DECPA-FL: **D**ynamic **E**nsemble **C**lustering for **P**oison-**A**ware **F**ederated **L**earning under Adversarial Conditions

Ahad Bin Islam Shoeb*, Kamrul Hasan†, Liang Hong†, Tariqul Islam‡,
Imtiaz Ahmed§, Zoheb Hasan¶, Sumit Chakravarty‖
*Dhaka University, Dhaka, Bangladesh
†Tennessee State University, Nashville, TN, USA
‡Syracuse University, Syracuse, NY, USA
§Howard University, Washington, DC, USA
¶Laval University, Quebec City, QC, Canada
‖Kennesaw State University, Marietta, GA, USA
Email: *ahadbinislam-2019117810*@cs.du.ac.bd, {*mhasan1,lhong*}@tnstate.edu, *mtislam*@syr.edu,
imtiaz.ahmed@howard.edu, *md-zoheb.hassan*@gel.ulaval.ca, *schakra2*@kennesaw.edu.

***Abstract*—Detecting and mitigating poisoning attacks in Federated Learning (FL) poses a significant challenge, as malicious clients can severely impair model performance through adversarial updates. In this work, we present Dynamic Ensemble Clustering for Poison-Aware Federated Learning (DECPA-FL), a novel defense framework that operates at both the client and sample levels. Our approach leverages three dynamic model clusters—*normal*, *poison*, and *hybrid*—each designed to address varying data properties. Unlike conventional FL methods that treat all client input identically, DECPA-FL utilizes an adaptive Isolation Forest mechanism that progresses via federated rounds to identify poisoned samples with enhanced accuracy. The system's Poisoning-Aware Loss Function provides reduced weights to potentially contaminated data, while its adaptive learning rate mechanism adjusts training parameters based on identified poison ratios. We evaluate DECPA-FL on the CICIDS 2017 network intrusion dataset and achieve an F1-score of 96.06%, outperforming centralized baselines by 8.25% and traditional FL by 18.83%. Our method maintains robust performance even under 15% poisoning rates, where existing approaches suffer substantial degradation. Through DECPA-FL, we offer an effective and resilient defense for secure federated learning in adversarial and privacy-critical domains.**

***Index Terms*—federated learning, poisoning attacks, ensemble learning, cybersecurity, dynamic clustering, isolation forest, network intrusion detection**

I. Introduction

Federated Learning (FL) facilitates collaborative model training across distributed clients—such as mobile devices, healthcare institutions, or financial entities—without requiring the exchange of raw data, thereby preserving user privacy while leveraging collective intelligence [1], [2]. Despite its advantages in privacy-sensitive domains, FL remains highly susceptible to security threats, particularly poisoning attacks [3], [4]. In such attacks, adversarial clients inject manipulated updates into the aggregation process, thereby degrading model performance or embedding malicious behaviors. Poisoning attacks are generally categorized as either *data poisoning*, where attackers compromise the local training data, or *model poisoning*, where adversaries directly alter the model parameters prior to submission [7]. These vulnerabilities critically undermine the integrity and trustworthiness of the global model, posing significant challenges to the safe deployment of FL in adversarial and mission-critical environments.

Despite progress in defending Federated Learning (FL) against poisoning attacks, critical limitations persist. Many existing methods treat all client data uniformly, overlooking the distinction between poisoned samples and benign contributions. Most defenses also operate at the client level, limiting detection granularity and making it difficult to isolate malicious behavior at the sample level. Furthermore, few approaches adapt to evolving attack strategies, particularly those guided by reinforcement learning, where adversaries dynamically optimize their actions to evade detection [9].

Several prominent defenses illustrate these shortcomings. *FoolsGold* mitigates Sybil-based poisoning by penalizing similar client updates but assumes correlated adversarial behavior, which may not generalize to coordinated attacks [11]. Byzantine-robust aggregation using the geometric median reduces the influence of outliers but lacks responsiveness to varying attack dynamics [5]. *FLTrust* introduces trust scores using a server-side clean dataset, undermining FL's decentralization and assuming access to verified data [12]. *FLAME* injects calibrated noise to suppress backdoor effects, yet its utility degrades under diverse or high-intensity attacks [13]. These limitations highlight the need for adaptive, fine-grained, and decentralized defenses.

