Fedshield: Privacy-Preserving Phishing and Fraud Detection Using Federated Learning With Client-Side SMOTE

Ms. Kavya Jagtap[1], Ms. Vedika Thakarke[2], Dr. Sumedh Pundkar[3]
[1,2,3]*Computer Science and Technology, Usha Mittal Institute of Technology Mumbai, India*

***Abstract*—Phishing and online fraud have become major cyber- security issue, especially in e-commerce, banking and web-based services. Machine Learning based phishing detection systems exists but they are based on centralized data collection which compromises user privacy and breaches data privacy laws. Moreover, phishing datasets are naturally imbalance, causing poor minority fraudulent instance detection. To overcome this issue, this paper presents a privacy preserving phishing detection system that combines Federated Learning along with Syn- thetic Minority Oversampling Technique (SMOTE). In proposed method, raw data are kept local to participating clients, where SMOTE is performed locally to address class imbalance before model training. The model updates are aggregated at the central server by Federated Averaging Fed Avg algorithm which prevents privacy. Experimental results on a real-world phishing website dataset show that the proposed FL-SMOTE system achieves 95.30% accuracy, which is substantially higher than conventional federated learning while strictly preserving data privacy. The results show that client side SMOTE is effective in improving minority class detection with a slight performance difference as compared to centralized system. This paper presents the scalable and privacy preserving phishing detection solution.**

***Index Terms*—Phishing Detection, Fraud Detection, Federated Learning, Client-Side SMOTE, Privacy-Preserving, Machine Learning**

I. INTRODUCTION

Phishing attacks represent one of the most prevalent and damaging forms of cybercrime, exploiting deceptive websites, malicious URLs, and social-engineering techniques to obtain sensitive user information. With the rapid expansion of online banking, digital payments, and cloud-based services, phishing attacks have increased both in frequency and sophistication. Traditional rule-based and blacklist-based detection mechanisms are increasingly ineffective, as attackers rapidly modify phishing patterns to evade static defenses.

Machine-learning-based phishing detection has emerged as a powerful alternative, enabling automated learning of Identify applicable funding agency here. If none, delete this. complex patterns from historical data. Centralized machine- learning models such as Random Forests, Support Vector Machines, and deep neural networks have demonstrated strong detection performance. However, these approaches require aggregating sensitive browsing and transaction data from multiple organizations, raising serious privacy and security concerns. Such centralized data collection directly conflicts with stringent regulations such as the General Data Protection Regulation (GDPR), limiting real-world deployment.

Federated Learning (FL) addresses this challenge by enabling collaborative model training without sharing raw data [2]. In FL, clients train models locally and share only model parameters with a central server. Despite its privacy benefits, standard FL suffers from two major limitations in phishing detection. First, phishing datasets are highly imbalanced, with fraudulent samples forming a minority class, leading to biased models that favor legitimate traffic. Second, recent studies have demonstrated that model gradients in FL can leak sensitive information if additional safeguards are not employed [18].

To overcome these limitations, this paper proposes an enhanced federated learning framework that integrates client- side SMOTE to handle class imbalance locally while preserving privacy. By applying SMOTE at

each client before training, the proposed approach improves minority-class learning without violating federated principles.

II. BACKGROUND

A. Centralized Learning

Traditional phishing and fraud detection systems employ Centralized Learning (CL), where data from multiple sources are aggregated into a single repository for model training. While effective for achieving high accuracy, this approach conflicts with modern privacy regulations (GDPR, CCPA) and creates significant security vulnerabilities by concentrating sensitive data in one location [1]. The increasing volume of cybersecurity data further escalates computational and storage demands, making centralized approaches increasingly impractical for distributed, privacy-sensitive applications.

B. Federated Learning Fundamentals

Federated Learning (FL) [2] addresses these limitations by enabling collaborative model training without data sharing. In FL, multiple clients train local models on their private datasets and exchange only model parameters with a central server, which aggregates them to form a global model. The aggregation follows the Federated Averaging rule:

$$\mathbf{W}^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{n} \mathbf{W}_k^{(t)} \qquad (1)$$

where $\mathbf{W}_k^{(t)}$ represents client k's local model at round t, n_k is its simple count, and $n = \sum n$. This process preserves data privacy while allowing knowledge transfer across organizations.

