SecureFed: A Two-Phase Framework for Detecting Malicious Clients in Federated Learning

Likhitha Annapurna Kavuri*, Akshay Mhatre†, Akarsh K Nair‡, Deepti Gupta§
*†§Dept. of Computer Information Systems, Texas A&M University - Central Texas, Texas, USA
‡Department of CSE, IIIT Kottayam, Kottayam, India
*lk040@my.tamuct.edu, †am271@my.tamuct.edu, ‡akarshkn@iiitkottayam.ac.in, §d.gupta@tamuct.edu

***Abstract*—Federated Learning (FL) protects data privacy while providing a decentralized method for training models. However, because of the distributed schema, it is susceptible to adversarial clients that could alter results or sabotage model performance. This study presents *SecureFed*, a two-phase FL framework for identifying and reducing the impact of such attackers. Phase 1 involves collecting model updates from participating clients and applying a dimensionality reduction approach to identify outlier patterns frequently associated with malicious behavior. Temporary models constructed from the client updates are evaluated on synthetic datasets to compute validation losses and support anomaly scoring. The idea of learning zones is presented in Phase 2, where weights are dynamically routed according to their contribution scores and gradient magnitudes. High-value gradient zones are given greater weight in aggregation and contribute more significantly to the global model, while lower-value gradient zones, which may indicate possible adversarial activity, are gradually removed from training. Until the model converges and a strong defense against poisoning attacks is possible, this training cycle continues Based on the experimental findings, *SecureFed* considerably improves model resilience without compromising model performance.**

***Index Terms*—Federated Learning, Anomaly Detection, Security, and Privacy.**

I. Introduction

Federated Learning (FL) is a decentralized machine learning paradigm that enables multiple clients to collaboratively train a shared global model while maintaining their raw data local [1]. This design provides significant privacy advantages, especially in domains such as healthcare, finance, and mobile applications, where sensitive data cannot be centrally aggregated due to privacy regulations like GDPR. Despite these benefits, FL is highly susceptible to adversarial attacks due to the lack of centralized control over individual client updates [2]. One of the major concerns in FL is the presence of malicious clients that send tainted or modified model updates to the server, aiming to degrade the overall model performance or generate targeted misclassifications [3]. Since malicious updates are usually designed to statistically mimic benign behavior, such attacks can be challenging to identify. Therefore, protecting FL systems against similar threats is crucial for preserving their effectiveness and trustworthiness [4].

Motivated by these challenges, we propose *SecureFed*, a two-stage FL framework designed to isolate and detect rogue clients. Initially, *SecureFed* leverages publicly available datasets to collect model updates from all participating clients. It then applies a dimensionality reduction techniques, Principal Component Analysis (PCA) to analyze weight vector patterns [5]. This analysis enables us to identify irregularities in client behavior. Further to assist in extensive anomaly detection, a temporary model is constructed using the gradients and validated on a synthetic datasets. The observed error rates and loss values computed from the reduced-dimensional representations further assist in the anomaly detection. This score is carry forwarded to phase 2, where *SecureFed* introduces the concept of learning zones, the zones group clients based on trustworthiness to guide training decisions, quantified using weight magnitudes and gradient values. Clients exhibiting low-gradient patterns, usually associated with adversarial manipulation, are progressively removed from the training loop. In contrast, clients with high-gradient values are deemed reliable and continue contributing to the global model [6]. The proposed zoning strategy enables targeted learning and enhances resistance to poisoning attacks.

The contributions of the article are as follows:

- A federated anomaly detection framework is proposed, leveraging dimensionality reduction to identify suspicious patterns in client weight updates.
- A preliminary scoring model is developed, trained on real-world attack datasets to support early-stage anomaly detection.
- A two-stage architecture termed *SecureFed* is presented, combining client scoring and trust-based filtering.
- A dynamic learning zone mechanism is introduced, adjusting the model training process based on client trustworthiness.
- Extensive experiments are conducted to demonstrate the effectiveness of *SecureFed* in mitigating adversarial impacts.

