

Privacy-Preserving Federated Fraud Detection System for Collaborative Banking

Akshaya A ¹, Akshayaa V R ², Priyadarshini S S ³, Sonika P ⁴, Mrs. C T Vidhya ⁵

^{1,2,3,4} Department of Computer Science and Engineering, ⁵ AP Department of Computer Science and Engineering, Avinashilingam Institute for Home Science and Higher Education for Women, School of Engineering

Abstract - Banks lose large amounts of money to fraud every year, yet the same privacy rules that protect customers also stop banks from comparing notes on how fraud happens. When one bank discovers a new fraud pattern, it usually has no way to warn other banks, because sharing raw transaction records is restricted by law and by competitive concerns. This paper presents a federated fraud detection system that lets several banks train one shared model together without any bank ever seeing another bank's data. Three simulated banks, modelled on Bank A, Bank B, and Bank C, each train a local neural network on their own transaction data and send only the trained weights to a central server. The server averages these weights using Federated Averaging to build a single global fraud model, which is then sent back to every bank. A simple web interface built with Streamlit lets a user enter a transaction and see, side by side, what the local bank model and the global model each predict. The results show that the global model can flag fraud patterns that a single bank's local model alone would have missed, while every bank keeps full control of its own data throughout. The system was implemented using PyTorch for model training and Flower for the federated learning pipeline.

Key words - Federated Learning, Fraud Detection, Data Privacy, Federated Averaging, Neural Network, PyTorch, Flower, Banking Security

I. INTRODUCTION

Digital banking makes moving money easy for both customers and fraudsters. When a scammer gets blocked by one bank, they simply try another. Because banks only see their own data, industry-wide fraud patterns often look like isolated incidents to a single institution.

The ideal solution is to pool all bank data to train a single fraud-detection model. However, privacy laws, corporate rivalry, and data breach risks make this impossible. Banks are forced to choose between protecting customer privacy or catching more fraud.

Federated learning solves this dilemma. Instead of moving private data to a central model, it brings the model to the data. Each bank trains a local AI model on its own transactions, then shares only the learned technical parameters never the raw data with a central server. The server combines these updates into a smarter global model and sends it back to everyone. No private data is ever exposed, but every bank benefits from the collective knowledge.

This paper tests that approach. We simulated three banks training individual neural networks on private data, using the Federated Averaging (FedAvg) algorithm to build a shared global model. We also created a web interface to compare local and global models side-by-side, proving the real-world benefit of this setup.

The rest of this paper is organized as follows: Section II reviews related work; Section III defines the problem and objectives; Section IV details the system architecture; Section V analyzes the results; Section VI presents our takeaways; Section VII discusses future scope; and Section VIII concludes the paper.

II. LITERATURE SURVEY

Federated learning has become an important research area for fraud detection because it allows banks and financial institutions to work together without sharing their customers' private data. Instead of sending transaction data to one central location, each bank trains a model on its own data and shares only the model updates. This helps improve fraud detection while protecting privacy. The following studies helped shape the design of the proposed system.

A. Transformer-Based Sequential Modelling with Federation

Tang and Liu [1] proposed a fraud detection system that combines transformer-based sequential learning with federated learning. In their approach, each bank learns the pattern and order of its own transactions before sharing the updated model with a central server. Their study also points out that keeping all participating banks synchronized with the server increases communication overhead. This is one of

the reasons why the proposed system uses a simple aggregation method.

B. Parameter Sharing for Banking Fraud Detection

Gawade, Hande, and Kene [2] developed a federated learning framework where every bank trains its model locally and sends only the model parameters to a central server. The server combines these updates to create a better global model. Their results show that this approach performs better than training each bank's model separately. However, they also mention that the system depends on a secure and reliable network connection between the banks and the server.

C. Blockchain-Based Federated Learning for Cryptocurrency Fraud

Ahmed and Alabi [3] reviewed fraud detection systems that combine blockchain technology with federated learning. Blockchain improves transparency and makes model updates easier to verify, but it also increases computation and communication costs. Their review suggests that such an approach is more suitable for large-scale systems. Therefore, the proposed work uses a simpler FedAvg server architecture instead of blockchain.

