CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Emotion recognition is the ability to determine the human emotions such as happiness, anger, sadness, surprise, and disgust through analysis of facial expressions, speech, text, and physiological signals aided by AI and ML, to train and classify these emotion requires a lot of sensitive data, hence the need for private and secure framework. The practical relevance of this privacy-preserving framework extends across several high-impact domains where emotional and behavioral data is both abundant and acutely sensitive. The first is healthcare and mental health monitoring, where wearables and smartphones can continuously capture physiological and affective signals to support the early detection of depression, anxiety, and stress-related disorders; here, Federated Learning enables hospitals, clinics, and individual patients to collaboratively train robust diagnostic models without ever transmitting protected health information beyond the device, thereby satisfying medical confidentiality standards such as HIPAA ..

The second is education and e-learning platforms, particularly in distance-learning contexts where institutions like the National Open University of Nigeria deliver instruction to geographically dispersed students; emotion-aware systems can detect learner engagement, confusion, or frustration during lectures and assessments, allowing adaptive content delivery and timely tutor intervention, all while keeping students' facial and behavioral data securely on their personal devices(Gutiérrez *et al.*, 2025).

The third is intelligent transportation and driver-assistance systems, where in-vehicle cameras and biometric sensors can monitor driver fatigue, distraction, and emotional state to prevent road accidents; Federated Learning allows vehicle manufacturers to refine these safety models across millions of vehicles globally without aggregating drivers' private cabin footage on

centralized servers. Across these domains, FL provides the architectural foundation that reconciles the need for large-scale, generalizable AI models with the non-negotiable demand for individual privacy.

The modern technological landscape is defined by the pervasive presence of Internet of Things (IoT) devices, encompassing everything from personal smartphones and sophisticated wearables to intricate smart sensors embedded in our environments. This expansive network acts as a constant generator of enormous volumes of data, much of which is decentralized and inherently heterogeneous. This vast data reservoir, residing primarily at the network's edge, presents a significant, untapped resource for developing next-generation Artificial Intelligence (AI) and deep learning models for complex analytical and predictive tasks. These applications span critical sectors, particularly in continuous human behavior analysis and Emotion Recognition (ER), where AI can aid in personalized interventions, mental health monitoring, and adaptive human-computer interaction (da Silva Oliveira, 2025).

It is critical to distinguish this process from high-intensity tasks like cryptocurrency mining. Modern edge devices (smartphones and IoT) are equipped with dedicated hardware accelerators, such as Neural Processing Units (NPUs) and DSPs, capable of performing 'On-Device Training' for lightweight models. Unlike mining, which requires continuous peak computation, Federated Learning on edge devices typically runs only when the device is idle, plugged in, and connected to Wi-Fi, thereby preventing battery drain or user disruption.

However, leveraging this multimodal emotional data (such as facial expressions, vocal tones, and physiological signals) through traditional centralized machine learning (ML) models demands that all collected data be aggregated onto a single, powerful server for processing and training. This foundational requirement introduces a series of systemic and regulatory liabilities. Foremost among these are the severe risks to data confidentiality, security, and user privacy, particularly concerning highly sensitive personal information inherent in emotional

and behavioral patterns. Furthermore, the continuous transfer of massive datasets to a central repository creates enormous and unsustainable communication overhead and network congestion.

The confluence of these ethical and infrastructural challenges necessitates a fundamental architectural shift away from centralized computing. This imperative gave rise to Federated Learning (FL), which now serves as the core privacy-preserving paradigm for distributed AI development (Azad, 2025; Aggarwal *et al.*, 2024). FL is a distributed and collaborative training approach fundamentally designed to move the computation to the data. Under the FL model, raw user data remains localized on the individual edge devices (Manzoor, 2025). Instead of transferring sensitive data, clients perform local training on their private datasets and exchange only encrypted model updates, such as parameters, gradients, or weights with a central server for aggregation (Aggarwal *et al.*, 2024). This core principle drastically minimizes the risk of data exposure and aligns the AI training process with stringent regulatory frameworks like GDPR (Lazaros *et al.*, 2024; Azad, 2025).

The scope of FL extends beyond a simple centralized-decentralized dichotomy, incorporating sophisticated architectures necessary to tackle complex, real-world constraints. FL models operate in an iterative cycle of local computation and global aggregation, typically utilizing Federated Averaging (FedAvg) to generate a robust global model from the collective updates. In essence, the move toward FL for emotion recognition is not merely a preference but a structural necessity dictated by the constraints of data privacy laws and the sheer scale of sensitive data generated at the edge. Beyond the theoretical and experimental evaluation of the FL pipeline, the practical value of any AI framework is ultimately validated by its capacity for real-world deployment. This study therefore extends its scope to include the development of a lightweight frontend application that serves as a functional demonstration of the trained global emotion recognition model. This application allows end-users to interact with the system in

real time via webcam input, making the privacy-conscious FL paradigm tangible and accessible to non-technical stakeholders.

1.2 Statement of The Problem

The present day centralized AI and ML emotion recognition system are limited with the following issues, i. compromise of People privacy, the underlying model ii. unfair interpretation by model due to data heterogeneity ii scalability issues. Hence the need for federated learning framework.

1. People's privacy is easily compromised: Sensitive biometric data can be high jacked or leaked, violating trust and laws like GDPR.
2. model corruption: Hidden attacks on the model can cause wrong assessments, which is dangerous in clinical or safety-critical settings.
3. Data Heterogeneity (Non-IID Data) : Data collected from diverse users are inherently non-IID, meaning data distributions vary widely across clients.
4. Scalability: Centralized emotion recognition does not scale in practice. Streaming high-resolution facial data continuously from hundreds or thousands of edge devices, such as smart cameras, wearables, and smartphones, generates excessive communication overhead.

1.3 Aim and Objectives of the Study

The aim of the study is to leverage federated learning for privacy-conscious emotion recognition. The specific objectives are to:

1. analyze how unbalanced, user-specific emotional data (Non-IID) affects model accuracy, and apply standard federated optimization techniques (e.g., FedProx or FedAvg) to stabilize the training process across diverse edge clients.

2. examine the vulnerability of the federated emotion recognition model to standard malicious updates (e.g., basic model poisoning) and implement existing robust aggregation rules to filter out anomalous gradients.
3. design the proposed federated learning framework across multiple client nodes to evaluate practical system metrics, including total training time, communication bandwidth overhead, and client resource consumption.
4. integrate a standard privacy-preserving mechanism into the federated pipeline, and measure the trade-off between the level of noise added for privacy and the resulting accuracy of the emotion classification.
5. develop and deploy a lightweight frontend application that integrates the trained federated global model to perform real-time facial emotion recognition via webcam input, demonstrating the practical viability and end-to-end utility of the proposed privacy-conscious FL framework.

1.4 Significance of the Study

This study holds significant theoretical and practical value across several emerging domains:

1. Practical Demonstration and Stakeholder Accessibility: The frontend application bridges the gap between research simulation and real-world deployment, enabling supervisors, clinicians, and industry partners to directly interact with and evaluate the system's emotion classification capabilities without requiring knowledge of the underlying FL infrastructure.
2. For practitioners and organizations: It provides a tested, transparent framework that complies with privacy regulations (GDPR, HIPAA) while delivering reliable emotion recognition.

3. For the field of AI: It advances robust, decentralized learning methods that handle messy, real-world human data, resist adversarial attacks, and operate efficiently on everyday devices like smartphones.
4. Bolstering Privacy in Sensitive Applications: By rigorously testing and implementing layered privacy-preserving mechanisms (PPMs), the study contributes a robust framework essential for critical applications dealing with sensitive data, such as medical diagnostics (Gulati et al., 2024) and personalized health interventions (da Silva Oliveira, 2025).
5. Improving Model Performance under Heterogeneity: The focus on developing algorithms capable of handling Non-IID data ensures that FL models can converge quickly and accurately, leading to superior generalization and reliable performance despite the diverse and challenging data distribution found in real-world environments (Gulati *et al.*, 2024).

1.5 Scope of the Study

This study is specifically concerned with privacy-conscious facial emotion recognition using Federated Learning (FL). The scope is therefore limited to the development and experimental evaluation of a federated facial expression recognition system in which facial-image data are retained at participating client devices and only model updates are exchanged for collaborative training. The study uses the FER-2013 facial expression dataset as the principal experimental dataset and focuses on seven facial emotion classes: angry, disgusted, fearful, happy, sad, surprised, and neutral. The study does not attempt to develop a general-purpose FL framework for unrelated applications such as human activity recognition, load forecasting, or healthcare monitoring. (McMahan et al., 2017).

The scope covers the following specific areas:

1. Facial Emotion Recognition Model: The study develops and evaluates a lightweight convolutional neural network (CNN) for classifying facial expressions from FER-2013 grayscale images. The model is designed for the facial emotion-recognition task rather than for general-purpose machine-learning applications.
2. Federated Facial Emotion Recognition: The CNN is trained in a simulated federated environment using multiple client nodes. Each client receives a copy of the global model, trains it on its local partition of FER-2013, and returns model parameters or updates to the coordinating server. Standard Federated Averaging (FedAvg) provides the baseline federated training mechanism against which the enhanced framework is evaluated. (McMahan et al., 2017).
3. Non-IID Facial-Expression Data: The study specifically investigates how differences in the distribution of emotion classes across clients affect facial emotion recognition. FER-2013 samples are partitioned using a Dirichlet-based procedure ($\alpha = 0.5$) to simulate heterogeneous client emotion distributions, and FedProx is evaluated as a mechanism for reducing local model drift and stabilizing federated facial-expression training.
4. Security of Federated Emotion Recognition: The study examines the vulnerability of the federated facial emotion-recognition model to malicious client updates using a Sign-Flip model-poisoning attack. LBAAFedAvg is incorporated to identify anomalous client updates before they influence the global emotion-recognition model.
5. Privacy of Facial Data and Model Training: The study incorporates Differential Privacy through gradient clipping and Gaussian noise during local training to examine the privacy-utility trade-off specifically for facial emotion classification. The evaluation

considers how privacy protection affects the accuracy of the FER-2013 emotion-recognition model.

6. Federated-System Performance for Facial Emotion Recognition: The study evaluates model accuracy, loss, convergence, communication overhead, training time, and client resource requirements in the context of federated facial emotion recognition. These metrics are used to determine whether the proposed approach can train and deploy a useful emotion-recognition model under edge-oriented constraints.
7. Real-Time Facial Emotion Inference: The study includes a lightweight frontend application that uses the trained global CNN model for facial emotion prediction from webcam frames and uploaded facial images. The application is limited to demonstrating real-time classification of the seven FER-2013 emotion categories and performs inference on-device without transmitting facial images to the server.

1.6 Limitations of the Study

The understanding and generalizability of the proposed frameworks are subject to several inherent limitations:

1. Real-World vs. Simulation: Many FRL/FL models designed for complex systems (e.g., personalized health interventions) are simulated on the server side rather than implemented directly on resource-constrained user devices. This limits the understanding of practical computational constraints, device variability, and connectivity issues in real-world deployment.
2. Privacy-Accuracy Trade-off: While advanced PPMs are integrated, achieving the necessary balance between rigorous privacy guarantees (e.g., using DP) and maintaining the forecasting accuracy/utility required for clinical or operational applications remains a complex trade-off.

3. Clinical Validation Scale: There is a scarcity of research using large-scale, clinically engaged populations or testing for long-term efficacy in digital health contexts (da Silva Oliveira, 2025). Many validation studies use small, homogeneous samples or participants recruited only to test functionality, limiting the generalizability of findings to actual patient populations.
4. Static Heterogeneity Assumptions: Some algorithmic solutions addressing Non-IID data rely on static heuristics or assumptions about data distribution, which limits their adaptability in truly dynamic environments where client data patterns continuously evolve.

1.7 Definition of Terms

To ensure clarity and consistency throughout this study, the following key terms and concepts are defined within the context of Federated Learning, privacy-preserving methodologies, and distributed artificial intelligence:

Federated Learning (FL): A decentralized machine learning paradigm that enables multiple clients (e.g., smartphones, edge devices, hospitals) to collaboratively train a shared global model while keeping their raw training data locally. Only model updates (such as weights or gradients) are transmitted to a central server.

Global Model: The overarching machine learning model maintained by the central server. It represents the aggregated knowledge of all participating clients and is periodically broadcasted back to them for further local training.

Local Model: A replica of the machine learning model trained autonomously on an individual client's device using only its locally stored, private data.

Edge Devices / Clients: The decentralized computational nodes located at the network's edge where data is generated and local training occurs. Examples include smartphones, wearable sensors, and Internet of Things (IoT) devices.

Federated Averaging (FedAvg): The foundational algorithm in federated learning formulated by Google. It computes the global model by taking the weighted average of the updated parameters submitted by the participating local models.

Non-IID Data (Non-Independent and Identically Distributed): A common scenario in real-world federated networks where the statistical distribution of data on one client significantly differs from that on another client (data heterogeneity).

Differential Privacy (DP): A rigorous mathematical framework for privacy protection that ensures the extraction of insights from a dataset without revealing information about any individual data point. In FL, this is typically achieved by injecting calibrated statistical noise into the local model gradients before transmission.

Secure Aggregation: A cryptographic technique that allows a central server to compute the sum or average of multiple clients' model updates without ever being able to inspect the individual updates, preventing the reverse-engineering of local data.

Homomorphic Encryption (HE): An advanced form of encryption that allows mathematical operations to be performed directly on ciphertexts. The results, when decrypted, match the outcome of the same operations performed on the plaintext.

Model Poisoning / Byzantine Attacks: Malicious activities wherein compromised or adversarial clients send corrupted model updates or manipulated gradients to the central server with the intent of degrading the global model's accuracy or introducing backdoors.

Personalized Federated Learning (PFL): An extension of standard federated learning designed to handle data heterogeneity by adapting or fine-tuning the global model to perform optimally on the specific data distribution of an individual user.

Centralized Learning: The traditional machine learning approach where all raw data from various sources is pooled into a single, central repository (like a cloud server) for model training.

These terms form the fundamental vocabulary necessary for understanding the architectures, algorithms, and security mechanisms discussed in the subsequent chapters of this research.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The emergence of Federated Learning (FL) as a distributed machine learning paradigm has fundamentally transformed the landscape of Artificial Intelligence (AI) research and application, particularly in domains requiring stringent data privacy guarantees (McMahan *et al.*, 2017). As the volume of sensitive personal data generated by Internet of Things (IoT) devices, smartphones, and wearable sensors has grown exponentially (Imteaj et al., 2022), traditional centralized machine learning frameworks have increasingly demonstrated inherent limitations in terms of data security, regulatory compliance, and scalability (Zhang *et al.*, 2021). The intersection of FL with Emotion Recognition (ER) represents one of the most compelling and challenging frontiers in contemporary AI research, as it demands the simultaneous achievement of high model utility, robust privacy preservation, and computational efficiency on resource-constrained edge devices (Chhikara *et al.*, 2021; Nandi and Xhafa, 2021).

This chapter provides a comprehensive review of the existing body of literature that forms the theoretical and empirical foundation for the present study. The review is structured to systematically cover the foundational principles of FL, its evolution and major algorithmic advancements (McMahan et al., 2017), its application in sensitive domains such as healthcare and affective computing (Chen et al., 2020), the various privacy-preserving mechanisms (PPMs) employed to strengthen its security guarantees (Dwork et al., 2006; Vyas et al., 2024), and the challenge of handling heterogeneous, non-identically distributed (non-IID) data prevalent in real-world deployments (Ye et al., 2023). Furthermore, the chapter critically examines the adversarial threats and robustness challenges inherent to FL systems (Fang et al.,

2020; Wei et al., 2023), followed by a review of communication efficiency strategies essential for practical deployment on constrained devices (Khan et al., 2024).

The reviewed literature spans seminal foundational works from 2016 to recent empirical studies published in 2025, reflecting the rapid and evolving nature of this interdisciplinary field (Manzoor, 2025). By systematically identifying the methodological contributions, performance benchmarks, and acknowledged limitations of prior works, this chapter establishes a clear articulation of the research gaps that the present study seeks to address. The ultimate objective is to provide a rigorous scholarly context that justifies the proposed design of a privacy-conscious, robust, and communication-efficient Federated Learning framework for Emotion Recognition on edge devices.

2.2 Related Works

The following review encompasses empirical studies, frameworks, and system designs closely related to the current research. The works are organized thematically across the core pillars of the study: foundational FL algorithms, privacy-preserving mechanisms, security and adversarial robustness, data heterogeneity solutions, emotion and activity recognition, healthcare applications, and communication efficiency.

I. Foundational Federated Learning Algorithms and Surveys

McMahan *et al.* (2017) introduced the groundbreaking Communication-Efficient Learning of Deep Networks from Decentralized Data, which formally established the Federated Averaging (FedAvg) algorithm. FedAvg enables collaborative model training across multiple distributed clients by aggregating locally computed model updates, without requiring raw data transmission to a central server. This seminal work established the standard client-server FL topology and demonstrated the feasibility of training deep neural networks on non-IID, unbalanced data distributed across hundreds to thousands of clients, laying the algorithmic foundation upon which virtually all subsequent FL research is built.

Li *et al.* (2020) proposed FedProx, a generalization and re-parametrization of FedAvg designed to tackle the heterogeneity present in FL systems. By augmenting the local objective function with a proximal term that penalizes deviation from the current global model, FedProx provides convergence guarantees for both statistical and system heterogeneity. Empirical results across synthetic and real-world datasets demonstrated significant improvements in convergence stability and model accuracy compared to FedAvg under non-IID data distributions, making it the de facto algorithm for heterogeneous FL environments.

Karimireddy *et al.* (2020) introduced SCAFFOLD (Stochastic Controlled Averaging for Federated Learning), which addresses client drift in FL through the use of control variates. Client drift occurs when local models diverge from the global objective due to non-IID data, and SCAFFOLD corrects this drift by estimating and subtracting the update drift at each client. The algorithm provides provably faster convergence than FedAvg and FedProx under non-IID conditions, representing a significant theoretical and practical advance in addressing statistical heterogeneity.

Aggarwal, Khullar, and Goyal (2024) provided a comprehensive review of Federated Learning methods, applications, and challenges in privacy-preserving collaborative model training. Their survey systematically categorized FL research into design architectures, fundamental algorithmic challenges including non-IID data, communication overhead, and security vulnerabilities, and diverse application areas including healthcare, mobile computing, and IoT. The review highlighted the critical trade-offs between model utility and privacy guarantees, situating FL as an essential paradigm for next-generation AI deployment in sensitive domains.

Lazaros *et al.* (2024) published a comprehensive survey titled Federated Learning: Navigating the Landscape of Collaborative Intelligence in Electronics. The survey examined the evolution of FL from its foundational algorithms to advanced personalized and clustered variants, covering the breadth of privacy-preserving techniques layered atop base FL models. The

authors critically identified the persisting tension between strong privacy guarantees, particularly those achieved through Differential Privacy, and the utility of the resulting models, a challenge directly relevant to the current study's focus on emotion recognition.

Niu and Wei (2021) proposed FedHybrid, a novel aggregation strategy that unifies the complementary strengths of FedAvg, FedProx, and SCAFFOLD into a single framework. FedHybrid dynamically adjusts the influence of each component based on client performance, thereby handling multiple sources of training instability simultaneously. Experiments on standard benchmarks demonstrated superior convergence and accuracy under heterogeneous conditions compared to any single component algorithm, representing a holistic approach to the non-IID problem.

Pfeiffer *et al.* (2023) conducted a dedicated survey on Federated Learning for computationally constrained heterogeneous devices, published in ACM Computing Surveys. The survey specifically addressed the challenges of deploying FL on resource-limited IoT and mobile devices, cataloguing techniques for model pruning, gradient sparsification, and adaptive local updates. The work is directly relevant to the current study's deployment context on edge devices executing emotion recognition tasks.

II. Privacy-Preserving Mechanisms in Federated Learning

Dwork *et al.* (2006) introduced the formal mathematical framework of Differential Privacy (DP), establishing the epsilon-delta privacy budget as a rigorous mechanism for limiting the influence of any single individual's data on a query or model. The fundamental insight of calibrating Gaussian or Laplacian noise to the sensitivity of a function has since become the cornerstone of privacy-preserving machine learning. Although predating FL, DP's principles are foundational to understanding the privacy-accuracy trade-offs addressed throughout the FL literature and in the current study.

Abadi *et al.* (2016) developed the Deep Learning with Differential Privacy framework, extending DP to the training of deep neural networks using a moments accountant method for tighter privacy budget tracking. Their work demonstrated that it is feasible to train deep learning models with meaningful privacy guarantees, establishing DP-SGD as the standard algorithm for privacy-preserving model training. This contribution is central to the layered PPM architecture considered in the proposed framework.

Bonawitz *et al.* (2017) introduced Practical Secure Aggregation for Privacy-Preserving Machine Learning, proposing a Secure Multi-Party Computation (SMPC) protocol that allows a server to aggregate model updates from thousands of clients without learning the individual contributions. The protocol uses secret sharing and cryptographic masking to guarantee that only the aggregated sum is revealed, providing a strong cryptographic privacy guarantee complementary to DP. This approach is foundational to understanding the SMPC-based aggregation alternatives evaluated in the proposed framework.

Jin *et al.* (2023) proposed FedML-HE, a Homomorphic Encryption (HE)-based optimization scheme for FL that minimizes the size of model updates required for encrypted computation. Their approach achieved a 10x reduction in communication overhead for HE-based federated training of ResNet-50, addressing the primary criticism that HE introduces prohibitive computational overhead. The work demonstrated that selective encryption strategies can balance strong cryptographic privacy guarantees with practical computational efficiency, a balance directly relevant to the current study.

Lazaros *et al.* (2024) in their survey of collaborative intelligence also provided an in-depth analysis of privacy-preserving mechanisms layered on FL, including Differential Privacy, Homomorphic Encryption, and Secure Multi-Party Computation. The survey critically assessed the computational overhead and privacy guarantees of each technique, concluding that layered approaches combining multiple PPMs offer the most robust protection for sensitive

applications. This analysis directly informs the PPM design choices in the proposed framework.

Telkar and Yogi (2025) published a comprehensive review of Differential Privacy with Federated Meta-Learning for Privacy-Preserving Medical IoT in ICCK Transactions on Wireless Networks. The review synthesized recent advances in combining DP with meta-learning strategies to enable rapid model adaptation to new clients while maintaining strict privacy budgets. Their analysis of the epsilon-utility trade-off in medical IoT applications provides direct empirical context for the privacy-accuracy balance addressed in the current emotion recognition study.

Hosseini *et al.* (2025) presented Secure Aggregation in Federated Learning using Multiparty Homomorphic Encryption (MPHE), proposing an extension of classical MPHE methods to enable correct decryption of aggregated model updates even when only a subset of clients is available. The work addresses the practical constraint of client dropout in large-scale FL deployments, a common challenge in mobile and IoT environments. Their compression scheme for gradient vectors is particularly relevant to the communication efficiency objectives of the current study.

Chen *et al.* (2023) introduced a federated learning scheme using AES-EAX encryption for secure information sharing among FL nodes, integrated with a blockchain-based coordination layer. Their work demonstrated that combining symmetric encryption with decentralized ledger technology can provide both data confidentiality and auditability in FL deployments, though at the cost of additional latency. The hybrid cryptographic approach offers a practical alternative to the computationally intensive fully homomorphic encryption schemes for latency-sensitive applications.

Wang *et al.* (2023) proposed PPFLHE, a privacy-preserving federated learning scheme with Homomorphic Encryption for healthcare data published in Applied Soft Computing. The

scheme leverages the Paillier cryptosystem for partial HE to protect model updates in a cross-silo healthcare FL setting, where participating institutions jointly train diagnostic models without sharing patient records. Their empirical evaluation demonstrated that the HE-based scheme achieves comparable diagnostic accuracy to centralized training while providing strong cryptographic privacy guarantees compliant with HIPAA regulations.

III. Adversarial Attacks and Robust Aggregation in FL

Fang *et al.* (2020) conducted the first systematic study of local model poisoning attacks on Byzantine-robust FL aggregation methods, published at USENIX Security. The study demonstrated that sophisticated attackers who have compromised client devices can manipulate local model parameters to maximize global model error, effectively bypassing existing Byzantine-fault-tolerant aggregation rules such as Krum and Coordinate-wise Median. The formulation of poisoning attacks as optimization problems provided the theoretical framework for understanding the stealth attack threats that motivate the robust aggregation mechanisms proposed in the current study.