The remainder of the article is structured as follows: Section II reviews the background and relevant FL security. Section III outlines the problem formulation followed by the introduction of *SecureFed* framework in Section IV. Section V presents the experimental setup, analysis, and results. Finally, Section VI presents the discussions and Section VII concludes the paper and presents future research directions.

II. Background and Related Work

This section reviews the existing literature on security threats in FL systems and highlights defense mechanisms

developed to detect and mitigate malicious client behavior.

A. Security Issues in Federated Learning

FL was introduced as a privacy-preserving training paradigm [7]. However, FL systems are susceptible to adversarial threats, including poisoning, leakage and backdoor attacks. In poisoning attacks, malicious clients injects adversarial data into the system, degrading model accuracy and sabotaging system performance. These attacks ultimately aim to shift the model's focus toward adversarial tasks and remain undetected to high extent. Backdoor attacks employ a more subtle approach, targeting specific inputs to trigger the model into executing adversarial behavior. These backdoors are stealthily embedded into the system in such a way that the model's performance on the primary task remains unaffected, allowing the adversarial activity to go undetected [4]. Additionally, the exchange of gradients or model weights during training may leak sensitive information (unintentionally or even intentionally), exposing client-side data. These leaks can be exploited by adversaries to launch powerful data leakage and reconstruction attacks. In such attacks, adversaries are able to reconstruct input data from the leaked information, effectively gaining access without directly participating in the system. Detecting and mitigating these threats remains a critical challenge in federated systems. Since malicious updates often mimic benign ones, traditional outlier-based defenses are often ineffective. This emphasizes the need for robust, adaptive defense mechanisms that can identify and isolate malicious behavior without disrupting the overall training process.

B. Methods to Detecting Malicious Clients in FL

Zhang et al. [8] presented one of the initial comprehensive reviews that discussed the challenges in FL, highlighting vulnerabilities such as model poisoning, data leakage, and high communication costs was presented by Zhang et al. [8]. The article emphasizes the complexity of ensuring both privacy and robustness in decentralized settings. It evaluates existing defense techniques like differential privacy, homomorphic encryption, and secret sharing, while identifying gaps in incentive mechanisms and personalization. This work presents the broader problem space where several FL security frameworks similar to *SecureFed* operate.

Li et al. [9] presented an article focusing on the detection of malicious clients, proposing a VAE-based framework for learning low-dimensional embeddings of model updates. Malicious updates generate higher reconstruction errors, enabling their detection without labeled data. The method works in both unsupervised and semi-supervised modes with dynamic thresholds. The framework emphasizes adaptive, data-driven detection of harmful contributions for both targeted and untargeted attacks. Similarly, FedDMC [10] is another framework focusing on identifying malicious clients in poisoning attack settings. The framework combines Binary Tree-Based Clustering with Noise, PCA, and a Self-Assemble Detection Correcting Module, forming a poisoning-resistant FL system. The system detects malicious clients without clean validation data by balancing high detection accuracy and low computational overhead. FedDMC's multi-tiered detection strategy has similarities to *SecureFed*'s use of dimensionality reduction and trust-based filtering.

Gupta et al. [11] have proposed an FL-based anomaly detection framework for healthcare systems. The framework combines Digital Twins and Edge Cloudlet Computing to avoid sensitive data transfer. In addition, several security models for protecting IoT devices are discussed in [12], [13].

Even though the primary focus is not on adversarial detection, the work highlights FL's potential in secure and privacy-preserving applications.

C. Comparison with existing works

To improve robustness against adversarial updates, several defenses rely on similarity or distance-based heuristics. Conventional mechanisms like Krum selects updates closest to the majority [14], while statistical aggregators like the median and trimmed mean offer resilience to outliers but overlook deeper structural patterns in client behavior [15]. FoolsGold [16] targets Sybil attacks by analyzing gradient similarity, but may struggle with more nuanced poisoning attempts. FLAME [17] is another popular mitigation strategy that uses adaptive weighting and gradient clipping. In contrast, *SecureFed* leverages dimensionality reduction to detect anomalies based on the underlying distribution of client updates, combining it with a trust-aware approach that dynamically filters clients based on anomaly and gradient-based scoring.