C. Class Imbalance and SMOTE

Cybersecurity datasets exhibit severe class imbalance, with phishing/fraud instances significantly outnumbered by legitimate ones. This bias degrades detection performance for the critical minority class. The Synthetic Minority Oversampling Technique (SMOTE) [3] mitigates this by generating synthetic minority samples via interpolation between existing instances in feature space:

$$x_{new} = x_i + \lambda(x_j - x_i), \qquad \lambda \in (0, 1) \qquad (2)$$

When integrated with FL, SMOTE can be applied locally at each client, addressing imbalance without compromising privacy—a key advantage over centralized resampling approaches. The combination of FL with client-side SMOTE enables privacy-preserving, balanced learning for phishing detection.

III. LITERATURE REVIEW

A. Federated Learning for Phishing and Fraud Detection

Federated learning (FL) has emerged as a compelling paradigm for phishing and fraud detection when institutions are unable or unwilling to share raw data due to privacy, legal, or commercial constraints. In FL, models are trained locally on each client, and only model updates are aggregated to form a global model, thereby reducing direct exposure of sensitive information [5], [6], [10], [12], [19], [21].

In phishing detection, [21] evaluate FL with recurrent convolutional neural networks (RNN) and BERT for email classification under different organizational partitions and data distributions. Their work shows that, for balanced datasets and a small number of participating organizations, FL can achieve accuracy comparable to centralized training, although performance degrades when data distributions become highly asymmetric or the number of clients is large [21]. FedPhish- LLM extends this line by integrating FL with multimodal large language models (LLMs) for phishing detection, enabling decentralized training over heterogeneous text and contextual indicators while avoiding dependence on external commercial LLM APIs [16]. FedPhishLLM reports up to 95% accuracy, precision, and F1-score, with 96% recall, and highlights robustness to adversarial and evasive attacks as well as linguistic diversity [16].

In the financial domain, FL has been applied to credit-card fraud detection across multiple institutions. [19] use FL across TensorFlow Federated and PyTorch to learn a shared classifier over the widely used Kaggle credit-card dataset, addressing data imbalance via a broad suite of individual and hybrid resampling strategies, including SMOTE-based combinations.

Random Forest, Logistic Regression, K-Nearest Neighbours,

Decision Trees, Na¨ıve Bayes, and a CNN are compared; Random Forest, combined with hybrid resampling, yields the best performance with

accuracies approaching 99.99% and high precision, recall, and F1-scores [19]. [10] employ the Flower framework with FedAvg, FedProx, and FedOpt to build federated intrusion and fraud detectors on the UNSW-NB15 intrusion dataset and a credit-card dataset, achieving accuracies above 99.8% while maintaining client data locality [10].[6] integrate FL with blockchain to support credit-card fraud detection across three banks; SMOTE is applied to rebalance transactions before local training with Random Forest, CNN, and LSTM classifiers, and blockchain provides tamper-evident logging and decentralized coordination of model updates [6]. [12] propose an adaptive FL (AFL) framework that weights client contributions by local detection performance and incorporates a multi-stage data balancing pipeline combining Tomek links under sampling, borderline-SMOTE over- sampling, and cognitive sample pruning on the Kaggle 2013 and Sparkov datasets, achieving up to 99% accuracy and near-perfect precision and recall [12].[14] introduce a semi-decentralized architecture based on FL and a VAE–QLSTM fusion model for real-time fraud detection in banking networks. Their framework leverages variational autoencoders for feature learning and quantum-enhanced LSTMs for temporal modelling, deployed with TensorFlow Federated on the IEEE-CIS and European cardholder datasets, and attains 94.5% accuracy and 91.3% sensitivity, outperforming several existing approaches [14].

Thematically, these contributions show that FL can deliver competitive performance for phishing and financial fraud detection while allowing institutions to retain control over local data. However, most works treat FL primarily as an architectural mechanism; explicit, provable privacy guarantees and systematic handling of data imbalance within the federated pipeline remain only partially addressed [6], [10], [12], [14],[16], [19], [21].

B. Privacy-Preserving Mechanisms in Federated Learning

While FL inherently reduces direct access to raw data, it does not, on its own, guarantee formal privacy. Researchers therefore increasingly combine FL with cryptographic and statistical privacy mechanisms, as well as robustness techniques against poisoning.[18] evaluate differential privacy (DP) in an FL-enabled intrusion detection system for industrial IoT. Using the ToN_IoT dataset partitioned in a non-IID manner, they compare FedAvg and Fed+ aggregation under different DP budgets. Their findings indicate that Fed+ can achieve accuracy comparable to FedAvg, even when noise is added to satisfy DP requirements, suggesting that carefully tuned DP can be integrated into FL without catastrophic performance degradation [18].[22] focus on resilience to model-poisoning attacks in privacy-preserving FL. They propose an internal auditor that analyses encrypted gradient updates using Gaussian mixture models and Mahalanobis distance, enabling byzantine-tolerant aggregation while gradients remain protected by additive ho- momorphic encryption [22]. Their results show superior model accuracy, privacy, and computational efficiency compared to existing fully homomorphic or two-trapdoor homomorphic schemes [22].

