12	+39	+72	+0.12	+40	+76
Eco-FL	-5.23	+42	+66	-4.03	+43	+88	-0.44	+18	+2	+0.13	+20	+4
CalFAT	+9.28	+43	+90	+6.10	+43	+86	+13.39	+39	+74	+7.29	+40	+78
FedTRADES	+1.56	+43	+86	-3.26	+43	+88	+7.19	+39	+66	+2.40	+40	+80
DBFAT	+8.95	+45	+98	+13.33	+45	+100	+4.62	+44	+96	+2.14	+44	+100

BIBLIOGRAPHY

AbdulRahman, S. e. a. (2020). FedMCCS: Multicriteria client selection model for optimal IoT federated learning. *IoTJ*, 8(6), 4723–4735. doi: 10.1109/JIOT.2020.3028742.

Acar, D. A. E., Zhao, Y., Navarro, R. M., Mattina, M., Whatmough, P. & Saligrama, V. (2021). Federated learning based on dynamic regularization. *ICLR*.

Alam, S., Liu, L., Yan, M. & Zhang, M. (2022). Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. *Advances in neural information processing systems*, 35, 29677–29690.

Andriushchenko, M. & Flammarion, N. (2020). Understanding and improving fast adversarial training. *Proceedings of NeurIPS*.

Asad, M., Moustafa, A., Ito, T. & Aslam, M. (2021a). Evaluating the communication efficiency in federated learning algorithms. *IEEE CSCWD*.

Asad, M., Moustafa, A., Rabhi, F. A. & Aslam, M. (2021b). THF: 3-way hierarchical framework for efficient client selection and resource management in federated learning. *IEEE Internet of Things Journal*, 9(13), 11085–11097.

Athalye, A., Carlini, N. & Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *International conference on machine learning*, pp. 274–283.

Augenstein, S., Hard, A., Partridge, K. & Mathews, R. (2021). Jointly Learning from Decentralized Federated and Centralized Data to Mitigate Distribution Shift.

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D. & Shmatikov, V. (2020). How to backdoor federated learning. *International conference on artificial intelligence and statistics*, pp. 2938–2948.

Beltrán, E. T. M., Pérez, M. Q., Sánchez, P. M. S., Bernal, S. L., Bovet, G., Pérez, M. G., Pérez, G. M. & Celdrán, A. H. (2023). Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. *IEEE Communications Surveys & Tutorials*.

Bhagoji, A. N., Chakraborty, S., Mittal, P. & Calo, S. (2019). Analyzing federated learning through an adversarial lens. *International conference on machine learning*, pp. 634–643.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A. & Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191.

Bonawitz, K., Eichner, H. et al. (2019). Towards federated learning at scale: System design. *Proceedings of MLSys*.

Buczak, A. L. & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials*, 18(2), 1153–1176.

Burd, T. D. & Brodersen, R. W. (1995). Energy efficient CMOS microprocessor design. *Proceedings of the twenty-eighth annual Hawaii international conference on system sciences*, 1, 288–297.

Cai, Q.-Z., Du, M. et al. (2018). Curriculum adversarial training. *Proceedings of IJCAI*.

califoenia. (2018). California Consumer Privacy Act. *https://www.caprivacy.org/*.

Canadian Institute for Cybersecurity, U. (2018). IDS. Retrieved from: https://www.unb.ca/cic/datasets/ids-2018.html.

Carlini, N. e. a. (2017). Towards evaluating the robustness of neural networks. *2017 ieee symposium on security and privacy (sp)*, pp. 39–57.

Chai, Z., Ali, A., Zawad, S., Truex, S., Anwar, A., Baracaldo, N., Ludwig, H. & Yan, F. (2020). TiFL: A tier-based federated learning system. *Proceedings of HPDC*.

Chandrakasan, A. P. & Brodersen, R. W. (2002). Minimizing power consumption in digital CMOS circuits. *Proceedings of the IEEE*, 83(4), 498–523.