D. Federated Learning for Credit Card Fraud Detection

El Hallal and El Mourabit [4] presented a survey on the use of federated learning for credit card fraud detection. They explain that federated learning allows organizations to train models together without sharing sensitive customer information. Their survey also discusses challenges such as handling different data distributions across institutions, which is an area that can be explored in future improvements of this project.

E. Explainable Federated Learning

Aljunaid et al. [5] combined federated learning with explainable AI methods such as SHAP and LIME. Their system not only detects fraud but also explains why a transaction is classified as fraudulent. This makes the system easier for banking officials to understand and trust. Although explainability is useful, it is not included in the current project and is considered future work.

F. Federated Meta-Learning for Imbalanced Data

Zheng et al. [6] proposed a federated meta-learning approach to improve fraud detection when fraudulent transactions are much fewer than normal transactions. Their method combines federated learning with metric learning to better identify rare fraud cases. Since fraud datasets are naturally imbalanced, their work is relevant to the proposed system.

G. Survey of Federated Learning in Financial Security

Kennedy, Hilal, and Momeni [7] reviewed different applications of federated learning in the financial sector. Their survey covers techniques such as FedAvg, secure

aggregation, and differential privacy. They conclude that federated learning is becoming an effective privacy-preserving solution for financial institutions. However, they also highlight practical challenges such as communication cost and synchronization between participants.

H. Centralized Machine Learning for Payment Fraud

Abid [8] compared several centralized machine learning models, including Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbour (KNN), and Convolutional Neural Network (CNN), for payment fraud detection. Among these models, CNN achieved the best performance after proper preprocessing. However, because all the data is stored in one central location, this approach does not address privacy concerns.

I. Federated Learning for Payment Security

Gudimetla [9] applied federated learning to payment fraud detection while ensuring that customer data remains private. The study reports good detection accuracy and better privacy protection. However, it also explains that federated learning requires more communication and computational resources compared to centralized systems. This trade-off was also considered while designing the proposed system.

J. AI-Based Fraud Detection in Financial Institutions

Islam and Rahman [10] used artificial intelligence and anomaly detection techniques to identify fraudulent financial transactions. Their model achieved high accuracy and fast detection. However, since their approach depends on centralized data, it does not address the privacy challenges faced by organizations that cannot share customer information. Federated learning helps solve this limitation.

III. PROBLEM STATEMENT AND OBJECTIVES

A. Problem Statement

Banks face a major challenge in detecting fraud because customer transaction data cannot be shared due to privacy and security regulations. As a result, each bank trains its fraud detection model using only its own data, limiting its ability to identify new fraud patterns that appear at other institutions. This delay allows fraudsters to exploit similar attacks across multiple banks. Therefore, there is a need for a privacy-preserving approach that enables banks to learn from each other without exposing sensitive customer information.

B. Objectives

The main objectives of the proposed system are:

- To develop a federated learning framework that allows multiple banks to collaboratively train a shared fraud detection model while keeping all transaction data within each bank.
- To ensure that only model parameters are exchanged between banks, protecting customer privacy throughout the training process.
- To evaluate whether the aggregated global model can detect fraud patterns more effectively than individual local models.
- To maintain good performance of each bank's local model while benefiting from collaborative learning.
- To build a real-time fraud detection interface that compares predictions from both local and global models, demonstrating the practical benefits of federated learning.

IV. PROPOSED SYSTEM ARCHITECTURE

The system is organised around four cooperating parts: dataset partitioning, local training at each bank, central aggregation, and inference through a web interface. At no point in this pipeline does a transaction record cross from one bank to another; only model weights travel between a bank and the central server. Three simulated banks, referred to as Bank A, Bank B, and Bank C, each hold an exclusive share of a transaction dataset. Because the global model absorbs patterns from all three shares, a bank whose own data never contained a particular fraud pattern can still catch it once the updated global model reaches that bank. Every incoming transaction is checked twice, once by the receiving bank's own model and once by the shared global model, and both results are shown together.