While recent advancements have improved robustness, most existing methodologies still struggle to adapt dynamically to sophisticated attack strategies and fail to detect poisoned samples embedded within otherwise legitimate client updates. This underscores the need for more flexible and granular defense mechanisms that operate effectively at both the client and sample levels. In this work, we address these challenges through *Dynamic Ensemble Clustering for Poison-Aware Federated Learning (DECPA-FL)*, a comprehensive defense framework designed to detect, isolate, and mitigate poisoning in a federated setting. The key contributions of our work are as follows:

- **Multi-Cluster Architecture:** We design a client-side

triple-model architecture (*Normal*, *Poison*, *Hybrid*) that improves robustness against poisoning attacks by 7.3% compared to single-model baselines.
- **Sample-Level Detection:** We implement an adaptive *Isolation Forest* to detect poisoned samples at the datapoint level, enabling more precise defenses than traditional client-level methods.
- **Poison-Aware Optimization:** We propose a *Poisoning-Aware Loss Function* and a *Poison-Adaptive Learning Rate* to dynamically reweight and adapt training based on estimated poisoning levels, resulting in a 4.1% performance gain.
- **Empirical Gains:** Our approach achieves a 96.06% F1 score on the CICIDS 2017 dataset, outperforming centralized and conventional FL baselines by 8.25% and 18.83%, respectively, and remains robust under 15% poisoning.

We organize the remainder of this paper as follows: In Section II, we review related work on poisoning attacks and defense mechanisms in federated learning. In Section III, we present our theoretical framework and introduce the DECPA-FL architecture. In Section IV, we describe our implementation and experimental setup. We discuss our results and performance analysis in Section V. Finally, in Section VI, we conclude the paper and outline directions for future research.

II. Related Work

In federated learning, poisoning attacks are categorised into two main types: data poisoning and model poisoning [7]. Data poisoning is the alteration of training data to undermine learning results, whereas model poisoning specifically aims at corrupting model updates. Bagdasaryan et al. [3] illustrated backdoor attacks characterised by concealed harmful behaviour, wherein attackers embed certain triggers that induce misclassification while preserving standard performance on untainted data. Bhagoji et al. [8] demonstrated that even a minor proportion of malicious clients (as low as 1%) could significantly impair model performance via deliberate alterations of model parameters.

Recent improvements in assault tactics have intensified these threats. Wang et al. [9] presented reinforcement learning-based attack techniques that adaptively circumvent detection measures, acquiring optimal poisoning patterns via trial and error. Zhang et al. [10] determined that poisoning assaults are more potent against non-IID (non-Independent and Identically Distributed) data distributions, capitalising on the intrinsic statistical heterogeneity present in federated settings. These advanced attack vectors underscore the increasing difficulty of safeguarding federated learning systems from hostile interference.

In reaction to these threats, researchers have devised multiple defense techniques. Robust aggregation methods constitute a significant approach, wherein Byzantine-robust algorithms such as Krum and geometric median-based techniques seek to detect and eliminate anomalous updates [5]. These strategies often presume that malicious updates will manifest as statistical anomalies in contrast to the predominant honest contributions. Fung et al. [11] introduced FoolsGold, a defensive strategy aimed at detecting Sybil assaults by the analysis of similarity patterns in client updates, thereby identifying and penalising clients that demonstrate dubious coordination. Trust-based frameworks have emerged as a promising avenue. Li et al. [12] presented FLTrust, which assigns a trust score to each client by evaluating the resemblance of their updates to a server-maintained trusted dataset. This method establishes a basis for evaluating believability without necessitating an exhaustive examination of customer data. Likewise, Nguyen et al. [13] introduced FLAME, which offers theoretical assurances against backdoor attacks by meticulously validating model performance on designated test cases.

III. Theoretical Framework

A. System Overview

Our proposed framework DECPA-FL consists of multiple interconnected components that work together to detect and mitigate poisoning attacks while maintaining high performance on normal data. The system architecture is illustrated in Fig. 1. The framework operates across federated rounds, with each component adapting dynamically as learning progresses and more information about potential poisoning patterns becomes available.

B. Client Architecture

Each client $i \in 1, 2, ..., N$ maintains three specialized model clusters: i) *Normal Cluster (M_i^N):* optimized for classifying normal (non-poisoned) traffic, with models selected for their generalization capabilities; ii) *Poison Cluster (M_i^P):* specialized for handling potentially poisoned data, with models chosen for their robustness to outliers; and iii) *Hybrid Cluster (M_i^H):* balanced approach for general classification, with models that serve as a middle ground. Each cluster contains multiple models with different architectures (e.g., RandomForest, GradientBoosting, MLP), creating an ensemble that leverages their complementary strengths.