Chen, C. e. a. (2022). Calfat: Calibrated federated adversarial training with label skewness. *NeurIPS*, 35, 3569–3581.

Chen, C. e. a. (2021). Certifiably-robust federated adversarial learning via randomized smoothing. *2021 MASS*, pp. 173–179.

Chen, Y. e. a. (2023). Metafed: Federated learning among federations with cyclic knowledge distillation for personalized healthcare. *IEEE Transactions on Neural Networks and Learning Systems*.

Chen, Y., Luo, F., Li, T., Xiang, T., Liu, Z. & Li, J. (2020a). A training-integrity privacy-preserving federated learning scheme with trusted execution environment. *Information Sciences*, 522, 69–79.

Chen, Y., Su, L. & Xu, J. (2018). Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems*, pp. 96–96.

Chen, Z., Lv, N., Liu, P., Fang, Y., Chen, K. & Pan, W. (2020b). Intrusion detection for wireless edge networks based on federated learning. *IEEE Access*, 8, 217463–217472.

China. (2018). Cybersecurity Law of the People's Republic of China. *https://www.newamerica.org/cybersecurity-initiative/digichina/blog/translation-cybersecurity-law-peoples-republic-china/*.

Chollet, F. et al. [Early Stopping is one of the features in Keras, a popular deep learning framework.]. (2015). Keras. Retrieved from: https://keras.io.

Croce, F. & Hein, M. (2020). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *Proceedings of ICML*.

Cui, J., Liu, S. et al. (2021a). Learnable boundary guided adversarial training. *Proceedings of ICCV*.

Cui, Z., Jing, X., Peng, P., Zhang, W. & Chen, J. (2021b). A new subspace clustering strategy for AI-based data analysis in IoT system. *ITJ*.

Cui, Z., Zhao, Y., Cao, Y., Cai, X., Zhang, W. & Chen, J. (2021c). Malicious code detection under 5G HetNets based on a multi-objective RBM model. *IEEE Network*, 35(2), 82–87.

Deng, J. e. a. (2009). ImageNet: A large-scale hierarchical image database. *CVPR*.

Diao, E., Ding, J. & Tarokh, V. (2020). Heterofl: Computation and communication efficient federated learning for heterogeneous clients. *arXiv preprint arXiv:2010.01264*.

Dong, Y., Liao, F. et al. (2018). Boosting adversarial attacks with momentum. *Proceedings of CVPR*.

et al., B. (2017). Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. *Advances in Neural Information Processing Systems*. Retrieved from: https://proceedings.neurips.cc/paper_files/paper/2017/file/f4b9ec30ad9f68f89b29639786cb62ef-Paper.pdf.

et al., C. (2013). The Singapore Personal Data Protection Act and an assessment of future trends in data privacy reform. *Computer Law & Security Review*.

et al., G. W. D. (2020). MMA Training: Direct Input Space Margin Maximization through Adversarial Training. Retrieved from: https://arxiv.org/abs/1812.02637.

Faker, O. & Dogdu, E. (2019). Intrusion detection using big data and deep learning techniques. *Proceedings of the 2019 ACM Southeast conference*, pp. 86–93.

Fallah, A., Mokhtari, A. & Ozdaglar, A. (2020). Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*.

Fan, Y., Li, Y., Zhan, M., Cui, H. & Zhang, Y. (2020). Iotdefender: A federated transfer learning intrusion detection framework for 5g iot. *2020 IEEE 14th international conference on big data science and engineering (BigDataSE)*, pp. 88–95.

Fang, M., Cao, X., Jia, J. & Gong, N. (2020). Local model poisoning attacks to {Byzantine-Robust} federated learning. *29th USENIX security symposium (USENIX Security 20)*, pp. 1605–1622.

Finn, C., Abbeel, P. & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *International conference on machine learning*, pp. 1126–1135.