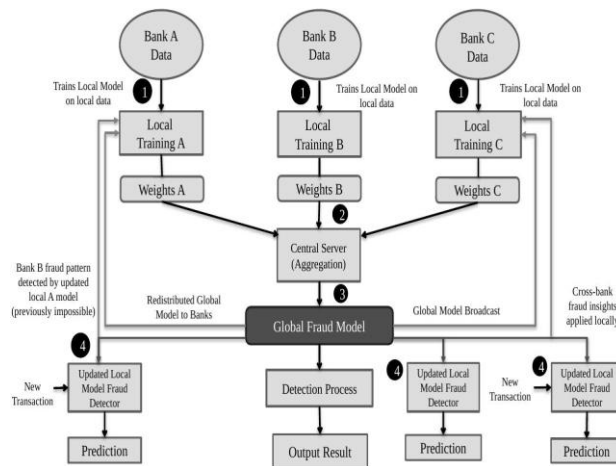


Fig. 1. Federated Fraud Detection System Architecture

A. Data Preparation and Partitioning

The dataset used for this project was collected from Kaggle and split into three separate, non-overlapping subsets, one for each simulated bank, so that no bank's data overlaps with another's. Each subset is cleaned and normalised on its own. Principal Component Analysis (PCA) is then applied to reduce the number of features to the ones that matter most for telling fraud apart from legitimate activity, which speeds up local training and helps prevent the model from overfitting to a comparatively small partition of data.

B. Local Model Training

Each bank trains its own feed-forward neural network classifier using standard forward propagation and backpropagation. The Adam optimiser adjusts the learning rate for each parameter as training goes on, and ReLU activation functions let the network represent the more complex decision boundaries needed to separate fraud from normal transactions. A Binary Cross-Entropy loss function measures how far the model's predictions are from the true labels at each step. Training runs locally for a fixed number of epochs, and only after that does the resulting set of weights get sent off for aggregation, never the underlying data.

C. Federated Averaging at the Central Server

After each round of local training, the central server collects the weight updates sent by all three banks and combines them using the Federated Averaging (FedAvg) algorithm, which works out a weighted average of the incoming parameters without ever looking at the transactions that produced them. The resulting global model is sent back to each bank, completing one federated round. Over several rounds, the global model keeps building on what it learned before while the local models keep refining themselves on fresh data.

D. Prediction and Web Interface

A Streamlit-based web application makes the trained models available to an end user. A transaction entered into the interface is sent to the chosen bank's local model and to the global model at the same time. Each model applies a threshold to its output to decide whether the transaction looks fraudulent or legitimate, and both decisions are shown side by side, so any disagreement between the local and global judgement is immediately visible to the user.

V. RESULTS AND ANALYSIS

We tested the system by training the three simulated bank models on their own separate data. For all three banks, error rates dropped steadily and the training stabilized. We didn't need to tweak the settings for individual banks, proving the system works well even when data sizes and distributions vary.

After each training round, the central server combined the updates into a master global model using the FedAvg algorithm. As a result, if a fraud pattern only appeared in Bank B's data, Bank A's updated model could now catch it too, something Bank A would have completely missed before the update.

Using the Streamlit web interface, we ran sample transactions through both the local and global models at the same time. Most of the time, the two models agreed because the global model reinforces what the local banks already know. However, when they disagreed, the global model was much better at flagging fraud than the local bank missed on its own.

Overall, the results prove that a federated setup lets banks team up against fraud without sharing raw customer data, noticeably improving fraud detection compared to banks working in isolation. The step-by-step results are shown in Figures 2 through 7 below.