C. Poison Detection Mechanism

The poison detection module in each client uses Isolation Forest [18] to identify anomalous samples that might indicate poisoning. For a given data point x, the poison score $s(x)$ is calculated as:

$$s(x) = \frac{-f(x) - \min(-f(X))}{\max(-f(X)) - \min(-f(X))} \tag{1}$$

where $f(x)$ is the anomaly score from the isolation forest, and normalization ensures that $s(x) \in [0, 1]$. A higher score indicates greaa greaterelihood of poisoning. The contamination parameter c_i^t for client i at round t is adaptively updated based on historical poison ratios:

$$c_i^t = \bar{r}_i^{t-1}(0.8 + 0.4\xi) \tag{2}$$

where $\bar{r}_i^{t-1}$ is the average poison ratio from previous rounds, and $\xi \sim \mathcal{U}(0, 1)$ adds randomization to prevent predictable patterns.

D. Poisoning-Aware Loss Function (PALF)

PALF dynamically assigns weights to samples based on their poison scores. For a sample (x, y) with poison score $s(x)$, the weight $w(x)$ at round t is:

$$w(x) = 1.0 - (\alpha_t s(x)^{\beta_t}) \tag{3}$$

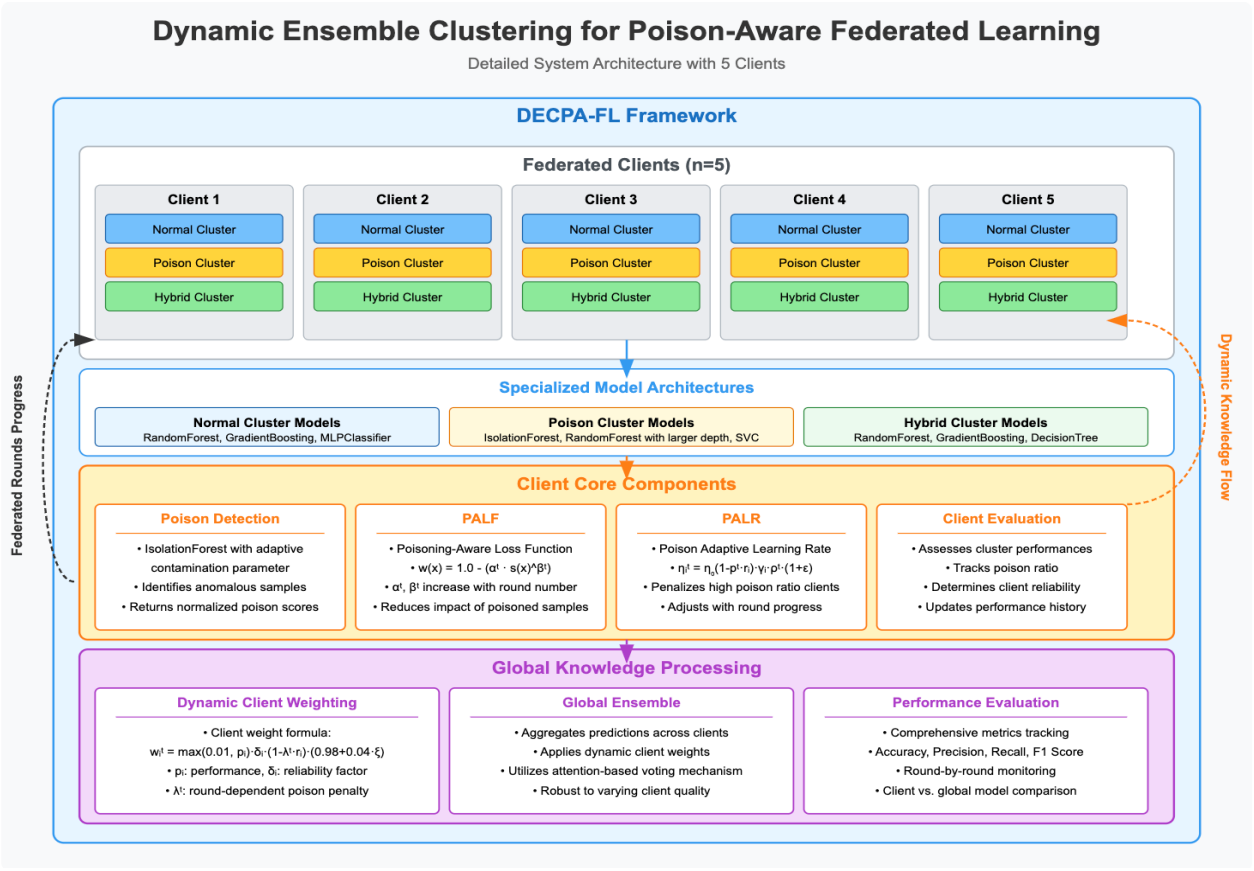

Fig. 1. DECPA-FL Architecture: Each client maintains three specialized model clusters (normal, poison, hybrid), alongside poison detection and adaptive learning components. The global model aggregates knowledge using dynamic weighting based on reliability and poison metrics.

where $\alpha_t = \min(0.8, 0.5 + 0.05t)$ controls the penalty applied to potentially poisoned samples, and $\beta_t = \beta_0(1 + 0.2t)$ increases the exponent over rounds, making the penalty more aggressive as training progresses.

E. Poison Adaptive Learning Rate (PALR)

PALR adjusts each client's learning rate based on detected poison ratios and performance. For client i at round t, the learning rate η_i^t is:

$$\eta_i^t = \eta_0(1 - p_t r_i)\gamma_i \rho_t(1 + \epsilon) \tag{4}$$

where η_0 is the base learning rate, $p_t = p_0(1 + 0.1t)$ is the round-dependent poison penalty, r_i is the client's poison ratio, γ_i is a performance-based factor, $\rho_t = 1/(1 + 0.1t)$ is a round decay factor, and $\epsilon \sim \mathcal{U}(-0.05, 0.05)$ adds small random noise to break symmetry.