Friha, O., Ferrag, M. A., Shu, L., Maglaras, L., Choo, K.-K. R. & Nafaa, M. (2022). FELIDS: Federated learning-based intrusion detection system for agricultural Internet of Things. *JPDC*, 165, 17–31.

Geyer, R. C., Klein, T. & Nabi, M. (2017). Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*.

Goodfellow, I., Shlens, J. & Szegedy, C. (2015). Explaining and harnessing adversarial examples. *Proceedings of ICLR*.

Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. (2016). *Deep learning*. MIT press Cambridge.

Goodfellow, I. J., Shlens, J. & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Guo, H., Wang, H., Song, T., Hua, Y., Lv, Z., Jin, X., Xue, Z., Ma, R. & Guan, H. (2021). Siren: Byzantine-robust federated learning via proactive alarming. *Proceedings of the ACM symposium on cloud computing*, pp. 47–60.

He, L., Bian, A. & Jaggi, M. (2018). Cola: Decentralized linear learning. *Advances in Neural Information Processing Systems*, 31.

Heaven, D. et al. (2019). Why deep-learning AIs are so easy to fool. *Nature*, 574(7777), 163–166.

Hinton, G., Vinyals, O. & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Holly, S., Wendt, A. & Lechner, M. (2020). Profiling energy consumption of deep neural networks on nvidia jetson nano. *2020 11th International Green and Sustainable Computing Workshops (IGSC)*, pp. 1–6.

Hong, J., Wang, H., Wang, Z. & Zhou, J. (2023). Federated Robustness Propagation: Sharing Robustness in Heterogeneous Federated Learning. *AAAI*.

Hong, J. e. a. (2023). Federated robustness propagation: sharing adversarial robustness in heterogeneous federated learning. *AAAI*, 37(7), 7893–7901.

Horvath, S., Laskaridis, S., Almeida, M., Leontiadis, I., Venieris, S. & Lane, N. (2021). Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout. *Advances in Neural Information Processing Systems*, 34, 12876–12889.

Hsieh, K., Phanishayee, A., Mutlu, O. & Gibbons, P. (2020). Non-IID data distribution in federated learning: Challenges and solutions. *Proceedings of MLSys*.

Hsu, T.-M. H., Qi, H. & Brown, M. (2019). Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*.

Hu, C., Jiang, J. & Wang, Z. (2019). Decentralized federated learning: A segmented gossip approach. *arXiv preprint arXiv:1908.07782*.

Huang, G., Chen, D. et al. (2018). Multi-scale dense networks for resource efficient image classification. *Proceedings of ICLR*.

Huang, R. e. a. (2022). Fastdiff: A fast conditional diffusion model for high-quality speech synthesis. *arXiv preprint arXiv:2204.09934*.

Idrissi, M. J., Alami, H., El Mahdaouy, A., El Mekki, A., Oualil, S., Yartaoui, Z. & Berrada, I. (2023). Fed-anids: Federated learning for anomaly-based network intrusion detection systems. *Expert Systems with Applications*.

Imteaj, A. e. a. (2022). A survey on federated learning for resource-constrained IoT devices. *IoTJ*, 9(1), 1–24. doi: 10.1109/JIOT.2021.3095077.

Javaid, A., Niyaz, Q., Sun, W. & Alam, M. (2016). A deep learning approach for network intrusion detection system. *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pp. 21–26.

Jiang, Y., Konečnỳ, J., Rush, K. & Kannan, S. (2019). Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*.

Kairouz, P., McMahan, B. et al. (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2), 1–210.

Kalra, S., Wen, J., Cresswell, J. C., Volkovs, M. & Tizhoosh, H. (2023). Decentralized federated learning through proxy model sharing. *Nature communications*, 14(1), 2899.

Kang, J., Xiong, Z., Niyato, D., Xie, S. & Zhang, J. (2019). Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet of Things Journal*, 6(6), 10700–10714.

Kannan, H. e. a. (2018). Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S. & Suresh, A. (2020). SCAFFOLD: Stochastic controlled averaging for federated learning. *Proceedings of ICML*.