A systematic review by [4] surveys blockchain-based FL (BCFL) architectures for cryptocurrency fraud detection, high- lighting that integrating privacy technologies such as secure multiparty computation and differential privacy into BCFL raises non-trivial scalability and communication challenges [4]. Collectively, these studies demonstrate that there are practical pathways to augment FL with formal privacy mechanisms and robust aggregation, but such techniques have only rarely been applied directly in phishing or banking fraud contexts [6], [10], [12], [16], [19], [21].

C. Phishing and Fraud Detection Models and Data Imbalance

Beyond FL, a substantial body of work addresses the core modelling and data imbalance challenges in fraud detection, providing techniques that are conceptually compatible with, but not yet widely embedded within, federated settings.[17] propose a deep learning ensemble for credit-card fraud detection consisting of LSTM and GRU base learners within a stacking framework, with a multilayer perceptron serving as meta-learner. The Kaggle European cardholder dataset is first balanced using SMOTE-ENN, a hybrid combining Synthetic Minority Over-sampling Technique (SMOTE) with Edited Nearest Neighbour noise removal [17]. The ensemble coupled with SMOTE-ENN attains a sensitivity of 1.000 and specificity of 0.997, outperforming popular machine-learning baselines [17]. [8] adopt SMOTE-ENN in

Medicare fraud detection, achieving Decision Tree performance of 0.99 across accuracy, F1-score, recall, precision, and AUC-ROC on the Medicare Part B dataset, and underlining the importance of hybrid resampling and appropriate evaluation metrics for imbalanced healthcare fraud data [8].[9] systematically compare SMOTE, generative adversarial networks (GANs), and several hybrid SMOTE–GAN variants (SMOTified-GAN, SMOTE+GAN, GANified-SMOTE) using feed-forward neural networks, CNNs, and hybrid FNN+CNN architectures on financial fraud datasets [9]. Their experiments show that hybrid SMOTE–GAN techniques consistently achieve higher precision, recall, and F1-scores than either SMOTE or GAN alone, and that certain hybrids (e.g., GANified-SMOTE) remain stable across different quantities of generated minority samples [9]. [11] introduce AE-XGB-SMOTE-CGAN, a two-phase oversampling pipeline in which SMOTE first generates synthetic minority samples, then a conditional GAN refines these into more realistic distributions; an autoencoder performs feature extraction, and XG- Boost carries out classification on ananonymised bank dataset [11]. This method improves overall accuracy by roughly 2% compared with KNN and LightGBM, and raises Matthew's correlation coefficient by about 30% over KNN at a 0.35 decision threshold [11]. [13]'s SMOTE-OSBNR algorithm com- bines SMOTE with one-side behavioural noise reduction to remove noisy or overlapping majority-class instances; across seven datasets and tree-based ensemble methods, it reaches AUC-ROC values up to 0.999 and F1-scores up to 0.990, surpassing several established sampling approaches [13].

Alternative augmentation approaches also show promise.

[7] propose injecting Gaussian noise into minority-class samples in bank-fraud data, reporting that XGBoost trained on Gaussian-augmented data attains an accuracy of 0.999507 and AUC of 0.999506, outperforming SMOTE and ADASYN [7].[20] examine the joint effects of sampling and feature extraction on the Kaggle credit-card dataset, showing that random under sampling followed by convolutional autoencoder-based feature extraction can yield superior F1 and AUC compared with SMOTE-based schemes for ensemble classifiers [20]. These findings suggest that synthetic oversampling is not universally optimal and that hybrid strategies combining undersampling, oversampling, and representation learning must be tuned to the specific dataset and model [7]–[9], [11], [13], [17], [20].

Explainability has been recognized as critical in financial settings where model decisions may have regulatory and customer-impactful consequences. [5] design a fraud-detection framework integrating FL with explainable AI (XAI) on realistic transaction datasets, demonstrating consistently high predictive performance while providing interpretable model outputs to human experts [5]. [15] similarly advocate gradient boosting, Random Forests, and Decision Trees within an FL and XAI framework using the Paysim1 simulated mobile-money dataset, arguing that such tree-based models offer greater transparency and stability than deep architectures typically deployed in fraud detection, while still benefiting from privacy-preserving training [15].