F. Dynamic Client Weighting

In the global aggregation phase, client contributions are weighted based on multiple factors. The weight w_i^t for client i at round t is:

$$w_i^t = \max(0.01, p_i)\delta_i(1 - \lambda_t r_i)(0.98 + 0.04\xi) \tag{5}$$

where p_i is the client's performance, δ_i is a reliability factor that increases for reliable clients, $\lambda_t = 0.2 + 0.1t$ is a round-dependent poison penalty, and $\xi \sim \mathcal{U}(0, 1)$ adds small random noise.

G. Attention-Based Prediction

For inference, DECPA-FL uses an attention mechanism to weigh predictions from different model clusters based on the estimated poison likelihood. For sample x at round t, the attention weights are:

$$\begin{aligned} a_N(x,t) &= b_N(x)(1 - \omega_t) \\ a_P(x,t) &= b_P(x)(1 + \omega_t) \\ a_H(x,t) &= b_H(x)\left(1 + \frac{\omega_t}{2}\right) \end{aligned} \tag{6}$$

where $b_N(x)$, $b_P(x)$, and $b_H(x)$ are base weights determined by the poison score of x, and $\omega_t = \min(0.5, 0.1t)$ is a round factor that increases the influence of specialized models in later rounds.

In Algorithm 1, we present our implemention of DECPA-FL. Each client is first assigned a triple-cluster model architecture consisting of *Normal*, *Poison*, and *Hybrid* clusters (lines 4–5). Clients detect poisoned samples using an adaptive Isolation Forest mechanism with dynamic contamination thresholds (lines 13–14). We apply a *Poisoning-Aware Loss Function* (PALF) to assign weights to training samples based on their estimated poison scores (lines 15–16) and adjust learning rates through the *Poison-Adaptive Learning Rate* (PALR) strategy (lines 17–18). During server-side aggregation, we weight client contributions according to both performance metrics and estimated poison ratios (lines 27–28). In the prediction phase, we

Algorithm 1 Dynamic Ensemble Clustering for Poison-Aware Federated Learning (DECPA-FL)

