Grenze International Journal of Engineering and Technology, June Issue

Improving Data Privacy and Performance in Collaborative Financial Fraud Detection with 3-Tiered Federated Learning with Differential Privacy

Richie Suresh Koshy[1], Dr. Chitra R[2], Caleb Stephen[3] and Joel Mathew[4]
[1-4]Karunya Institute of Technology and Sciences, Coimbatore, India
Email: richiesuresh@karunya.edu.in, chitrar@karunya.edu, {calebstephen, joelmathew20}@karunya.edu.in

***Abstract*— Amidst the rise of the new digital economy, combating financial fraud has become a critical challenge, requiring numerous collaborative efforts to detect fraudulent transactions in real-time. However, any collaborative solution needs to take privacy considerations into account while sharing sensitive data, which is a crucial hindrance to obtain optimal performance. This paper presents an in-depth analysis of the benefits of utilizing a 3-tiered Federated Learning architecture with DPFedAvg aggregation strategy as a collaborative alternative means to tackle financial fraud, allowing institutions to collaborate while maintaining privacy and further improving security. Additionally, we assess the effectiveness of this approach in detecting fraudulent transactions, comparing its performance to traditional methodologies. Our findings reveal a notable performance improvement using the Federated Learning model, surpassing conventional models.**

***Index Terms*— federated learning, detecting financial fraud, data privacy, ADASYN, collaborative learning, DPFedAvg, deep learning**

I. INTRODUCTION

With rapid upgrades to technology, the financial industry is seeing unprecedented growth as more transactions now depend on financial institutions and cash becomes more obsolete by the day. However, a natural cost of this pace is the failure of security as financial fraud is becoming a greater concern. This is all the more the case as financial literacy has failed to keep up with the rapidly advancing technology. The complexity involving the factors that affect fraud does not allow for traditional checks or automated scans to discover these frauds. Furthermore, the mere scale and volume of such transactions discredits the possibility of using trained experts to manually identify these incidents.

In light of this scenario, Machine Learning has been proposed as a valid and scalable approach to combat financial fraud [1] and credit card fraud [2]. Hybrid fraud detection models can be used to tackle various kinds of financial fraud like securities and commodities fraud, financial statement fraud, credit card fraud and so on [3]. Also, as there is a noticeable data drift in the spending patterns of people, Machine Learning can identify these patterns as they occur in real-time. However, conventional traditional methodologies require data, and sharing sensitive data is an infeasible solution for various institutions. Therefore, options have been limited to limiting the scope of the learning to in-house data or achieving limited collaboration by performing various techniques to achieve

Grenze ID: 01.GIJET.10.2.236

anonymization and de-identification of the data. This could be achieved using Dimensionality Reduction techniques like Linear Discriminant Analysis(LDA), t-Distributed Stochastic Neighbor Embedding (t-SNE), Principal Component Analysis(PCA), Differential Privacy, Data Masking, Data Perturbation, Encryption, and so on.

Federated Learning is a novel approach that promises to address this challenge. It allows disparate organizations to pool their expertise in fraud detection without compromising the confidentiality of sensitive information, thus serving as a bridge between collaboration and privacy preservation. For example, it has been sed effectively in various domains such as open banking [4], loan default prediction [5], and credit scoring [6].

This paper will compare the effectiveness of this approach to the conventional approach using a simulated dataset of financial fraud transactions that happened in the duration of 1st January 2019 to 31st December 2020.

II. METHODOLOGY

A. Overview of Federated Learning

Federated learning is a set-up in which multiple clients collaborate to solve machine learning problems, which is under the coordination of a central aggregator. This setting also allows the training data decentralized to ensure the data privacy of each device [7]. It aims to create a global model using distributed datasets. The goal as illustrated in (1), is to minimize the following objective function to minimize the average training loss for all the clients.

$$F_k(\theta) = \frac{1}{N_k}\sum_{i=1}^{N_k} L(M(x_i;\theta), y_i) \quad (1)$$

In (1), $F_k(\theta)$ is the local objective or loss function on client k, $N_k(\theta)$ is the number of samples or data points on client k, $M(x_i;\theta)$ is the model's prediction on the i-th sample, y_i is the true label of the i-th sample, and $L(M((x_i;\theta), y_i)$ is the loss function measuring the difference between the prediction and the true label.

In the domain of banking, we have similar features that contain sensitive information like name, amount transferred, location of merchant, and so on. In such a scenario, Horizontal Federated Learning is preferred as all the clients share similar data formats.