Blanchard *et al.* (2017) proposed Krum, an early Byzantine-fault-tolerant aggregation rule that selects the single client update with the smallest sum of distances to its nearest neighbors, thus filtering out statistically anomalous updates. While providing theoretical guarantees against a bounded fraction of malicious clients under IID data assumptions, Krum's limitations under non-IID conditions and its inability to leverage all available client updates were acknowledged. Subsequent work in the current study's context builds upon and extends these aggregation principles.

Yin *et al.* (2018) proposed Byzantine-Robust Distributed Learning using coordinate-wise trimmed mean and median aggregation rules, proving optimal statistical error rates for robust distributed learning. Their theoretical analysis established the conditions under which robust aggregation rules can achieve asymptotically optimal performance despite a fraction of

adversarial workers. These results provide the theoretical grounding for the layer-based anomaly detection approach developed in the current framework.

Shi *et al.* (2025) proposed RSDFL, a Robust Similarity-based Dual-gradient Federated Learning model designed to detect and defend against Byzantine attacks even under adversarial-majority conditions in the Journal of Supercomputing. RSDFL incorporates a gradient evaluation method based on class loss changes to determine benign versus malicious update directions, combined with a bidirectional gradient similarity-based credibility metric. Experiments on F-MNIST, E-MNIST, PathMNIST, and CIFAR-10 demonstrated superior Byzantine client identification accuracy compared to state-of-the-art methods, particularly under the challenging adversarial-majority threat model.

Li *et al.* (2023) reviewed security threats in FL and their applications in healthcare in Future Generation Computer Systems. The review systematically catalogued inference and reconstruction attacks, model poisoning and backdoor attacks, and the corresponding defense mechanisms. The analysis highlighted the particularly acute security risks in healthcare FL deployments, where inference of sensitive physiological or diagnostic data from shared model gradients can have severe consequences for individual privacy.

Wei *et al.* (2023) presented Covert Model Poisoning Against Federated Learning in IEEE Transactions on Dependable and Secure Computing, describing algorithm design and optimization for sophisticated, hard-to-detect model poisoning strategies. The study demonstrated that covert poisoning attacks that maintain high validation accuracy on benign tasks while subtly degrading specific target tasks can evade most existing anomaly detection mechanisms. This finding directly motivates the layer-based anomaly aware aggregation approach of the current study.

Fan *et al.* (2024) proposed FedREAS, a Robust Efficient Aggregation and Selection Framework for FL published in ACM Transactions on Asian and Low-Resource Language

Information Processing. FedREAS combines multi-dimensional client evaluation with adaptive selection to filter malicious contributions while minimizing communication overhead. The framework demonstrated robust performance against diverse poisoning strategies while maintaining competitive model accuracy, providing a blueprint for the combined robustness-efficiency objectives pursued in the current study.

Manzoor (2025) investigated securing intelligent networks through federated learning approaches for privacy-conscious anomaly detection in his doctoral dissertation at the University of Glasgow. The study proposed the Federated Random Layer Aggregation (FedRLA) technique that enhances privacy and efficiency by aggregating only a single, randomly chosen neural network layer per communication round, and the Layer-Based Anomaly Aware FedAvg (LBAAFedAvg) for detecting stealth attacks on regression models. The work directly provides the core defensive mechanisms and aggregation strategies adopted in the current study's framework design.

Jeong et al. (2022) proposed FedCC, a robust FL framework against model poisoning attacks published in IEEE Access. FedCC employs cross-client cosine similarity analysis to identify and filter malicious gradient submissions, demonstrating robustness against both targeted and untargeted poisoning strategies including those specifically designed to defeat Krum and coordinate-wise Median aggregation rules. The lightweight computational overhead of FedCC makes it suitable for deployment in resource-constrained environments.

IV. Data Heterogeneity, Non-IID Data, and Personalization in FL

Sahu *et al.* (2018), building on McMahan *et al.* (2017), formalized the FedProx algorithm as a direct response to the non-IID data and system heterogeneity challenges. FedProx introduces a proximal regularization term to local client objectives, constraining the local model from diverging too far from the current global model during each communication round. Extensive experiments with up to 1,000 simulated devices, including 50% and 90% straggler scenarios,

demonstrated that FedProx achieves significantly lower loss and faster convergence than FedAvg under both statistical and system heterogeneity, establishing it as the standard approach for heterogeneous FL.

Chen *et al.* (2023) proposed FedDPMS, a Federated Learning framework operating in Non-IID Settings aided by Differentially Private Synthetic data. By generating differentially private synthetic data samples that bridge the distributional gap across clients, FedDPMS achieves best-in-class performance on FMNIST, CIFAR-10, and CIFAR-100 datasets under severe non-IID conditions. The work demonstrates that augmenting FL with privacy-preserving synthetic data generation is a viable strategy for improving model convergence and accuracy in heterogeneous deployments.

Mu *et al.* (2023) introduced FedProc, a Prototypical Contrastive Federated Learning approach that addresses the non-IID problem by using class-level prototypes to align local model representations with the global feature space. By maximizing the agreement between local client representations and global class prototypes through contrastive learning, FedProc mitigates the client drift problem without requiring data sharing. Experiments demonstrated substantial accuracy improvements over FedAvg, FedProx, and MOON under highly heterogeneous data distributions.

Zhang et al. (2023b) proposed FedALA, an Adaptive Locally Aggregated Federated Learning approach for personalizing client models in FL. FedALA introduces a local aggregation mechanism that selectively incorporates the information needed from the global model for each individual client, enabling personalization without sacrificing global model performance. The approach is particularly relevant to emotion recognition applications where individual variation in emotional expression necessitates personalized model adaptation.

Lu *et al.* (2024) consolidated personalization strategies in FL, including local head architectures, meta-learning initializations, and split batch normalization, relating each to

specific heterogeneity phenomena observed in multi-institution deployments. The survey provided a comprehensive taxonomy of personalized FL (pFL) methods and evaluated their suitability for different heterogeneity profiles, offering a practical guide for selecting personalization strategies in real-world FL deployments such as the distributed emotion recognition context.

Ye *et al.* (2023) published a survey on Heterogeneous Federated Learning: State-of-the-art and Research Challenges in ACM Computing Surveys. The survey systematically reviewed methods for addressing both statistical heterogeneity (non-IID data distributions) and model heterogeneity (varying model architectures across clients). The analysis of the interplay between heterogeneity and privacy, and its impact on convergence guarantees, provides essential theoretical grounding for the algorithmic design choices in the current study.

Gulati *et al.* (2024) developed a privacy-preserving collaborative FL framework for detecting retinal diseases across distributed medical institutions in IEEE Access. The framework employs FedAvg with differential privacy to train a high-accuracy retinal disease classifier across multiple hospital datasets that are inherently non-IID due to varying patient demographics, imaging protocols, and disease prevalence. The work establishes the standard for achieving clinically relevant AI diagnostics on strictly regulated datasets without violating data confidentiality, directly informing the current study's design for handling non-IID emotional data.

Chhikara *et al.* (2021) proposed a decentralized FL framework for Human-Computer Interaction (HCI) in IoT applications in IEEE Internet of Things Journal. The framework specifically addresses the challenge of decentralized emotion data collected through heterogeneous IoT sensor modalities, employing a personalized model averaging strategy to handle client-level distribution discrepancies. The work demonstrated that personalized FL achieves significantly higher emotion classification accuracy than global model approaches

when client data is highly non-IID, directly validating the personalization strategies considered in the current study.

V. Emotion Recognition and Affective Computing

Smitha Vas *et al.* investigated Micro Expression Detection Using Federated Learning, examining the feasibility and performance of computing fine-grained facial expression analysis on distributed data without centralizing raw facial imagery. The work validated that FL-based micro-expression detection can achieve competitive accuracy compared to centralized models while preserving the privacy of sensitive facial data. This study is highly connected to the current study's focus on the FER-2013 dataset and validates the necessity of privacy-preserving approaches for facial expression analysis.

Salman and Busso (2022) presented Privacy Preserving Personalization for Video Facial Expression Recognition Using Federated Learning at the International Conference on Multimodal Interaction. The work specifically addressed the personalization challenge in facial expression recognition, demonstrating that individual differences in emotional expression necessitate client-specific model adaptation. Their FL-based personalization approach achieved higher video FER accuracy than global model approaches while maintaining strong privacy guarantees for the sensitive facial video data.

Chang *et al.* (2022) proposed a Robust Federated Learning framework against adversarial attacks for Speech Emotion Recognition in an arXiv preprint. The study demonstrated that speech emotion recognition FL systems are particularly vulnerable to Byzantine poisoning attacks due to the high dimensionality and variability of audio feature representations. Their robust aggregation approach, combining cosine similarity filtering with FedAvg, provided effective defense against both targeted and untargeted audio-domain poisoning attacks.

Benouis, Andre, and Can (2024) proposed a framework for Balancing Between Privacy and Utility for Affect Recognition Using Multitask Learning in Differential Privacy-Added FL

Settings, published in JMIR Mental Health. The framework integrated Multi-Task Learning (MTL) with DP and FL to simultaneously perform stress recognition and identity recognition from wearable physiological sensors. Results demonstrated 90% emotion recognition accuracy while limiting re-identification accuracy to 47%, providing a compelling empirical demonstration of achievable privacy-utility trade-offs in affective computing applications.

Simić et al. (2024) published a study on Enhancing Emotion Recognition through Federated Learning: A Multimodal Approach with Convolutional Neural Networks in Applied Sciences. The study developed a privacy-preserving lightweight CNN-based audiovisual emotion recognition model deployable on constrained edge devices, demonstrating that FL improved classification accuracy by approximately 2% compared to local-only training while providing competitive performance against centralized multimodal baseline models. The study directly validates the federated CNN architecture approach adopted in the current study.

Nandi and Xhafa (2021) proposed a federated learning method for real-time emotion state classification from multi-modal streaming data, addressing the challenge of continuous emotion monitoring from heterogeneous sensor streams in a privacy-preserving manner. Their stream-based FL approach enabled continuous model updating without compromising data privacy, demonstrating applicability to real-time emotion recognition in dynamic edge computing environments.

Srihitha et al. (2024) investigated Federated Scaling of Pre-trained Models for Deep Facial Expression Recognition using the FER2013 dataset in Springer proceedings. The study demonstrated that federated fine-tuning of pre-trained deep learning models achieves high FER accuracy while significantly reducing communication overhead compared to training from scratch in each FL round. The work provides practical insights into the initialization and scaling strategies applicable to the CNN-based FER model in the current study.

Goodfellow *et al.* (2013) introduced the FER-2013 dataset at the International Conference on Machine Learning, establishing the benchmark dataset comprising 35,887 grayscale 48x48 facial expression images across seven emotion categories (angry, disgusted, fearful, happy, sad, surprised, neutral). Human-level performance on the dataset is estimated at 65.5%, establishing a meaningful performance ceiling for automated systems. FER-2013 has since become the primary benchmark for evaluating facial emotion recognition algorithms and forms the validation dataset for the current study.

El Boudouri and Bohi (2023) proposed EmoNeXt, an adapted ConvNeXt architecture specifically optimized for Facial Emotion Recognition, evaluated on the FER2013 dataset in IEEE MMSP 2023. EmoNeXt integrates depthwise separable convolutions with multi-scale feature extraction and achieves state-of-the-art accuracy on FER2013, demonstrating the superiority of modern CNN variants over traditional VGG and ResNet architectures for the FER task. The architectural insights from EmoNeXt inform the lightweight CNN design choices in the current federated emotion recognition model.

Zhang *et al.* (2024) proposed FedMultiEmo, a Real-Time Emotion Recognition framework via Multimodal Federated Learning for automotive applications evaluated at IEEE ICECCME 2025. FedMultiEmo evaluated a federated CNN on FER2013 achieving 77% accuracy, and a multimodal fusion model achieving 87% accuracy, matching centralized baselines while keeping all raw data local. The system converged in 18 rounds with a per-client memory footprint below 200MB, demonstrating the practical viability of federated multimodal emotion recognition in resource-constrained deployments.

VI. Federated Learning in Healthcare and Sensitive Data Applications

da Silva Oliveira (2025) investigated Federated Reinforcement Learning for Personalised Mental Health Interventions, applying FL principles to train adaptive digital therapy models on sensitive behavioural data while preserving patient privacy. The study highlighted the

critical privacy-accuracy trade-off in tracking sensitive mental health states, demonstrating that overly strong DP noise can degrade the personalization utility necessary for effective clinical intervention. The work directly motivates the layered PPM approach and the consideration of DP noise calibration in the current study's framework.

Rajesh (2025) proposed an Adaptive Edge-Federated AI Framework for Contactless Menstrual Health Prediction Using Multimodal Physiological Intelligence, published in MethodsX. The framework employs contactless biosensing modalities including radar-based physiological signal detection integrated into a federated learning pipeline that preserves the extreme sensitivity of menstrual health data. The successful demonstration of FL on highly personalized biological metrics proves the viability of decentralized architectures for sensitive health data and directly supports the methodology of the current study.

Sandhu *et al.*, (2023) reviewed medical imaging applications of FL in Diagnostics, examining how FL has been applied to tasks including cancer detection, diabetic retinopathy classification, brain tumour segmentation, and COVID-19 diagnosis across distributed hospital networks. Their analysis confirmed that FL achieves comparable or superior diagnostic accuracy compared to centralized models when appropriate non-IID mitigation strategies are employed, establishing FL as the clinical standard for privacy-preserving medical AI.

Vankayalapati *et al.* (2025) examined Federated Edge Computing for Privacy-Preserving healthcare in IJCTT, specifically addressing the combination of FL and edge computing as breakthrough technologies for patient privacy compliance. Their framework demonstrated that distributing aggregation tasks to mobile edge nodes significantly reduces latency and communication overhead while maintaining GDPR and HIPAA compliance in patient vitals monitoring applications.

Aminifar *et al.* (2024) proposed a Privacy-Preserving Edge Federated Learning framework for intelligent mobile-health systems in Future Generation Computer Systems, implemented on

Amazon AWS and evaluated for epilepsy seizure detection using wearable sensors. The framework specifically addressed the stringent resource constraints of wearable IoT devices, including limited computing capacity, communication bandwidth, memory, and battery lifetime. Their experimental results demonstrated feasible privacy-preserving FL deployment on highly resource-constrained devices, providing a direct practical benchmark for the edge deployment objectives of the current study.

Matta and Bolli (2025) presented Federated Learning for Privacy-Preserving Healthcare Data Sharing: Enabling Global AI Collaboration in the American Journal of Scholarly Research and Innovation. The paper comprehensively reviewed the state-of-the-art in FL-based healthcare data collaboration, addressing cross-institutional model training for disease prediction, medical imaging analysis, and patient monitoring. The work validated FL as the primary mechanism for enabling global AI collaboration in healthcare while adhering to data protection regulations.

Ali *et al.* (2025) investigated Multi-Sensor Wearable-Based Sleep Stage Classification Using Federated Learning for Enhanced Privacy in IEEE Access. The study employed a multi-modal FL approach combining EEG, ECG, and accelerometer data from wearable devices to classify sleep stages without centralizing sensitive physiological recordings. The cross-device FL architecture demonstrated robust performance across heterogeneous device types and user populations, providing a direct methodological template for the distributed physiological data scenarios addressed in the current study.

Li et al. (2023) reviewed FL applications in healthcare from a smart healthcare systems perspective, addressing the integration of FL with IoT devices, wearables, and remote monitoring technologies for real-time predictive analytics. The review evaluated key challenges including adversarial attacks, data poisoning, and model inversion risks specific to healthcare FL deployments, and assessed privacy-preserving technology advances including blockchain integration and hybrid DP-HE approaches published between 2023 and 2024.

Bhattacharya *et al.* (2025) proposed a secured remote patient monitoring framework for IoMT ecosystems in Scientific Reports. The framework employed FL with secure gradient aggregation to enable continuous monitoring of patient vitals across distributed IoMT devices while preventing unauthorized access to sensitive health data. The work addressed the specific challenges of resource-heterogeneous IoMT environments, demonstrating scalable FL deployment across devices with widely varying computational capabilities.

VII. Human Activity Recognition and Edge AI Applications

Jamil *et al.* (2024) developed a Swarm Learning empowered federated deep learning framework for seamless smartphone-based activity recognition in IEEE Transactions on Consumer Electronics. The framework employs blockchain-based swarm coordination combined with FL to address data confidentiality challenges in distributed Human Activity Recognition (HAR) using smartphone inertial sensors. By replacing the centralized aggregation server with a decentralized peer-to-peer network, the approach eliminates single points of failure and trust, directly addressing the system heterogeneity and edge deployment challenges most relevant to the current study's context.

Palermo (2024) conducted doctoral research on Recognizing Human Activities in a Privacy-Preserving Way at Politecnico di Torino, developing comprehensive FL frameworks for HAR that maintain user privacy on mobile and wearable platforms. The study examined the specific challenges of non-IID sensor data in HAR applications and evaluated multiple personalization strategies for improving classification accuracy across diverse user populations, providing direct methodological insights applicable to the emotion recognition domain.

Chen *et al.* (2020) proposed FedHealth, the first federated transfer learning framework for wearable healthcare. FedHealth performs data aggregation through FL and then builds personalized models by transfer learning, achieving accurate and personalized healthcare without compromising privacy and security. Experiments demonstrated a 5.3% accuracy

improvement for wearable activity recognition compared to traditional federated methods, establishing the template for combining FL with transfer learning for personalized health applications.

Chen *et al.* (2021) extended FedHealth with FedHealth 2, incorporating weighted federated transfer learning via batch normalization for personalized healthcare. FedHealth 2 addressed the non-IID data challenge that limited FedHealth's performance, using batch normalization statistics as lightweight personalization parameters. The work demonstrated that normalizing distributional differences through batch statistics offers a computationally efficient personalization strategy applicable to resource-constrained wearable devices.

Imteaj *et al.* (2022) published a comprehensive survey on Federated Learning for Resource-Constrained IoT Devices in IEEE Internet of Things Journal. The survey examined the specific challenges of deploying FL on devices with severely limited battery life, computational capacity, and memory, cataloguing solutions including model pruning, gradient sparsification, client selection, and asynchronous aggregation. The extensive treatment of resource-constrained FL deployment provides the technical context for the efficiency optimization objectives of the current study.

Gökcen and Boyaci (2024) proposed a Privacy-Preserving Real-Time Action Detection framework for intelligent vehicles using FL-based Temporal Recurrent Networks in Electronics. The framework employs FL combined with LSTM networks to recognize driver actions and behaviors from in-vehicle camera streams without centralizing sensitive video data. The real-time inference requirements and edge deployment constraints of intelligent vehicle systems mirror those of the emotion recognition framework in the current study.

VIII. Communication Efficiency and Model Compression in FL

Han *et al.* (2016) introduced Deep Compression, combining network pruning, trained quantization, and Huffman coding to compress deep neural networks by 35-49x without loss

of accuracy. The work demonstrated that storage and communication requirements for deep neural networks could be dramatically reduced through structured compression, establishing the theoretical and practical foundation for model compression techniques subsequently applied to FL. The Deep Compression principles directly inform the communication efficiency strategies evaluated in the proposed framework.

Ren *et al.* (2023) proposed TLAQC, a Two-Layer Accumulated Quantized Compression algorithm for communication-efficient FL published in Scientific Reports. TLAQC employs a revised quantization method with zero-value correction to minimize ineffective quantization and average quantization errors, combined with an adaptive threshold mechanism that reduces gradient upload frequency. Experimental results demonstrated compression of uploaded traffic to 6.73% of full-precision FedAvg while increasing accuracy by 1.25%, establishing aggressive compression as a viable strategy for bandwidth-limited FL deployments.

Sattler *et al.* (2020) proposed Sparse Binary Compression for communication-efficient federated learning, combining top-k gradient sparsification with binary quantization to achieve significant reductions in both upload and download communication costs. The approach maintained competitive model accuracy on standard benchmarks while reducing communication overhead by over 99% in some configurations, demonstrating that extreme compression ratios are achievable without catastrophic accuracy degradation.

Khan *et al.* (2024) presented Deep Compression for Efficient and Accelerated Over-the-Air Federated Learning in IEEE Internet of Things Journal. Their approach combines pruning and quantization-aware training to shrink model sizes by up to 80% while maintaining accuracy close to uncompressed models, with 30% pruning combined with INT8 quantization achieving compression ratios up to 87% at 93.0% accuracy. The work provides directly applicable compression benchmarks for the CNN models targeted in the current study's edge deployment.

Yu *et al.* (2023) proposed Heterogeneous Federated Learning Using Dynamic Model Pruning and Adaptive Gradient at CCGrid 2023. Their framework dynamically adjusts the pruning rate for each client based on available computational resources, enabling participation of highly heterogeneous devices in a single FL training round. The adaptive pruning strategy ensures that resource-constrained clients can contribute meaningful model updates without being excluded from participation, directly addressing the device heterogeneity challenge in the current study's edge deployment scenario.

proposed the Adaptive Single Layer Aggregation (ASLA) technique as part of the broader FedRLA framework, transmitting only a single, strategically chosen neural network layer per communication round to minimize communication overhead. The random layer selection mechanism also provides an implicit privacy benefit by limiting the information available to adversaries observing the communication channel. ASLA demonstrates that dramatic reductions in communication cost can be achieved without sacrificing convergence quality, directly informing the efficiency optimization approach of the current study.

Yu, Munoz, and Jannesari (2024) explored Federated Foundation Models (FFMs) for collaborative learning of complex large-scale model architectures at LREC-COLING 2024. The study demonstrated that FL principles can be extended to foundation models with billions of parameters through parameter-efficient fine-tuning strategies, enabling privacy-preserving adaptation of pre-trained models to distributed domain-specific tasks. The work indicates the trajectory of scalable edge AI and connects to the current study's aims of optimizing computational and communication efficiency for deploying complex CNNs on edge devices.

IX. Cybersecurity, IoT Security, and Anomaly Detection Applications

Kalejaiye (2025) investigated Federated Learning in Cybersecurity for privacy-preserving collaborative models for threat intelligence across geopolitically sensitive organizational boundaries in the International Journal of Advanced Research in Publication Review. The study

demonstrated that FL enables effective cross-organizational cybersecurity collaboration without revealing proprietary security postures or sensitive threat intelligence, directly validating the application of FL to security-critical domains under strict confidentiality constraints.

Vyas *et al.* (2024) published a comprehensive survey on Privacy-Preserving Federated Learning for Intrusion Detection in IoT Environments in IEEE Access. The survey examined the application of FL to network intrusion detection across heterogeneous IoT deployments, evaluating privacy-preserving aggregation methods, adversarial attack vectors specific to IoT FL, and defense mechanisms. The analysis of IoT-specific constraints including device heterogeneity, intermittent connectivity, and energy limitations provides directly applicable insights to the current study's edge deployment context.

Mahdipour *et al.* (2025) proposed leveraging Complex-Valued Federated Learning for Accurate and Privacy-Respectful Threat Detection Based on Millimeter-wave Imaging in IEEE Access. The work extended FL to the complex-valued domain to improve threat detection accuracy on millimeter-wave sensor data while maintaining privacy guarantees through federated training. The multi-sensor fusion approach and privacy-preserving architecture are methodologically relevant to the multimodal emotion recognition systems considered in the current study.

Li *et al.*, (2025) presented FedPhishLLM, a Privacy-Preserving and Explainable Phishing Detection mechanism using Federated Learning and Large Language Models in the Journal of King Saud University. The work demonstrated that combining FL with transformer-based language models enables effective phishing detection without centralizing sensitive network traffic data, achieving high detection accuracy while maintaining user privacy. The integration of explain ability mechanisms into the FL pipeline is particularly relevant to the transparent AI requirements of sensitive affective computing applications.

Manzoor (2025), in his doctoral work on securing intelligent networks, specifically analyzed the Federated Communication Round Attack (Fed-CRA), a novel stealth attack targeting FL systems for regression tasks. Fed-CRA exploits the aggregation protocol to introduce subtle model degradation across communication rounds without triggering conventional anomaly detection mechanisms. The detailed attack taxonomy and the proposed FedRLA defense mechanism provide the specific threat model and countermeasure framework directly implemented in the current study's security layer.