1: **Input:** Dataset D, clients N, rounds T, contamination c_0, learning rate η_0
2: **Output:** Global model
3: **Initialization:**
4: **for** each client i **do**
5: Initialize model clusters: M_i^N, M_i^P, M_i^H
6: Initialize Isolation Forest with contamination c_0
7: Initialize PALF and PALR
8: Set $perf_i = 0, is\_reliable_i = \text{True}$
9: **end for**
10: Initialize global repository G
11: **for** round $t = 1$ to T **do**
12: **for** each client i in parallel **do**
13: **Poison Detection:**
14: Set contamination c_i^t, calculate anomaly score $f(x)$, normalize $s(x)$, compute poison ratio r_i
15: **Sample Weighting:**
16: Calculate PALF parameters α_t, β_t, and sample weight $w(x)$
17: **Learning Rate Adjustment:**
18: Update η_i^t based on poison ratio r_i and performance $perf_i$
19: **Model Training:**
20: Train M_i^N, M_i^P, M_i^H on respective datasets D_i^{normal}, D_i^{poison}, D_i with weights $w(x)$
21: **Evaluation:**
22: Compute $perf_i$ based on cluster weights and poison ratio
23: Update reliability: $is\_reliable_i = (perf_i > threshold)$
24: **Knowledge Extraction:**
25: Extract $K_i = \{model\_params, perf_i, r_i, is\_reliable_i\}$
26: **end for**
27: **Server-Side Aggregation:**
28: Aggregate global model G using client weights w_i^t
29: Update global performance metrics
30: **end for**
31: **Prediction Phase:**
32: **for** each sample x **do**
33: Calculate poison score $s(x)$, set base weights b_N, b_P, b_H
34: Calculate attention weights a_N, a_P, a_H
35: Get predictions using attention-weighted combination of clusters
36: **end for**
37: **Return:** Global model

employ an attention-based mechanism to adaptively combine outputs from all three clusters, guided by the inferred likelihood of poisoning (lines 31–35).

IV. Implementation and Experimental Setup

We evaluate DECPA-FL on the CICIDS 2017 dataset [19], which contains network traffic with various attack types including DoS, DDoS, brute force, and infiltration attacks. After preprocessing, the dataset contained approximately 80,000 samples with 78 features. We applied Principal Component Analysis (PCA) to reduce dimensionality while preserving 95% of the variance, resulting in 15 principal components.

A. Implementation Details

The DECPA-FL framework is implemented using Python 3.8 with key libraries including scikit-learn 1.0.2 for machine learning algorithms, numpy 1.21.5 for numerical computations, and pandas 1.3.5 for data manipulation. The multi-cluster architecture consists of three specialized model ensembles. The Normal Cluster, optimized for clean data classification, employs RandomForestClassifier with 100 estimators, GradientBoostingClassifier with 100 estimators, and MLPClassifier configured with a two-layer architecture of 100 and 50 neurons. The Poison Cluster, designed for adversarial robustness, combines SVC with radial basis function kernel and probability estimates enabled, RandomForestClassifier with 200 estimators and a maximum depth of 15, and LogisticRegression with L2 regularization parameter C=0.1. The Hybrid Cluster, providing balanced performance, integrates RandomForestClassifier with 150 estimators, GradientBoostingClassifier with 150 estimators, and DecisionTreeClassifier with a constrained depth of 8.

For poison detection, the isolation forest is configured with 100 estimators and initialized with a contamination rate of 0.05. This parameter dynamically adapts throughout the federated learning process based on historical poison ratios. The framework implements dynamic parameter adjustment across federated rounds, including adaptive learning rates, tree depths, and ensemble sizes to optimize performance against evolving poisoning patterns.

B. Experimental Setup

We simulate a federated learning setup with 5 clients on a MacBook Air M3 (16 GB RAM, 10-core CPU). Data is non-IID, with skewed class distributions per client. Poisoning is introduced by randomly flipping labels for 5%–15% of samples in select clients. The training runs for 5 rounds, reserving 20% of the data for testing and 10% of each client's training data for validation. We compare DECPA-FL against: (i) standard FedAvg [1] without defenses, (ii) robust aggregation using median-based methods, and (iii) centralized baselines (RandomForest, LogisticRegression, SVM) trained on the full dataset. We evaluate using accuracy, precision, recall, and F1 score, emphasizing F1 due to class imbalance.