B. Dataset Description and Exploratory Data Analysis

The original data used for this experiment was taken from https://www.kaggle.com/datasets/kartik2112/fraud-detection. Due to the nature of how sensitive financial fraudulent data is, this data is a simulated credit card transaction dataset containing legitimate and fraud transactions from the duration of 1st January 2019 to 31st December 2020. It covers credit cards of 1000 customers doing transactions with a pool of 800 merchants over a period of 2 years.

The simulated dataset was obtained using the Sparkov Data Generation tool. The GitHub link for this tool is https://github.com/namebrandon/Sparkov_Data_Generation.

The original dataset had a total of 1.85 million records, which were split into 1.29 million records for the train dataset and 0.55 million records for the test dataset in an approximate 70:30 ratio. The original 23 columns were increased to 26 columns after the preliminary feature engineering. As we had 10 clients, the training data was evenly distributed amongst the 10 clients for this experiment after train-test split.

To capture the trend within the dataset, many changes were made. Some of these are as follows:

- Transform timestamp to month, week, weekday, and hour to detect temporal patterns. Fig. 1. illustrates a sample of this difference.
- Derive distance from customer to current merchant at the time of purchase.
- Derive distance from the current merchant to the previous merchant at the time of purchase.
- Perform one hot encoding on category attribute as Fraudulent events are more likely in certain categories than others. See Fig. 2.
- Remove redundant columns like State and City which have been encoded within latitude and longitude.

From Fig. 1 and Fig. 2, there is a marked influence of category and time of transaction over the dataset. This is especially the case for the categories “Shopping”, “Grocery” and “Misc”. Although there are slight differences between client datasets, most of the datasets of the clients indicate the same overall trend.

Other features that were shown to be extremely important are weekday, age, distance from customer to merchant, distance from current merchant to prev merchant, and so on, but their influence was lesser compared to the trends shown in Fig. 1 and Fig. 2.

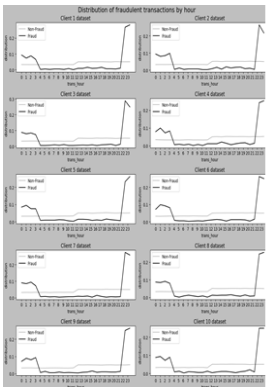

Fig. 1. Distribution of Fraudulent Transactions by Hour

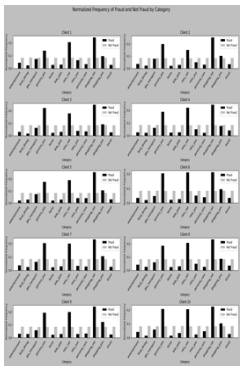

Fig. 2. Distribution of Fraudulent Transactions by Category

C. Three-Tiered Architecture

To improve the security and anonymity of the clients, we are proposing a novel three-tiered FL Architecture as illustrated in Fig. 3 which involves the following:

1. *Server:* The global model aggregator responsible for aggregating weights and implementing Differential Privacy.
2. *Client*: Each institution can be represented as a client in this architecture. They are responsible for training the global model.
3. *Client Group*: A layer introduced between the Server and Client, responsible for selecting random clients during the training process. Ensures the server does not know which client contributed to the result and at the same time, inhibits any reverse engineering attacks to discover the identity of clients involved in training. Furthermore, it introduces the scope to employ Selective Aggregation.

The presence of a server and multiple clients is the norm in conventional Federated Learning methodologies. The addition of a Client Group does not contribute to performance in any manner, but it does allow for anonymity in the training process by preventing any client from knowing which other clients are involved in the training process. At the same time, it prevents the server from also learning about which clients have contributed to the training.

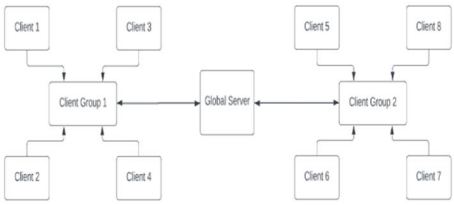

Fig. 3. Sample Architecture with Client Groups

Another benefit of utilizing the approach visualized in Fig. 3. is the possibility of introducing control at the Client Group layer. This can be done with Differential Privacy at this layer and it adds another layer of privacy and security to this entire architecture. At the same time, it reduces the effect of any individual client.

D. Feature Engineering

The data involved in banking applications tend to be complex and multi-faceted, containing diverse and sensitive information such as transaction history and customer profiles. Effective feature engineering is a necessity to optimize predictive accuracy, augment discernment in risk assessment, ameliorate model sophistication, and accommodate the inherent imbalance prevalent in the finance industry. A minority of the techniques employed in this study are listed below:

- *ADASYN*: ADASYN focuses on mitigating class imbalance in banking data by adaptively generating synthetic samples for the minority class, thus promoting a more balanced dataset for robust model training. There are studies showing greater performance on credit card fraud detection with ADASYN [8].