A. Data Preparation Output

The credit card transaction dataset was first loaded and divided into three nearly equal partitions to simulate data owned by three different banks. Bank A received 94,935 transactions with 149 fraud cases, Bank B received 94,936 transactions with 173 fraud cases, and Bank C received 94,936 transactions with 170 fraud cases. In total, the original dataset contained 284,807 transactions, including 492 fraudulent transactions. This partitioning allows each bank to train its model independently while maintaining data privacy, closely reflecting a real-world federated learning environment. The console output generated during the data-splitting process is shown in Fig. 2.

```
Loading dataset...
Total rows loaded: 284807
Total fraud cases: 492
Total legit cases: 284315
Total fraud cases: 492
Total legit cases: 284315

— Split Complete —
Bank A (SBI) → 94935 rows | 149 fraud cases
Bank B (HDFC) → 94936 rows | 173 fraud cases
Bank C (ICICI) → 94936 rows | 170 fraud cases

Files saved to data/ folder
PS C:\Users\apson\OneDrive\Desktop\federated-fraud>
```

Fig. 2. Data Preparation and Bank-Wise Split Output

B. Local Model Training Output

Each bank's local model was trained for three epochs, with the training loss dropping in every case; for example, Bank A's loss fell from 0.0743 in the first epoch to 0.0039 by the third. Evaluation accuracy after local training exceeded 99% for all three banks (Bank A: 99.98%, Bank B: 99.96%, Bank C: 99.95%). Fig. 3 shows the training logs for Bank A and Bank B, and Fig. 4 (top) shows the corresponding log for Bank C.

- Bank A training (epochs + loss dropping)

```
INFO : Received: get_parameters message 411f1b45-f3ab-4e94-9ba2-eadb7a70071
INFO : Sent reply
INFO : Received: train message 07526406-b945-4664-b5ef-b08cc42d69d3
Bank A | Epoch 1/3 | Loss: 0.0743
Bank A | Epoch 2/3 | Loss: 0.0054
Bank A | Epoch 3/3 | Loss: 0.0039
INFO : Sent reply
INFO : Received: evaluate message 9f2de233-7d11-4215-a44a-7a78bc17e5cb
Bank A | Accuracy: 99.92%
INFO : Sent reply
INFO : Received: train message 4d12f9ba-c796-4d47-adc2-c9b8d2d4861a
Bank A | Epoch 1/3 | Loss: 0.0035
Bank A | Epoch 2/3 | Loss: 0.0033
Bank A | Epoch 3/3 | Loss: 0.0030
INFO : Sent reply
INFO : Received: evaluate message c23ef06c-77a1-4aae-aba4-00c5bbcaefaa
Bank A | Accuracy: 99.98%
```

- Bank B training (epochs + loss dropping)

```
INFO : Received: train message c875875f-cd8c-4ed0-9120-66f3480a0088
Bank B | Epoch 1/3 | Loss: 0.0756
Bank B | Epoch 2/3 | Loss: 0.0065
Bank B | Epoch 3/3 | Loss: 0.0050
INFO : Sent reply
INFO : Received: evaluate message d750dcc7-c848-4cbc-9633-0fd76e0bd202
Bank B | Accuracy: 99.90%
INFO : Sent reply
INFO : Received: train message a3c91295-9e32-4e58-8647-591fed5980ae
Bank B | Epoch 1/3 | Loss: 0.0053
Bank B | Epoch 2/3 | Loss: 0.0046
Bank B | Epoch 3/3 | Loss: 0.0042
INFO : Sent reply
INFO : Received: evaluate message 07959f69-4349-4605-aaa2-505f5ee3ffbe
Bank B | Accuracy: 99.96%
```

Fig. 3. Local Model Training Output for Bank A and Bank B

C. Central Server Aggregation Output

Once all three banks completed local training, the central server aggregated their weights using FedAvg over five federated rounds. Global model accuracy rose from 99.91% after round 1 to 99.97% by round 4 and stayed above 99.9% through round 5, confirming that the aggregated model converges to a stable, high-performing state. Fig. 4 (bottom) shows the aggregation log for these five rounds.