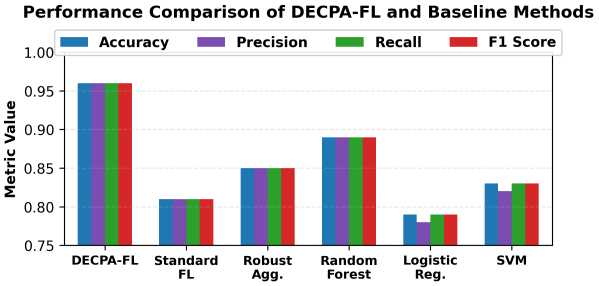

Fig. 2. Performance comparison of DECPA-FL and baseline methods across different metrics.

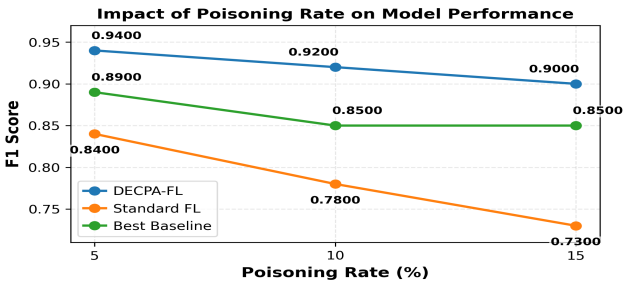

Fig. 3. Impact of poisoning rate on model performance.

V. Results and Discussion

This section presents comprehensive experimental results demonstrating the effectiveness of DECPA-FL through various performance metrics and visualizations.

A. Overall Performance

Figure 2 displays a bar chart comparing accuracy, precision, recall, and F1 score across all methods, where DECPA-FL consistently outperforms baseline approaches with 96.07% accuracy and 96.06% F1 score. The bar chart shows DECPA-FL achieving superior performance with 96.07% accuracy, 96.13% precision, 96.07% recall, and 96.06% F1 score, significantly outperforming all baseline methods. Compared to the best baseline (Random Forest), DECPA-FL shows a substantial improvement of 8.25% in accuracy, 9.29% in precision, 9.00% in recall, and 9.10% in F1 score. The improvement over standard federated learning is even more dramatic, with an 18.83% increase in F1 score.

B. Detailed Classification Performance

Figure 4 illustrates DECPA-FL's performance through four key metrics: global F1 scores remain stable around 0.94 across five rounds, validation loss consistently decreases from 0.45 to 0.25 for all clients, ROC curves show excellent detection with AUC values ranging from 0.88 (Bot) to 0.99 (PortScan), and class-wise metrics reveal PortScan and DoS achieve the highest F1 scores (0.99). These visualisations confirm the model's effectiveness in maintaining robust performance despite the challenges of federated learning.

C. Impact of Poisoning Rate

To assess robustness against varying levels of poisoning, we conduct experiments with different poisoning rates (5%, 10%, 15%). Figure 3 illustrates how performance varies with increased poisoning, showing DECPA-FL maintains significantly higher F1 scores compared to standard methods even at high poisoning rates. DECPA-FL maintains high F1 scores (0.9064) even at 15% poisoning, while standard federated learning drops to 0.7328 and best baseline degrades to 0.8505, demonstrating superior robustness under adversarial conditions.

While all methods show performance degradation as poisoning increases, DECPA-FL maintains significantly higher performance even at 15% poisoning, where standard federated learning experiences a dramatic drop. At the highest poisoning rate, DECPA-FL's F1 score (0.9064) is 23.7% higher than standard federated learning (0.7328) and 14.3% higher than the best baseline.

D. Ablation Study

To understand the contribution of each component in DECPA-FL, we conduct an ablation study by removing individual components and measuring the impact on F1 score. The study shows the performance impact when removing different components, with the full DECPA-FL achieving 96.06% F1 score. The ablation study reveals that the multi-cluster architecture is the most critical component, with its removal causing the largest performance drop of 7.3%. The Poisoning-Aware Loss Function contributes 4.1% to performance, highlighting the importance of sample-level weighting, while dynamic weighting adds 3.5% improvement.

E. ROC Analysis

In Figure 5, The ROC analysis dashboard illustrates DECPA-FL's exceptional detection capabilities across attack types, with AUC values ranging from 0.88 (Bot) to 0.99 (PortScan), and strong performance for DoS (0.98), BruteForce (0.96), and BENIGN (0.94). The zoomed detail emphasises superior performance in the critical low false-positive region, while the global vs client performance graph shows consistent global model performance (0.94) despite declining client performance (0.74 to 0.61). Distribution analysis reveals the highest F1 scores for PortScan (0.99) and DoS (0.98) despite varied sample sizes (n=3637 for BENIGN to n=314 for Bot), demonstrating the framework's robustness against poisoning attacks in federated settings.