- *SMOTE*: SMOTE is another technique used to handle imbalances in datasets by generating synthetic samples for the minority class. SMOTE is more simplistic and does not adapt to the data distribution, unlike ADASYN. This has been utilized often with great effectiveness in the domain of finance [9].
- *Yeo-Johnson transformation*: This is a versatile technique for addressing both kinds of skewness in banking data. This transformation ensures that model assumptions are met to enhance the effectiveness of subsequent statistical analyses. It has been used successfully in areas such as Intrusion Detection [10].
- *Random Under Sampler*: This technique is utilized to address the imbalance by randomly eliminating instances from the majority class, thereby enhancing the model's ability to discern patterns in the minority class.
- *Feature Engineering on specific features*: This includes transforming certain features and deriving new features while removing redundant attributes and attributes that do not contribute to the model's performance.

E. Evaluation Metrics

In assessing the performance of the proposed methodology given the context of financial fraud, the evaluation metrics are limited to the most important figures to judge the efficacy of the model in identifying fraudulent transactions.

Recall: Recall quantifies the model's ability to capture all relevant instances of the fraudulent class as shown in (2). This is the most important metric as the goal is to reduce the number of False Negatives.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (2)$$

Accuracy: This measures the overall correctness of predictions, offering a general assessment of the model's performance as shown in (3). However, due to the class imbalance of the test sets, this figure is not nearly as important as recall.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Predictions} \quad (3)$$

F. Proposed Solution

The Federated Learning solution prescribed involves an authenticated server with multiple authenticated clients. The clients have access to their own data, which they process via a standard set of guidelines prescribed by the central server.

For the current experiment, a server and 10 clients have been chosen. The clients have been divided these 2 client groups of 5 clients each. This is to achieve ease of implementation of training and can be configured. Each of these clients has an in-house dataset of about 129,000 data points.

The framework used to run this experiment is Flower [11], which is a user-friendly approach to running Federated Learning experiments.

The server starts by initializing the global model and sends the weights to all the clients involved. The server picks random clients from each group to perform training and waits for the results. The selected clients train the global model on their dataset and return the updated weights to the central server. These weights are aggregated by the server, updating the global model. Then, the updated global model is shared with all the clients involved. In case of any individual client failure, the server shall ignore that particular client and proceed with the successful weights. This procedure is repeated until the global model converges or a satisfactory performance has been obtained, with new clients being selected each round. This workflow has been visualized in Fig. 4.

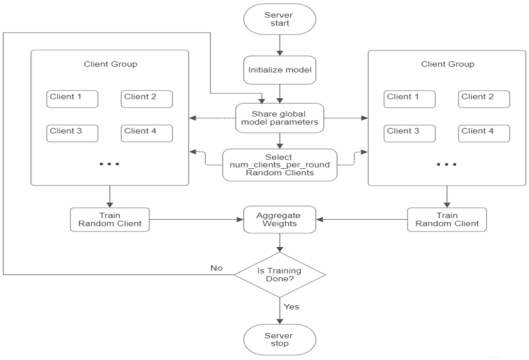

Fig. 4. Federated Learning with Client Groups

G. Model Architecture and Hyperparameters

The collaborative financial fraud detection model was implemented using a Feedforward Neural Network architecture. The input layer consists of 26 nodes, corresponding to the features of our dataset. This was followed by three hidden layers, each containing 128, 64, and 32 neurons respectively.

To introduce non-linearity, ReLU activation functions were employed, and kernel initialization was performed using the He normal initializer. The He normal initializer is a weight initialization technique used to address the issue of vanishing or exploding gradients during the training of deep neural networks.

The He normal initializer specifically initializes the weights of neurons by drawing random values from a Gaussian distribution with a mean of 0 and a standard deviation calculated as the square root of 2 divided by the number of input units in the weight tensor. Mathematically, for a weight tensor W with shape (fan_in, fan_out), the He normal initialization sets the weights as illustrated in (4).

$$W_{ij} \sim N\left(0, \sqrt{\frac{2}{n_{in}}}\right) \quad (4)$$

In (4), N represents a Gaussian distribution and n_{in} is the number of input units in the weight tensor.

To mitigate the chances of overfitting, dropout layers with a dropout rate of 0.5 were inserted between each pair of dense and batch normalization layers. Finally, the output was a single neuron with a sigmoid activation function, which identifies whether the corresponding input data point belonged to the "Fraudulent" class or not. This aforementioned architecture is illustrated in Fig. 5 and Fig. 6.