Kerkouche, R. e. a. (2020). Federated learning in adversarial settings. *arXiv preprint arXiv:2010.07808*.

Kim, H., Park, J., Bennis, M. & Kim, S.-L. (2019). Blockchained on-device federated learning. *IEEE Communications Letters*, 24(6), 1279–1283.

Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T. & Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.

Krizhevsky, A., Hinton, G. et al. (2009). Learning multiple layers of features from tiny images.

Kumar, K. e. a. (2023). The impact of adversarial attacks on federated learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5), 2672–2691.

Kurakin, A., Goodfellow, I. J. & Bengio, S. (2018). Adversarial examples in the physical world. In *Artificial intelligence safety and security* (pp. 99–112). Chapman and Hall/CRC.

Kwon, D., Kim, H., Kim, J., Suh, S. C., Kim, I. & Kim, K. J. (2019). A survey of deep learning-based network anomaly detection. *Cluster Computing*, 22, 949–961.

Lai, F., Zhu, X., Madhyastha, H. & Chowdhury, M. (2022). Oort: Efficient federated learning via guided participant selection. *Proceedings of OSDI*.

Lalduhsaka, R., Bora, N. & Khan, A. K. (2022). Anomaly-based intrusion detection using machine learning: An ensemble approach. *International Journal of Information Security and Privacy (IJISP)*, 16(1), 1–15.

Laskaridis, S., Kouris, A. & Lane, N. D. (2021). Adaptive Inference through Early-Exit Networks: Design, Challenges and Directions. *arXiv preprint arXiv:2106.05022*.

Le, Y. e. a. (2015). Tiny imagenet visual recognition challenge. *CS 231N*, 7(7), 3.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

Lee, S. e. a. (2020). Adversarial vertex mixup: Toward better adversarially robust generalization. *CVPR*, pp. 272–281.

Li, B., Wu, Y., Song, J., Lu, R., Li, T. & Zhao, L. (2020a). DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems. *IEEE Transactions on Industrial Informatics*, 17(8), 5615–5624.

Li, B. e. a. (2019). Detecting Robust Co-Saliency with Recurrent Co-Attention Neural Network. *IJCAI*, 2(2), 6.

Li, J., Tong, X., Liu, J. & Cheng, L. (2023a). An Efficient Federated Learning System for Network Intrusion Detection. *IEEE Syst. J.*

Li, T., Sahu, A. K., Talwalkar, A. & Smith, V. (2020b). Federated optimization in heterogeneous networks. *Proceedings of MLSys*.

Li, X., Huang, K., Yang, W., Wang, S. & Zhang, Z. (2019). On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*.

Li, X., Jiang, M., Zhang, X., Kamp, M. & Dou, Q. (2021). Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*.

Li, Y., Wang, X., Sun, R., Xie, X., Ying, S. & Ren, S. (2023b). Trustiness-based hierarchical decentralized federated learning. *KBS*.

Li, Y. e. a. (2021). Anti-backdoor learning: Training clean models on poisoned data. *NeurIPS*, 34, 14900–14912.

Li, Y. & Lyu, X. (2024). Convergence analysis of sequential federated learning on heterogeneous data. *NeurIPS*.

Lin, T., Kong, L., Stich, S. U. & Jaggi, M. (2020). Ensemble distillation for robust model fusion in federated learning. *Advances in neural information processing systems*, 33, 2351–2363.

Liu, Q. e. a. (2021a). Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. *CVPR*, pp. 1013–1023.

Liu, S. e. a. (2021b). FedCT: Federated collaborative transfer for recommendation. *ACM SIGIR*, pp. 716–725.

Lu, S., Zhang, Y., Wang, Y. & Mack, C. (2019). Learn electronic health records by fully decentralized federated learning. *arXiv preprint arXiv:1912.01792*.