SecureFed differs from FLTrust [18], which uses a static reference model without revalidating client behavior, by incorporating synthetic validation prior to aggregation. Unlike prior works that apply dimensionality reduction solely for post-hoc analysis [19], *SecureFed* integrates it directly into the FL pipeline to enable iterative anomaly detection.

III. Problem Identification

Even though FL enhances privacy by keeping data local, its decentralized architecture introduces significant security vulnerabilities. Particularly, malicious clients may inject poisoned updates that degrade model performance or embed backdoors into the global model.

A. Vulnerabilities in Federated Learning

FL is vulnerable to adversarial clients whose poisoned updates can degrade global model performance. Traditional aggregation methods such as Federated Averaging (FedAvg) are vulnerable to such attacks as they assume all clients to be benign. To address this, several robust aggregation strategies have been proposed. One such framework, FLTrust [18] introduces a server-side trust bootstrapping mechanism, where the server maintains a small, trusted dataset. Client updates are scored based on their similarity to this baseline, helping to reduce the impact of malicious clients. MAB-RFL [20] employs a multi-armed bandit strategy to adaptively select clients for aggregation. By formulating client selection as

a bandit problem, the framework balances exploration and exploitation to identify and prioritize reliable clients over time.

However, these defenses face limitations, particularly in high-dimensional settings, where benign anomalies and adversarial manipulations are difficult to distinguish.

B. Dimensionality Reduction for Malicious Client Detection

High-dimensional model updates in FL can make it challenging to distinguish between benign and malicious behavior. To address this, dimensionality reduction techniques such as autoencoders and PCA have been explored:

- **Autoencoder-Based Anomaly Detection:** Autoencoders can learn compressed representations of benign client updates by capturing their underlying structure. Deviations from this structure (reflected as high reconstruction errors) usually indicate anomalous or malicious activity.
- **Gradient and Reconstruction Analysis:** Existing literature has proven that combining gradient information with autoencoder-based reconstruction improves anomaly detection accuracy in FL. This hybrid approach leverages both gradient deviations and reconstruction errors to enhance robustness against poisoning attacks [21].

However, integrating these techniques into federated settings remains challenging, particularly in ensuring an optimal trade-off between representation reduction, client privacy, and potential model performance degradation.

C. Secure Aggregation Protocols

Secure aggregation algorithms aim to preserve client anonymity by aggregating model updates without revealing individual contributions. However, the privacy guarantee can unintentionally conceal malicious activity.

- **ELSA** introduces a secure aggregation scheme that distributes trust management between two non-colluding servers. This design ensures the privacy of client updates while enabling detection of malicious behavior [7].
- **SeaFlame** enhances communication efficiency using share conversion and sharing techniques. It reduces communication overhead while maintaining robustness against malicious clients [22].

These protocols highlight the inherent trade-off between protecting client privacy and retaining the ability to detect and mitigate adversarial activities.

D. Feature Engineering and Client Behavior Analysis

Feature engineering plays a critical role in enhancing the detection of malicious clients.

- **Feature Selection Techniques:** Methods such as recursive feature elimination, chi-square tests, and mutual information help identify relevant features that differentiate between benign and adversarial behaviors. Effective feature selection reduces dimensionality and improves model interpretability.
- **Modeling Client Behavior:** Understanding and modeling client behavior can assist in identifying anomalies. By analyzing patterns in client updates over time, certain deviations can be observed, indicative of malicious intent.

E. Research Gap

Even with the advancements in secure aggregation, anomaly detection, and robust aggregation strategies, several key challenges can be identified:

- **High-Dimensional Data:** Existing defenses often fail in high-dimensional settings, where distinguishing adversarial behavior from benign outliers becomes unreliable.
- **Integration of Techniques:** There is a lack of unified frameworks that effectively combines dimensionality reduction, anomaly detection, and secure aggregation to address malicious client identification.
- **Adaptive Learning Zones:** Current systems are primarily static, failing to dynamically adjust to evolving client behavior, limiting their long-term robustness.