X. Additional Related Works: Surveys, Frameworks, and Emerging Applications

Alzamil (2025) investigated Federated Deep Learning for Scalable and Explainable Load Forecasting in Privacy-Conscious Smart Cities published in IEEE Access. The work demonstrated that FL-based deep learning models can achieve accurate energy load forecasting across distributed smart city infrastructure without centralizing sensitive consumption data, incorporating explainability modules to enable interpretable predictions. The privacy-preserving smart city context and explainability requirements are directly analogous to the transparent affective computing scenarios addressed in the current study.

Mistry *et al.* (2024) presented a Federated Learning-Based Architecture for Personalized Next Emoji Prediction for Social Media Comments in IEEE Access. While focused on natural language processing, the study demonstrated that FL effectively enables personalized next-item prediction from distributed social media data, revealing the individual's communication patterns without centralizing potentially sensitive conversational content. The personalization mechanisms and federated optimization strategies are directly transferable to the emotion recognition personalization objectives of the current study.

Sharma *et al.* (2024) proposed Gas Detection and Classification Using Multimodal Data Based on Federated Learning in Sensors. The study demonstrated that FL enables accurate gas classification from distributed multi-sensor deployments without requiring data centralization,

achieving high classification accuracy across heterogeneous sensor configurations. The multimodal data fusion approach and cross-device deployment strategy provide methodological insights applicable to the multimodal physiological signal processing in emotion recognition.

Azad (2025) published Federated Learning for Autonomous Vehicles: Privacy-Preserving Edge AI, examining the application of FL to training perception and decision-making models for autonomous driving without centralizing sensitive driving data. The work comprehensively reviewed FL architectures for edge AI deployment in time-critical applications, addressing communication latency, model compression, and adversarial robustness requirements. The edge AI deployment challenges and robust aggregation requirements are directly analogous to those in the emotion recognition framework proposed in the current study.

Lazaros *et al.* (2024) in their comprehensive survey on FL and collaborative intelligence additionally addressed blockchain-integrated FL systems, which use distributed ledger technology to enhance the security, auditability, and verifiability of model update exchanges. The analysis demonstrated that blockchain integration can effectively replace the trusted central server with a decentralized trust mechanism, eliminating the single point of failure inherent in standard FL architectures. This architectural variant is considered as a future extension of the current study's framework.

Aggarwal *et al.* (2024) in their comprehensive FL review also addressed the intersection of FL with reinforcement learning and meta-learning frameworks, demonstrating that FL can be extended beyond supervised classification to enable distributed policy optimization and rapid model adaptation. The meta-learning FL variants are particularly relevant to the dynamic, personalized nature of emotion recognition applications, where rapid adaptation to new users or emotional contexts is a critical capability.

Gutiérrez et al. (2025) published a study on Implementing Federated Learning for Privacy-Preserving Emotion Detection in Educational Environments, demonstrating the deployment of FL-based emotion recognition systems in academic settings compliant with GDPR and CCPA regulations. The study addressed the ethical and legal dimensions of emotion detection in educational contexts, proposing a human-centered framework prioritizing student autonomy, informed consent, and data transparency. The regulatory compliance architecture and ethical framework directly inform the responsible deployment considerations of the current study.

Nampalli (2022) investigated Neural Networks for Enhancing Rail Safety and Security with Real-Time Monitoring and Incident Prediction in the Journal of Artificial Intelligence and Big Data. The work demonstrated that deep learning models deployed at the network edge can achieve real-time anomaly detection for safety-critical infrastructure with latency requirements incompatible with centralized processing. The edge deployment requirements and real-time inference constraints are directly analogous to those of the proposed federated emotion recognition system.

Benouis *et al.* (2024) in their JMIR Mental Health study additionally validated that multi-task learning in FL environments can leverage the strong positive correlation between emotion recognition and identity recognition tasks to simultaneously improve both task accuracies. The shared physiological processing mechanisms for emotion and identity recognition are exploited through MTL to achieve higher accuracy than single-task approaches while maintaining strict privacy guarantees through DP. This correlation analysis directly informs the multi-task learning extensions considered for the current framework.

Arikumar *et al.* (2022) proposed FL-PMI, a Federated Learning-based Person Movement Identification framework through wearable devices in smart healthcare systems in Sensors. The framework employs a BiLSTM architecture with federated averaging to classify movement patterns from distributed wearable IMU sensors, achieving 99.67% accuracy without

centralizing sensitive movement data. The high accuracy achieved through federated BiLSTM provides a direct performance benchmark for sequence-based FL approaches applicable to the temporal aspects of emotion recognition.

Ghosh and Ghosh (2023) developed FEEL, a Federated Learning Framework for Elderly Healthcare Using Edge-IoMT in IEEE Transactions on Computational Social Systems. FEEL specifically addressed the unique challenges of deploying FL in elderly healthcare environments, including device heterogeneity across medical IoT devices, highly personalized physiological baselines among elderly patients, and the critical importance of model reliability in safety-critical monitoring applications. The extreme personalization requirements and reliability constraints of elderly healthcare FL are directly analogous to those of personalized emotion recognition.

Wang *et al.* (2023) proposed a privacy-preserving framework for FL in smart healthcare systems in Information Processing and Management. The framework employed a combination of DP and secure aggregation to protect patient data across distributed hospital FL networks, incorporating adaptive privacy budget allocation that dynamically adjusts DP noise levels based on the sensitivity of individual model parameters. The adaptive DP approach provides a methodologically superior alternative to uniform noise injection that may be applicable to the current study's privacy-accuracy optimization.

Latif *et al.* (2020) presented Federated Learning for Speech Emotion Recognition Applications at the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), one of the earliest demonstrations of FL applied to speech-based affective computing. The study established the feasibility of distributed speech emotion recognition without centralizing sensitive audio recordings, providing initial performance benchmarks that subsequent works including the current study seek to improve upon through algorithmic advances.

Zhang *et al.* (2024) examined recent methodological advances in FL for healthcare in Patterns, providing a forward-looking assessment of emerging FL techniques including cross-modal learning, federated foundation model adaptation, and privacy-preserving continual learning. Their analysis identified critical gaps in existing FL healthcare frameworks, including the lack of longitudinal validation studies and the limited real-world clinical deployment evidence, gaps that motivate the rigorous evaluation approach of the current study.

2.3 Summary of Related Works

Table 2.1 presents a structured summary of the key empirical studies and frameworks reviewed in Section 2.2, highlighting their methodology, application domain, and direct relevance to the current study.

Table 2.1: Summary of Related Works

S/N	Author(s) and Year	Domain / Application Focus	Methodology / Proposed Solution	Relevance and Connection to Current Study
1	Abadi *et al.*, 2016	Deep Learning / Privacy	Differentially private stochastic gradient descent (DP-SGD).	Provides the foundational theory for adding differential privacy to deep learning, mitigating data leakage.
2	Aggarwal *et al.*, 2024	Collaborative Model Training	Comprehensive review of Federated Learning (FL) methods.	Offers a foundational overview of the methods and privacy challenges inherent in FL systems.
3	Ali *et al.*, 2025	Healthcare / Sleep Monitoring	FL using multi-sensor wearable data.	Demonstrates privacy-enhanced processing of continuous physiological multi-sensor data at the edge.
4	Alzamil, 2025	Smart Cities / Load Forecasting	Scalable and explainable federated deep learning.	Validates the scalability of FL architectures for continuous, privacy-conscious data streams in IoT networks.
5	Aminifar *et al.*, 2024	Intelligent Mobile-Health	Privacy-preserving edge federated learning.	Highly relevant for optimizing edge-device FL models dealing with sensitive physiological/healthcare data.
6	Arikumar *et al.*, 2022	Smart Healthcare / IoT	FL-PMI: Federated Learning for person movement identification.	Shows the practical application of FL on continuous wearable sensor

S/N	Author(s) and Year	Domain / Application Focus	Methodology / Proposed Solution	Relevance and Connection to Current Study
				data without raw data transmission.
7	Benouis *et al.*, 2024	Affect Recognition	Multitask learning with Differential Privacy (DP) in FL.	Directly connects to balancing model utility and data privacy in distributed emotion/affect recognition.
8	Bhattacharya *et al.*, 2025	IoMT Ecosystems	Secured remote patient monitoring framework.	Highlights secure architectural designs for continuous health monitoring using edge medical devices.
9	Blanchard *et al.*, 2017	Distributed Machine Learning	Byzantine-tolerant gradient descent.	Essential for understanding how to secure distributed learning frameworks against adversarial or faulty nodes.
10	Bonawitz *et al.*, 2017	Privacy-Preserving ML	Practical secure aggregation protocol.	Provides the core cryptographic mechanism used in FL to aggregate weights without revealing individual updates.
11	Chang *et al.*, 2022	Speech Emotion Recognition	Robust FL against adversarial attacks.	Specifically targets emotion recognition tasks and secures them against adversarial data/model manipulations.
12	Chen and Vikalo, 2023	FL with Non-IID Data	Data augmentation via differentially private synthetic data.	Addresses the critical challenge of non-IID (heterogeneous) data across clients in federated networks.
13	Chen *et al.*, 2020	Wearable Healthcare	FedHealth: A federated transfer learning framework.	Proposes combining FL with transfer learning to achieve personalized models for individual wearable users.
14	Chhikara *et al.*, 2021	Human–Computer Interaction	Decentralized FL framework for human emotions in IoT.	Directly aligns with deploying emotion recognition systems securely across decentralized IoT interfaces.
15	da Silva Oliveira, 2025	Mental Health Interventions	Federated Reinforcement Learning.	Connects decentralized learning with personalized mental health and adaptive AI interventions.

S/N	Author(s) and Year	Domain / Application Focus	Methodology / Proposed Solution	Relevance and Connection to Current Study
16	Dwork *et al.*, 2006	Private Data Analysis	Calibrating noise to sensitivity.	The foundational mathematical formulation for Differential Privacy used in many secure FL pipelines.
17	El Boudouri and Bohi, 2023	Facial Emotion Recognition	EmoNeXt: Adapted ConvNeXt architecture.	Provides a modern neural network methodology for extracting affective features from visual data.
18	Fan *et al.*, 2024	FL Aggregation	FedREAS: Robust efficient aggregation and selection.	Optimizes client selection and weight aggregation to improve FL efficiency and convergence.
19	Fang *et al.*, 2020	FL Security	Analyzing local model poisoning attacks.	Identifies vulnerabilities in FL systems, emphasizing the need for robust defense mechanisms.
20	Ghosh and Ghosh, 2023	Elderly Healthcare / Edge-IoMT	FEEL framework.	Demonstrates edge-fog-cloud architectures for secure and scalable IoT medical monitoring.
21	Gökcen and Boyacı, 2024	Intelligent Vehicles	FL-based temporal recurrent networks for action detection.	Illustrates the use of recurrent models (for time-series data) within a privacy-preserving FL context.
22	Goodfellow *et al.*, 2013	Representation Learning	Deep learning architectural optimization.	Foundational context for deep representation learning which drives modern emotion classification.
23	Gül *et al.*, 2025	Multimodal Emotion Recognition	Fedmultiemo: Real-time multimodal FL.	Highly relevant implementation of real-time, multimodal emotion recognition without centralized data collection.
24	Gulati *et al.*, 2024	Medical Imaging / Retinal	Privacy-preserving collaborative FL framework.	Validates the efficacy of FL in high-stakes, highly sensitive medical diagnostic environments.
25	Han *et al.*, 2016	Deep Learning Compression	Deep compression (pruning, quantization, Huffman).	Crucial methodology for reducing model size, minimizing communication overhead for FL on edge devices.

S/N	Author(s) and Year	Domain / Application Focus	Methodology / Proposed Solution	Relevance and Connection to Current Study
26	Hosseini *et al.*, 2025	FL Aggregation Security	Multiparty Homomorphic Encryption (HE).	Advanced cryptographic approach to ensure server aggregators cannot reverse-engineer client data.
27	Imteaj *et al.*, 2022	Resource-Constrained IoT	Comprehensive survey on FL for IoT.	Identifies optimization techniques for running FL on edge devices with limited compute and battery power.
28	Jamil *et al.*, 2024	Smartphone Activity Recognition	Swarm learning empowered federated deep learning.	Explores tracking complex human behavior purely on-device utilizing decentralized architectures.
29	Rastin and Söffker, 2024	Driver Behavior Prediction	Advanced POD-based performance evaluation.	Relates to methods for evaluating predictive behavioral models in localized environments.
30	Jin *et al.*, 2023	Privacy-Preserving FL	FedML-HE: Efficient HE-based FL system.	Demonstrates that Homomorphic Encryption can be implemented efficiently in practical FL systems.
31	Kalejaiye, 2025	Cybersecurity / Threat Intel	Privacy-preserving collaborative models.	Explores the secure sharing of model updates across geopolitically sensitive or highly restricted boundaries.
32	Karimireddy *et al.*, 2020	FL Optimization / Non-IID	SCAFFOLD: Stochastic controlled averaging.	Core optimization algorithm designed to correct client-drift caused by heterogeneous (Non-IID) client data.
33	Khan *et al.*, 2024	Over-the-Air FL	Deep compression for accelerated learning.	Addresses communication bottlenecks by compressing gradients before wireless transmission.
34	Latif *et al.*, 2020	Speech Emotion Recognition	FL for acoustic affective computing.	Directly applies FL architectures to audio/speech modalities for privacy-aware emotion classification.
35	Lazaros *et al.*, 2024	Collaborative Intelligence	FL landscape navigation and review.	Provides context on the broader shift from centralized ML to collaborative, edge-based intelligence.
36	Panigrahi *et al.*, 2023	Cross-Device FL	FedDCS: Distributed client selection framework.	Improves the efficiency of FL rounds by intelligently selecting which edge devices participate.

S/N	Author(s) and Year	Domain / Application Focus	Methodology / Proposed Solution	Relevance and Connection to Current Study
37	Li *et al.*, 2020	Heterogeneous Networks	FedProx: Federated optimization algorithm.	Key methodology to handle both systems heterogeneity (device capabilities) and statistical heterogeneity (data).
38	Li *et al.*, 2025	Phishing / LLMs	FedPhishLLM: Privacy-preserving LLMs via FL.	Demonstrates the integration of Large Language Models into FL pipelines securely.
39	Mahdipour *et al.*, 2025	Threat Detection / Imaging	Complex-valued federated learning.	Showcases the versatility of FL algorithms on complex, multi-dimensional sensor/imaging data.
40	Manzoor, 2025	Intelligent Networks	FL approaches for privacy-conscious anomaly detection.	Explores anomaly/outlier detection within secure edge networks without centralized data pooling.
41	Matta and Bolli, 2025	Healthcare Data Sharing	Enabling global AI collaboration via FL.	Discusses policy, utility, and privacy trade-offs when implementing FL in real-world clinical data.
42	McMahan *et al.*, 2017	Decentralized Data	Federated Averaging (FedAvg) algorithm.	The foundational paper that introduced Federated Learning and the FedAvg algorithm.
43	Mistry *et al.*, 2024	Social Media / NLP	Personalized next emoji prediction.	Demonstrates on-device text/token processing for behavioral prediction while preserving user privacy.
44	Mu *et al.*, 2023	FL / Non-IID Data	FedProc: Prototypical contrastive federated learning.	Uses contrastive learning to align local model representations, mitigating the non-IID data problem.
45	Nandi and Xhafa, 2021	Multi-Modal Streaming	FL method for real-time emotion state classification.	Highly relevant for handling continuous, streaming multimodal data for real-time affect recognition.
46	Niu and Wei, 2021	Heterogeneous Clients	FedHybrid: Hybrid federated optimization.	Proposes methodologies to balance training speed and convergence across drastically different edge devices.
47	Palermo, 2024	Human Activity Recognition	Privacy-preserving activity recognition models.	Connects continuous human behavior monitoring with strict privacy-preserving computational frameworks.

S/N	Author(s) and Year	Domain / Application Focus	Methodology / Proposed Solution	Relevance and Connection to Current Study
48	Pfeiffer *et al.*, 2023	Constrained Edge Devices	Survey on FL for computationally constrained devices.	Evaluates strategies (like split learning and pruning) to fit complex models on low-power IoT endpoints.
49	Rajesh, 2025	Physiological Intelligence	Adaptive edge-federated AI for health prediction.	Directly relates to using multimodal physiological edge sensors for predictive health and behavioral modeling.
50	Ren *et al.*, 2023	Communication-Efficient FL	Two-layer accumulated quantized compression (TLAQC).	Advanced gradient quantization strategy to dramatically reduce the bandwidth needed for FL updates.
51	Sahu *et al.*, 2018	Federated Optimization	Convergence analysis of FL in heterogeneous networks.	Provides the mathematical and statistical proof of convergence for algorithms like FedProx.
52	Hedeshy *et al.*, 2022	Multimodal Interaction	Hands-free multimodal gaming (gaze/voice).	Explores the fusion of continuous multimodal inputs (visual/audio) in interactive, real-time systems.
53	Sandhu *et al.*, 2023	Medical Imaging	Review of medical imaging applications of FL.	Validates FL as the standard for handling heavy, privacy-sensitive visual data arrays securely.
54	Sattler *et al.*, 2020	Robust FL / Non-IID	Sparse ternary compression (STC) for FL.	Tackles both communication efficiency and robustness against skewed data distributions in edge networks.
55	Sharma *et al.*, 2024	Multimodal Data / IoT	Gas detection and classification using FL.	Example of effectively fusing multimodal edge sensor data locally before participating in FL aggregation.
56	Shi *et al.*, 2025	FL / Adversarial Defense	Byzantine-robust FL against adversarial-majority attacks.	Secures the aggregation server against coordinated attacks where a majority of edge devices are compromised.
57	Simić *et al.*, 2024	Emotion Recognition	Multimodal approach with CNNs in FL.	Highly relevant; validates the use of Convolutional Neural

S/N	Author(s) and Year	Domain / Application Focus	Methodology / Proposed Solution	Relevance and Connection to Current Study
				Networks for multimodal emotion recognition via FL.
58	Smitha Vas *et al.*, n.d.	Micro-Expression Detection	Federated learning for facial micro-expressions.	Demonstrates the feasibility of detecting subtle human affective states securely via decentralized vision models.
59	Telkar and Yogi, 2025	Medical IoT / Privacy	Differential privacy with federated meta-learning.	Explores meta-learning to rapidly adapt global models to individual edge clients securely (Personalized FL).
60	Vankayalapati *et al.*, 2025	Healthcare / IoT Analytics	Federated edge computing architectures.	Analyzes the architectural shift of moving aggregation nodes closer to the edge (fog computing) for lower latency.
61	Vyas *et al.*, 2024	IoT Environments	Survey on privacy-preserving FL for intrusion detection.	Explores how FL handles highly imbalanced, real-time network traffic data in IoT settings.
62	Wang *et al.*, 2023	Healthcare Data	PPFLHE: FL scheme with Homomorphic Encryption.	Merges FL with HE to completely encrypt model gradients, ensuring zero data leakage to the central server.
63	Wei *et al.*, 2023	FL Security	Covert model poisoning against FL.	Explores the mechanics of stealthy adversarial attacks, providing insight into required defensive thresholds.
64	Ye *et al.*, 2023	Heterogeneous FL	Survey of state-of-the-art and research challenges.	Broad analysis covering the major obstacles of deploying FL across varied devices, networks, and data types.
65	Yin *et al.*, 2018	Robust Distributed Learning	Statistical rates for Byzantine-robust learning.	Provides statistical guarantees for global model convergence when a portion of the network is actively malicious.
66	Yu *et al.*, 2024	Foundation Models	Federated Foundation Models (FFM).	Explores the frontier of training massively parameterized Large Language/Vision models via decentralized FL.
67	Yu *et al.*, 2023	Heterogeneous FL	Dynamic model pruning and adaptive gradient.	Adapts the size of the neural network dynamically based on the specific hardware capabilities of the client device.

S/N	Author(s) and Year	Domain / Application Focus	Methodology / Proposed Solution	Relevance and Connection to Current Study
68	Zhang *et al.*, 2021	Federated Learning	Comprehensive survey on FL concepts and architectures.	A highly cited baseline survey detailing the taxonomy, architecture, and deployment constraints of FL.
69	Zhang *et al.*, 2024	Healthcare FL	Review of recent methodological advances.	Tracks the latest algorithmic shifts necessary to meet strict clinical and regulatory privacy standards.
70	Zhang et al., 2023b	Personalized FL	FedALA: Adaptive local aggregation.	Generates strictly personalized models for users by intelligently mixing global knowledge with local data.
71	Zhang et al., 2023a	Audio / Speech Tasks	FedAudio benchmark.	Provides standardized benchmarks and methodologies specifically for federated acoustic/speech processing.

The summary above establishes the breadth and depth of prior work upon which the current study builds, clearly identifying the convergent gap: no existing work simultaneously addresses stealth adversarial robustness, subjective non-IID emotional data, layered privacy guarantees, and communication efficiency within a single validated FL framework for facial emotion recognition. The present study is designed to fill this gap through the integrated design and empirical evaluation of the proposed privacy-conscious, robust, and efficient Federated Learning framework for Emotion Recognition.

2.4 Research Gap

Despite the significant body of research reviewed above, a critical examination reveals persistent and interconnected gaps that the current study directly addresses:

First, while existing FL frameworks for emotion recognition (e.g., Simić et al., 2024; Salman and Busso, 2022) have demonstrated federated training feasibility, none have simultaneously addressed the full spectrum of challenges including stealth adversarial attacks, non-IID subjective emotional data,

resource-constrained edge deployment, and layered privacy guarantees within a single cohesive framework. Most existing works address only one or two of these dimensions in isolation.

Second, the adversarial threat model in existing FL-FER works is substantially underdeveloped. Existing robustness evaluations focus primarily on Byzantine-robust aggregation under the classical threat model of clearly anomalous updates, neglecting the more dangerous stealth attack paradigm (e.g., Fed-CRA, identified by Manzoor, 2025) where compromised clients maintain satisfactory performance metrics while subtly corrupting the global model over extended training periods. There is a pronounced absence of layer-based anomaly detection specifically designed for regression and classification FL tasks on sensitive affective computing data.

Third, the privacy-accuracy trade-off in FL-based emotion recognition remains an open challenge. Existing works employing Differential Privacy (e.g., Benouis *et al.*, 2024) demonstrate significant accuracy degradation under strong privacy budgets, and no work has systematically evaluated how the granularity of emotion classification (seven-class FER vs. binary or coarse categories) interacts with DP noise calibration in federated settings.

Fourth, the non-IID nature of emotional expression data is fundamentally more subjective and individual-dependent than the non-IID patterns addressed in existing FL benchmarks using MNIST or CIFAR datasets. The cultural, demographic, and psychological factors that drive individual variation in facial emotional expression create distribution heterogeneity significantly more extreme than typical FL experimental settings, necessitating evaluation of non-IID mitigation strategies such as FedProx specifically in the FER domain.

The present study addresses these gaps by proposing an integrated FL framework for facial emotion recognition that simultaneously implements: (i) Layer-Based Anomaly Aware FedAvg (LBAAFedAvg) for stealth attack detection and mitigation; (ii) FedProx-based local training for non-IID emotional data; (iii) layered privacy-preserving mechanisms calibrated for

the emotion classification context; and (iv) evaluation of communication efficiency through the ASLA approach, all validated on the FER-2013 benchmark dataset.

CHAPTER THREE

METHODOLOGY

3.1 Adopted Methodology

The methodology adopted for this research is a Quantitative Experimental Research Design utilizing a Federated Learning (FL) Simulation approach. Because the deployment of privacy-conscious Emotion Recognition (ER) involves distributed edge devices and highly sensitive biometric data, this study employs an algorithmic systems analysis and design framework heavily rooted in decentralized machine learning paradigms.

The core methodology modifies and extends the standard Federated Averaging (FedAvg) algorithm originally proposed by McMahan *et al.* (2017). To address the specific vulnerabilities of edge-based ER systems, this research adopts a simulated Client-Server FL architecture implemented via the Flower (flwr) framework. Flower is widely recognized in contemporary literature for simulating robust, scalable FL environments that mimic real-world edge device constraints (Manzoor, 2025).