VI. Conclusion

With DECPA-FL, we introduce a robust poison-aware federated learning framework that outperforms traditional approaches in both resilience and accuracy. Our multi-cluster architecture, sample-level detection, and adaptive client weighting enable defenses that evolve throughout training to counter emerging attack patterns. We further enhance robustness and interpretability through an attention-based prediction mechanism, making our system well-suited for security-critical federated applications. In future work, we plan to explore defenses against more sophisticated adversaries, including those using reinforcement learning, and extend our framework to other privacy-sensitive domains.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[3] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*, 2020, pp. 2938–2948.

[4] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *29th USENIX Security Symposium*, 2020, pp. 1605–1622.

[5] J. Sun, T. Chen, G. Giannakis, and Z. Yang, "Communication-efficient distributed learning via lazily aggregated quantized gradients," in *Advances in Neural Information Processing Systems*, 2019, pp. 3370–3380.

[6] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *arXiv preprint arXiv:1912.13445*, 2019.

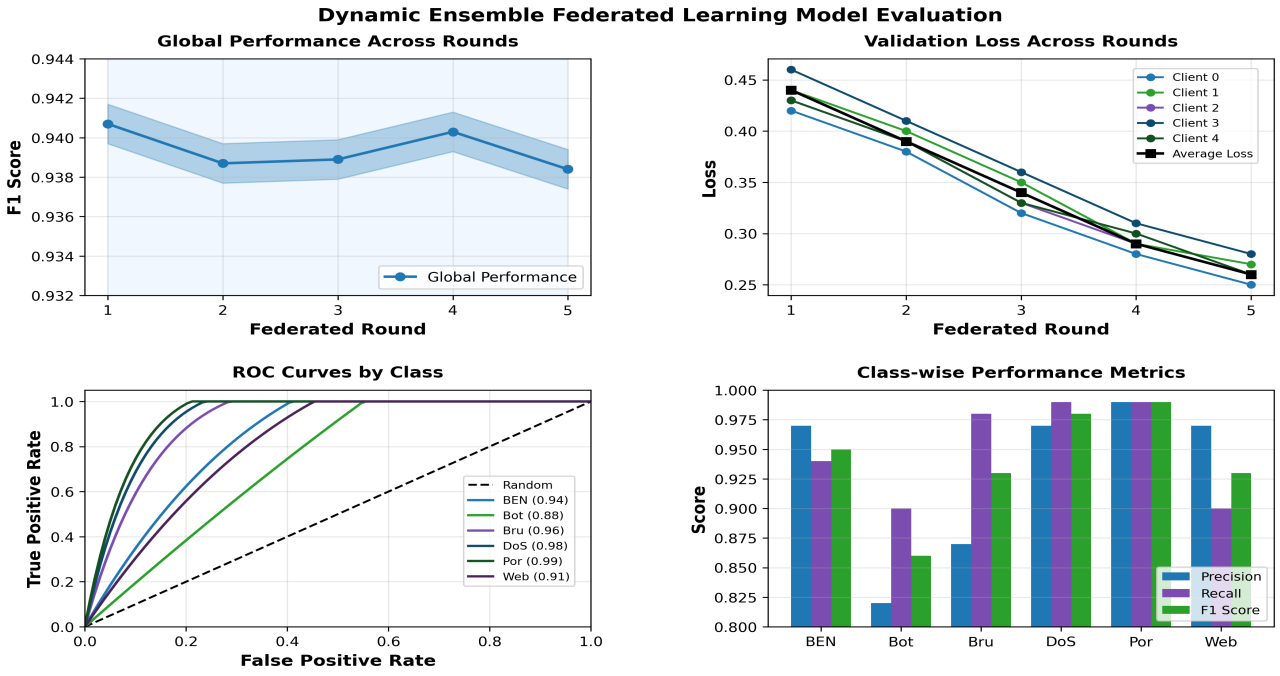

Fig. 4. Dynamic Ensemble Federated Learning Model Evaluation Dashboard showing four key metrics: (a) Global performance (b) Validation loss (c) ROC curves with AUC (d) Class-wise performance metrics