F. Proposed Approach

Motivated by these research gaps, the study proposes *SecureFed*, a two-stage framework that combines anomaly detection using dimensionality reduction with adaptive client filtering based on trust scores. The proposed framework aims to strengthen FL against adversarial attacks by improving adaptive security via learning zone based client screening.

IV. Proposed Framework

The *SecureFed* framework is designed to enhance the robustness of FL systems via a multi stage schema, presented in Figure 1. It operates through two core phases: an anomaly detection module and an adaptive aggregation module.

A. First layer: Client Side

To simulate the FL setup and extract client-side model updates for subsequent analysis, the system begin by using publicly available attack datasets. Data is then partitioned in IID settings, followed by standard preprocessing approhces such as reshaping and normalization. After preprocessing, FL training is initiated by transferring the global model to clients, who then begin their initial round of training. After training, clients send their updated model weights to the central server for aggregation and anomaly analysis.

.

B. Second Layer: Server Side

Server-side processing in *SecureFed* operates in two phases: anomaly detection and adaptive zone-based aggregation.

*1) **Phase 1 (Anomaly Detection)**:* After receiving weight updates $\{\mathcal{W}_c^r\}$ from all clients in round r, the server applies PCA to reduce each client's update into a lower-dimensional space:

$$\tilde{\mathcal{W}}_c^r = \text{PCA}(\mathcal{W}_c^r)$$

An anomaly score A_c is computed for each client based on deviations in the reduced representation. To calibrate the detection process, a threshold τ^* is estimated using a synthetic validation dataset D_s. For the scope of this work, a standard dataset with similar feature vectors to the training dataset was used [23] in the place of synthetic data:

$$\tau^* = \text{Validate}(D_s, \{A_c\})$$

This threshold is further used to normalize and scale trust computation in Phase IV-B2.

*2) **Phase 2 (Adaptive Learning Zones)**:* Each client's update $\mathcal{W}_c^r$ is temporarily applied to the current global model to form a personalized model $\mathcal{M}_c^r$:

$$\mathcal{M}_c^r = \mathcal{M}_g^r + \mathcal{W}_c^r$$

This temporary model is evaluated on the synthetic dataset D_s to calculate a validation loss L_c, which helps assess the behavioral consistency of each client's contribution. Simultaneously, the gradient magnitude is computed as:

$$G_c = \|\nabla \mathcal{W}_c^r\|$$

Next, a trust score T_c is calculated by combining the normalized anomaly score, validation loss, and gradient magnitude:

$$T_c = \alpha \cdot \left(1 - \frac{A_c}{\tau^*}\right) + \beta \cdot \left(1 - \frac{L_c}{\max(L)}\right) + \gamma \cdot \frac{G_c}{\max(G)}$$

Based on this score, clients are dynamically assigned to one of three learning zones determined by a preset threshhold value: Zone 1 (High Trust) , Zone 2 (Uncertain) and Zone 3 (Low Trust). Each zone is assigned a weighting factor $\alpha_z(c)$ which determines its influence during aggregation. The global model is updated through zone-weighted aggregation as follows:

$$\mathcal{W}_{r+1} = \frac{\sum_{c \in \mathcal{C}} \alpha_z(c) \cdot n_c \cdot \mathcal{W}_c^r}{\sum_{c \in \mathcal{C}} \alpha_z(c) \cdot n_c}$$

$$\mathcal{M}_g^{r+1} = \mathcal{M}_g^r + \mathcal{W}_{r+1}$$

The updated model $\mathcal{M}_g^{r+1}$ is then sent to all clients for the next training round. This process continues until model convergence. The overall framework of the proposed system is illustrated in Fig. 1. This mechanism ensures that only trustworthy clients contribute to the global model. The detailed working of the framework is presented in Algorithm 1.