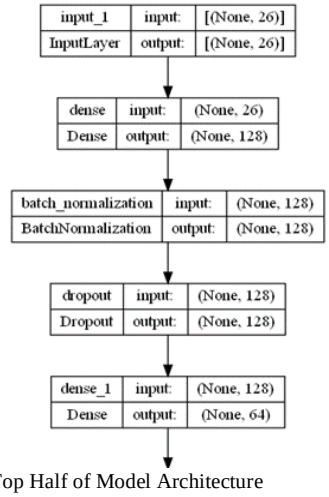

Fig. 5. Top Half of Model Architecture

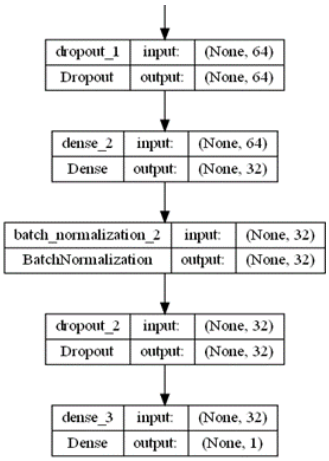

Fig. 6. Bottom Half of Model Architecture

The model used an Adam optimizer with a learning rate of 0.001 to ensure a smooth balance between speed and stability. The training process was carried out for 50 epochs to ensure convergence and the model was saved at each epoch in the scenario of potential overfitting.

III. Experiments And Results

A. Experiment Set-Up

To analyze the performance of the Federated Learning global model in context, there is a need to compare it with benchmarks. For this experiment, individual models have been trained on each of the individual clients, comparing the best models with the global model. The individual models reflect the traditional methodology used in training models in-house.

To remove other confounding factors arising from varying the model architecture and hyperparameters, we've used only the model mentioned in Fig. 5 and Fig. 6. This decision has been made to illustrate the performance of the Federated Learning approach versus in-house training using the same model.

This experiment uses 10 clients, each having their own dataset of approximately 129,000 data points each. The client groups connect to the global central Federated Learning server. As benchmarks, we train the model on each of the client datasets for 10 epochs. The resulting models are evaluated on the test dataset.

To evaluate the performance of the FL model, the clients are split equally between 2 client groups. After running the FL algorithm for 10 epochs, we shall use the final global model's performance to evaluate the performance of this algorithm with the in-house client models. The expectation of this analysis is that as the Federated Learning

model has seen a far greater number of samples from varying clients, its performance will far surpass each of the individual clients.

B. Experiment Results

The Federated Learning model was trained across 50 epochs, with 2 random clients involved in each epoch. Fig. 7 illustrates the change in accuracy and recall of this experiment on the test set.

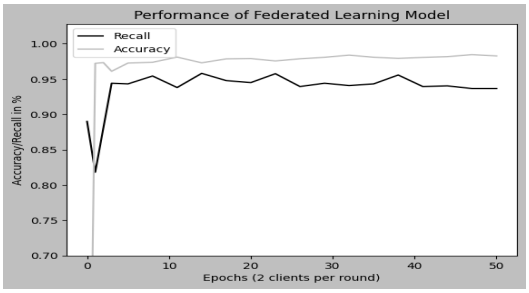

Fig. 7. Performance of Global Model across epochs

The FL implementation converged to a Recall of about 0.95 after about 5 five rounds. The reason behind this is that the model had seen the entire dataset by the 5th epoch and learned the patterns from the entire dataset.
The model converged to a Recall of about 0.95 after about 5 five rounds. The reason behind this is that the model had seen the entire dataset by the 5th epoch and learned the patterns from the entire dataset.
For the baseline models, each of the 10 clients had their own custom models, trained with about 129,000 data points each. For the simulation, the original dataset had about 1.29 million data points which were split across 10 clients equally. The comparison of the recall of these models versus the Federated Learning model has been illustrated in Fig. 8.
As seen from the figure, the highest recall obtained from the Federated Learning model is about 0.95. The models for each of the 10 clients have the highest recall of about 0.8. This result has been obtained on the same model using both approaches.
The performance visualized lies within expectations as this is a natural consequence of the federated model having gained insights from more data. Moreover, this differential will only expand further as more data becomes available, which is a realistic assumption as more clients participate in this system.
The biggest takeaway is that the 3-tiered architecture did not hamper the accuracy of the model while enabling performance greater than each client's performance on their in-house dataset.