Bank C training (epochs + loss dropping)

```
INFO : Received: train message 821e4b9c-2750-4944-a7d8-1ef305be5c4e
Bank C | Epoch 1/3 | Loss: 0.0746
Bank C | Epoch 2/3 | Loss: 0.0053
Bank C | Epoch 3/3 | Loss: 0.0046
INFO : Sent reply
INFO : Received: evaluate message 36d6f14e-94a8-4b48-aa64-133065631713
Bank C | Accuracy: 99.91%
INFO : Sent reply
INFO : Received: train message f8e30bfe-e98e-4387-b9db-d3aac9669775
Bank C | Epoch 1/3 | Loss: 0.0040
Bank C | Epoch 2/3 | Loss: 0.0036
Bank C | Epoch 3/3 | Loss: 0.0032
INFO : Sent reply
INFO : Received: evaluate message 569ae799-aa62-446b-970b-250a36f7dbc4
Bank C | Accuracy: 99.95%
```

```
— Training Complete —
Accuracy per round:
Round 1: 99.91%
Round 2: 99.96%
Round 3: 99.96%
Round 4: 99.97%
Round 5: 99.96%
(.venv) PS C:\Users\apson\OneDrive\Desktop\federated-fraud>
```

Fig. 4. Bank C and Central Server Aggregation Output

D. Prediction and Interface Output

We used the Streamlit dashboard to test transactions against all four AI models at the same time. Here is how they performed:

- **Bank A Dashboard Test:** A transaction submitted through Bank A was classified as legitimate by the local models of Bank B and Bank C. However, Bank A's local model and the Global Model correctly identified it as fraudulent, with the Global Model predicting fraud with 97.78% confidence.
- **Bank B Dashboard Test:** During testing with Bank B, all three local models classified a fraudulent transaction as legitimate. In contrast, the Global Model successfully detected the fraud with 84.29% confidence, demonstrating its ability to identify fraud patterns learned from collaborative training.
- **Bank C Dashboard Test:** When a genuine transaction was tested through Bank C, all three local models and the Global Model correctly classified it as legitimate. This indicates that the federated approach improves fraud detection without increasing false alarms for normal transactions.

These side-by-side tests clearly demonstrate the practical value of the federated system. By catching hidden scams without disrupting everyday banking, the platform proves it can protect both customer privacy and financial security.

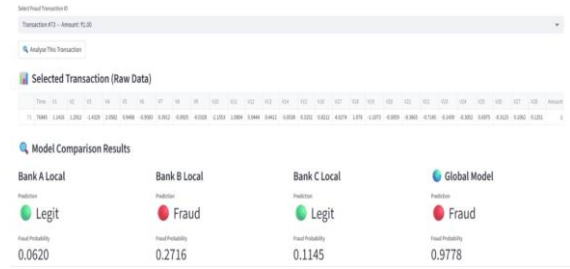


Fig. 5. Prediction Output for a Sample Bank B Transaction

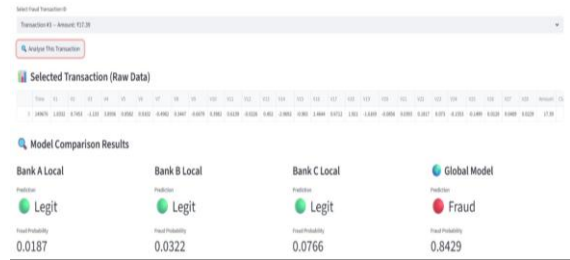


Fig. 6. Prediction Output for a Sample Bank C Transaction

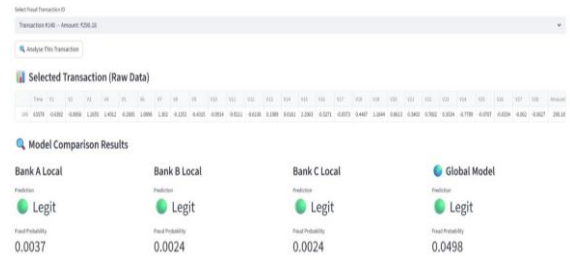


Fig. 7. Prediction Output for a Sample Bank A Transaction

VI. INFERENCES

The implementation and evaluation of the proposed system show that federated learning is an effective approach for privacy-preserving fraud detection. By sharing only model parameters, banks can collaboratively improve fraud detection without exposing customer transaction data. The FedAvg algorithm successfully combines knowledge from multiple local models into a global model that captures broader fraud patterns while preserving the strengths of each bank's local model. Comparing local and global predictions also demonstrates the practical benefits of collaborative learning. Overall, the proposed approach improves fraud detection while reducing the privacy risks associated with centralized data storage.