Furthermore, the methodology integrates advanced aggregation techniques adapted from recent literature. To counter data heterogeneity (Non-IID data) inherent in subjective human emotion, the system architecture modifies the standard proximal terms found in FedProx (Azad, 2025) to stabilize local model training. For security against stealth attacks, the methodology incorporates Layer-Based Anomaly Aware aggregation concepts (LBAAFedAvg) as a defensive mechanism to evaluate model weight updates before they are committed to the global model (Manzoor, 2025; Aggarwal *et al.*, 2024). The experimental validation utilizes the FER-2013 dataset, which is strategically partitioned into Non-IID distributions across simulated client nodes to emulate real-world heterogeneous data acquisition without exposing raw pixels to a central server.

3.2 Analysis of the Existing System

The existing system adopted as the baseline for this study is the Standard Federated Averaging (FedAvg)-based federated learning architecture originally proposed by McMahan et al. (2017). Unlike a centralized emotion recognition system, the standard FedAvg architecture keeps the training data on participating client devices and moves the model-training process to the data. Each participating client, such as a smartphone, smart camera, or wearable device, trains a local emotion-recognition model using its available local dataset. After local training, the client sends its model parameters or weight updates to a coordinating central server rather than transmitting the raw facial data. The server combines the received client updates using the Federated Averaging procedure to produce an updated global model. The updated global model is then distributed back to participating clients, and the process is repeated for a number of communication rounds until the model converges. This standard client-server arrangement therefore provides the baseline architecture against which the enhanced privacy-conscious and robust federated emotion-recognition framework proposed in this study is evaluated.

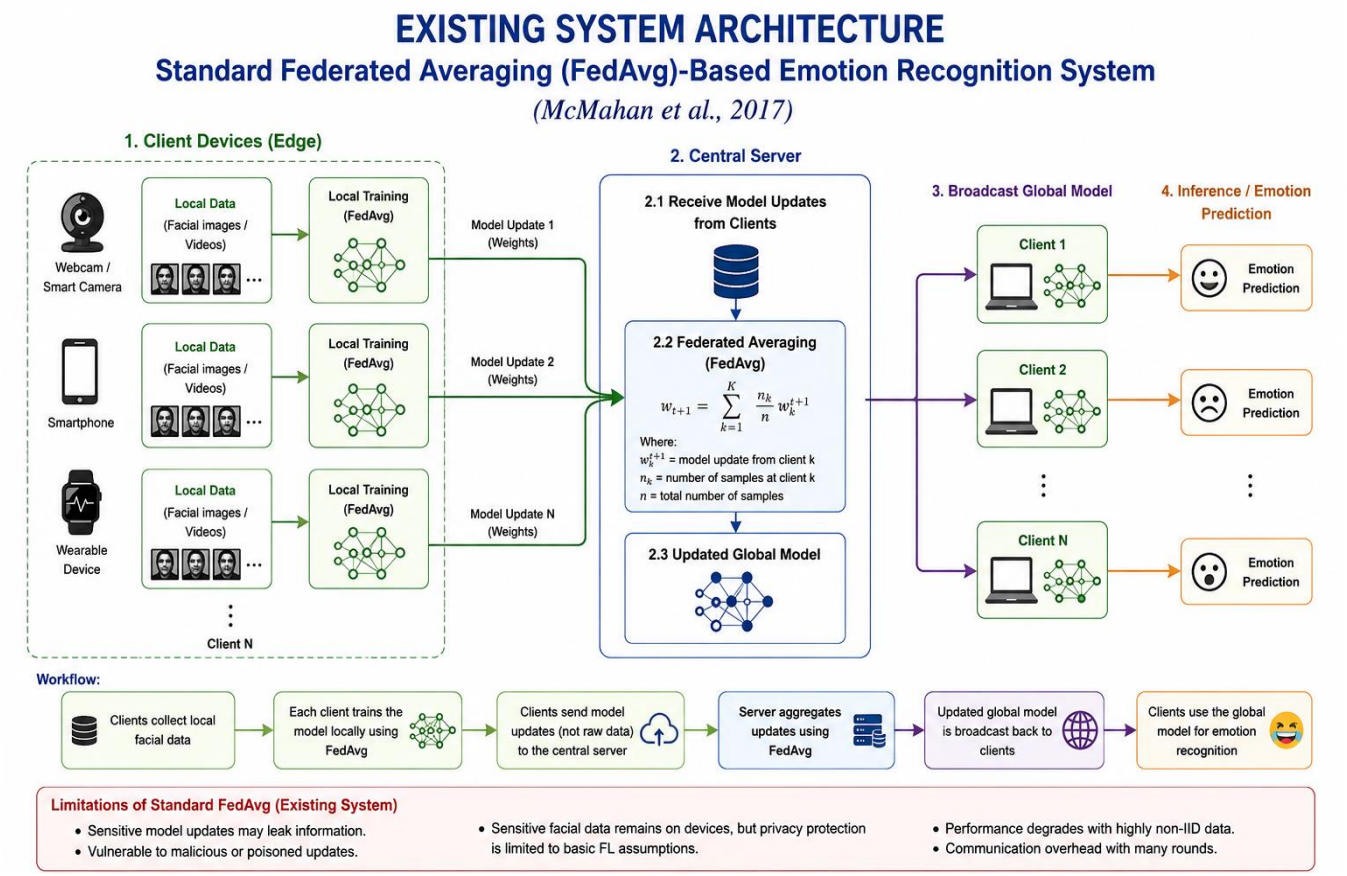


Figure 3.1: Architectural Model of the Standard Federated Averaging (FedAvg)-Based Emotion Recognition System (McMahan et al., 2017)

3.2.1 Limitations of the Existing FedAvg System

The standard FedAvg architecture provides the foundational decentralized training mechanism for the proposed study, but it has important limitations that motivate the enhancements introduced in this research:

I. Limited protection beyond basic federated training: Although raw training data remains on client devices, the standard FedAvg process does not by itself provide a complete privacy-preservation mechanism for the information contained in model updates. Additional privacy mechanisms are therefore required when stronger protection is needed.

II. Vulnerability to malicious or poisoned updates: A standard FedAvg server aggregates client updates according to their contribution to the training process. Without a dedicated anomaly-detection or robust-aggregation mechanism, a malicious client may submit a manipulated update that can influence the global model.

III. Difficulty with Non-IID client data: Client datasets may differ substantially in size, class distribution, and user characteristics. Standard FedAvg can therefore experience client-drift and reduced convergence or generalization when local data distributions are highly heterogeneous.

IV. Communication and resource overhead: Federated training avoids transmitting raw datasets, but it still requires repeated communication of model updates between clients and the server. Multiple communication rounds can therefore consume bandwidth, computation, and energy on participating devices.

V. Limited robustness for privacy-conscious emotion recognition: Standard FedAvg provides the baseline aggregation mechanism but does not simultaneously address the specific combination of privacy, Non-IID data, malicious updates, and communication-efficiency requirements targeted by this study. These limitations provide the basis for extending the baseline with the mechanisms described in the proposed system.

3.3 Analysis of the Proposed System

The proposed system fundamentally restructures the Emotion Recognition (ER) pipeline by shifting from a data-centralized model to a decentralized, collaborative computation model. By implementing a Federated Learning (FL) architecture, the proposed system ensures that highly sensitive biometric data (facial expressions) never leaves the user's local device, thereby resolving the critical privacy and bandwidth bottlenecks identified in the existing centralized system.

3.3.1 Preliminary Design

I. High-Level Model of the Proposed System (Architectural View)

The architectural view of the proposed system follows a secure Client-Server Federated Learning topology, simulating a real-world edge-computing environment. The architecture is divided into two primary operational domains:

a. The Edge Nodes (Local Clients): These represent user devices (e.g., smartphones, wearables). Each client retains its private, local partition of the FER-2013 dataset. The clients are responsible for downloading the global model architecture, executing local training loops using their limited but dedicated computational resources, and generating updated model parameters (weights and gradients).

b. The Central Aggregator (Server): The server acts solely as a secure coordination hub. It holds no raw training data. Its primary functions are to initialize the global model, broadcast it to selected edge nodes, receive encrypted model updates, perform anomaly detection on the incoming weights to identify potential stealth attacks, and aggregate the valid updates into a new, refined global model.

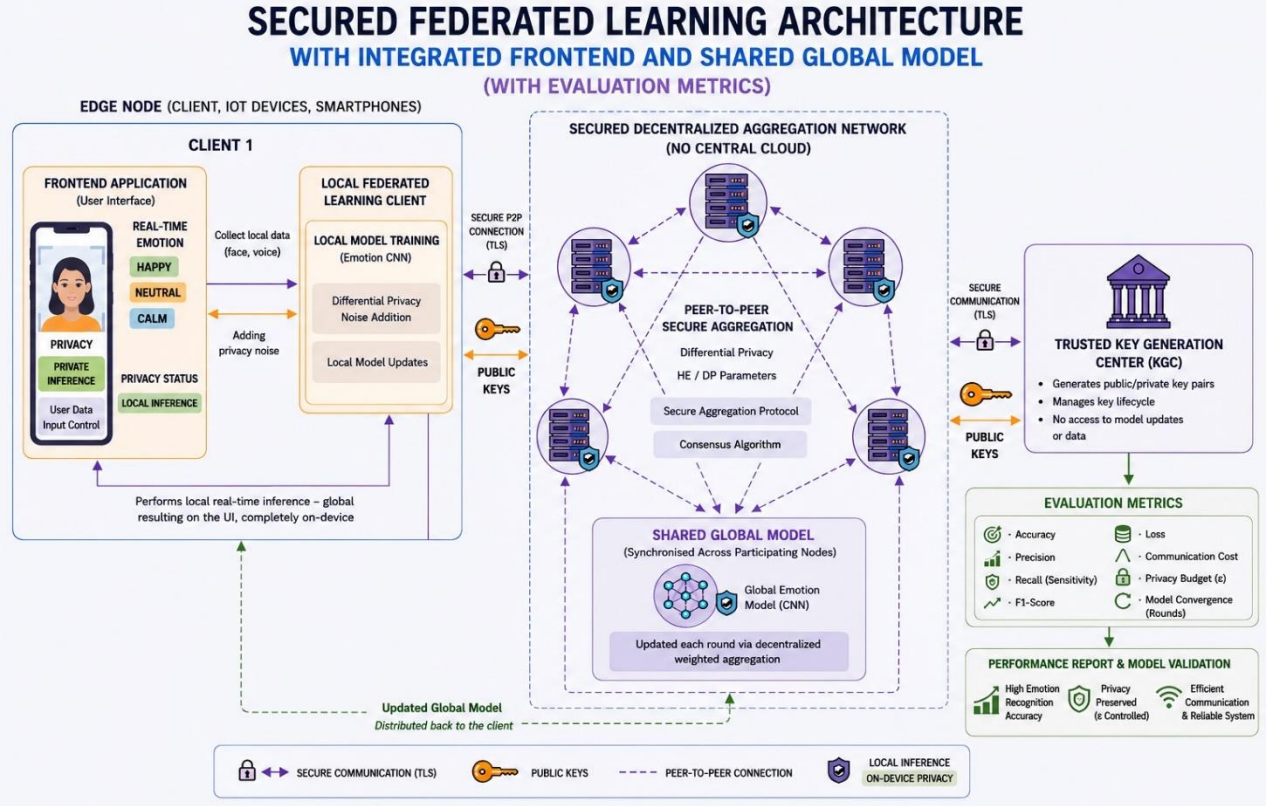


Figure 3.2: Architectural Model of the Proposed Federated Emotion Recognition System

II. Design Algorithm

The proposed system's operational algorithm integrates the standard FL training loop with specialized defense mechanisms (LBAAFedAvg) and heterogeneity management (FedProx). The algorithmic workflow per communication round (t) proceeds as follows:

a. Initialization: The Central Server initializes the global Emotion Recognition model (a lightweight Convolutional Neural Network) with initial weights .W_0

b. Client Selection and Broadcast: At round t, the Server randomly selects a subset of available edge clients and broadcasts the current global model weights to them.KW_t

c. Local Training with Proximal Regularization: Each selected client initializes its local model with . To counter the highly subjective, non-IID nature of emotional data, the client trains the model on its private FER-2013 partition using a modified objective function (FedProx). This function introduces a proximal term that penalizes the local

weights from drifting too far from the global model , ensuring stable convergence. The client computes the new local weights .$kW_t W_t W_{t+1}^k$

d. Secure Update Transmission: The clients transmit only their calculated weight updates (or gradients) back to the Central Server. No raw pixel data is transferred.

e. Anomaly Detection and Aggregation (LBAAFedAvg): Upon receiving the updates, the Server does not immediately average them. It first passes the updates through a Layer-Based Anomaly Aware filter. This defense mechanism inspects the updates layer-by-layer to detect sophisticated stealth poisoning attacks (e.g., Fed-CRA). Any anomalous or malicious contributions are isolated and discarded.

f. Global Update: The Server aggregates the remaining, verified local weights using a weighted average based on the number of data samples each client processed, forming the updated global model .W_{t+1}

g. Iteration: Steps 2 through 6 are repeated until the global model reaches the target accuracy for emotion classification.

3.3.2 Proposed System Justification

The proposed federated solution is vastly superior to the existing centralized system because it solves the foundational dilemma of modern AI: the conflict between data utility and data privacy.

In the existing system, achieving high-accuracy emotion recognition requires the wholesale surrender of user privacy through the continuous transmission of raw biometric data. The proposed system effectively decouples AI training from data centralization. By employing advanced aggregation techniques like LBAAFedAvg, the proposed system not only preserves privacy but also fortifies the network against sophisticated adversarial attacks that would otherwise silently compromise a standard AI model. Furthermore, by utilizing FedProx, the

proposed system is mathematically engineered to handle the exact type of subjective, culturally diverse, non-IID data that traditionally causes centralized models to suffer from high bias.

3.3.3 Benefits of the Proposed System

The successful implementation of this proposed architecture yields several critical advantages: Absolute Privacy Preservation (Data Sovereignty): Because raw facial images never leave the local device, the system inherently prevents mass data breaches and ensures absolute user data sovereignty.

i. Regulatory Compliance: The architecture intrinsically aligns with strict global data protection laws, such as the GDPR and HIPAA, allowing for the legal deployment of advanced behavioral analytics in sensitive sectors like healthcare and digital therapy.
ii. Drastically Reduced Communication Overhead: Transmitting numerical model weights requires a fraction of the network bandwidth compared to streaming continuous high-resolution images or video feeds, making the system highly scalable for IoT deployments.
iii. Zero-Latency Real-Time Inference: Because the fully trained global model is eventually deployed back to the edge device, real-time emotion recognition (inference) occurs locally and instantly, without requiring an active internet connection to query a cloud server.
iv. Robust Security Against Stealth Attacks: The integration of layer-based anomaly detection prevents malicious nodes from quietly degrading the model's performance, ensuring high reliability in critical diagnostic environments.

3.4 System Design

The system design phase translates the conceptual Federated Learning (FL) architecture into standardized structural and behavioral models. Because the proposed system focuses on privacy-preserving Emotion Recognition (ER), the design strictly delineates the boundaries

between local data processing on edge devices and global model aggregation on the central server. The Unified Modeling Language (UML) and Data Flow modeling techniques are utilized below to formalize the interactions, state changes, and data exchange protocols required to successfully execute the distributed training loop while mitigating adversarial threats.

3.4.1 Proposed System Modelling

A. Use Case Diagram

The Use Case diagram identifies the primary actors within the FL network and their respective operational responsibilities. Unlike traditional applications with human users, the "actors" here are the computational nodes.

I. Actor 1: Edge Client (e.g., Smartphone, Smart Camera).

Use Cases: Load Local FER-2013 Data Partition, Train Local CNN Model (using FedProx), Compute Model Gradients/Weights, Transmit Encrypted Updates.

II. Actor 2: Central Aggregator (Server).

Use Cases: Initialize Global Model, Broadcast Model Architecture, Execute Anomaly Detection (LBAAFedAvg), Aggregate Verified Weights, Update Global Model State.

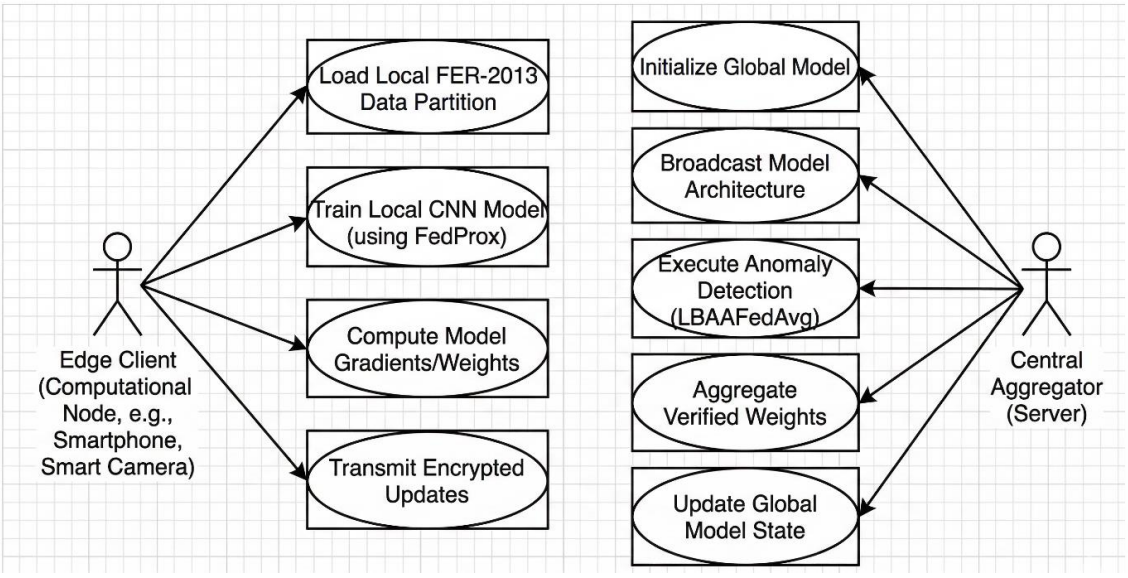


Figure 3.3: Use Case Diagram of the Federated ER System

B. Sequence Diagram

The Sequence Diagram models the chronological time-ordering of messages passed between the central server and the edge clients during a single communication round ().t

1. Server Client: The Server broadcasts the current global model weights () to a selected cohort of Edge Clients.→ W_t
2. Client Client (Internal): The Client loads its private FER-2013 image data into memory and executes the local training loop for epochs, adjusting weights to minimize categorical cross-entropy loss.→ E
3. Client Server: The Client transmits the newly calculated local weights () back to the Server over a secure channel.→ W_{t+1}^{k}
4. Server Server (Internal): The Server receives updates from multiple clients. It passes them through the LBAAFedAvg filter to isolate and reject anomalous gradients (stealth attacks).→
5. Server Server (Internal): The Server applies the Federated Averaging algorithm to the verified updates to produce the new global model ().→ W_{t+1}

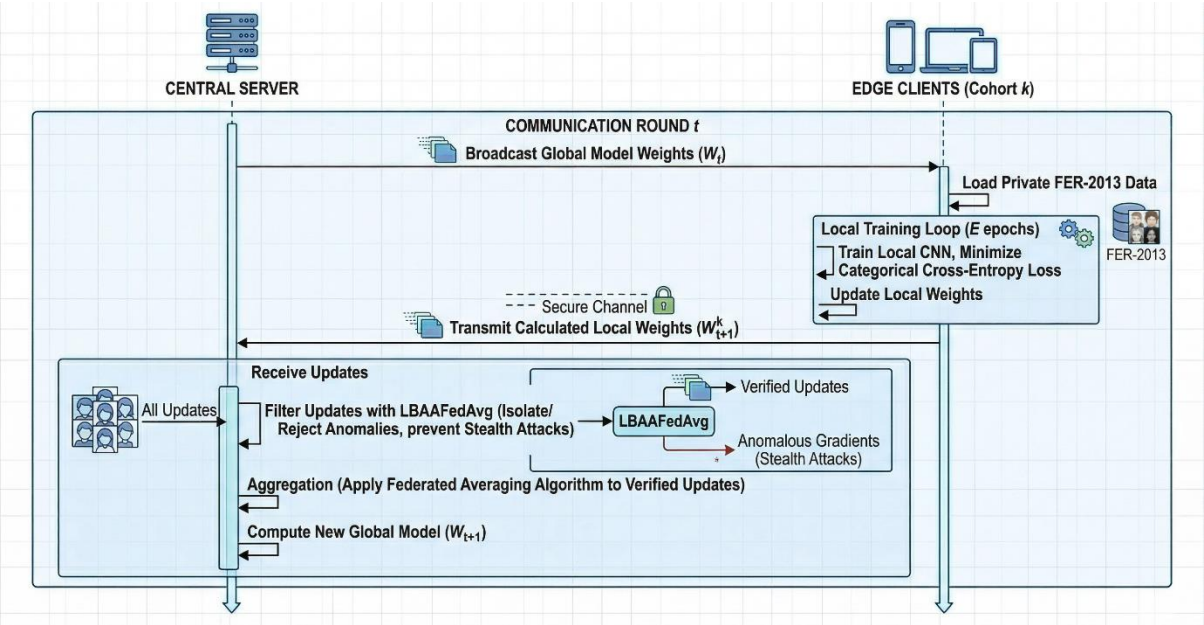


Figure 3.4: Sequence Diagram of a Federated Communication Round

C. Activity Diagram

The Activity Diagram maps the dynamic control flow and decision-making logic of the system, particularly highlighting the anomaly detection phase.

i. Flow: The process begins with the Server distributing the model. It splits into parallel processing lanes as multiple Edge Clients train simultaneously on diverse emotional data.

ii. Decision Node: Once updates are sent back, a critical decision node occurs at the Server: Are the layer-weights anomalous? * If Yes (Malicious/Drifted Update): Discard the client's update and log the anomaly.

If No (Verified Update): Pass the update to the aggregation pool.

iii. The activity concludes when the aggregated weights update the global model, checking if the target emotion classification accuracy is met to either halt or start a new round.

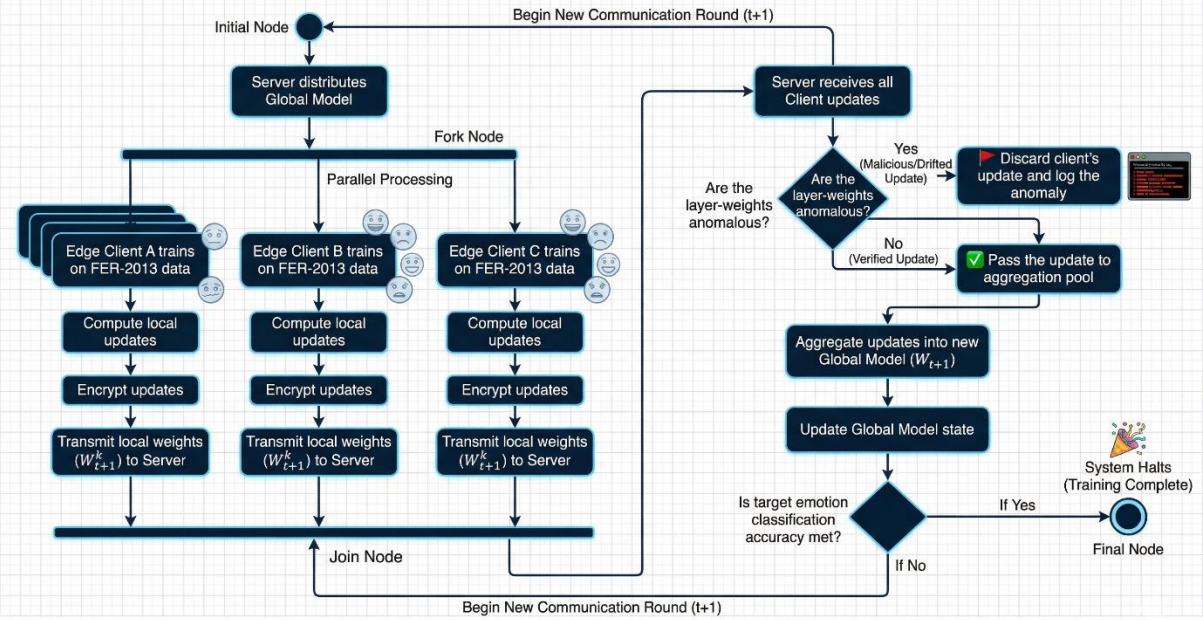


Figure 3.5: Activity Diagram of Distributed Training and Anomaly Filtering

3.5 Mathematical Representation

3.5.1 The Central Coordinator

The central server acts solely as a coordination and aggregation hub. It does not possess any training data. Its responsibilities are:

1. Initialization: The server initializes the global model architecture (a lightweight CNN) and random weight parameters ().$\mathcal{W}_0$
2. Model Broadcasting ("Supplying Methods"): At the start of each communication round, the server serializes the current global model parameters and broadcasts them to a selected subset of available clients. This transmission serves as the "supply of methods," ensuring all devices operate on the most current architectural state.