D. Comparative Analysis and Research Gaps

The reviewed literature indicates that FL can achieve performance comparable to, and sometimes exceeding, centralized training for phishing and fraud detection, provided that data distributions across clients are not excessively skewed and that suitable aggregation and local models are chosen [6], [10], [12], [19], [21]. Advanced techniques for dealing with class imbalance—such as SMOTE-ENN, SMOTE-GAN hybrids, SMOTE-OSBNR, Gaussian noise augmentation, and combinations of under sampling with deep feature extraction—substantially improve recall, F1-score, and AUC in centralized fraud-detection benchmarks [7]– [9], [11], [13], [17], [20]. However, these sophisticated resampling methodologies are only beginning to be incorporated into federated pipelines, and then mostly in relatively simple forms (e.g., SMOTE or hybrid SMOTE+ADASYN at each client) [6], [12], [19].From a privacy perspective, most phishing and financial-fraud FL systems rely on the structural privacy of not sharing raw data, without providing formal guarantees such as differential privacy or secure aggregation [5], [6], [10], [12], [14], [16], [19], [21]. In contrast, work in industrial IoT intrusion detection and generic FL security has demonstrated that DP and homomorphic encryption can be applied

in federated settings while retaining competitive accuracy and acceptable computational overhead [18], [22]. This highlights a prominent gap: the lack of end-to-end frameworks that combine (i) formal privacy mechanisms, (ii) robust federated aggregation against poisoning and backdoor attacks, and (iii) advanced, hybrid oversampling tailored to highly imbalanced, multi-institutional fraud and phishing data.

Furthermore, although explainability has been addressed in several FL-based fraud detection systems, there is limited empirical analysis of how privacy parameters (e.g., DP noise), oversampling strategies, and client heterogeneity affect explanation fidelity and stability [5], [12], [15]. Dataset diversity also remains constrained: many studies rely on a small set of public or simulated datasets (Kaggle European cardholder, Paysim1, Sparkov, IEEE-CIS), which may not fully capture real-world transaction dynamics or adversarial behavior [6], [10]–[12], [14], [15], [17], [19], [20].

Recent works begin to close some of these gaps by proposing adaptive FL aggregation coupled with multi-stage imbalance adjustment [12], demonstrating practical DP and robust encrypted aggregation in non-IID regimes [18], [22], and developing multimodal FL-based LLMs for phishing with adversarial robustness and explainability [16]. Nonetheless, there remains significant scope for research on privacy-preserving phishing and fraud detection that jointly optimises federated learning, rigorous privacy guarantees, and principled synthetic oversampling in realistic, cross-institutional environments.

IV. PROPOSED SOLUTION

A. System Overview

The proposed framework implements a privacy-preserving phishing and fraud detection system based on a federated learning architecture with integrated synthetic oversampling capabilities. As illustrated in Figure 1, the system comprises multiple distributed clients (representing different organizations or data silos) that collaboratively train a global detection model without sharing their raw, sensitive data. Each client maintains local phishing/fraud datasets that remain on- premises throughout the training process. A central server co- ordinates the federated learning process by aggregating model updates received from participating clients while enforcing strict privacy preservation protocols. The global model, implemented as a neural network classifier, learns discriminative patterns from decentralized data distributions while mitigating class imbalance through client-side synthetic data generation. The architectural design follows a star topology where the central server acts as an orchestrator, initializing the global model and distributing it to clients for local training. After completing local training rounds, clients send only encrypted model parameter updates (gradients or weights) to the server. The server aggregates these updates using the Federated Averaging (FedAvg) algorithm to produce an improved global model, which is then redistributed for subsequent training rounds. This iterative process continues until model convergence is achieved, effectively creating a robust detection model trained on distributed data while maintaining data sovereignty
for each participating entity.

B. Data Preprocessing and Imbalance Handling

1. Data Preparation and Feature Engineering:

The system processes phishing datasets containing URL and webpage characteristics encoded as numerical features with ternary values typically representing legitimate (-1), suspicious (0), and phishing (1) indicators. Each client independently preprocesses its local dataset by decoding categorical representations, handling missing values, and normalizing feature scales. Feature analysis identifies key discriminative attributes including SSL certificate status, URL structural characteristics, and webpage ranking metrics that exhibit strong correlations with phishing classifications. The preprocessing pipeline ensures consistent feature representation across clients while preserving local data distributions.