Algorithm 1 SecureFed: Two-Phase Framework with Trust-Zone Weighted Aggregation

Require: Initial global model $\mathcal{M}_g^0$, client set $\mathcal{C}$, public dataset D_p, synthetic dataset D_s, number of rounds R
Ensure: Final global model $\mathcal{M}_g^R$

for each round $r = 1$ to R **do**
  Server broadcasts $\mathcal{M}_g^r$ to all clients $\mathcal{C}$
  for each client $c \in \mathcal{C}$ in parallel **do**
    Client c trains and returns $\mathcal{W}_c^r$
  end for
  Phase 1: Anomaly Detection
  Collect all client updates: $\{\mathcal{W}_c^r\}$
  Apply dimensionality reduction: $\tilde{\mathcal{W}}_c^r = DR(\mathcal{W}_c^r)$
  Compute anomaly scores: $A_c = f_{\text{anomaly}}(\tilde{\mathcal{W}}_c^r)$
  Estimate detection threshold via synthetic evaluation: $\tau^* = \text{Validate}(D_s, \{A_c\})$
  Phase 2: Adaptive Learning Zones
  for each client c **do**
    Temporarily update model: $\mathcal{M}_c^r = \mathcal{M}_g^r + \mathcal{W}_c^r$
    Evaluate $\mathcal{M}_c^r$ on synthetic data D_s to compute validation loss: L_c
    Compute gradient magnitude: $G_c = \|\nabla \mathcal{W}_c^r\|$
    Compute trust score: $T_c = \alpha \cdot \left(1 - \frac{A_c}{\tau^*}\right) + \beta \cdot \left(1 - \frac{L_c}{\max(L)}\right) + \gamma \cdot \frac{G_c}{\max(G)}$
    if $T_c \geq \tau_{\text{high}}$ **then**
      Assign c to Zone 1 (High Trust), set $\alpha_z(c) = \alpha_1$
    else if $\tau_{\text{low}} \leq T_c < \tau_{\text{high}}$ **then**
      Assign c to Zone 2 (Uncertain), set $\alpha_z(c) = \alpha_2$
    else
      Assign c to Zone 3 (Low Trust), set $\alpha_z(c) = \alpha_3$
    end if
  end for
  Weighted Aggregation from All Zones:
  $\mathcal{W}_{r+1} \leftarrow \frac{\sum_{c \in \mathcal{C}} \alpha_z(c) \cdot n_c \cdot \mathcal{W}_c^r}{\sum_{c \in \mathcal{C}} \alpha_z(c) \cdot n_c}$
  Update global model: $\mathcal{M}_g^{r+1} \leftarrow \mathcal{M}_g^r + \mathcal{W}_{r+1}$
end for
return Final global model $\mathcal{M}_g^R$

V. Results

A. Experimental Setup

To evaluate the performance of *SecureFed*, we simulate a FL environment using the standard MNIST dataset, distributed under IID schema. The system is trained for three global rounds with 20 clients, with varying malicious client ratios (30-48%) injecting the model with poisoning updates. The poisoning attack primarily focuses on single-class label flipping mechanism [24]. Each client trains their model on local data, and the aggregation is initially performed using the vanilla FL schema and then using the proposed *SecureFed* schema. The *SecureFed* schema uses PCA for dimensionality reduction (retaining top-5 components) combined with K-Means based anomaly clustering, followed by a synthetic dataset based validation for trust-based scoring.

B. Adversarial Robustness Evaluation

The primary focus of the *SecureFed* framework is attaining adversarial robustness. Thus, the system's ability to isolate malicious clients is measured using precision, recall, and F1-score of the aggregated model. Additionally, the aggregated model's accuracy is also compared against baselines trained under both benign and poisoned environments. As shown in Table I, *SecureFed* achieves the highest scores compared to vanilla FL, due to the usage of low-dimensional behavior patterns and synthetic data-based model evaluation.