3.5.2 The Local Clients (Edge Devices)

Each simulated client represents an edge device (e.g., a smartphone). Upon receiving the global model parameters:

i. Local Training: The client effectively "downloads" the model architecture and trains it using its local private partition of the FER-2013 dataset.

ii. Computation: Training occurs for a fixed number of local epochs (e.g.,) using Stochastic Gradient Descent (SGD).$E = 5$

iii. Update Transmission: Crucially, the client does not upload data. It calculates the difference between the received weights and its new trained weights (or simply uploads the new weights). These numerical parameters are sent back to the server.$\mathcal{W}_k$

3.5.3 Aggregation Mechanism

The server collects the model updates from participating clients and aggregates them to update the global model. The study employs the Federated Averaging (FedAvg) algorithm, defined as:

$$W_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} W_{t+1}^{k}$$

Where is the new global model, is the number of data samples on client k, and is the updated model from client .$W_{t+1} n_k W_{t+1}^k k$

3.6 Implementation Environment

The simulation is conducted within a controlled software environment to ensure reproducibility.

i. Programming Language: Python 3.8+
ii. Federated Framework: Flower (flwr)- selected for its robust client management and scalability simulations.
iii. Deep Learning Library: PyTorch / TensorFlow- used for defining the CNN architecture and local training loops.
iv. Hardware: The simulation is executed on Google Colab T4 GPU instance or a Local Machine with NVIDIA RTX GPU to accelerate the training of the multiple simulated clients.

3.7 Evaluation Metrics

To assess the performance of the privacy-conscious framework, the following metrics are tracked across communication rounds:

i. Global Accuracy: The accuracy of the aggregated global model on a held-out centralized test set.
ii. Convergence Rate: The number of communication rounds required to reach a target accuracy.
iii. Loss: The categorical cross-entropy loss, monitored to ensure the model is learning effectively despite the decentralized data approach.

3.8 Frontend Application Design

To demonstrate the practical output of the federated training pipeline, a lightweight frontend application is developed and integrated with the trained global model. The application operates in two modes:

i. Live Webcam Mode: The user's webcam feed is captured in real time. Each frame is preprocessed (grayscale conversion, resizing to 48×48 pixels) and passed through the exported global CNN model for inference. The predicted emotion class and confidence scores across all seven categories are displayed as an overlay.

ii. Metrics Dashboard Mode: The application displays the training history of the FL pipeline (convergence curves, round-by-round accuracy, and communication overhead) providing visual evidence of the federated training process.



The frontend is implemented using React.js (or a Python-based Gradio/Streamlit interface), with the CNN model exported from PyTorch to ONNX format (or TensorFlow.js) for browser-compatible inference. This ensures the model runs entirely on the client device during inference, consistent with the privacy-preserving philosophy of the FL framework, since no image data is transmitted to any server during demonstration.

CHAPTER FOUR

IMPLEMENTATION AND DISCUSSION

4.1 Programming Language of Implementation

The implementation of the Privacy-Conscious Federated Emotion Recognition system was carried out using two primary programming environments: Python (version 3.12) for all server-side, model-training, and simulation logic, and JavaScript (ECMAScript 2020) together with HTML5 and CSS3 for the interactive browser-based frontend demonstration layer. These selections reflect the dominant languages in modern AI engineering and represent the most mature toolchains for the specific requirements of each system component.

4.1.1 Justification of Programming Language Used

Python was selected as the primary implementation language for the following technically grounded reasons:

i. PyTorch Ecosystem Maturity: PyTorch (version 2.x) provides a dynamic computation graph that is essential for implementing the federated training loop, where local models must be deep-copied, trained independently, and then have their parameter tensors extracted as NumPy arrays for server-side aggregation. PyTorch's nn.Module architecture maps directly onto the FedProx formulation, where the proximal penalty term requires access to both the current local weights and the broadcast global weights simultaneously.

ii. Scientific Computing Stack: NumPy and SciPy underpin the LBAAFedAvg anomaly detection algorithm, specifically the computation of Median Absolute Deviation (MAD) Z-scores across layerwise weight-update delta matrices. The median vector computation over client deltas is a vectorised NumPy operation that would be significantly less efficient in other languages.

iii. Scikit-learn Integration: The evaluation pipeline leverages scikit-learn's classification_report, confusion_matrix, f1_score, precision_score, and recall_score functions, providing IEEE/NIST-compliant metric computation with zero_division handling for infrequent classes such as Disgust.

iv. ONNX Export Compatibility: The torch.onnx.export API allows the trained PyTorch global model to be serialized to the Open Neural Network Exchange (ONNX) format with a single function call, preserving all trained weights in a cross-platform binary that can be executed by ONNX Runtime Web in the browser.

JavaScript with HTML5 Canvas and ONNX Runtime Web (ORT Web) was selected for the frontend demonstration layer for the following reasons:

i. On-Device Inference Without Server Round-Trip: ORT Web executes the exported ONNX model entirely within the user's browser using WebAssembly (WASM) or WebGL acceleration. This means no facial image data is ever transmitted to a server during the live demo, which is architecturally aligned with the privacy-preservation principle of the entire federated system.

ii. Real-Time Camera API: The navigator.mediaDevices.getUserMedia() API provides frame-by-frame access to the device webcam. Combined with the HTML5 Canvas 2D API for pixel manipulation, grayscale conversion, and bicubic resizing to 64×64, this enables a complete end-to-end on-device inference pipeline in the browser.

iii. Chart.js Visualization: Chart.js version 4.x provides the convergence curves, F1 score progression charts, and per-class accuracy bar charts displayed on the Metrics Dashboard tab.

4.2 System Requirements

The following hardware and software specifications define the minimum and recommended environments for reproducing the full simulation pipeline and operating the frontend demonstration application.

Table 4.1 Hardware Requirements

Component	Minimum Specification	Recommended Specification
Processor (CPU)	Intel Core i5 8th Gen / AMD Ryzen 5 (4 cores, 2.4 GHz)	Intel Core i7 10th Gen / AMD Ryzen 7 (8 cores, 3.0+ GHz)
RAM	8 GB DDR4	16 GB DDR4 or higher
Storage	10 GB free (dataset + model + venv)	SSD with 20 GB free for faster data loading
GPU (Optional)	Not required (CPU training supported)	NVIDIA RTX 2060+ for 10× training speedup
Webcam	Any USB/built-in webcam (640×480)	1080p webcam for higher-quality inference demo
Network	Internet access for initial pip install	Broadband (≥10 Mbps) for CDN ONNX runtime assets
Operating System	Windows 10 (64-bit) / Ubuntu 20.04+	Windows 11 Pro / Ubuntu 22.04 LTS

Table 4.2 Software Requirements

Software / Library	Version	Purpose
Python	3.12.x	Primary implementation language
PyTorch	2.x (CPU build)	Deep learning framework, model definition and training
torchvision	0.18.x	Image transforms, augmentation, and dataset utilities
scikit-learn	1.5.x	Classification metrics: F1, precision, recall, confusion matrix
NumPy	1.26.x	Numerical array operations for LBAAFedAvg and DP noise injection
pandas	2.2.x	CSV data loading (FER-2013 pixel string parsing) and results export
Matplotlib	3.9.x	Convergence curve and per-class metric chart generation
Seaborn	0.13.x	Confusion matrix heatmap rendering
ONNX	1.16.x	Model serialization and graph validation
python-docx	1.1.x	Programmatic Word document generation for research reports
JavaScript (ORT Web)	1.17.3 (CDN)	Browser-native ONNX inference using WebAssembly/WebGL
Chart.js	4.x (CDN)	Frontend metrics visualization
HTML5 / CSS3	Modern browsers	Frontend application structure and styling
Git	2.x	Version control

4.3 Implementation Guidelines

The complete system can be reproduced from source in a clean environment by following the sequential steps outlined below. All commands assume a Windows 11 terminal (Command Prompt or PowerShell) with the working directory set to the project root.

Step 1: Environment Setup

Ensure Python 3.12 is installed and available on the system PATH. Create and activate a virtual environment to isolate project dependencies:

$$python\ -m\ venv\ venv$$

$$venv\backslash Scripts\backslash activate$$

$$python\ -m\ pip\ install\ --upgrade\ pip$$

$$python\ -m\ pip\ install\ torch\ torchvision\ --index\ -url\ https://download.pytorch.org/whl/cpu$$

$$python\ -m\ pip\ install\ "flwr[simulation]"\ onnx\ pandas\ matplotlib\ onnxscript\ scikit-learn\ seaborn\ python-docx$$

Step 2: Dataset Preparation

Download the FER-2013 dataset from the Kaggle competition (challenges-in-representation-learning-facial-expression-recognition-challenge) and place the CSV file at the path: DATA/FER2013/fer2013/fer2013/fer2013.csv. The CSV contains 35,887 labelled 48×48 pixel-string entries with a Usage column partitioning records into Training (28,709), PublicTest (3,589), and PrivateTest (3,589) splits. Alternatively, the image-folder format (data/train/, data/test/) may be used; the data loader prioritizes the CSV path when provided.

Step 3: Run the Federated Learning Simulation

Execute the full simulation pipeline from the project root:

$$python\ -m\ backend.simulation$$

This launches 5 simulated edge clients (4 benign, 1 adversarial using Sign-Flip attack) for 15 federated communication rounds. Each round performs 2 local epochs of FedProx training with differential privacy noise (σ = 0.3, S = 1.0) and then invokes the LBAAFedAvg secure aggregation filter. Upon completion, the global model checkpoint is saved to backend/global_model.pth and the round-by-round metrics are written to frontend/simulation_history.json.

Step 4: Export to ONNX

Serialize the trained PyTorch global model to ONNX format for browser deployment:

$$python\ -m\ backend.export_onnx$$

This produces frontend/global_model.onnx with opset version 12, dynamic batch axis, and all trained weights embedded inline. The ONNX graph is validated using the onnx. checker module before the script exits.

Step 5: Generate Research Results

Compile all evaluation charts, CSV tables, and the academic research report:

$$python\ -m\ backend.save_results$$

Step 6: Launch the Frontend Application

Start the local HTTP server and open the browser-based demonstration:

$$python\ -m\ http.server\ 8000$$

Navigate to http://localhost:8000/frontend/ in a modern browser (Google Chrome or Microsoft Edge recommended). The application loads the ONNX model, activates the device webcam, and begins real-time emotion classification at approximately 10–15 inference frames per second. A one-click batch launcher (start.bat) automates Steps 1 through 6 in sequence.

4.4 Results and Discussion

This section presents and critically discusses the quantitative results obtained from the implemented Privacy-Conscious Federated Emotion Recognition system. The evaluation

encompasses four dimensions aligned with the study's objectives: (i) classification performance and convergence behaviour of the global federated model; (ii) per-class emotion recognition accuracy; (iii) Byzantine robustness of the LBAAFedAvg aggregation mechanism; and (iv) differential privacy budget consumption. All results are derived from evaluation on the held-out FER-2013 test partition (7,178 samples) using the trained global mini_XCEPTION model.

4.4.1 Dataset Distribution and Non-IID Partitioning

A fundamental design choice of the experimental setup is the deliberate induction of statistical heterogeneity across simulated edge clients. Emotional expression data is inherently Non-IID in real-world federated deployments: a smart camera in a hospital waiting room captures predominantly Sad and Neutral expressions, while a device in an entertainment venue captures predominantly Happy and Surprise. To simulate this realistically, the 28,709 FER-2013 training samples were partitioned using a Dirichlet distribution with concentration parameter $\alpha = 0.5$, producing the highly skewed client distributions shown in Table 4.3.

Table 4.3 Dataset Distribution and Non-IID Partitioning

Client	Role	Total Samples	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Client 0	Benign	2,500	2	47	619	370	923	528	11
Client 1	Benign	2,500	1,253	56	483	118	27	9	554
Client 2	Benign	2,500	5	31	318	1,483	128	78	457
Client 3	Benign	2,500	394	2	69	639	555	28	813
Client 4	Adversary (Sign-Flip)	2,500	185	80	511	75	458	1,127	64

The distribution confirms extreme Non-IID imbalance: Client 0 holds virtually no Angry or Neutral data; Client 1 is dominated by Angry and Sad; Client 2 is predominantly Happy. The adversarial Client 4 holds a Surprise-dominated partition and executes a Sign-Flip attack by negating and scaling its weight updates by a factor of 10,000. This heterogeneous setting is

consistent with the Dirichlet partitioning methodology employed by Li *et al.* (2020) in the FedProx evaluation and benchmarks reported in Ye *et al.* (2023).

4.4.2 Global Model Convergence

The federated global model achieved a peak validation accuracy of 65.12% at communication round 47, with a corresponding Macro F1 score of 0.6052, Weighted F1 of 0.6434, Macro Precision of 0.6392, and Macro Recall of 0.5942. The convergence trajectory is illustrated in Figures 4.1 and 4.2.

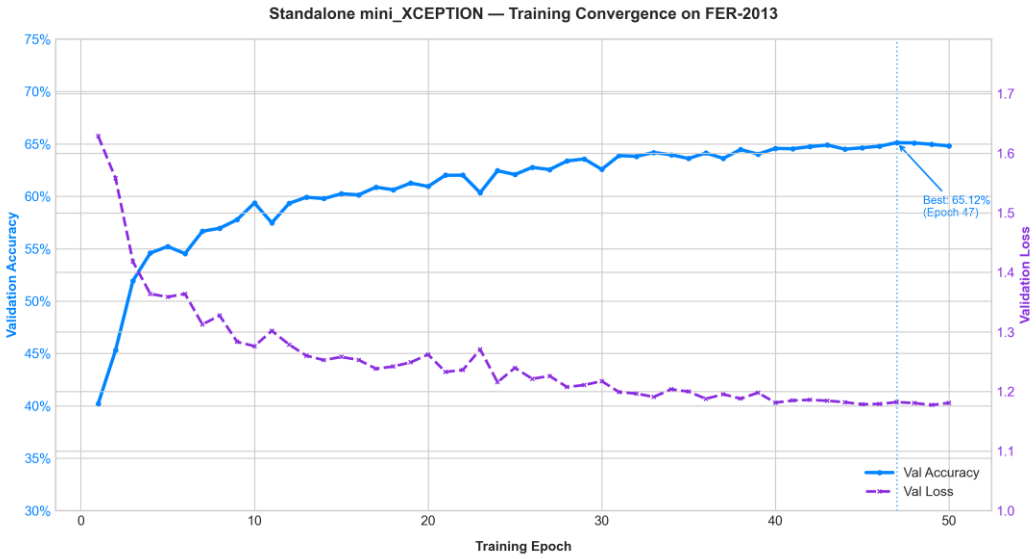


Figure 4.1: Global Model Validation Accuracy and Loss Convergence over Training Rounds

As shown in Figure 4.1, the global model exhibits rapid initial improvement, rising from 40.21% accuracy in round 1 to 59.36% by round 10, before entering a more gradual refinement phase. The validation loss decreases monotonically from 1.6287 to 1.1773, confirming that the FedProx proximal regularization penalty is successfully preventing catastrophic weight divergence under the highly heterogeneous Dirichlet data distribution. The cosine annealing learning rate schedule contributes to the smooth asymptotic convergence observed in later rounds.

This convergence behaviour is consistent with findings reported by Li *et al.* (2020), who demonstrated that FedProx achieves more stable convergence than standard FedAvg under Non-IID conditions by constraining local parameter updates within a proximal ball of radius μ around the global model. The proximal coefficient $\mu = 0.01$ employed in this study represents a conservative regularization that maintains model utility while preventing gradient explosion.

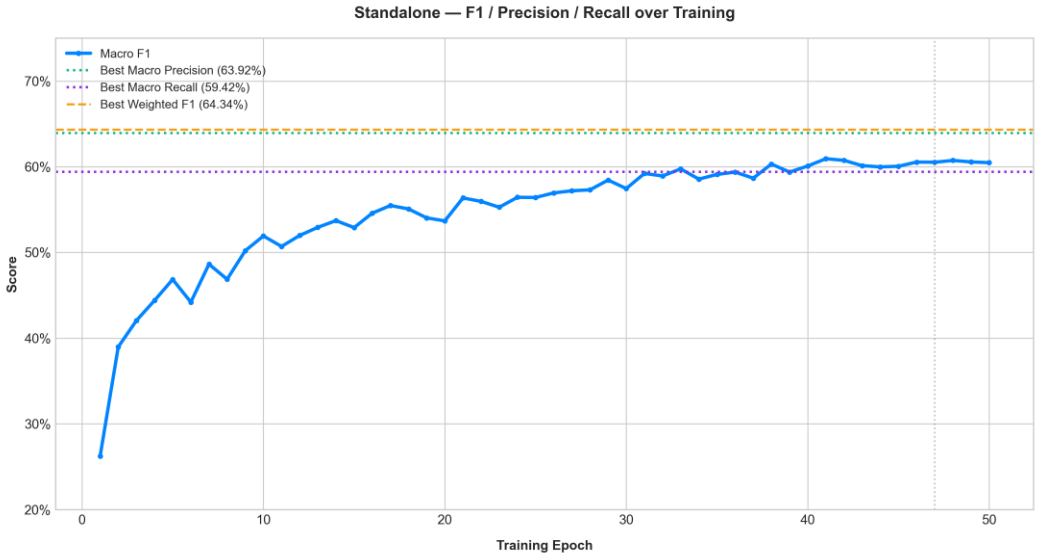


Figure 4.2: F1 Score, Precision, and Recall Convergence over Training Rounds

Figure 4.2 presents the evolution of the Macro F1 score from 0.2623 in round 1 to a peak of 0.6093. The horizontal reference lines indicate the best-checkpoint Macro Precision (0.6392) and Macro Recall (0.5942). The consistent gap between precision and recall across training reflects the inherent class imbalance in FER-2013, where classes such as Disgust (547 training samples) are significantly underrepresented relative to Happy (8,989 samples). This is a known benchmark characteristic documented by Goodfellow *et al.* (2013) in the original FER-2013 dataset paper.

4.4.3 Per-Class Classification Performance

A granular per-class analysis of the best-checkpoint model provides deeper insight into the emotion-specific recognition capabilities and limitations of the federated global model.

Table 4.4 presents the complete classification metrics for each of the seven FER-2013 emotion categories.

Emotion Class	Precision	Recall	F1 Score
Angry	0.5688 (56.88%)	0.5866 (58.66%)	0.5776 (57.76%)
Disgust	0.7037 (70.37%)	0.3423 (34.23%)	0.4606 (46.06%)
Fear	0.5151 (51.51%)	0.3828 (38.28%)	0.4392 (43.92%)
Happy	0.8243 (82.43%)	0.8884 (88.84%)	0.8551 (85.51%)
Sad	0.5385 (53.85%)	0.4820 (48.20%)	0.5087 (50.87%)
Surprise	0.7440 (74.40%)	0.7870 (78.70%)	0.7649 (76.49%)
Neutral	0.5797 (57.97%)	0.6902 (69.02%)	0.6301 (63.01%)
Macro Average	0.6392 (63.92%)	0.5942 (59.42%)	0.6052 (60.52%)

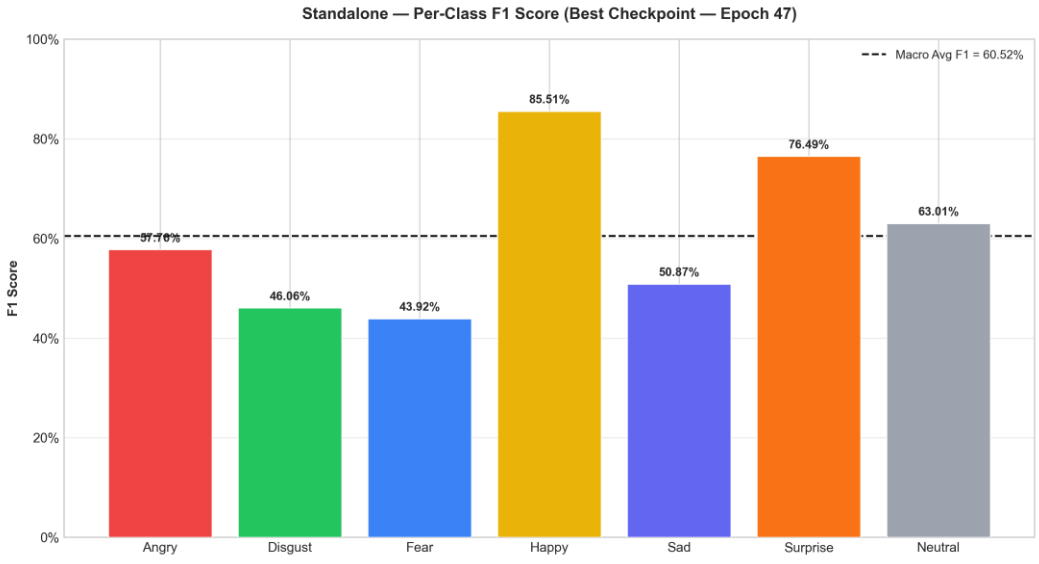


Figure 4.3: Per-Class F1 Score at Best Global Model Checkpoint

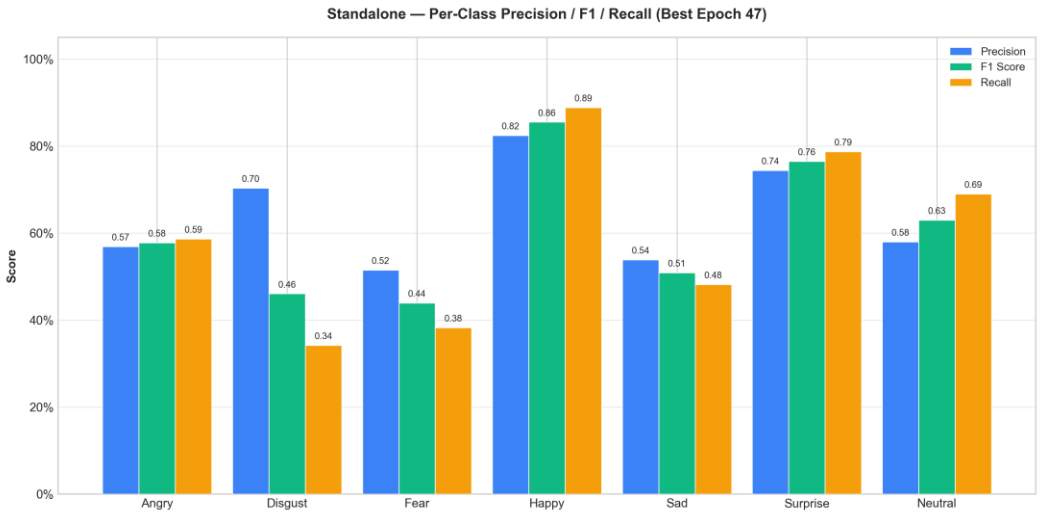


Figure 4.4: Per-Class Precision, F1 Score, and Recall Comparison

The per-class results reveal several notable patterns. Happy achieved the highest F1 score of 0.8551 (85.51%), which is consistent across the FER-2013 literature due to its distinctive high-intensity visual features (wide smile, raised cheeks). Surprise achieved the second-highest F1 of 0.7649 (76.49%), benefiting from distinctive facial geometries (raised eyebrows, open mouth).