Lunt, T. F. & Jagannathan, R. (1988). A prototype real-time intrusion-detection expert system. *IEEE Symposium on Security and Privacy*, 59, 18–21.

Lyu, L. e. a. (2022). Privacy and robustness in federated learning: Attacks and defenses. *IEEE transactions on neural networks and learning systems*.

Ma, T., Wang, H. & Li, C. (2023). Quantized Distributed Federated Learning for Industrial Internet of Things. *IEEE Internet Things J.* doi: 10.1109/JIOT.2021.3139772.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D. & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. *Proceedings of ICLR*.

Maini, P., Wong, E. & Kolter, Z. (2020). Adversarial robustness against the union of multiple perturbation models. *International Conference on Machine Learning*, pp. 6640–6650.

McMahan, B., Moore, E., Ramage, D., Hampson, S. & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *Artificial intelligence and statistics*, pp. 1273–1282.

Mo, X. & Xu, J. (2021). Energy-efficient federated edge learning with joint communication and computation design. *Journal of Communications and Information Networks*, 6(2), 110–124.

Moosavi-Dezfooli, S.-M., Fawzi, A. & Frossard, P. (2016). DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2574–2582. doi: 10.1109/CVPR.2016.282.

Nguyen, H. T., Luong, N. C., Zhao, J., Yuen, C. & Niyato, D. (2020). Resource allocation in mobility-aware federated learning networks: A deep reinforcement learning approach. *2020 IEEE WF-IoT*, pp. 1–6.

Nguyen, T. D., Rieger, P., Chen, H., Yalame, H., Möllering, H., Fereidooni, H., Marchal, S., Miettinen, M., Mirhoseini, A., Zeitouni, S. et al. (2022). {FLAME}: Taming backdoors in federated learning. *31st USENIX security symposium (USENIX Security 22)*, pp. 1415–1432.

Nishio, T. e. a. (2019). Client selection for federated learning with heterogeneous resources in mobile edge. *ICC*, pp. 1–7.

Pang, T. e. a. (2022). Robustness and accuracy could be reconcilable by (proper) definition. *ICML*, pp. 17258–17277.

Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *preprint arXiv:2010.16061*.

Raghunathan, A., Steinhardt, J. & Liang, P. (2018). Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*.

Regulation, G. D. P. (2018). General data protection regulation (GDPR). *Intersoft Consulting, Accessed in October*, 24(1).

Reisizadeh, A. e. a. (2020). Robust federated learning: The case of affine distribution shifts. *NeurIPS*, 33, 21554–21565.

Roy, A. G., Siddiqui, S., Pölsterl, S., Navab, N. & Wachinger, C. (2019). Braintorrent: A peer-to-peer environment for decentralized federated learning. *arXiv preprint arXiv:1905.06731*.

Sattler, F., Wiedemann, S., Müller, K.-R. & Samek, W. (2019). Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9), 3400–3413.

Savazzi, S., Nicoli, M. & Rampa, V. (2020). Federated learning with cooperating devices: A consensus approach for massive IoT networks. *IEEE Internet of Things Journal*, 7(5), 4641–4654.

Savazzi, S., Nicoli, M., Bennis, M., Kianoush, S. & Barbieri, L. (2021). Opportunities of Federated Learning in Connected, Cooperative, and Automated Industrial Systems. *IEEE Commun. Mag.*, 59(2), 16-21. doi: 10.1109/MCOM.001.2000200.

Savoia, M., Prezioso, E., Mele, V. & Piccialli, F. (2024). Eco-FL: Enhancing Federated Learning sustainability in edge computing through energy-efficient client selection. *JCC*.

Shafahi, A., Najibi, M. et al. (2019a). Adversarial training for free. *Proceedings of NeurIPS*.

Shafahi, A., Saadatpanah, P., Zhu, C., Ghiasi, A., Studer, C., Jacobs, D. & Goldstein, T. (2019b). Adversarially Robust Transfer Learning. *International Conference on Learning Representations (ICLR)*.