VII. FUTURE SCOPE

Our current system shows that federated fraud detection works well on a small scale using simulated bank data. To upgrade this into a real-world, production-ready system, a few key additions are needed:

- **Differential Privacy:** We can add extra privacy guards to the weight-sharing step. This ensures that even the shared technical

parameters leak absolutely zero clues about individual customer habits.

- **Real-Time Streaming:** Moving from analyzing transactions in batches to continuously scoring live transaction streams will allow the system to block scammers instantly, rather than catching them after the damage is done.
- **Handling Uneven Data:** Real banks have vastly different customer bases and data sizes. Upgrading to smarter merging algorithms, like FedProx, will help the system handle these uneven, real-world data differences much better than the basic setup we used.
- **Secure Aggregation:** We can introduce high-level encryption protocols so that even the central server cannot see an individual bank's model updates. It would only ever see the final, blended result.
- **Clear Explanations (Explainability):** Finally, we need to add tools that explain *why* the AI flagged a transaction. This will help compliance, audit, and regulatory teams understand the decisions, rather than just showing a simple "fraud" or "legit" label on a screen.

VIII. CONCLUSION

- This paper presented a federated learning system that lets multiple banks work together to strengthen a shared fraud detection model while each bank keeps exclusive control of its own transaction data. By limiting all communication between banks to model parameters, and combining those parameters through Federated Averaging at a central server, the system lets fraud knowledge learned at one bank help detection at another bank, something a conventional, siloed fraud detection setup cannot do.
- Built across three simulated bank nodes using PyTorch, Flower, and a Streamlit interface, the system trains reliably at the local level and produces a global model that genuinely reflects knowledge pooled from all three banks. The results support the central claim of this project: banks do not have to choose between protecting customer privacy and detecting fraud effectively. Extending the system with differential privacy, secure aggregation, and explainability, as outlined in Section VII, would move it closer to something a real bank could deploy.

REFERENCES

- [1] Y. Tang and Z. Liu, "A Credit Card Fraud Detection Algorithm Based on SDT and Federated Learning," IEEE Access, 2024.
- [2] P. Gawade, J. Hande, and P. Kene, "Federated Learning in Banking: A Secure and Intelligent Approach to Financial Fraud Detection," Journal of Emerging Technologies and Innovative Research (JETIR), June 2025.
- [3] A. A. Ahmed and O. O. Alabi, "Secure and Scalable Blockchain-Based Federated Learning for Cryptocurrency Fraud Detection: A Systematic Review," IEEE Access, July 2024.
- [4] T. El Hallal and Y. El Mourabit, "Federated Learning for Credit Card Fraud Detection: Key Fundamentals and Emerging Trends," IEEE, June 2024.
- [5] S. K. Aljunaid, S. J. Almheiri, H. Dawood, and M. A. Khan, "Secure and Transparent Banking: Explainable AI-Driven Federated Learning Model for Financial Fraud Detection," Journal of Risk and Financial Management, March 2025.
- [6] W. Zheng, L. Yan, C. Gou, and F.-Y. Wang, "Federated Meta-Learning for Fraudulent Credit Card Detection," IJCAI, 2019.
- [7] C. H. Kennedy, A. Hilal, and M. Momeni, "The Role of Federated Learning in Improving Financial Security: A Survey," IEEE, October 2025.
- [8] N. Abid, "Improving Accuracy and Efficiency of Online Payment Fraud Detection and Prevention with Machine Learning Models," International Journal of Innovative Science and Research Technology, December 2025.
- [9] A. R. Gudimetla, "Decentralized Fraud Detection: Advancing Payment Security Through Federated Learning," IJITMIS, January 2025.
- [10] M. S. Islam and N. Rahman, "AI-Driven Fraud Detections in Financial Institutions: A Comprehensive Study," Journal of Computer Science and Technology Studies, January 2025.