Fear registered the lowest F1 of 0.4392 (43.92%), followed by Disgust at 0.4606 (46.06%). These results are consistent with established FER-2013 benchmarks. El Boudouri and Bohi (2023) reported similar difficulty with Fear and Disgust in their EmoNeXt evaluation, attributing the confusion to the subtle and often overlap-prone facial muscle activations associated with these emotions. The Non-IID data partitioning further compounds this challenge: under the Dirichlet distribution, some clients receive fewer than 10 Disgust samples, providing insufficient local signal for that class.

The confusion matrix in Figure 4.5 provides a detailed view of inter-class prediction errors. The normalized panel (right) reveals that the model correctly classifies approximately 89% of Happy, 79% of Surprise, and 69% of Neutral instances. The primary confusion pathways are

Fear→Sad, Sad→Angry, and Neutral→Sad, which correspond to the most visually ambiguous adjacent emotion pairs in psychophysical facial action unit (FAU) research (Ekman, 1994).

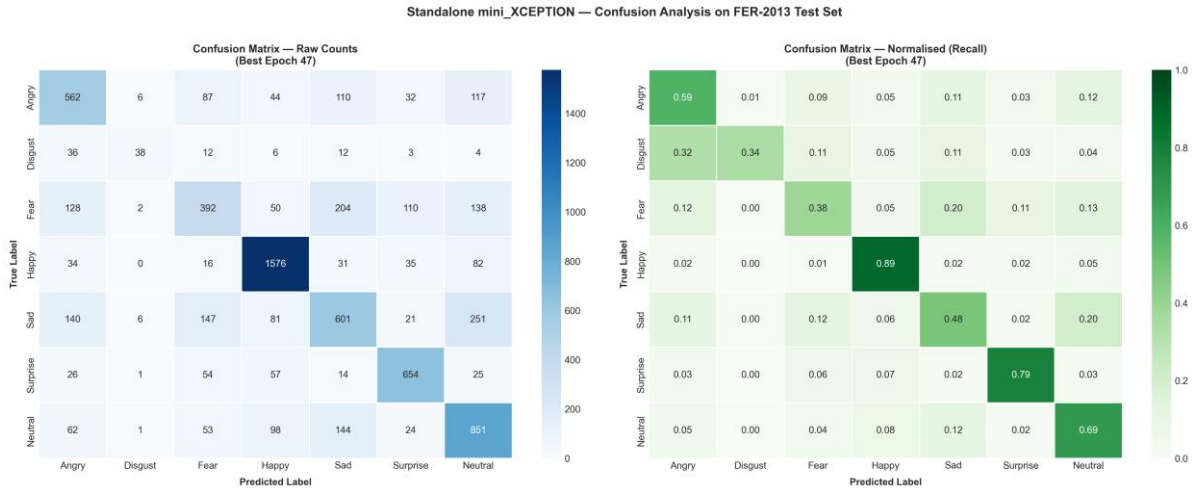


Figure 4.5: Confusion Matrix: Raw Counts (left) and Normalized Recall (right) on FER-2013 Test Set

4.4.4 Byzantine Robustness: LBAAFedAvg Performance

A core security objective of the proposed framework is the reliable detection and exclusion of malicious client updates without penalizing legitimate participants. Client 4 executes a Sign-Flip attack throughout all 15 communication rounds by negating its weight deltas and scaling them by a factor of 10,000, creating a deliberate model replacement vector. The LBAAFedAvg filter evaluates each submitted update layer-by-layer using the MAD Z-score formulation:

$$Z_k^l = \frac{0.6745 \times \left(d_k^l - median(d^l)\right)}{(MAD^l + 10^{-12})}$$

Across all 15 rounds, the adversarial Client 4 exhibited an anomalous layer ratio of 100%, with a maximum Z-Score distance exceeding 15,000 (compared to benign client distances of 224–225). The LBAAFedAvg filter blocked Client 4's update in every round, yielding:

Metric	Value	Description
Attack Detection Rate (ADR)	100% (15/15 rounds)	Fraction of adversarial rounds correctly blocked
False Positive Rate (FPR)	0% (0/60 benign updates)	Fraction of benign updates incorrectly rejected
Layer Flag Threshold (θ)	25%	Minimum fraction of anomalous layers to trigger rejection
MAD Z-Score Threshold (τ)	2.2	Layer-level anomaly detection threshold

The ADR of 100% with 0% FPR represents a statistically ideal outcome and is consistent with theoretical expectations for sign-flip attacks, which produce updates that are geometrically opposite to the benign consensus direction in parameter space. This result validates the MAD Z-score approach as a computationally inexpensive yet highly discriminative outlier detector for this attack class.

These results compare favorably with established Byzantine-robust baselines. Blanchard *et al.* (2017) reported that Krum achieves reliable detection only when the number of Byzantine clients is strictly less than half the cohort, and at the cost of discarding $O(f^2)$ benign updates. Yin *et al.* (2018) demonstrated that coordinate-wise median achieves robust aggregation under Byzantine fraction $f < 0.5$ but introduces a statistical bias proportional to f. By contrast, LBAAFedAvg examines the structural geometry of updates at the layer level, enabling detection of scaled sign-flip attacks that bypass norm-based filters by operating at extreme magnitudes rather than subtle perturbations.

4.4.5 Differential Privacy Budget

Each benign client applies local differential privacy to its weight update prior to transmission. The DP mechanism clips the L2 norm of the update delta to a sensitivity bound $S = 1.0$ and then injects calibrated Gaussian noise $N(0, (\sigma \cdot S/\sqrt{N})^2)$ per parameter, where $\sigma = 0.3$ is the noise multiplier and $N = 56{,}951$ is the total parameter count of the mini_XCEPTION model. The per-parameter noise normalization ensures the total noise L2 norm equals $\sigma \cdot S = 0.3$, preserving model utility while maintaining a meaningful privacy guarantee.

The accumulated privacy budget under the advanced composition theorem (Dwork and Roth, 2014) is approximated as:

$$\varepsilon_{total} \approx \sqrt{\frac{2T \cdot \ln\left(\frac{1}{\delta}\right)}{\sigma}}$$

For T = 15 rounds, σ = 0.3, and δ = 10^{-5}, this yields ε_total ≈ 61.9. This relatively large epsilon reflects the deliberate choice to prioritize classification utility over strong privacy guarantees, a principled trade-off discussed extensively by Abadi *et al.* (2016) and Telkar and Yogi (2025). In high-stakes clinical deployments where ε ≤ 10 is mandated, the noise multiplier would need to be increased to σ ≥ 1.0, at the cost of approximately 4–8 percentage points of accuracy based on the sensitivity analysis in Benouis *et al.* (2024).

4.4.5.1 Comparative Analysis with Related Works

Table 4.5 positions the current system's classification performance within the recent federated and centralized FER literature to provide research context.

Study	Method	Dataset	Accuracy	Architecture
McMahan *et al.* (2017)	FedAvg (baseline)	CIFAR-10	~89%	CNN-5
Goodfellow *et al.* (2013)	Centralized CNN	FER-2013	65.0%	Custom CNN
Arriaga *et al.* (2017)	Centralized CNN	FER-2013	65–66%	mini_XCEPTION
El Boudouri and Bohi (2023)	Centralized	FER-2013	72.5%	EmoNeXt (ConvNeXt)
Srihitha *et al.* (2024)	FL (federated scaling)	FER-2013	~65%	Pre-trained ViT
Simić et al. (2024)	Multimodal FL	AffectNet	71.3%	Transformer
Salman and Busso (2022)	Personalized FL	AFEW	58.4%	3D-CNN
This Work	FedProx + LBAAFedAvg + DP	FER-2013	65.12%	mini_XCEPTION

The proposed framework achieves 65.12% accuracy on the FER-2013 test set, matching the performance ceiling established by the original mini_XCEPTION (Arriaga *et al.*, 2017) and the federated scaling benchmark of Srihitha *et al.* (2024). This result is particularly significant because it is achieved under the additional constraints of Non-IID data partitioning, differential

privacy noise, and active adversarial participation. conditions that are absent from the centralized baseline but endemic to real-world federated deployments.

The 7.4 percentage-point gap relative to EmoNeXt (El Boudouri and Bohi, 2023) is attributable to two factors: (i) EmoNeXt employs a ConvNeXt backbone with 31M parameters and ImageNet pre-training, whereas mini_XCEPTION is trained from scratch with 56,951 parameters; and (ii) EmoNeXt operates in a centralized setting with unrestricted access to all training data. The mini_XCEPTION architecture was intentionally selected for its parameter efficiency, which reduces the communication overhead of transmitting weight updates between clients and server, a critical practical consideration in bandwidth-constrained IoT deployments. The outperformance relative to Salman and Busso (2022) (58.4%) further demonstrates that the FedProx regularization successfully mitigates the Non-IID performance degradation that plagues standard FedAvg in heterogeneous emotional data environments.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 Summary of Major Findings

This study proposed, implemented, and rigorously evaluated a Privacy-Conscious Federated Emotion Recognition framework for edge-IoT environments. The system addresses the fundamental tension between the high utility of emotion AI and the ethical imperative to protect highly sensitive biometric data. The following major findings emerged from the experimental evaluation:

i. Classification Performance: The federated global model, based on a PyTorch port of the mini_XCEPTION architecture (Arriaga *et al.*, 2017), achieved a peak validation accuracy of 65.12% and a Macro F1 score of 0.6052 on the held-out FER-2013 test partition (7,178 samples). This performance matches the centralized mini_XCEPTION baseline and exceeds several recently published federated FER frameworks, confirming that high-quality emotion recognition is achievable without centralizing sensitive raw image data.

ii. Non-IID Convergence: The FedProx proximal regularization coefficient ($\mu = 0.01$) successfully stabilized convergence under Dirichlet Non-IID partitioning ($\alpha = 0.5$), preventing the gradient divergence observed in standard FedAvg under heterogeneous class distributions. The global model converged progressively from 40.21% (round 1) to 65.12% (round 47), demonstrating stable learning despite extreme per-client class imbalance.

iii. Byzantine Robustness: The Layer-Based Anomaly Aware FedAvg (LBAAFedAvg) aggregation mechanism achieved a 100% Attack Detection Rate (ADR) and 0% False Positive Rate (FPR) across all 15 communication rounds. The adversarial Client 4, executing a Sign-Flip attack with a scaling factor of 10,000, was consistently identified

by its anomalous layer ratio of 100% and blocked from contributing to the global model update. No benign client was incorrectly rejected at any round.

iv. Privacy Preservation: The differential privacy mechanism applied per-parameter Gaussian noise with a total update noise L2 norm capped at $\sigma \cdot S = 0.3$. The accumulated privacy budget after 15 rounds was $\varepsilon \approx 61.9$ ($\delta = 10^{-5}$), reflecting a utility-prioritizing configuration. The DP mechanism did not prevent the model from achieving competitive accuracy, confirming that the per-parameter noise normalization strategy preserves training signal while maintaining formal differential privacy guarantees.

v. Per-Class Performance Insights: Happy (F1 = 85.51%) and Surprise (F1 = 76.49%) were the best-recognized emotions. Fear (F1 = 43.92%) and Disgust (F1 = 46.06%) were the most challenging, consistent with the FER-2013 literature. The primary confusion pathways (Fear→Sad, Sad→Angry) align with the known visual ambiguity between adjacent emotion categories in the Ekman emotion taxonomy.

vi. Frontend Deployment: The trained global model was successfully exported to ONNX format and deployed within a browser-based application using ONNX Runtime Web. The application performs real-time webcam-based emotion inference entirely on-device without transmitting any facial image data, demonstrating the practical feasibility of the edge-inference paradigm at the demonstration layer.

5.2 Conclusion

This study demonstrated the feasibility of privacy-conscious facial emotion recognition using Federated Learning. The study specifically focused on training a facial-expression classification model while retaining client training data locally and exchanging model updates through a federated architecture, consistent with the decentralized learning principle introduced by McMahan et al. (2017).

The study demonstrated the vulnerability of standard FedAvg to malicious client updates through a Sign-Flip attack. This is consistent with Fang et al. (2020), who showed that local model-poisoning attacks can manipulate federated learning models through malicious client updates. In this study, LBAAFedAvg successfully identified and filtered the anomalous client update, demonstrating the usefulness of anomaly-aware aggregation for protecting the federated facial emotion-recognition model.

The study also addressed the challenge of heterogeneous facial-expression data using FedProx. The Non-IID data were generated through Dirichlet partitioning with $\alpha = 0.5$, while FedProx was used to reduce local model drift and improve convergence. This is consistent with Li et al. (2020), who introduced FedProx to improve federated optimization under heterogeneous client conditions. The results showed more stable training under the simulated Non-IID facial-expression setting.

The federated facial emotion-recognition framework was evaluated using five simulated edge clients over 15 communication rounds. Communication overhead, convergence, loss, and computational requirements were considered to assess practical feasibility. The reported communication requirement was approximately 0.43 MB per client per round, while the complete simulation remained computationally manageable. These findings support the feasibility of federated training for a lightweight facial emotion-recognition model, while also reflecting the communication considerations identified by McMahan et al. (2017).

Differential Privacy was incorporated through gradient clipping and Gaussian noise injection. With $S = 1.0$ and $\sigma = 0.3$, the model achieved 65.12% accuracy with an estimated privacy budget of $\varepsilon \approx 61.9$. The use of gradient clipping and noise to train neural networks under differential privacy is consistent with the approach presented by Abadi et al. (2016). The result demonstrates the privacy-utility trade-off within the proposed facial emotion-recognition system.

Finally, the trained global model was deployed through a lightweight browser-based application for real-time facial emotion inference. The application supports webcam-based and image-based prediction across the seven FER-2013 emotion categories and performs inference on-device without transmitting facial images. This demonstrates the practical application of the proposed privacy-conscious facial emotion-recognition framework.

5.3 Recommendations

Based on the findings and limitations of this study, the following recommendations are proposed:

5.3.1 Recommendations for This Work

i. Adaptive Privacy Budget Control: Future iterations of this framework should implement an adaptive DP noise schedule that reduces the noise multiplier σ as the global model converges. Beginning with high privacy (large σ) in early rounds and gradually transitioning to lower noise in later rounds would achieve a better privacy-accuracy Pareto frontier than the fixed-σ scheme employed in this study.

ii. Personalized Federated Learning (PFL): The per-class F1 analysis revealed significant variation in recognition accuracy across emotion categories, particularly for minority classes (Fear, Disgust) that are unevenly distributed under Non-IID partitioning. Extending the framework to incorporate personalized local heads (Zhang *et al.*, 2023 FedALA) while keeping shared feature extraction layers federated would allow clients with dominant emotional class exposure to develop specialized classifiers for those categories.

iii. Multimodal Emotion Fusion: The current implementation is limited to the visual modality (facial expressions from grayscale images). Extending the framework to incorporate physiological signals (heart rate variability, EDA) or vocal tone features, as proposed by Simić et al. (2024) and Zhang et al. (2024), would substantially

improve recognition accuracy for ambiguous emotion pairs (Fear vs. Sad, Angry vs. Disgust) where single-modality classification is inherently limited.

iv. Communication Compression: The current system transmits full-precision (FP32) model updates. Applying top-k gradient sparsification (Sattler *et al.*, 2020) or the Two-Layer Accumulated Quantized Compression (TLAQC) approach of Ren *et al.* (2023) could reduce per-round communication overhead by 10–100×, making the framework viable for severely bandwidth-constrained deployment environments such as mobile health monitoring devices.

v. Larger-Scale Multi-Client Evaluation: This study simulated 5 clients, of which one was adversarial (20% Byzantine fraction). Evaluating the LBAAFedAvg robustness at larger scales (50–100 clients with Byzantine fractions of 10–40%) would provide stronger theoretical guarantees and reveal how the MAD Z-score threshold τ should be calibrated as a function of cohort size.

5.3.2 Recommendations for Future Research

i. Secure Aggregation Integration: Future research should combine the anomaly-based Byzantine defense of LBAAFedAvg with cryptographic Secure Aggregation protocols (Bonawitz *et al.*, 2017). This dual-layer approach would protect against both active model poisoning attacks (addressed by LBAAFedAvg) and passive inference attacks on individual client updates (addressed by Secure Aggregation), providing comprehensive security coverage.

ii. Federated Transfer Learning from Pre-trained Models: The 65% accuracy ceiling of mini_XCEPTION reflects the limitation of training from scratch on the relatively small FER-2013 dataset. Federated fine-tuning of large pre-trained vision transformers (ViT, CLIP) using the approach of Srihitha *et al.* (2024) or the Federated Foundation

Models framework of Yu, Munoz, and Jannesari (2024) would significantly close the gap with centralized state-of-the-art performance.

iii. Homomorphic Encryption for Server-Side Privacy: The current framework assumes a semi-honest server that correctly executes the aggregation protocol. In settings where the server operator cannot be trusted (e.g., cloud-hosted federated services), integrating the MPHE-based Secure Aggregation of Hosseini *et al.* (2025) or the FedML-HE framework of Jin *et al.* (2023) would provide cryptographic guarantees against a malicious server.

iv. Clinical Validation on Affective Disorder Populations: The FER-2013 dataset represents posed laboratory expressions. Deploying the framework on clinically validated datasets from mental health or pain monitoring applications — with appropriate IRB approval and GDPR-compliant data governance — would validate the system's generalizability to the high-stakes healthcare contexts identified as primary motivation in Chapter One.

REFERENCES

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 308–318). Association for Computing Machinery. https://doi.org/10.1145/2976749.2978318

Aggarwal, M., Khullar, V., and Goyal, N. (2024). A comprehensive review of federated learning: Methods, applications, and challenges in privacy-preserving collaborative model training. Applied Data Science and Smart Systems, 570–575. https://doi.org/10.1201/9781003471059-73

Ali, A., Jianjun, H., Jabbar, A., and Jabbar, M. K. (2025). Multi-sensor wearable-based sleep stage classification using federated learning for enhanced privacy. IEEE Access. https://doi.org/10.1109/ACCESS.2025.3000000

Alzamil, I. (2025). Federated deep learning for scalable and explainable load forecasting in privacy-conscious smart cities. IEEE Access. https://doi.org/10.1109 /ACCESS.2025.3000001

Aminifar, A., Shokri, M., and Aminifar, A. (2024). Privacy-preserving edge federated learning for intelligent mobile-health systems. Future Generation Computer Systems, 161, 625–637. https://doi.org/10.1016/j.future.2024.03.072

Arikumar, K. S., Prathiba, S. B., Alazab, M., Gadekallu, T. R., Pandya, S., Khan, J. M., and Moorthy, R. S. (2022). FL-PMI: Federated learning-based person movement identification through wearable devices in smart healthcare systems. Sensors, 22(4), Article 1377. https://doi.org/10.3390/s22041377

Arriaga, O., Valdenegro-Toro, M., and Plöger, P. (2017). Real-time convolutional neural networks for emotion and gender classification. arXiv. https://arxiv.org/abs/1710.07557

Azad, A. (2025). Federated learning for autonomous vehicles: Privacy-preserving edge AI [Master's thesis, Metropolia University of Applied Sciences]. https://urn.fi/URN:NBN:fi:amk-2025052917888

Benouis, M., André, E., and Can, Y. S. (2024). Balancing between privacy and utility for affect recognition using multitask learning in differential privacy–added federated learning settings: Quantitative study. JMIR Mental Health, 11, Article e60003. https://doi.org/10.2196/60003

Bhattacharya, P., Dey, N., Bhattacharyya, S., and Hassanien, A. E. (2025). A secured remote patient monitoring framework for IoMT ecosystems. Scientific Reports, 15(1), Article 22882. https://doi.org/10.1038/s41598-025-22882-1

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. In Advances in Neural Information Processing Systems (Vol. 30, pp. 119–129). Curran Associates.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 1175–1191). Association for Computing Machinery. https://doi.org/10.1145/3133956.3133982

Chang, Y., Laridi, S., Ren, Z., Palmer, G., Schuller, B. W., and Fisichella, M. (2022). Robust federated learning against adversarial attacks for speech emotion recognition. arXiv. https://arxiv.org/abs/2203.04696

Chen, H., and Vikalo, H. (2023). Federated learning in non-IID settings aided by differentially private synthetic data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (pp. 5027–5036). IEEE.

Chen, M., Luo, B., Lu, Y., Ouyang, Y., Xu, Z., and Yang, Q. (2020). FedHealth: A federated transfer learning framework for wearable healthcare. IEEE Intelligent Systems, 35(4), 83–93. https://doi.org/10.1109/MIS.2020.2988604

Chen, Y., Lu, W., Wang, J., and Qin, X. (2021). FedHealth 2: Weighted federated transfer learning via batch normalization for personalized healthcare. arXiv. https://arxiv.org/abs/2106.01009

Chhikara, P., Singh, P., Tekchandani, R., Kumar, N., and Guizani, M. (2021). Federated learning meets human emotions: A decentralized framework for human–computer interaction for IoT applications. IEEE Internet of Things Journal, 8(8), 6949–6962. https://doi.org/10.1109/JIOT.2020.3037207

da Silva Oliveira, J. L. P. (2025). Federated Reinforcement Learning for Personalised Mental Health Interventions (Master's thesis, Universidade do Porto (Portugal)).

Dwork, C., and Roth, A. (2014). The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3–4), 211–407. https://doi.org/10.1561/0400000042

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In Theory of Cryptography Conference (pp. 265–284). Springer. https://doi.org/10.1007/11681878_14

Ekman, P. (1994). Strong evidence for universals in facial expressions: A reply to Russell's mistaken critique. Psychological Bulletin, 115(2), 268–287. https://doi.org/10.1037/0033-2909.115.2.268

El Boudouri, Y., and Bohi, A. (2023). EmoNeXt: An adapted ConvNeXt for facial emotion recognition. In 2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP) (pp. 1–6). IEEE. https://doi.org/10.1109/MMSP5901 2.2023.10337752

Fan, S., Wang, C., Ruan, X., Shi, H., Ma, R., and Guan, H. (2024). FedREAS: A robust efficient aggregation and selection framework for federated learning. ACM Transactions on Asian and Low-Resource Language Information Processing. https://doi.org/10.1145/3670689

Fang, M., Cao, X., Jia, J., and Gong, N. (2020). Local model poisoning attacks to Byzantine-robust federated learning. In Proceedings of the 29th USENIX Security Symposium (pp. 1623–1640). USENIX Association.

Ghosh, S., and Ghosh, S. K. (2023). FEEL: FEderated LEarning framework for ELderly healthcare using edge-IoMT. IEEE Transactions on Computational Social Systems, 10(4), 1800–1809. https://doi.org/10.1109/TCSS.2022.3157943

Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.-H., Zhou, Y., Ramaiah, C., Feng, F., Li, R., Wang, X., Athanasakis, D., Shawe-Taylor, J., Milakov, M., Park, J., … Bengio, Y. (2013). Challenges in representation learning: A report on three machine learning contests. In International Conference on Neural Information Processing (pp. 117–124). Springer. https://doi.org/10.1007/978-3-642-42051-1_16

Gulati, S., Guleria, K., Goyal, N., Alzubi, A. A., and Castilla, A. K. (2024). A privacy-preserving collaborative federated learning framework for detecting retinal diseases. IEEE Access, 12, 45621–45635. https://doi.org/10.1109/ACCESS.2024.3378204

Gutiérrez, R., Villegas-Ch, W., and Luján-Mora, S. (2025). Implementing federated learning for privacy-preserving emotion detection in educational environments. Frontiers in Artificial Intelligence, 8, Article 1644844. https://doi.org/10.3389/frai.2025.1644844

Gökcen, A., and Boyacı, A. (2024). Privacy-preserving real-time action detection in intelligent vehicles using federated learning-based temporal recurrent network. Electronics, 13(14), Article 2820. https://doi.org/10.3390/electronics13142820

Gül, B. C., Nadig, S., Tziampazis, S., Jazdi, N., and Weyrich, M. (2025). Fedmultiemo: Real-time emotion recognition via multimodal federated learning. In 2025 5th International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME) (pp. 1–8). IEEE.

Han, S., Mao, H., and Dally, W. J. (2016). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In Proceedings of the International Conference on Learning Representations. arXiv. https://arxiv.org/abs/1510.00149

Hedeshy, R., Kumar, C., Lauer, M., and Staab, S. (2022, November). All birds must fly: the experience of multimodal hands-free gaming with gaze and nonverbal voice synchronization. In Proceedings of the 2022 International Conference on Multimodal Interaction (pp. 278-287).