C. Trust Score Dynamics and Zone Distribution

Figure 2 illustrates the functioning of trust score based zones and how they evolve over rounds. Benign clients gradually converge to high-trust zones (Zone 1), while malicious ones

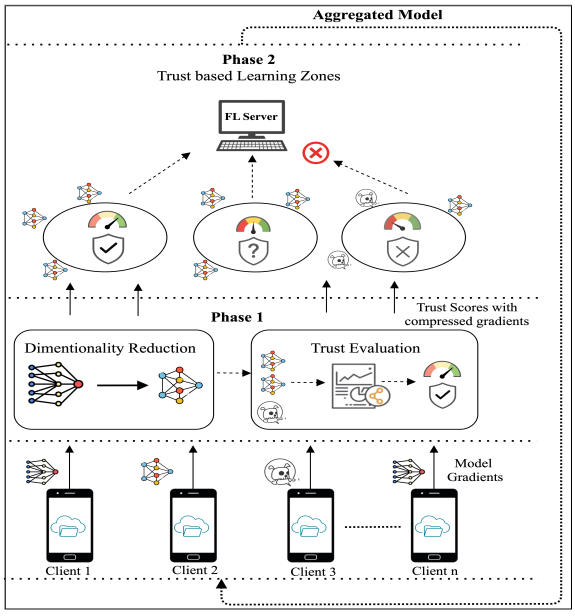

Fig. 1: Secure_Fed : Data flow diagram

TABLE I: Model performance metrics under varying malicious client ratios.

FL Method	Client Status	A (%)	P	R	F1
Vanilla FL (FedAvg)	Benign	95.49%	0.95	0.95	0.95
	30% Malicious	92.71%	0.91	0.93	0.93
	48% Malicious	84.42%	0.90	0.86	0.84
SecureFed	Benign	95.49%	0.95	0.95	0.95
	30% Malicious	93.11%	0.93	0.94	0.93
	48% Malicious	92.50%	0.91	0.92	0.92

A → Accuracy, P → Precision, R → Recall, F1 → F1 Score

remain in or migrate to Zone 3. Based on the zones, a weighted averaging is employed, ensuring primary contributions from clients in zone 1. The observations validates the effectiveness of adaptive learning zones in filtering malicious behavior without compromising useful contributions.

D. Ablation Study

Further, to analyze the contribution of each *SecureFed* component, an ablation study was performed and the observsation are as noted in Table II.

TABLE II: Component-wise Ablation Study

Configuration	Accuracy	Detection Rate
SecureFed (Full)	92.50%	75%
w/o PCA	91.61%	66.75%
w/o Synthetic Validation	88.54%	38.25%
w/o Trust Score (Binary Filter)	89.27%	45.05%

Observations showcase that PCA-based detection, synthetic validation, and trust-driven aggregation together ensure the robustness and reliability offered by the *SecureFed* framework.

VI. Discussion

The evaluation results demonstrate the effectiveness of the proposed *SecureFed* framework for mitigating the influence of malicious clients while preserving overall model performance. As shown in Table I, *SecureFed* consistently outperforms the baseline Vanilla FL (FedAvg) under increasing proportions of adversarial clients. specifically, with 48% of malicious clients, *SecureFed* achieves an accuracy of 92.50%, compared to only 84.42% with Vanilla FL, an improvement of 8.08%. Similarly, the F1 score improves from 0.84 to 0.92, showcasing an increase of 9.5%, indicating better balance between precision and recall in adversarial scenarios. This highlights the resilience of *SecureFed* in malicious settings.

In scenarios with 30% malicious clients, *SecureFed* maintains its robustness, improving accuracy from 92.71% (FedAvg) to 93.11%, and increasing the F1 score from 0.93 to 0.93 (and a slight increase in precision and recall), indicating that the framework does not compromise performance even when with lesser adversarial clients. The ablation results in Table II provide a comprehensive overview of the contribution of each framework component. When PCA-based is removed, the detection rate drops from 75% to 66.75%, and accuracy declines by nearly 0.9%, indicating that PCA plays a moderate but critical role in outlier isolation. A more significant performance degradation is observed when synthetic validation is excluded as accuracy drops to 88.54% and detection rate reduces by half to 38.25%. These observations confirms the importance of cross-verification with synthetic data for extensive anomaly detection.