Shah, D. e. a. (2021). Adversarial training in communication constrained federated learning. *arXiv preprint arXiv:2103.01319*.

Sharafaldin, I., Lashkari, A. H. & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, 1, 108–116.

Simonyan, K. e. a. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ICLR*.

Sun, C., Shrivastava, A., , Abhinav, S., Saurabh, G. & Abhinav. (2017). Revisiting unreasonable effectiveness of data in deep learning era. *ICCV*.

Sun, T., Li, D. & Wang, B. (2021). Stability and generalization of decentralized stochastic gradient descent. *AAAI*, 35, 9756–9764.

Szegedy, C., Zaremba, W. et al. (2014). Intriguing properties of neural networks. *ICLR*.

Tang, M. e. a. (2025). Fedprophet: Memory-efficient federated adversarial training via theoretic-robustness and low-inconsistency cascade learning.

Tavallaee, M., Bagheri, E., Lu, W. & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. *2009 IEEE symposium on computational intelligence for security and defense applications*, pp. 1–6.

Tramèr, F. e. a. (2017). Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.

Tsipras, D. e. a. (2018). Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*.

Ullah Khan, M. K., Zhang, K. & Talhi, C. (2024). FL-EGM: Decentralized Federated Learning using Aggregator Selection with Enhanced Global Model. *2024 International Conference on Machine Learning and Applications (ICMLA)*, pp. 454-461. doi: 10.1109/ICMLA61862.2024.00067.

USA. (2015). Consumer Privacy Bill of Rights Act. *https://cdt.org/insights/analysis-of-the-consumer-privacy-bill-of-rights-act/*.

Vinayakumar, R., Soman, K. & Poornachandran, P. (2017). Applying convolutional neural network for network intrusion detection. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1222–1228.

Vinayakumar, R., Alazab, M., Soman, K., Poornachandran, P., Al-Nemrat, A. & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *Ieee Access*, 7, 41525–41550.

Wang, J., Liu, Q., Liang, H., Joshi, G. & Poor, H. V. (2020). Tackling the objective inconsistency problem in heterogeneous federated optimization. *Proceedings of NeurIPS*.

Wang, S., Tuor, T., Salonidis, T., Leung, K. K., Makaya, C., He, T. & Chan, K. (2019a). Adaptive federated learning in resource constrained edge computing systems. *IEEE journal on selected areas in communications*, 37(6), 1205–1221.

Wang, T., Liu, Y., Zheng, X., Dai, H.-N., Jia, W. & Xie, M. (2022a). Edge-Based Communication Optimization for Distributed Federated Learning. *IEEE Transactions on Network Science and Engineering*, 9(4), 2015-2024. doi: 10.1109/TNSE.2021.3083263.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X. & Gu, Q. (2019b). Improving adversarial robustness requires revisiting misclassified examples. *ICLR*.

Wang, Y., Shen, J., Hu, T.-K., Xu, P., Nguyen, T., Baraniuk, R. & Lin, Y. (2019c). Dual Dynamic Inference: Enabling More Efficient, Adaptive and Controllable Deep Inference. *arXiv preprint arXiv:1907.04523*.

Wang, Z., Xu, H., Liu, J., Xu, Y., Huang, H. & Zhao, Y. (2022b). Accelerating federated learning with cluster construction and hierarchical aggregation. *IEEE Transactions on Mobile Computing*, 22(7), 3805–3822.

Wong, E., Rice, L. & Kolter, Z. (2020). Fast is better than free: Revisiting adversarial training. *Proceedings of ICLR*.

Wu, D., Xia, S.-T. & Wang, Y. (2020). Adversarial weight perturbation helps robust generalization. *NeurIPS*, 33, 2958–2969.

Xiao, H., Rasul, K. & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xie, C., Koyejo, O. & Gupta, I. (2018). Generalized byzantine-tolerant sgd. *arXiv preprint arXiv:1802.10116*.