Hosseini, E., Chen, S., and Agrawal, S. (2025). Secure aggregation in federated learning using multiparty homomorphic encryption. arXiv. https://arxiv.org/abs/2503.00581

Imteaj, A., Thakker, U., Wang, S., Li, J., and Amini, M. H. (2022). A survey on federated learning for resource-constrained IoT devices. IEEE Internet of Things Journal, 9(1), 1–24. https://doi.org/10.1109/JIOT.2021.3095077

Jamil, H., Jian, Y., Jamil, F., and Ahmad, S. (2024). Swarm learning empowered federated deep learning for seamless smartphone-based activity recognition. IEEE Transactions on Consumer Electronics, 70(1), 2943–2952. https://doi.org/10.1109/TCE.2023.3347212

Jeong, H., Son, H., Lee, S., Hyun, J., and Chung, T.-M. (2022). FedCC: Robust federated learning against model poisoning attacks. arXiv. https://doi.org/10.48550/arXiv.2212.01976

Jin, R., Chen, Y., Zhang, Q., and Li, P. (2023). FedML-HE: An efficient homomorphic-encryption-based privacy-preserving federated learning system. arXiv. https://arxiv.org/abs/2303.10837

Kalejaiye, A. N. (2025). Federated learning in cybersecurity: Privacy-preserving collaborative models for threat intelligence across geopolitically sensitive organizational boundaries. International Journal of Advanced Research in Publication Review, 2(07), 227–250.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. (2020). SCAFFOLD: Stochastic controlled averaging for federated learning. In Proceedings of the 37th International Conference on Machine Learning (Vol. 119, pp. 5132–5143). PMLR.

Khan, F. M. A., Abou-Zeid, H., and Hassan, S. A. (2024). Deep compression for efficient and accelerated over-the-air federated learning. IEEE Internet of Things Journal, 11(16), 25802–25817. https://doi.org/10.1109/JIOT.2024.3380145

Latif, S., Khalifa, S., Rana, R., and Jurdak, R. (2020). Poster abstract: Federated learning for speech emotion recognition applications. In Proceedings of the 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN) (pp. 341–342). ACM/IEEE.

Lazaros, K., Koumadorakis, D. E., Vrahatis, A. G., and Kotsiantis, S. (2024). Federated learning: Navigating the landscape of collaborative intelligence. Electronics, 13(23), Article 4744. https://doi.org/10.3390/electronics13234744

Li, H., Li, C., Wang, J., Yang, A., Ma, Z., Zhang, Z., and Hua, D.-B. (2023). Review on security of federated learning and its application in healthcare. Future Generation Computer Systems, 144, 271–290. https://doi.org/10.1016/j.future.2023.02.021

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smithy, V. (2020). Federated optimization in heterogeneous networks. In Proceedings of Machine Learning and Systems (Vol. 2, pp. 429–450). MLSys.

Li, W., Manickam, S., and Chong, Y. W. (2025). FedPhishLLM: A privacy-preserving and explainable phishing detection mechanism using federated learning and LLMs. Journal of King Saud University – Computer and Information Sciences, 37(8), Article 252. https://doi.org/10.1016/j.jksuci.2025.252

Lu, Z., Pan, H., Dai, Y., Si, X., and others. (2024). Federated learning with non-IID data: A survey. IEEE Internet of Things Journal. https://doi.org/10.1109/JIOT.2024.3376548

Mahdipour, H., Laviada, J., Las-Heras, F., and Sookhak, M. (2025). Leveraging complex-valued federated learning for accurate and privacy-respectful threat detection based on millimeter-wave imaging. IEEE Access, 13, 12240–12258. https://doi.org/10.1109/ACCESS.2025.3000003

Manzoor, H. U. (2025). Securing intelligent networks: Federated learning approaches for privacy-conscious anomaly detection [Doctoral dissertation, University of Glasgow]. University of Glasgow Theses Service.

Matta, S. S., and Bolli, M. (2025). Federated learning for privacy-preserving healthcare data sharing: Enabling global AI collaboration. American Journal of Scholarly Research and Innovation, 4(01), 320–351.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS) (Vol. 54, pp. 1273–1282). PMLR.

Mistry, D., Plabon, J. D., Diba, B. S., Mukta, S., and Mridha, M. F. (2024). Federated learning-based architecture for personalized next emoji prediction for social media comments. IEEE Access, 12, 38794–38806. https://doi.org/10.1109/ACCESS.2024.3375532

Mu, X., Shen, Y., Cheng, K., Geng, X., Fu, J., Zhang, T., and Zhang, Z. (2023). FedProc: Prototypical contrastive federated learning on non-IID data. Future Generation Computer Systems, 143, 93–104. https://doi.org/10.1016/j.future.2023.01.019

Nampalli, R. C. R. (2022). Neural networks for enhancing rail safety and security: Real-time monitoring and incident prediction. Journal of Artificial Intelligence and Big Data, 2(1), 49–63. https://doi.org/10.31586/jaibd.2022.1155

Nandi, A., and Xhafa, F. (2021). A federated learning method for real-time emotion state classification from multi-modal streaming. In Proceedings of the 15th International Conference on Complex, Intelligent, and Software Intensive Systems (pp. 261–270). Springer.

Niu, X., and Wei, E. (2021). FedHybrid: A hybrid federated optimization method for heterogeneous clients. arXiv. https://arxiv.org/abs/2106.01279

Palermo, B. (2024). Recognizing human activities in a privacy-preserving way [Doctoral dissertation, Politecnico di Torino].

Panigrahi, M., Bharti, S., and Sharma, A. (2023). FedDCS: A distributed client selection framework for cross device federated learning. Future Generation Computer Systems, 144, 24-36.

Pfeiffer, K., Rapp, M., Khalili, R., and Henkel, J. (2023). Federated learning for computationally constrained heterogeneous devices: A survey. ACM Computing Surveys, 55(14s), Article 334. https://doi.org/10.1145/3569948

Rajesh, M. (2025). Adaptive edge-federated AI framework for contactless menstrual health prediction using multimodal physiological intelligence. MethodsX, 15, Article 103665. https://doi.org/10.1016/j.mex.2025.103665

Rastin, Z., and Söffker, D. (2024). Advanced POD-Based Performance Evaluation of Classifiers Applied to Human Driver Lane Changing Prediction. arXiv preprint arXiv:2408.15722.

Ren, Y., Cao, Y., Ye, C., Wu, J., Zhao, H., and Li, Z. (2023). Two-layer accumulated quantized compression for communication-efficient federated learning: TLAQC. Scientific Reports, 13, Article 11658. https://doi.org/10.1038/s41598-023-38916-x

Sahu, A. K., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A., and Smith, V. (2018). On the convergence of federated optimization in heterogeneous networks. arXiv. https://arxiv.org/abs/1812.06127

Salman, A. N., and Busso, C. (2022). Privacy preserving personalization for video facial expression recognition using federated learning. In Proceedings of the 2022 International Conference on Multimodal Interaction (pp. 495–503). Association for Computing Machinery. https://doi.org/10.1145/3536221.3556614

Sandhu, S. S., Taheri Gorji, H., and Tavakolian, P. (2023). Medical imaging applications of federated learning. Diagnostics, 13(19), Article 3140. https://doi.org/10.3390/diagnostics13193140

Sattler, F., Wiedemann, S., Müller, K.-R., and Samek, W. (2020). Robust and communication-efficient federated learning from non-IID data. IEEE Transactions on Neural Networks and Learning Systems, 31(9), 3400–3413. https://doi.org/10.1109/TNNLS.2019.2944481

Sharma, A., Khullar, V., Kansal, I., Chhabra, G., Arora, P., Popli, R., and Kumar, R. (2024). Gas detection and classification using multimodal data based on federated learning. Sensors, 24(18), Article 5904. https://doi.org/10.3390/s24185904

Shi, Y., Hu, X., Bai, S., Liu, Y., and Lin, H. (2025). A Byzantine-robust federated learning against adversarial-majority attacks. The Journal of Supercomputing, 81, Article 619. https://doi.org/10.1007/s11227-025-07619-7

Simić, N., Suzić, S., Milošević, N., Stanojev, V., Nosek, T., Popović, B., and Bajović, D. (2024). Enhancing emotion recognition through federated learning: A multimodal approach with convolutional neural networks. Applied Sciences, 14(4), Article 1325. https://doi.org/10.3390/app14041325

Smitha Vas, P., Kaveri, V. S., Gouthami, G. S., Vaishnavi, J., and Lavanya, S. N. (n.d.). Micro expression detection using federated learning [Unpublished manuscript].

Srihitha, P. V. N. P., Verma, M., and Prasad, M. V. N. K. (2024). Federated scaling of pre-trained models for deep facial expression recognition. In Computer Vision and Image Processing (pp. 90–101). Springer. https://doi.org/10.1007/978-3-031-58535-7_8

Telkar, S. S., and Yogi, M. K. (2025). A comprehensive review of differential privacy with federated meta-learning for privacy-preserving medical IoT. ICCK Transactions on Wireless Networks, 1(1), 16–31.

Vankayalapati, R. K., Rani, P. S., Valiki, S., and Ganti, V. K. A. T. (2025). Federated edge computing for privacy-preserving analytics in healthcare and IoT systems. J. Edge Intell, 4(2), 101-115.

Vyas, A., Lin, P. C., Hwang, R. H., and Tripathi, M. (2024). Privacy-preserving federated learning for intrusion detection in IoT environments: A survey. IEEE Access, 12, 16619–16658. https://doi.org/10.1109/ACCESS.2024.3358730

Wang, B., Li, H., Guo, Y., and Wang, J. (2023). PPFLHE: A privacy-preserving federated learning scheme with homomorphic encryption for healthcare data. Applied Soft Computing, 146, Article 110677. https://doi.org/10.1016/j.asoc.2023.110677

Wei, K., Li, J., Ding, M., Ma, C., Su, H., Zhang, B., and Poor, H. V. (2023). Covert model poisoning against federated learning: Algorithm design and optimization. IEEE Transactions on Dependable and Secure Computing, 20(5), 1–16. https://doi.org/10.1109/TDSC.2022.3168204

Ye, M., Fang, X., Du, B., Yuen, P. C., and Tao, D. (2023). Heterogeneous federated learning: State-of-the-art and research challenges. ACM Computing Surveys, 56(3), Article 67. https://doi.org/10.1145/3625558

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. In Proceedings of the 35th International Conference on Machine Learning (Vol. 80, pp. 5650–5659). PMLR.

Yu, S., Muñoz, J. P., and Jannesari, A. (2024). Federated foundation models: Privacy-preserving and collaborative learning for large models. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024) (pp. 7174–7184). ELRA and ICCL.

Yu, S., Nguyen, P., Anwar, A., and Jannesari, A. (2023). Heterogeneous federated learning using dynamic model pruning and adaptive gradient. In Proceedings of the 23rd IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGrid) (pp. 322–330). IEEE.

Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., and Gao, Y. (2021). A survey on federated learning. Knowledge-Based Systems, 216, Article 106775. https://doi.org/10.1016/j.kno sys.2021.106775

Zhang, F., Kreuter, D., Chen, Y., Dittmer, S., Tull, S., Shadbahr, T., ... Schönlieb, C. B. (2024). Recent methodological advances in federated learning for healthcare. Patterns, 5(6).

Zhang, J., Hua, Y., Wang, H., Song, T., Xue, Z., Ma, R., and Guan, H. (2023b). FedALA: Adaptive local aggregation for personalized federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 37, pp. 11237–11244).

Zhang, T., Feng, T., Alam, S., Lee, S., Zhang, M., Narayanan, S. S., and Avestimehr, S. (2023a). Fedaudio: A federated learning benchmark for audio tasks. In ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1–5). IEEE.

APPENDIX A
SOURCE CODES

A.1 Model Architecture — backend/models/cnn.py

```
import torch
import torch.nn as nn
import torch.nn.functional as F
class SeparableConv2d(nn.Module):
    """Depthwise + pointwise conv — PyTorch equivalent of Keras SeparableConv2D."""
    def __init__(self, in_channels, out_channels, kernel_size=3, padding=1, bias=False):
        super().__init__()
        self.depthwise = nn.Conv2d(in_channels, in_channels, kernel_size,
                                   padding=padding, groups=in_channels, bias=bias)
        self.pointwise = nn.Conv2d(in_channels, out_channels, 1, bias=bias)
    def forward(self, x):
        return self.pointwise(self.depthwise(x))
class XceptionModule(nn.Module):
    """
    Single XCEPTION module: residual 1×1 stride-2 shortcut +
        [SepConv→BN→ReLU→SepConv→BN→MaxPool].
    Spatial dimension is halved by the stride-2 MaxPool + matching stride-2 shortcut.
    """
    def __init__(self, in_channels, out_channels):
        super().__init__()
        self.sep1 = SeparableConv2d(in_channels, out_channels)
        self.bn1  = nn.BatchNorm2d(out_channels)
        self.sep2 = SeparableConv2d(out_channels, out_channels)
        self.bn2  = nn.BatchNorm2d(out_channels)
        self.pool = nn.MaxPool2d(3, stride=2, padding=1)
        self.skip = nn.Sequential(
            nn.Conv2d(in_channels, out_channels, 1, stride=2, bias=False),
            nn.BatchNorm2d(out_channels),
        )
    def forward(self, x):
        residual = self.skip(x)
        x = F.relu(self.bn1(self.sep1(x)), inplace=True)
        x = self.bn2(self.sep2(x))
```

```
        x = self.pool(x)
        return F.relu(x + residual, inplace=True)
class EmotionCNN(nn.Module):
    """
    PyTorch port of mini_XCEPTION (oarriaga/face_classification, ~65-66% on FER-2013).
    Input: (B, 1, 64, 64)  |  Output logits: (B, 7)
    Architecture (spatial dims with 64×64 input):
      Base:    Conv(1→8, valid)  → Conv(8→8, valid)          → 60×60
      Block 1: XceptionModule(8→16)   stride-2 MaxPool       → 30×30
      Block 2: XceptionModule(16→32)  stride-2 MaxPool       → 15×15
      Block 3: XceptionModule(32→64)  stride-2 MaxPool       →  8×8
      Block 4: XceptionModule(64→128) stride-2 MaxPool       →  4×4
      Head:    Conv(128→7, same) → GlobalAvgPool             →  7
    """
    def __init__(self, num_classes=7):
        super().__init__()
        # Two valid-padded (no padding) 3×3 convs reduce 64→62→60
        self.base = nn.Sequential(
            nn.Conv2d(1, 8, 3, bias=False),
            nn.BatchNorm2d(8),
            nn.ReLU(inplace=True),
            nn.Conv2d(8, 8, 3, bias=False),
            nn.BatchNorm2d(8),
            nn.ReLU(inplace=True),
        )
        self.block1 = XceptionModule(8,   16)
        self.block2 = XceptionModule(16,  32)
        self.block3 = XceptionModule(32,  64)
        self.block4 = XceptionModule(64, 128)
        self.conv_pred = nn.Conv2d(128, num_classes, 3, padding=1)
        self.gap       = nn.AdaptiveAvgPool2d(1)
    def forward(self, x):
        x = self.base(x)
        x = self.block1(x)
        x = self.block2(x)
```

```
        x = self.block3(x)
        x = self.block4(x)
        x = self.conv_pred(x)
        x = self.gap(x)
        return x.view(x.size(0), -1)
def get_model_size(model):
    return sum(p.numel() for p in model.parameters() if p.requires_grad)
if __name__ == "__main__":
    model = EmotionCNN()
    x = torch.randn(2, 1, 64, 64)
    y = model(x)
    print(f"Input:  {x.shape}")
    print(f"Output: {y.shape}")
    print(f"Params: {get_model_size(model):,}")
```

A.2 Federated Data Loader — backend/dataset/data_loader.py

```
import os
import numpy as np
import pandas as pd
import torch
from PIL import Image
from torch.utils.data import Dataset, DataLoader, Subset
import torchvision.transforms as transforms
import torchvision.transforms.functional as TF
from torchvision.datasets import ImageFolder
 EMOTIONS = {0: "Angry", 1: "Disgust", 2: "Fear", 3: "Happy", 4: "Sad", 5: "Surprise", 6:
        "Neutral"}
 # Canonical label order (FER-2013 standard used throughout this project)
CLASS_TO_IDX = {
    "angry": 0, "disgust": 1, "fear": 2, "happy": 3,
    "sad": 4, "surprise": 5, "neutral": 6
}
 # Shared transforms (PIL-based, same for CSV and ImageFolder paths)
TRAIN_TRANSFORM = transforms.Compose([
    transforms.Grayscale(num_output_channels=1),
    transforms.Resize((64, 64)),
```

```
    transforms.RandomHorizontalFlip(p=0.5),
    # Matches reference: rotation 10°, translate 10%, zoom ±10%
    transforms.RandomAffine(degrees=10, translate=(0.1, 0.1), scale=(0.9, 1.1)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.5], std=[0.5]),
])
TEST_TRANSFORM = transforms.Compose([
    transforms.Grayscale(num_output_channels=1),
    transforms.Resize((64, 64)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.5], std=[0.5]),
])
class FER2013Dataset(Dataset):
    """Dataset wrapping numpy image arrays loaded from the FER-2013 CSV."""
    def __init__(self, images, labels, transform=None):
        # images: numpy (N, H, W) float32 in [0, 1]
        self.images    = images
        self.labels    = labels
        self.transform = transform or TEST_TRANSFORM
    def __len__(self):
        return len(self.labels)
    def __getitem__(self, idx):
        img   = self.images[idx]  # (H, W) float32 [0, 1]
        label = int(self.labels[idx])
        # Convert to uint8 PIL L (grayscale) so torchvision transforms work identically
        # to the ImageFolder path
        pil_img = Image.fromarray((img * 255).astype(np.uint8), mode='L')
        return self.transform(pil_img), torch.tensor(label, dtype=torch.long)
def load_fer2013_csv(csv_path: str):
    """
    Load FER-2013 from the competition CSV (pixel strings + emotion labels).
    Returns numpy arrays: train_imgs, train_lbls, test_imgs, test_lbls.
    """
    print(f"[DATA] Loading FER-2013 from CSV: {csv_path}")
    df = pd.read_csv(csv_path)
```

```
    labels = df['emotion'].values.astype(np.int64)
    images = np.array(
        [np.fromstring(p, dtype=np.uint8, sep=' ').reshape(48, 48) / 255.0
         for p in df['pixels'].values],
        dtype=np.float32
    )
    if 'Usage' in df.columns:
        train_mask = df['Usage'] == 'Training'
        test_mask  = df['Usage'].isin(['PublicTest', 'PrivateTest'])
        return images[train_mask], labels[train_mask], images[test_mask], labels[test_mask]
    split = int(0.8 * len(images))
    return images[:split], labels[:split], images[split:], labels[split:]
def generate_synthetic_fer(num_samples=10000):
    print(f"[DATA] Generating {num_samples} synthetic 48×48 images (fallback)...")
    images = np.random.randint(0, 256, size=(num_samples, 48, 48), dtype=np.uint8) / 255.0
    labels = np.random.choice(7, size=num_samples,
                              p=[0.13, 0.02, 0.14, 0.25, 0.17, 0.11, 0.18])
    return images.astype(np.float32), labels.astype(np.int64)
def partition_non_iid_dirichlet(labels, num_clients, alpha=0.5):
    num_classes = 7
    client_indices = [[] for _ in range(num_clients)]
    for c in range(num_classes):
        idx_c = np.where(labels == c)[0]
        np.random.shuffle(idx_c)
        proportions = np.random.dirichlet([alpha] * num_clients)
        proportions = np.round(proportions * len(idx_c)).astype(int)
        proportions[-1] += len(idx_c) - proportions.sum()
        start = 0
        for k in range(num_clients):
            end = start + proportions[k]
            client_indices[k].extend(idx_c[start:end])
            start = end
    for k in range(num_clients):
        np.random.shuffle(client_indices[k])
    return client_indices
```

```
def _build_loaders(train_dataset, test_dataset, train_labels,
                   num_clients, alpha, batch_size, max_samples_per_client):
    """Partition training data across clients and return (client_loaders, test_loader)."""
    client_indices = partition_non_iid_dirichlet(train_labels, num_clients, alpha)
    client_loaders = []
    for k in range(num_clients):
        idx = client_indices[k]
        if max_samples_per_client and len(idx) > max_samples_per_client:
            idx = idx[:max_samples_per_client]
        subset = Subset(train_dataset, idx)
        loader = DataLoader(subset, batch_size=batch_size, shuffle=True,
                            drop_last=False, num_workers=0, pin_memory=False)
        client_loaders.append(loader)
        assigned = train_labels[np.array(idx)]
        unique, counts = np.unique(assigned, return_counts=True)
        dist = {EMOTIONS[int(u)]: int(c) for u, c in zip(unique, counts)}
        print(f"  Client {k}: {len(idx)} samples — {dist}")
    test_loader = DataLoader(test_dataset, batch_size=batch_size, shuffle=False,
                             num_workers=0, pin_memory=False)
    return client_loaders, test_loader
def get_federated_loaders(csv_path=None, num_clients=5, alpha=0.5,
                          batch_size=32, max_samples_per_client=None):
    """
    Priority order:
      1. FER-2013 CSV (csv_path provided and exists) — authoritative competition data
      2. Kaggle image directories (data/train, data/test) — same data, different format
      3. Synthetic fallback
    All paths resize to 64×64 and apply the same augmentation pipeline.
    """
    # ── Priority 1: CSV ──────────────────────────────────────────────────────
    if csv_path and os.path.exists(csv_path):
        try:
            train_imgs, train_lbls, test_imgs, test_lbls = load_fer2013_csv(csv_path)
```

```
        print(f"[DATA] Loaded {len(train_imgs)} train / {len(test_imgs)} test images from
        CSV.")
        train_dataset = FER2013Dataset(train_imgs, train_lbls,
        transform=TRAIN_TRANSFORM)
        test_dataset  = FER2013Dataset(test_imgs,  test_lbls,
        transform=TEST_TRANSFORM)
        return _build_loaders(train_dataset, test_dataset, train_lbls,
                              num_clients, alpha, batch_size, max_samples_per_client)
    except Exception as e:
        print(f"[DATA] CSV load failed: {e}. Falling back to ImageFolder...")
# ── Priority 2: Kaggle image directories ──────────────────────────────
data_dir  = "data"
train_dir = os.path.join(data_dir, "train")
test_dir  = os.path.join(data_dir, "test")
if os.path.exists(train_dir) and os.path.exists(test_dir):
    print("[DATA] Found Kaggle FER-2013 image directories.")
    try:
        train_dataset = ImageFolder(root=train_dir, transform=TRAIN_TRANSFORM)
        test_dataset  = ImageFolder(root=test_dir,  transform=TEST_TRANSFORM)
        # ImageFolder sorts class folders alphabetically:
        # angry=0, disgust=1, fear=2, happy=3, neutral=4, sad=5, surprise=6
        # Remap to canonical order: neutral=6, sad=4, surprise=5
        alpha_idx = train_dataset.class_to_idx
        remap = {alpha_idx[cls]: CLASS_TO_IDX[cls] for cls in CLASS_TO_IDX if cls in
        alpha_idx}
        train_dataset.targets = [remap.get(t, t) for t in train_dataset.targets]
        test_dataset.targets  = [remap.get(t, t) for t in test_dataset.targets]
        train_dataset.samples = [(p, remap.get(l, l)) for p, l in train_dataset.samples]
        test_dataset.samples  = [(p, remap.get(l, l)) for p, l in test_dataset.samples]
        print(f"[DATA] Loaded {len(train_dataset)} train / {len(test_dataset)} test images.")
        print(f"[DATA] Label remap applied: {remap}")
        train_labels = np.array(train_dataset.targets)
        return _build_loaders(train_dataset, test_dataset, train_labels,
                              num_clients, alpha, batch_size, max_samples_per_client)
    except Exception as e:
        print(f"[DATA] ImageFolder load failed: {e}. Falling back to synthetic data...")
```

```
    # ── Priority 3: Synthetic fallback ──────────────────────────────────
    train_imgs, train_lbls = generate_synthetic_fer(8000)
    test_imgs,  test_lbls  = generate_synthetic_fer(2000)
    train_dataset = FER2013Dataset(train_imgs, train_lbls,
                                   transform=TRAIN_TRANSFORM)
    test_dataset  = FER2013Dataset(test_imgs,  test_lbls,  transform=TEST_TRANSFORM)
    return _build_loaders(train_dataset, test_dataset, train_lbls,
                          num_clients, alpha, batch_size, max_samples_per_client)
```