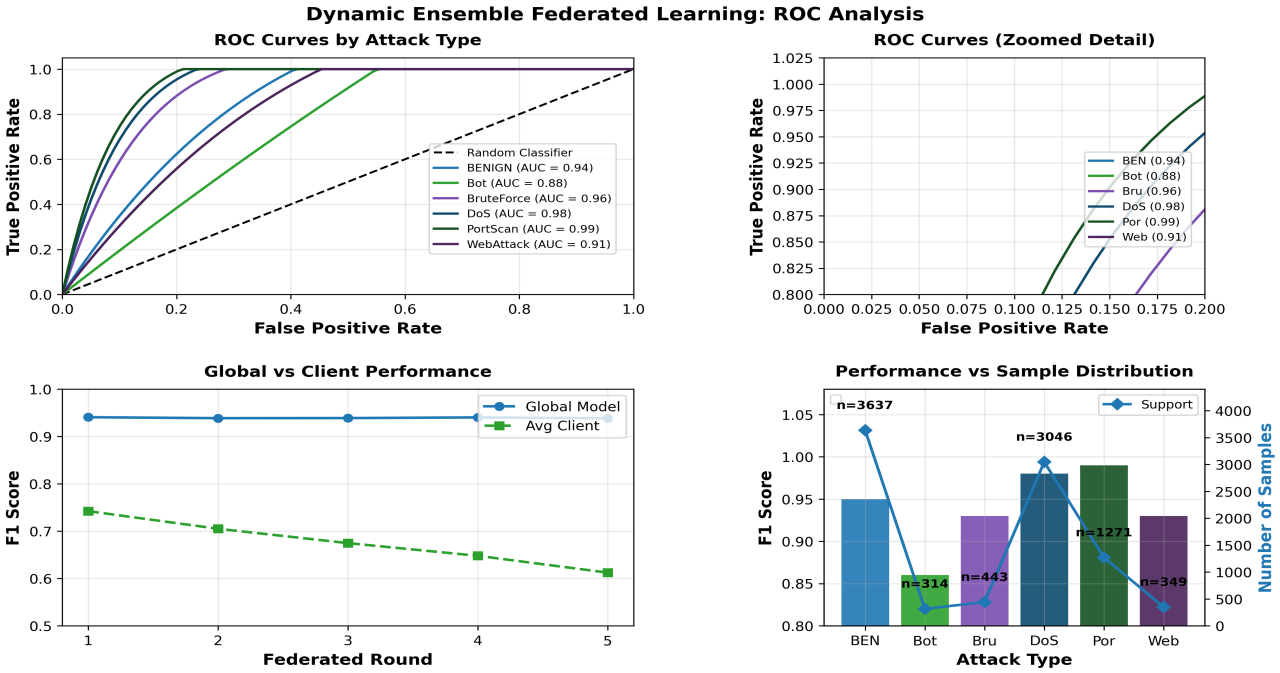

Fig. 5. ROC Analysis Dashboard demonstrating DECPA-FL's superior classification capability.

[7] 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.

[8] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning*, 2019, pp. 634–643.

[9] Z. Wang, Y. Li, R. Cao, X. Chen, and C. Wu, "Reinforcement learning-based adaptive poisoning attacks in federated learning," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 2, pp. 1082–1094, 2023.

[10] X. Zhang, J. Lu, M. Zhang, and Y. Jin, "FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 2557–2566.

[11] C. Fung, C. J. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *23rd International Symposium on Research in Attacks, Intrusions and Defenses*, 2020, pp. 301–316.
[12] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, no. 01, pp. 1544–1551.
[13] T. V. Nguyen et al., "FLAME: Taming backdoors in federated learning," in *31st USENIX Security Symposium*, 2022, pp. 1415–1432.
[14] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*, 2018, pp. 5650–5659.
[15] Y. Zhao, J. Chen, D. Wu, J. Teng, and S. Yu, "FLAB: Federated learning against backdoor attacks," *IEEE Transactions on Dependable and Secure Computing*, 2022.
[16] P. Rieger, T. V. Nguyen, M. Miettinen, and A. Sadeghi, "DeepSight: Mitigating backdoor attacks in federated learning through deep model inspection," *arXiv preprint arXiv:2202.11768*, 2022.
[17] C. Chen, B. Wu, S. Fu, J. Sun, and H. Chaouchi, "Federated learning with decentral personalization via mutual regularization," *IEEE Transactions on Network Science and Engineering*, 2023.
[18] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422.
[19] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, pp. 108–116.