Finally, removing the trust score mechanism and relying solely on binary filters result in a reduced accuracy (89.27%) and incomplete detection capability, highlighting the value of trust-based aggregation over rigid thresholding.

Overall, these observations validate the functionality of *SecureFed*. Each module contributes significantly to both performance and defense. The dynamic learning zone strategy enables graceful degradation in adversarial scenarios without abrupt exclusions.

VII. Conclusion and Future Work

In this research, we introduced *SecureFed*, a new two-phase architecture that detects and isolates malicious clients to improve the robustness of FL systems. *SecureFed* incorporates dimensionality reduction techniques and a dynamic learning zone strategy to systematically analyze client inputs and protect the model training process from adversarial participants. In Phase 1, we used dimensionality reduction to collect and analyze client weight vectors from publicly available datasets, allowing the early detection of client behavior anomalies. Further, the idea of learning zones is presented in Phase 2, which enable the framework to filter out possibly dangerous clients based on gradient importance and route reliable updates for further training. Validations on standard attack datasets show that *SecureFed* enhances the federated model's overall performance and security while successfully reducing the effects of poisoned updates. The method preserves model integrity by dynamically adjusting to adversarial behavior without necessitating major changes to the fundamental FL methodology.

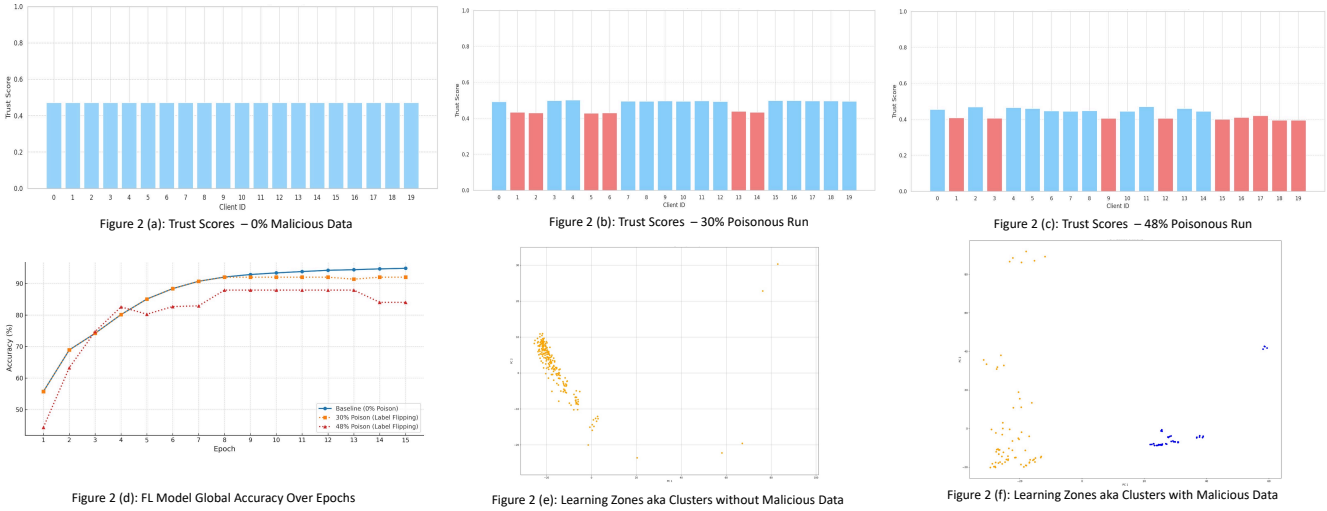

Figure 2 (a): Trust Scores – 0% Malicious Data

Figure 2 (b): Trust Scores – 30% Poisonous Run

Figure 2 (c): Trust Scores – 48% Poisonous Run

Figure 2 (d): FL Model Global Accuracy Over Epochs

Figure 2 (e): Learning Zones aka Clusters without Malicious Data

Figure 2 (f): Learning Zones aka Clusters with Malicious Data

Fig. 2: Results

However, a key limit observed in the current framework is its high dependency gradient divergence for anomaly classification. Since the system has only been evaluated under IID data distribution schemes, the underlying assumption holds true. However, in highly divergent settings, the assumption of divergent clients being malicious becomes less reliable. Thus, in the future work, we propose developing a robust aggregation mechanism that combines gradient divergence with additional behavioral features to support non-IID scenarios. Furthermore, the trust-based filtering mechanism used in *SecureFed* is limited to iteration-specific observations. Thus, in the future works, we plan to formulate an enhanced credibility metrics that integrate trust scores with historical performance trends for more comprehensive and adaptive client evaluation.

VIII. Acknowledgment

This work is partially supported by the US National Science Foundation grant 2431531.

References

[1] K. Bonawitz *et al.*, "Towards federated learning at scale: System design," *Proceedings of machine learning and systems*, vol. 1, pp. 374–388, 2019.