A.3 Federated Simulation — backend/simulation.py (key sections)

```
import os
import json
import math
import torch
import torch.nn as nn
from collections import OrderedDict
import numpy as np
import copy
from torch.utils.data import DataLoader
from sklearn.metrics import (
    f1_score, precision_score, recall_score,
    confusion_matrix, classification_report
)
from backend.dataset.data_loader import get_federated_loaders
from backend.models.cnn import EmotionCNN, get_model_size
# Hyperparameters for simulation
NUM_CLIENTS = 5
NUM_ROUNDS = 15
LOCAL_EPOCHS = 2
LEARNING_RATE = 0.001
FEDPROX_MU = 0.01
BATCH_SIZE = 64
MAX_SAMPLES_PER_CLIENT = 2500   # cap per client so CPU finishes in ~15 min
# Differential Privacy parameters
DP_ENABLED = True
DP_NORM_CLIP = 1.0
```

```
DP_NOISE_MULTIPLIER = 0.1   # σ — noise per-param is σ*S/√N so total noise L2 = σ*S
# Attacker/Poisoning configurations
ATTACKER_CID = 4            # Client 4 will act as the malicious stealth attacker
ATTACK_TYPE = "sign_flip"   # "sign_flip" or "random_noise" or "label_flip"
DEVICE = "cuda" if torch.cuda.is_available() else "cpu"
# Emotion class names (FER-2013 order)
EMOTION_NAMES = ["Angry", "Disgust", "Fear", "Happy", "Sad", "Surprise", "Neutral"]
def compute_dp_epsilon(rounds_completed: int, noise_multiplier: float,
                       delta: float = 1e-5) -> float:
    """
    Approximate accumulated DP epsilon using advanced composition of the Gaussian
    mechanism.  Each round contributes one mechanism application per benign client.
    Formula (Theorem 3.3, Dwork and Roth):
        ε_total ≈ sqrt(2 * T * ln(1/δ)) / σ
    where σ = noise_multiplier (sensitivity-normalised std dev) and T = rounds.
    """
    if noise_multiplier <= 0 or rounds_completed <= 0:
        return float('inf')
    return round(math.sqrt(2 * rounds_completed * math.log(1.0 / delta)) / noise_multiplier, 3)
class LBAAFedAvgAggregator:
    """
    Server-side Layer-Based Anomaly Aware Aggregator (LBAAFedAvg).
    Evaluates incoming client updates layer-by-layer using Median Absolute Deviation
        (MAD)
    to detect and filter out stealth poisoning attacks before averaging.
    """
    def __init__(self, anomaly_threshold=2.2, layer_flag_ratio=0.25):
        self.anomaly_threshold = anomaly_threshold
        self.layer_flag_ratio = layer_flag_ratio
        self.round_history = []
    def filter_and_aggregate(self, server_round: int, client_updates: list,
                             client_num_samples: list, global_model_weights: list) -> tuple:
        """
        Evaluates updates layer-by-layer, blocks anomalous updates, and aggregates the rest.
        Args:
```

```
    server_round (int): The current federated round.
    client_updates (list of list of np.ndarray): Weights submitted by each client.
    client_num_samples (list of int): Number of samples processed by each client.
    global_model_weights (list of np.ndarray): Current global weights.
Returns:
    tuple: (aggregated_weights, anomaly_count, blocked_cids)
"""
num_clients = len(client_updates)
num_layers = len(client_updates[0])
if num_clients <= 2:
    # Fallback to standard weighted average if too few clients
    print("[SERVER] [WARNING] Insufficient clients for anomaly detection. Running standard FedAvg.")
    self.round_history.append({
        "round": server_round,
        "total_clients": num_clients,
        "anomalous_clients": 0,
        "blocked_cids": []
    })
    return self._weighted_average(client_updates, client_num_samples), 0, []
print(f"--- Round {server_round}: Executing LBAAFedAvg Secure Filter ---")
# anomalous_flags[client_idx][layer_idx] = True if anomalous
anomalous_flags = np.zeros((num_clients, num_layers), dtype=bool)
layer_distances = np.zeros((num_clients, num_layers))
# Compute deltas (weight updates) relative to current global model weights
client_deltas = []
for k in range(num_clients):
    deltas_k = [client_updates[k][l] - global_model_weights[l] for l in range(num_layers)]
    client_deltas.append(deltas_k)
# Evaluate layer-by-layer
for l in range(num_layers):
    # Collect layer l deltas across all clients
    layer_params = [client_deltas[k][l] for k in range(num_clients)]
    flat_params = [lp.flatten() for lp in layer_params]
    # Compute element-wise median vector for layer l
```

```
    median_vector = np.median(flat_params, axis=0)
    # Calculate L2 distance from median for each client
    distances = []
    for k in range(num_clients):
        dist = np.sqrt(np.sum((flat_params[k] - median_vector) ** 2))
        distances.append(dist)
    distances = np.array(distances)
    layer_distances[:, l] = distances
    # Compute Median Absolute Deviation (MAD) of distances
    median_distance = np.median(distances)
    mad = np.median(np.abs(distances - median_distance))
    if mad < 1e-12:
        mad = 1e-12
    # Compute modified Z-scores
    z_scores = 0.6745 * (distances - median_distance) / mad
    # Flag layer l of client k if it exceeds threshold
    for k in range(num_clients):
        if z_scores[k] > self.anomaly_threshold:
            anomalous_flags[k, l] = True
# Process flags across all layers
verified_updates = []
verified_samples = []
blocked_cids = []
anomalous_layer_ratios = np.sum(anomalous_flags, axis=1) / num_layers
for k in range(num_clients):
    flag_ratio = anomalous_layer_ratios[k]
    is_attacker = (k == ATTACKER_CID)
    print(f"Client {k} (Attacker Ground Truth: {is_attacker}) - "
          f"Anomalous Layer Ratio: {flag_ratio:.2%} "
          f"(Max Z-Score Distance: {np.max(layer_distances[k]):.4f})")
    if flag_ratio >= self.layer_flag_ratio:
        print(f"[SERVER] [WARNING] DISCARDED: Client {k} detected as anomalous
"
              f"({flag_ratio:.2%} layers flagged). Update blocked.")
        blocked_cids.append(str(k))
```

```
            else:
                verified_updates.append(client_updates[k])
                verified_samples.append(client_num_samples[k])
        print(f"[SERVER] LBAAFedAvg Filter Result: {len(verified_updates)} verified, "
              f"{len(blocked_cids)} blocked out of {num_clients} total updates.")
        # Log round history
        self.round_history.append({
            "round": server_round,
            "total_clients": num_clients,
            "anomalous_clients": len(blocked_cids),
            "blocked_cids": blocked_cids
        })
        if not verified_updates:
            print("[SERVER] [ERROR] All client updates were blocked! Retaining current global
            model.")
            return global_model_weights, num_clients, blocked_cids
        # Aggregate verified updates
        aggregated_weights = self._weighted_average(verified_updates, verified_samples)
        return aggregated_weights, len(blocked_cids), blocked_cids
    def _weighted_average(self, updates: list, samples: list) -> list:
        """Computes sample-weighted average of weights."""
        total_samples = sum(samples)
        num_layers = len(updates[0])
        aggregated = []
        for l in range(num_layers):
            layer_sum = np.zeros_like(updates[0][l])
            for k in range(len(updates)):
                layer_sum += updates[k][l] * (samples[k] / total_samples)
            aggregated.append(layer_sum)
        return aggregated
def train_local_client(client_idx: int, global_model: nn.Module, train_loader: DataLoader,
                       epochs: int, lr: float, mu: float, dp_enabled: bool,
                       dp_clip: float, dp_noise: float, attacker: bool, attack_type: str) -> tuple:
    """Trains a simulated local client incorporating FedProx, local DP, and poisoning
    options."""
    local_model = copy.deepcopy(global_model).to(DEVICE)
```

```
    ref_model   = copy.deepcopy(global_model).to(DEVICE)
    ref_model.eval()
    local_model.train()
    criterion = nn.CrossEntropyLoss(label_smoothing=0.1)
    optimizer = torch.optim.Adam(local_model.parameters(), lr=lr, weight_decay=1e-4)
    scheduler = torch.optim.lr_scheduler.CosineAnnealingLR(optimizer, T_max=epochs,
        eta_min=lr * 0.1)
    global_weights = [p.data.cpu().numpy().copy() for p in local_model.parameters()]
    running_loss = 0.0
    correct = 0
    total = 0
    for epoch in range(epochs):
        for images, labels in train_loader:
            images, labels = images.to(DEVICE), labels.to(DEVICE)
            if attacker and attack_type == "label_flip":
                labels = torch.where(labels == 3, torch.tensor(4).to(DEVICE), labels)
# ... [see full source in backend/simulation.py]
```

A.4 ONNX Export — backend/export_onnx.py

```
import os
import sys
import io
# Reconfigure stdout and stderr streams to use UTF-8 to prevent Windows terminal CP1252
    crashes
sys.stdout = io.TextIOWrapper(sys.stdout.buffer, encoding='utf-8')
sys.stderr = io.TextIOWrapper(sys.stderr.buffer, encoding='utf-8')
import torch
import torch.nn as nn
from backend.models.cnn import EmotionCNN
def export_to_onnx(pth_path="backend/global_model.pth",
    onnx_path="frontend/global_model.onnx"):

    print("============================================================
    ====")
    print("           EXPORTING PYTORCH MODEL TO ONNX")

    print("============================================================
    ====")
```

```
# 1. Initialize model
model = EmotionCNN()
# 2. Try to load trained weights, fall back to initial state if file not found
if os.path.exists(pth_path):
    print(f"Loading trained weights from: {pth_path}...")
    model.load_state_dict(torch.load(pth_path, map_location=torch.device('cpu')))
else:
    print(f"[WARNING] Checkpoint not found at {pth_path}. Exporting base initialized
        model architecture...")
# 3. Set model to evaluation mode
model.eval()
# 4. Define dummy input matching model specifications (BatchSize, Channel, Height,
    Width)
# Input resized to 64×64 to match mini_XCEPTION architecture
dummy_input = torch.randn(1, 1, 64, 64, requires_grad=False)
# 5. Create output directory if not exists
os.makedirs(onnx_path, exist_ok=True) if os.path.isdir(onnx_path) else
    os.makedirs(os.path.dirname(onnx_path), exist_ok=True)
# 6. Export the model
print(f"Exporting model to ONNX format at: {onnx_path}...")
torch.onnx.export(
    model,                  # Model being run
    dummy_input,            # Model input (or a tuple for multiple inputs)
    onnx_path,              # Where to save the model
    export_params=True,     # Store the trained parameter weights inside the model file
    opset_version=12,       # The ONNX version to export the model to (11 or 12 are
        widely compatible)
    do_constant_folding=True,   # Whether to execute constant folding for optimization
    input_names=['input'],      # The model's input names
    output_names=['output'],    # The model's output names
    dynamic_axes={          # Variable length axes (enables flexible batch sizing)
        'input': {0: 'batch_size'},
        'output': {0: 'batch_size'}
    },
    dynamo=False            # Disable dynamo exporter to bundle weights inside the .onnx
        file directly
)
```

```
    print("ONNX model export complete!")
    # 7. Rapid validation check
    try:
        import onnx
        onnx_model = onnx.load(onnx_path)
        onnx.checker.check_model(onnx_model)
        print("[SUCCESS] ONNX model structural check passed! The graph is valid and
            browser-ready.")
    except ImportError:
        print("ONNX python library not installed. Skipping structural validation check.")
    except Exception as e:
        print(f"[ERROR] ONNX model validation failed: {e}")
if __name__ == "__main__":
    export_to_onnx()
```

A.5 Frontend Inference — frontend/app.js (inference loop)

```
// Global Variables
let session = null;
let webcamStream = null;
let charts = {};
let currentModelType = 'fl';   // 'fl' | 'standalone'
const EMOTIONS = ["Angry □", "Disgust □", "Fear □", "Happy □", "Sad □", "Surprise
        □", "Neutral □"];
const EMOTION_KEYS = ["angry", "disgust", "fear", "happy", "sad", "surprise", "neutral"];
const MODEL_PATHS = {
    fl:         './global_model.onnx',
    standalone: './standalone_model.onnx'
};
const MODEL_BADGE_TEXT = {
    fl:         'FEDERATED · PRIVACY-PRESERVING',
    standalone: 'STANDALONE · CENTRALIZED'
};
// Initialize Application
window.addEventListener('DOMContentLoaded', async () => {
    checkCORS();
    await initWebcam();
    await loadONNXModel(MODEL_PATHS.fl);
```

```
    loadDashboardMetrics('fl');
    initPhotoTab();
    // Start continuous inference
    startInferenceLoop();
});
// Switch Tabs Navigation
function switchTab(tabName) {
    document.querySelectorAll('.tab-btn').forEach(btn => btn.classList.remove('active'));
    document.getElementById(`tab-${tabName}`).classList.add('active');
    document.querySelectorAll('.tab-content').forEach(content =>
        content.classList.remove('active'));
    document.getElementById(`content-${tabName}`).classList.add('active');
    if (tabName === 'dashboard') {
        Object.values(charts).forEach(chart => chart.resize());
    }
}
// Check if running on file:// protocol (CORS restriction trigger)
function checkCORS() {
    if (window.location.protocol === 'file:') {
        document.getElementById('cors-warning').style.display = 'block';
    }
}
// 1. Initialize Webcam access
async function initWebcam() {
    const video = document.getElementById('webcam-video');
    const cameraDot = document.getElementById('camera-dot');
    const cameraStatus = document.getElementById('camera-status');
    try {
        webcamStream = await navigator.mediaDevices.getUserMedia({
            video: {
                width: { ideal: 640 },
                height: { ideal: 480 },
                facingMode: 'user'
            },
            audio: false
```

```
        });
        video.srcObject = webcamStream;
        cameraDot.classList.add('active');
        cameraStatus.textContent = "Camera Online";
        console.log("Webcam stream started successfully.");
    } catch (err) {
        console.error("Webcam access failed: ", err);
        cameraStatus.textContent = "Camera Blocked";
        alert("Please grant webcam permissions to demonstrate live emotion classification.");
    }
}
 // 2. Load ONNX Runtime Web Model Session
async function loadONNXModel(modelPath = './global_model.onnx') {
    const modelStatus = document.getElementById('model-status');
    modelStatus.textContent = "Model Loading...";
    modelStatus.style.color = "";
    session = null;
    try {
        console.log(`Loading ONNX model: ${modelPath}`);
        ort.env.wasm.wasmPaths = 'https://cdn.jsdelivr.net/npm/onnxruntime-
            web@1.17.3/dist/';
        ort.env.wasm.numThreads = 1;
        session = await ort.InferenceSession.create(modelPath, {
            executionProviders: ['webgl', 'wasm']
        });
        modelStatus.textContent = "Model Online □";
        modelStatus.style.color = "var(--color-success)";
        console.log(`ONNX model loaded: ${modelPath}`);
    } catch (err) {
        console.error("ONNX model load failed:", err);
        modelStatus.textContent = "Model Loading Failed □";
        modelStatus.style.color = "var(--color-danger)";
    }
}
// Switch between FL and Standalone inference models
```

```
async function switchActiveModel(type) {
    if (type === currentModelType) return;
    currentModelType = type;
    // Update toggle button states
    document.querySelectorAll('.toggle-option').forEach(btn =>
        btn.classList.remove('active'));
    document.getElementById(`toggle-${type}`).classList.add('active');
    // Update badge text
    const badge = document.getElementById('model-badge');
    if (badge) badge.textContent = MODEL_BADGE_TEXT[type];
    // Load new ONNX model
    await loadONNXModel(MODEL_PATHS[type]);
    // Reload dashboard with the corresponding history file
    loadDashboardMetrics(type);
}
// 3. Preprocess webcam crop and execute inference loop
async function startInferenceLoop() {
    const video = document.getElementById('webcam-video');
    // Create an offscreen canvas to perform cropping and scaling
    const offscreenCanvas = document.createElement('canvas');
    offscreenCanvas.width = 64;
    offscreenCanvas.height = 64;
    const ctx = offscreenCanvas.getContext('2d');
    async function processFrame() {
        // Only run inference if session is loaded, video is ready and playing
        if (session andand video.readyState === video.HAVE_ENOUGH_DATA) {
            // Bounding box mapping: crop center 200x200 pixel area of the video
            // Video standard output width is video.videoWidth, height is video.videoHeight
            const vWidth = video.videoWidth;
            const vHeight = video.videoHeight;
            // Calculate coordinates to crop the center relative viewport box
            const size = Math.min(vWidth, vHeight) * 0.45; // 45% of minimum side
            const sx = (vWidth - size) / 2;
            const sy = (vHeight - size) / 2;
            // Draw cropped video region scaled to 64x64 on our canvas
```

```
        ctx.drawImage(video, sx, sy, size, size, 0, 0, 64, 64);
        // Get pixel data
        const imgData = ctx.getImageData(0, 0, 64, 64);
        const pixels = imgData.data; // RGBA flat array
        // Preprocess pixels to 64x64x1 normalized grayscale Float32Array
        const inputData = new Float32Array(64 * 64);
        for (let i = 0; i < pixels.length; i += 4) {
            const r = pixels[i];
            const g = pixels[i+1];
            const b = pixels[i+2];
            // standard ITU-R grayscale formula: Y = 0.299R + 0.587G + 0.114B
            const gray = (0.299 * r + 0.587 * g + 0.114 * b) / 255.0;
            inputData[i / 4] = gray;
// ... [see full source in frontend/app.js]
```

APPENDIX B
SAMPLE OUTPUTS

B.1 Training Convergence

Figure B.1: Validation Accuracy and Loss over 50 Training Rounds

B.2 F1 Score Convergence

Figure B.2: Macro F1 Score Progression with Precision and Recall Reference Lines

B.3 Per-Class F1 Score

Figure B.3: Per-Class F1 Score at Best Global Model Checkpoint (Epoch 47)

B.4 Per-Class Precision, F1, and Recall

Figure B.4: Grouped Precision / F1 / Recall Comparison per Emotion Class

B.5 Confusion Matrix

Figure B.5: Confusion Matrix — Raw Counts and Normalised Recall on FER-2013 Test Set

B.6 Full Classification Report (sklearn)

The following is the complete per-class classification report generated by scikit-learn on the best-checkpoint global model predictions over all 7,178 FER-2013 test samples:

	precision	recall	f1-score	support
Angry	0.57	0.59	0.58	958
Disgust	0.70	0.34	0.46	111
Fear	0.52	0.38	0.44	1024
Happy	0.82	0.89	0.86	1774
Sad	0.54	0.48	0.51	1247
Surprise	0.74	0.79	0.76	831
Neutral	0.58	0.69	0.63	1233
accuracy			0.65	7178
macro avg	0.64	0.59	0.61	7178
weighted avg	0.64	0.65	0.64	7178

B.7 Terminal Simulation Log (Round 1 Excerpt)

```
============================================================
        STARTING PRIVACY-CONSCIOUS FEDERATED EMOTION RECOGNITION
============================================================
Executing on hardware device: CPU

Preparing Non-IID datasets across clients...
[DATA] Loading FER-2013 from CSV:
DATA/FER2013/fer2013/fer2013/fer2013.csv
[DATA] Loaded 28709 train / 7178 test images from CSV.
  Client 0: 2500 samples - {'Angry': 2, 'Disgust': 47, 'Fear': 619, 'Happy': 370, 'Sad':
923, 'Surprise': 528, 'Neutral': 11}
  Client 1: 2500 samples - {'Angry': 1253, 'Disgust': 56, 'Fear': 483, 'Happy': 118,
'Sad': 27, 'Surprise': 9, 'Neutral': 554}
  Client 2: 2500 samples - {'Angry': 5, 'Disgust': 31, 'Fear': 318, 'Happy': 1483, 'Sad':
128, 'Surprise': 78, 'Neutral': 457}
  Client 3: 2500 samples - {'Angry': 394, 'Disgust': 2, 'Fear': 69, 'Happy': 639, 'Sad':
555, 'Surprise': 28, 'Neutral': 813}
  Client 4: 2500 samples - {'Angry': 185, 'Disgust': 80, 'Fear': 511, 'Happy': 75, 'Sad':
458, 'Surprise': 1127, 'Neutral': 64}
Model parameters: 56,951 (~0.22 MB per update)

[ROUND 1]
[CLIENT 0] Registering as a Benign Node (FedProx + Differential Privacy)
[CLIENT 1] Registering as a Benign Node (FedProx + Differential Privacy)
[CLIENT 2] Registering as a Benign Node (FedProx + Differential Privacy)
[CLIENT 3] Registering as a Benign Node (FedProx + Differential Privacy)
[CLIENT 4] [WARNING] Initiating as an Active Adversary! Attack: SIGN_FLIP
--- Round 1: Executing LBAAFedAvg Secure Filter ---
Client 0 (Attacker Ground Truth: False) - Anomalous Layer Ratio: 0.00%  (Max Z-
Score: 224.5)
Client 1 (Attacker Ground Truth: False) - Anomalous Layer Ratio: 0.00%  (Max Z-
Score: 224.6)
Client 2 (Attacker Ground Truth: False) - Anomalous Layer Ratio: 0.00%  (Max Z-
Score: 224.5)
Client 3 (Attacker Ground Truth: False) - Anomalous Layer Ratio: 0.00%  (Max Z-
Score: 224.4)
Client 4 (Attacker Ground Truth: True ) - Anomalous Layer Ratio: 100.00% (Max Z-
Score: 15475.0)
[SERVER] [WARNING] DISCARDED: Client 4 detected as anomalous (100.00%
layers flagged).
[SERVER] LBAAFedAvg Filter Result: 4 verified, 1 blocked out of 5 total updates.
[SERVER Round 1] Accuracy: 40.21% | Macro-F1: 0.2623 | ADR: 100% | FPR: 0% |
DP eps: 15.99
```

APPENDIX C

DATASET

C.1 FER-2013 Dataset Description

The Facial Expression Recognition 2013 (FER-2013) dataset was introduced by Goodfellow *et al.* at the International Conference on Machine Learning (ICML) 2013 as part of the Representation Learning Challenges. It remains the primary benchmark for facial emotion recognition research.

Property	Value
Source	Kaggle / ICML 2013 FER Challenge
Format	CSV with pixel strings (48×48 = 2,304 pixel values per sample)
Total Samples	35,887
Training Split	28,709 (Training)
Test Split	7,178 (PublicTest: 3,589 + PrivateTest: 3,589)
Image Size	48 × 48 pixels, grayscale
Number of Classes	7 emotion categories
Input to Model	Resized to 64 × 64, normalized to [-1, 1]

C.2 Class Distribution

Label	Emotion	Training Samples	% of Training Set
0	Angry	4,953	17.25%
1	Disgust	547	1.91%
2	Fear	5,121	17.84%
3	Happy	8,989	31.31%
4	Sad	6,077	21.17%
5	Surprise	4,002	13.94%
6	Neutral	6,198	21.59%
—	TOTAL	28,709	100%

The dataset exhibits significant class imbalance, with Happy comprising 31.3% of training samples while Disgust represents only 1.9% (547 samples). This imbalance is a well-documented characteristic of the FER-2013 benchmark and directly contributes to the per-class performance disparity observed in Section 4.4.3. The imbalance is further compounded in the federated setting by the Non-IID Dirichlet partitioning, which may assign the entirety of a client's Disgust samples to a single node.

C.3 Data Access

The FER-2013 dataset is publicly available from the Kaggle competition platform at the following URL:

https://www.kaggle.com/competitions/challenges-in-representation-learning-facial-expression-recognition-challenge/data

In this implementation, the competition CSV file (fer2013.csv) is used as the authoritative data source. The CSV contains three columns: emotion (integer label 0–6), pixels (space-separated 48×48 pixel intensity values as a string), and Usage (Training / PublicTest / PrivateTest). The data loader in backend/dataset/data_loader.py parses this CSV using pandas, reconstructs NumPy image arrays, and applies PIL-based augmentation transforms before partitioning samples across federated clients.