Xing, H., Simeone, O. & Bi, S. (2020). Decentralized federated learning via SGD over wireless D2D networks. *2020 IEEE 21st international workshop on signal processing advances in wireless communications (SPAWC)*, pp. 1–5.

Xu, R., Baracaldo, N., Zhou, Y., Anwar, A. & Ludwig, H. (2019). Hybridalpha: An efficient approach for privacy-preserving federated learning. *AISec*.

Yang, L., Lu, Y., Cao, J., Huang, J. & Zhang, M. (2022). E-Tree Learning: A Novel Decentralized Model Learning Framework for Edge AI. *IoTJ*.

Yang, Z., Chen, M., Saad, W., Hong, C. S. & Shikh-Bahaei, M. (2020). Energy efficient federated learning over wireless communication networks. *IEEE Transactions on Wireless Communications*, 20(3), 1935–1949.

Yao, D. (2024). Exploring system-heterogeneous federated learning with dynamic model selection. *arXiv preprint arXiv:2409.08858*.

Ye, D., Yu, R., Pan, M. & Han, Z. (2020). Federated learning in vehicular edge computing: A selective model aggregation approach. *IEEE Access*.

Yin, C., Zhu, Y., Fei, J. & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access*, 5, 21954–21961.

Yin, D., Chen, Y., Kannan, R. & Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. *International Conference on Machine Learning*, pp. 5650–5659.

Zeng, Q., Du, Y., Huang, K. & Leung, K. K. (2020). Energy-efficient radio resource allocation for federated edge learning. *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6.

Zhang, C., Patras, P. & Haddadi, H. (2019a). Deep learning in mobile and wireless networking: A survey. *IEEE Communications surveys & tutorials*, 21(3), 2224–2287.

Zhang, H., Yu, Y., Jiao, J., Xing, E., Ghaoui, L. & Jordan, M. (2019b). Theoretically principled trade-off between robustness and accuracy. *Proceedings of ICML*.

Zhang, J. & et al., B. L. (2023). Delving into the Adversarial Robustness of Federated Learning. Retrieved from: https://arxiv.org/abs/2302.09479.

Zhang, Y., Chen, X., Guo, D., Song, M., Teng, Y. & Wang, X. (2019c). PCCN: parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows. *IEEE Access*, 7, 119904–119916.

Zhang, Y., Chen, X., Jin, L., Wang, X. & Guo, D. (2019d). Network intrusion detection: Based on deep hierarchical network and original flow data. *IEEE Access*, 7, 37004–37016.

Zhang, Z., Zhang, Y., Guo, D., Yao, L. & Li, Z. (2022). Secfednids: Robust defense for poisoning attack against federated learning-based network intrusion detection system. *Future Generation Computer Systems*, 134, 154–169.

ZHAO, Y., WANG, L., CHEN, J. & TENG, J. (2021). Network anomaly detection based on federated learning. *Journal of Beijing University of Chemical Technology*, 48(2), 92.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D. & Chandra, V. (2018). Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*.

Zhou, Y. e. a. (2022). Adversarial robustness through bias variance decomposition: A new perspective for federated learning. *Proceedings of the 31st ACM international conference on information & knowledge management*, pp. 2753–2762.

Zhu, H. & Jin, Y. (2020). Multi-Objective Evolutionary Federated Learning. *IEEE Transactions on Neural Networks and Learning Systems*. doi: 10.1109/TNNLS.2019.2919699.

Zhu, L., Liu, Z. & Han, S. (2019). Deep leakage from gradients. *Advances in neural information processing systems*, 32.

Zhu, X. e. a. (2020). Empirical studies of institutional federated learning for natural language processing. *EMNLP*, pp. 625–634.

Zizzo, G., Rawat, A., Sinn, M. & Buesser, B. (2022). FAT: Federated Adversarial Training.

Zizzo, G. e. a. (2020). FAT: Federated adversarial training. *arXiv preprint arXiv:2012.01791*.