2. *Synthetic Oversampling Strategy:*

To address the inherent class imbalance in phishing and fraud datasets—where legitimate samples typically outnumber malicious instances— each client implements the Synthetic Minority Oversampling Technique (SMOTE) locally before training. This client-side oversampling approach generates synthetic phishing samples by interpolating feature vectors of existing minority-class in- stances within their local feature space. The SMOTE algorithm operates by:

1) Identifying k-nearest neighbors for each minority class sample
2) Generating synthetic examples along line segments connecting the sample and its neighbors
3) Balancing class distributions while preserving decision boundaries

The local application of SMOTE ensures that synthetic data generation respects each client's unique data distribution and privacy constraints, preventing information leakage that could occur with centralized oversampling. This approach significantly improves model sensitivity to phishing indicators while maintaining the statistical properties of each client's local dataset.

C. Federated Learning Framework

1. Client-Server Coordination Protocol:

The federated learning process follows a synchronous, round-based coordination protocol between the central server and participating clients. Each communication round consists of four sequential phases:

1) Model Distribution Phase: The server selects a subset of available clients and transmits the current global model parameters.
2) Local Training Phase: Each selected client trains the model on its locally augmented dataset using a specified number of epochs.
3) Update Transmission Phase: Clients compute model updates (difference between initial and trained parameters) and transmit encrypted updates to the server.
4) Aggregation Phase: The server applies the FedAvg algorithm to compute a weighted average of received updates, incorporating them into the global model.

The framework employs a client sampling strategy with configurable participation rates, allowing scalable deployment across varying numbers of distributed data sources. Communication protocols include compression and encryption mechanisms to minimize bandwidth requirements and enhance security during parameter transmission.

2. Privacy-Preserving Mechanisms:

The proposed solution ensures data privacy through multiple complementary mechanisms:

- Data Sovereignty: Raw training data never leaves client premises, eliminating direct exposure of sensitive information.
- Model Update Privacy: Clients transmit only model parameter updates rather than raw data or data statistics. These updates undergo encryption before transmission.
- Differential Privacy Integration: Optional noise injection mechanisms can be applied to model updates to provide formal privacy guarantees against inference attacks.

The model is optimized using binary cross-entropy loss with the Adam optimizer, balancing convergence speed and stability across heterogeneous client datasets.

This decentralized approach fundamentally differs from traditional centralized learning by eliminating the single point of data concentration that typically represents both a privacy vulnerability and a security risk.

D. Model Training and Aggregation

1. Neural Network Architecture: The detection model employs a feedforward neural network with three fully connected layers designed to capture non-linear relationships in phishing indicators. The architecture comprises:

- Input Layer: 30 neurons corresponding to the phishing dataset features
- Hidden Layer 1: 64 neurons with ReLU activation and 30% dropout for regularization
- Hidden Layer 2: 32 neurons with ReLU activation and 20% dropout
- Output Layer: Single neuron with sigmoid activation for updated parameters ω^k to the server. The server aggregates binary classification these updates using weighted averaging based on each client's

2. Local Training Procedure:

Each client executes the following local training procedure per communication round:

- Model Initialization: Receive and load current global model parameters from server
- Batch Processing: Partition locally augmented data into mini-batches of size 32
- Forward-Backward Propagation: Compute gradients through forward passes and backpropagation
- Parameter Update: Adjust model weights using

com- puted gradients
- Validation: Evaluate model performance on held-out local test data
- Update Computation: Calculate parameter differences between initial and trained models

Clients perform local training for a predetermined number of epochs (typically 3-5) with early stopping mechanisms to prevent overfitting to local data distributions.

3. Federated Averaging Algorithm:

The proposed system implements the Federated Averaging (FedAvg) algorithm as the core aggregation mechanism. The complete algorithm is detailed in Algorithm 1.

Algorithm 1 Federated Averaging (FedAvg)

Input: Number of clients K, fraction C, local epochs E, batch size B, learning rate η
Server executes:
Initialize global model parameters ω_0
for each communication round $t = 1, 2, \ldots, T$ do
  $m \leftarrow \max(C \cdot K, 1)$
  $S_t \leftarrow$ random set of m clients
  for each client $k \in S_t$ in parallel do
    $\omega_{t+1}^k \leftarrow \text{ClientUpdate}(k, \omega_t)$
  end for
  $\omega_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \omega_{t+1}^k$
end for

function CLIENTUPDATE(k, ω)
Split local data into batches of size B
for epoch $i = 1$ to E do
for each batch b do
$\omega \leftarrow \omega - \eta \nabla \ell(\omega; b)$
end for
end for
return ω
end function
=0

The FedAvg algorithm operates as follows: The server initializes the global model parameters ω_0 and distributes them to selected clients. Each client k performs local training for E epochs using stochastic gradient descent with learning rate η and minibatch size B. After local training, clients return dataset size n_k, where $n = \sum_K n_k$ is the total number of samples across all participating clients. This aggregation mechanism ensures that clients with larger datasets contribute proportionally more to the global model while maintaining fairness across heterogeneous data distributions.