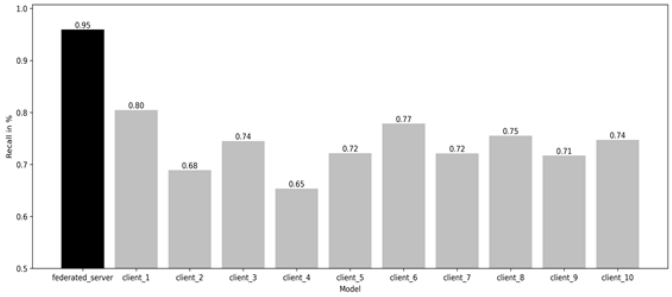

Fig. 8. 3-tiered FL vs Conventional In-house training

IV. Conclusions

In conclusion, the Collaborative Financial Fraud Detection project lays a strong foundation for the integration of federated learning in the finance sector. From the results, we can see that the proposed methodology of using Client Groups to form a 3-tiered architecture can be leveraged for privacy enhancement and noise injection, contributing to a more secure Federated Learning process. The performance of the resulting model surpasses the conventional methodology of training solely with in-house datasets without requiring the sharing of any private data.
Furthermore, the integration of additions like ADASYN and Yeo-Johnson transformer has bolstered the performance of these fraud detection models. This performance differential visualized in Fig. 8 will only increase as the number of clients increases. Moreover, the anonymity introduced by the Client Group can prevent the server

and other clients from knowing which clients contributed to the training at any particular round without compromising on security.

By embracing a collaborative paradigm, this project not only demonstrates the efficacy of federated learning in enhancing accuracy while preserving data privacy but also underscores the need for continual refinement of methodologies to tackle this problem. Many adjustments can be made to the basic Federated Learning approach, which can improve security and anonymity without compromising on performance. As the financial landscape evolves, future endeavors should prioritize tackling financial fraud using a collaborative approach that considers privacy while improving security and performance.

REFERENCES

[1] M. Lokanan and K. Sharma, “Fraud prediction using machine learning: The case of investment advisors in Canada,” Machine Learning With Applications, vol. 8, p. 100269, Jun. 2022, doi: 10.1016/j.mlwa.2022.100269.

[2] Dubey S. C., Mundhe K. S., and Kadam A. A., “Credit card fraud detection using artificial neural network and backpropagation,” in Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 268–273, Madurai, India, 2020. 10.1109/iciccs48265.2020.9120957.

[3] Sadgali, N. Sael, and F. Benabbou, “Performance of machine learning techniques in the detection of financial frauds,” Procedia Computer Science, vol. 148, pp. 45–54, Jan. 2019, doi: 10.1016/j.procs.2019.01.007.

[4] G. Long, Y. Tan, J. Jiang, and C. Zhang, “Federated Learning for open Banking,” in Lecture Notes in Computer Science, 2020, pp. 240–254. doi: 10.1007/978-3-030-63076-8_17.

[5] G. Shingi, "A federated learning based approach for loan defaults prediction," 2020 International Conference on Data Mining Workshops (ICDMW), Sorrento, Italy, 2020, pp. 362-368, doi: 10.1109/ICDMW51313.2020.00057.

[6] F. Zheng, Erihe, K. Li, J. Tian, and X. Xiang, “A Vertical Federated Learning Method for Interpretable Scorecard and Its Application in Credit Scoring,” arXiv (Cornell University), Sep. 2020, [Online]. Available: https://arxiv.org/pdf/2009.06218.pdf.

[7] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, “A survey on federated learning,” Knowledge-Based Systems, vol. 216, p. 106775, Mar. 2021, doi: 10.1016/j.knosys.2021.106775.

[8] S. Bagga, A. K. Goyal, N. Gupta, and A. K. Goyal, “Credit Card Fraud Detection using Pipeling and Ensemble Learning,” Procedia Computer Science, vol. 173, pp. 104–112, Jan. 2020, doi: 10.1016/j.procs.2020.06.014.

[9] B. Baesens, S. Höppner, and T. Verdonck, “Data engineering for fraud detection,” Decision Support Systems, vol. 150, p. 113492, Nov. 2021, doi: 10.1016/j.dss.2021.113492.

[10] J. Zhao, X. Rao, J. Liu, Y. Guo, and B. Yang, “CVAR-FL IOV Intrusion Detection Framework,” in Lecture Notes in Computer Science, 2023, pp. 123–137. doi: 10.1007/978-981-99-7032-2_8.

[11] D. J. Beutel et al., “FLOWER: A FRIENDLY FEDERATED LEARNING FRAMEWORK,” HAL (Le Centre Pour La Communication Scientifique Directe), Mar. 2022, [Online]. Available: https://hal.archives-ouvertes.fr/hal-03601230.