[2] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[3] E. Bagdasaryan *et al.*, "How to backdoor federated learning," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020, pp. 2938–2948.

[4] X. Sun *et al.*, "Can you really backdoor federated learning?" *arXiv preprint arXiv:2011.01767*, 2020.

[5] I. Jolliffe, *Principal Component Analysis*, 2nd ed. Springer, 2002.

[6] L. Zhang *et al.*, "Gradient similarity-based defense against model poisoning attacks in federated learning," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[7] M. Rathee *et al.*, "Elsa: Secure aggregation for federated learning with malicious actors," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 1961–1979.

[8] K. Zhang *et al.*, "Challenges and future directions of secure federated learning: a survey," *Frontiers of computer science*, vol. 16, pp. 1–8.

[9] S. Li *et al.*, "Learning to detect malicious clients for robust federated learning," *arXiv preprint arXiv:2002.00211*, 2020.

[10] X. Mu *et al.*, "Feddmc: Efficient and robust federated learning via detecting malicious clients," *IEEE Transactions on Dependable and Secure Computing*, 2024.

[11] D. Gupta *et al.*, "Hierarchical federated learning based anomaly detection using digital twins for smart healthcare," in *2021 IEEE 7th international conference on collaboration and internet computing (CIC)*. IEEE, 2021, pp. 16–25.

[12] L. Praharaj *et al.*, "Hierarchical federated transfer learning and digital twin enhanced secure cooperative smart farming," in *2023 IEEE International Conference on Big Data (BigData)*. IEEE, 2023, pp. 3304–3313.

[13] H. Karim *et al.*, "Securing llm workloads with nist ai rmf in the internet of robotic things," *IEEE Access*, 2025.

[14] P. Blanchard *et al.*, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[15] D. Yin *et al.*, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*, 2018, pp. 5650–5659.

[16] C. Fung *et al.*, "The limitations of federated learning in sybil settings," in *Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, 2020, pp. 301–316.

[17] S. Ma *et al.*, "Flame: Taming backdoors in federated learning," in *USENIX Security Symposium*, 2022, pp. 295–312.

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

[19] C. Zhao *et al.*, "Differentially private pca in federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, 2021, pp. 9118–9126.

[20] W. Wan *et al.*, "Shielding federated learning: Robust aggregation with adaptive client selection," *arXiv preprint arXiv:2204.13256*, 2022.

[21] Z. Alsulaimawi, "Federated learning with anomaly detection via gradient and reconstruction analysis," *arXiv preprint arXiv:2403.10000*, 2024.

[22] J. Tang *et al.*, "Seaflame: Communication-efficient secure aggregation for federated learning against malicious entities," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2025, no. 2, pp. 69–93, 2025.

[23] "Handwritten digits dataset (not in mnist)," https://www.kaggle.com/datasets/jcprogjava/handwritten-digits-dataset-not-in-mnist, 2021, accessed: 2025-05-27.

[24] V. Tolpegin *et al.*, "Data Poisoning Attacks Against Federated Learning Systems," *arXiv e-prints*, p. arXiv:2007.08432, Jul. 2020.