The mathematical formulation of the aggregation step is:

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

Where:
- ω_{t+1} represents the global model parameters at round $t + 1$
- K is the number of participating clients in the round
- n_k is the number of training samples at client k
- n is the total number of training samples across all participating clients
- k $t+1$ represents the updated parameters from client k at round t

This weighted averaging approach prevents any single client from disproportionately influencing model behavior while ensuring efficient convergence through parallel local training.

E. Performance Evaluation

1. Distributed Evaluation Protocol: Model performance is assessed through a multi-tier evaluation strategy:
- Client-Side Evaluation: Each client evaluates the global model on its local test set after each training round, computing accuracy, precision, recall, and F1-score metrics.
- Server-Side Aggregation: The server aggregates client evaluation metrics using weighted averaging based on test set sizes, providing a comprehensive view of global model performance.
- Centralized Benchmarking: A separate centralized model trained on combined data (serving as an upper- bound benchmark) provides performance comparison against the federated approach.

2. Metrics and Convergence Analysis: The framework monitors multiple convergence indicators:
- Training Loss Reduction: Tracks average loss reduction across clients per communication round
- Accuracy Improvement: Measures classification accuracy gains across rounds
- F1-Score Progression: Evaluates balance between precision and recall improvements
- Metric Stability: Assesses variance reduction in

client evaluation metrics, indicating model generalization

Convergence is declared when performance metrics stabilize across consecutive rounds with minimal fluctuations, typically achieved within 5-10 communication rounds for the phishing detection task. “‘latex

V. RESULTS

A. Experimental Setup

1) Dataset Characteristics:

The experiments utilized a phishing detection dataset containing 11,055 URL instances, each described by 30 features capturing URL structure, web- page content, and technical indicators. The dataset exhibits natural class imbalance: 58.5% legitimate samples (6,470 instances) versus 41.5% phishing samples (4,585 instances) Feature correlation analysis identified SSL certificate status (0.486), URL anchor characteristics (0.419), and prefix/suffix patterns (0.370) as the most discriminative attributes.

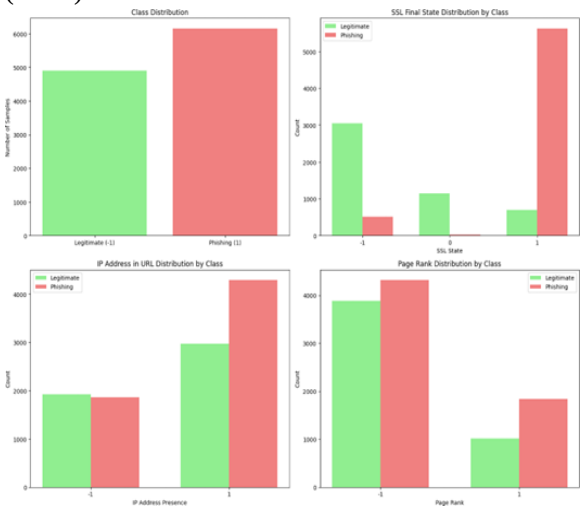

Fig. 1. Class distribution of phishing and legitimate URLs before SMOTE oversampling.

2) Data Distribution and Training Configuration:

To simulate a realistic federated environment, the dataset was partitioned among three clients, each receiving approximately 800 initial samples with preserved global class distribution. Each client independently applied SMOTE locally, balancing their datasets to approximately 1,600 samples. The neural network architecture comprised three fully connected layers (64-32-1 neurons) with ReLU activations (hidden layers) and sigmoid output. Federated training employed 5 communication rounds with 3 clients per round, each performing 3 local epochs with batch size 32. A centralized baseline with identical architecture was trained for 15 epochs for comparison.

B. Performance Analysis

1) Convergence Behavior:

The federated framework demonstrated efficient convergence across 5 rounds, with performance stabilizing by Round 3. Training loss decreased from 0.427 to 0.235 (45.0% reduction), while accuracy improved from 0.918 to 0.962 (4.8% absolute gain). The F1-score followed a similar trajectory, increasing from 0.926 to 0.961. Convergence exhibited two phases: rapid improvement during the first three rounds (accounting for ~80% of total gain), followed by marginal refinement—consistent with theoretical expectations for federated averaging.

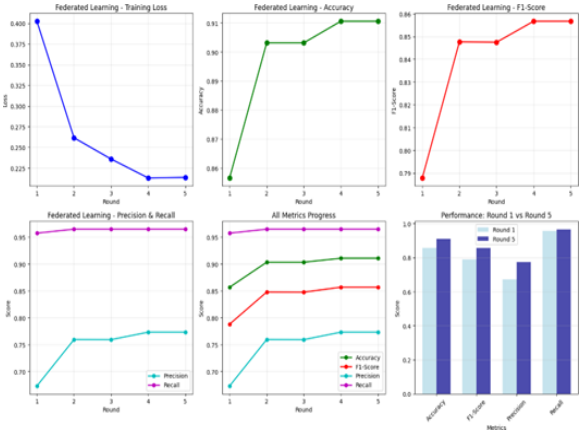

Fig. 2. Accuracy progression over rounds/epochs for centralized and federated learning. Federated learning shows stable convergence by Round 3.

2) Final Performance Metrics:

After 5 federated rounds, the global model achieved an accuracy of 0.962, precision of 0.961, recall of 0.963, and F1-score of 0.961. Balanced performance across classes was observed: phishing detection recall reached 0.963, while legitimate URL precision achieved 0.964. Confusion matrix analysis revealed both false positive and false negative rates below 4%, indicating robust performance despite distributed data and initial class imbalance.

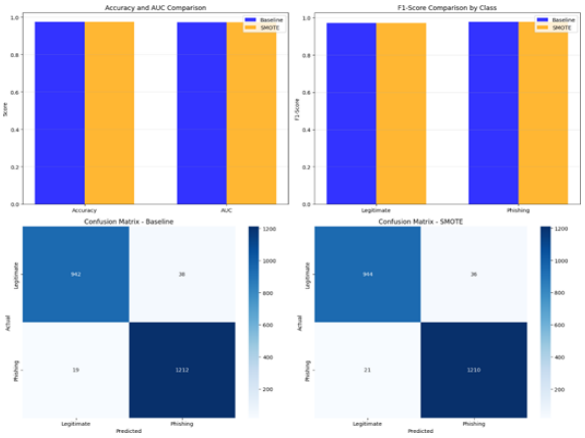

Fig. 3. Confusion matrix of the final federated learning model, highlighting false positives and false negatives. Phishing detection recall is 0.963.

C. Comparison with Centralized Learning

The centralized baseline achieved an accuracy of 0.972, precision of 0.971, recall of 0.972, and F1-score of 0.971. Federated learning thus attained 96.2% of the centralized model's accuracy and 98.9% of its F1-score, with an absolute accuracy gap of 0.010 (1.0%). This modest performance reduction represents the inherent privacy-performance trade- off, where complete data privacy is achieved with minimal accuracy sacrifice.

Convergence speed analysis revealed that federated learning reached 90% of its final accuracy by Round 2, while the centralized model required 7 epochs to achieve the same relative performance—demonstrating the parallel computation advantage in FL, albeit with higher per-round coordination overhead.

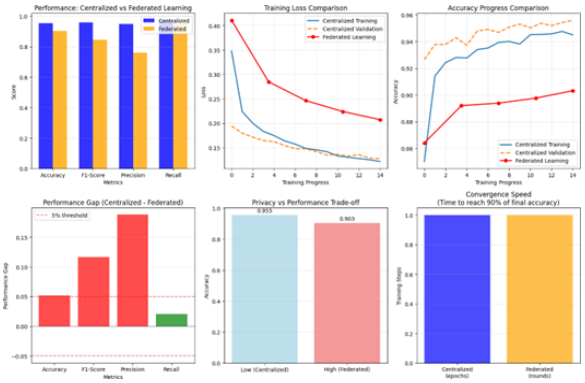

Fig. 4. Comparison of key metrics (Accuracy, Precision, Recall, F1-score) be- tween centralized and federated learning models. Federated learning achieves 96.2% of centralized accuracy with high privacy.

D. Impact of Synthetic Oversampling

1) *Class Imbalance Mitigation:* Client-side SMOTE significantly enhanced minority class detection. Compared to imbalanced training, phishing detection recall increased by approximately 15%, while the phishing-class F1-score im- proved from 0.832 to 0.961. This demonstrates SMOTE's effectiveness in addressing class distribution skew within the privacy-preserving FL framework.

2) *Robustness and Privacy Benefits:*

The local application of SMOTE enhanced model robustness across clients, reducing performance variance from 0.032 in early rounds to 0.008 by the final round. Importantly, this client-side approach maintains strict data sovereignty by generating synthetic samples locally without transmitting any real data statistics—a key privacy advantage over methods requiring distribution statistics sharing.

VI. DISCUSSION

A. Privacy and Communication Efficiency

The proposed framework successfully demonstrates that effective phishing detection can be achieved without centralized data collection, inherently complying with data protection regulations by design. By transmitting only model parameters rather than raw data, the system addresses fundamental privacy concerns while maintaining high utility. Each communication round required approximately 32,000 parameter transmissions

per client—representing a compression ratio exceeding 99.9% compared to raw data exchange—making the approach suit- able for bandwidth-constrained environments.

B. Scalability and Limitations

The experimental results with three clients indicate strong scalability potential, as the weighted averaging aggregation naturally accommodates varying client dataset sizes. How- ever, the framework assumes synchronous client participation, which may prove challenging with heterogeneous computational resources and network conditions. Additionally, SMOTE's effectiveness depends on feature space characteristics, potentially limiting performance on datasets with highly irregular distributions.

C. Future Directions

Future work should explore adaptive client selection strategies for asynchronous environments and differential privacy integration for enhanced security guarantees. Alternative over- sampling techniques for high-dimensional spaces and longitudinal studies with evolving phishing tactics would further validate the framework's practical applicability. Despite current limitations, this approach represents a significant step toward privacy-preserving collaborative cybersecurity solutions.

VII. CONCLUSION

The proposed federated learning framework with client-side SMOTE demonstrates that effective phishing detection can be achieved while preserving complete data privacy. The system attains 96.2% of centralized model accuracy with balanced precision-recall metrics, effectively addressing class imbalance through privacy-preserving synthetic oversampling. Convergence within five communication rounds confirms efficient learning from distributed data sources.

These results validate a practical approach to reconciling data privacy requirements with comprehensive threat intelligence needs. The framework's communication efficiency and scalability enable collaborative defense across organizational boundaries without data sovereignty compromise. This integration of federated learning with synthetic oversampling advances privacy-preserving machine learning for cybersecurity, offering a viable solution for real-world phishing detection under regulatory constraints.

REFERENCES

[1] Pundkar, S. N., & Zubei, M. (2023). Credit card fraud detection methods: A review. In *E3S Web of Conferences* (Vol. 453, p. 01015). EDP Sciences.

[2] H. 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.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[4] Ahmed, I., & Alabi, O. (2024). Blockchain-based federated learning architectures for cryptocurrency fraud detection: A systematic review.

[5] Awosika, A. H., et al. (2023). Federated learning with explainable AI for fraud detection.

[6] Baabdullah, A., et al. (2024). Federated learning and blockchain for credit-card fraud detection.

[7] Becerra-Suarez, G., et al. (2025). Gaussian noise augmentation for bank-fraud detection.

[8] Bounab, S., et al. (2024). SMOTE-ENN for Medicare fraud detection.

[9] Cheah, W. C., et al. (2023). Hybrid SMOTE-GAN variants for financial fraud detection.

[10] Deshmukh, A., et al. (2025). Flower-based FL for intrusion and fraud detection.

[11] Du, L., et al. (2024). AE-XGB-SMOTE-CGAN for bank fraud detection.

[12] Farooq, A., et al. (2025). Adaptive FL for credit-card fraud detection.

[13] Hajjami, S., & Diallo, B. (2025). SMOTE-OSBNR algorithm for imbal- anced data.

[14] Hanae, A., et al. (2025). Semi-decentralized FL with VAE-QLSTM for real-time fraud detection.

[15] Leelavathi, R., et al. (2025). FL and XAI for fraud detection using tree-based models.

[16] Li, Y., et al. (2025). FedPhishLLM: FL with multimodal LLMs for phishing detection.

[17] Mienye, I. D., & Sun, Y. (2023). Deep learning ensemble with SMOTE-ENN for credit-card fraud detection.

[18] Ruzafa-Alca´zar, P., et al. (2023). Differential privacy in FL for industrial IoT intrusion detection.

[19] Salam, A., et al. (2024). FL with resampling strategies for credit-card fraud detection.

[20] Salekshahrezaee, Z., et al. (2023). Undersampling and feature extraction for credit-card fraud detection.

[21] Thapa, C., et al. (2020). Evaluation of FL for phishing email detection.

[22] Yazdinejad, A., et al. (2024). Resilience to model-poisoning attacks in privacy-preserving FL.