A PROJECT REPORT
on

"FINANCIAL CRIME DETECTION USING FEDERATED LEARNING"

Submitted to
KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE & ENGINEERING

BY

SREEJA SANYAL	**22051200**
AYANTIKA BARDHAN	**23057012**
SAKET KUMAR	**22054212**
ADITYA MAJUMDER	**22054214**
ANANYA BISWAL	**2205874**
ANUSA DE	**22052969**

UNDER THE GUIDANCE
OF
Prof. JHALAK HOTA

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024

November 2025

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA
751024

CERTIFICATE

This is certify that the project entitled

"FINANCIAL CRIME DETECTION USING FEDERATED LEARNING"

submitted by

SREEJA SANYAL	**22051200**
AYANTIKA BARDHAN	**23057012**
SAKET KUMAR	**22052412**
ADITYA MAJUMDER	**22054214**
ANANYA BISWAL	**2205874**
ANUSA DE	**22052969**

is a record of bonafide work carried out by then, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science and Engineering) at KIIT Deemed to be University, Bhubaneshwar.
This work is done during the year 2025-2026, under our guidance.

Date: 20/11/2025

Prof. JHALAK HOTA
(Project Guide)

Acknowledgement

We are profoundly grateful to Prof. Jhalak Hota for his expert guidance and constant support throughout, without which the completion of this project would not have been possible. His continuous encouragement has helped us in completing the project in the provided time.

SREEJA SANYAL
AYANTIKA BARDHAN
SAKET KUMAR
ADITYA MAJUMDER
ANANYA BISWAL
ANUSA DE

ABSTRACT

Financial crime continues to evolve alongside global digital payments, creating a critical need for fraud detection methods that can learn from diverse transaction streams without compromising customer privacy. Federated Learning (FL) offers a promising direction by enabling multiple financial institutions to collaboratively train a shared fraud detection model while keeping all sensitive data decentralized. This report examines the applicability of FL for detecting suspicious transactions, with particular attention to two defining challenges in financial crime analytics: highly unbalanced fraud data and heterogeneous (non-IID) data distributions across different sources, currencies, and customer segments.

The study highlights how these characteristics affect model performance and discusses techniques reported in recent literature to improve learning under such constraints. The findings suggest that FL can enhance fraud-detection intelligence across organizations while adhering to regulatory and confidentiality requirements. At the same time, the work identifies the need for improved validation practices and algorithmic strategies that better accommodate rare fraud events and diverse transactional behaviors. Overall, the project reinforces FL as a viable foundation for privacy-preserving financial crime detection and provides direction for future research in secure, collaborative fraud analytics.

Keywords: Federated Learning (FL), Flower Framework, PyTorch, Federated Averaging (FedAvg), Model Aggregation, Centralized Learning, Privacy-Preserving AI, Machine Learning, Fraud Detection, Data Preprocessing,Python, MLP.

Contents

Chapter 1: Introduction

Financial crime, specifically credit card fraud, identity theft, and unauthorized digital transactions continues to grow in complexity as global payment ecosystems expand across countries, currencies, and digital channels. Banks, payment networks, and fintech institutions are under increasing pressure to detect suspicious activities quickly while complying with strict privacy and data-protection regulations. Conventional fraud detection pipelines rely on centralized data collection, where transaction records from multiple sources are aggregated into a single system for model training. However, modern privacy policies, including GDPR and financial data governance standards, frequently restrict such direct sharing of customer information. This creates a critical need for privacy-preserving fraud detection models capable of learning from distributed financial data without exposing sensitive client information.

Federated Learning (FL) has emerged as a promising solution to this challenge. Instead of pooling raw data, FL allows participating financial organizations to train a shared fraud detection model collaboratively, while data remains securely stored at each institution. Only model weights or gradient updates are exchanged, significantly reducing the privacy risk and legal barriers associated with cross-border financial information transfer. As recent research on federated financial analytics has underlined, this paradigm enables greater fraud-pattern intelligence across global transactions with full confidentiality and regulatory compliance.

1.1 Unbalanced Data in Financial Fraud Detection

A defining feature of financial crime datasets is their **highly unbalanced distribution**. Genuine transactions often represent more than 99% of the data, while fraudulent transactions form only a small minority. Such disproportion causes traditional machine learning models to become biased toward predicting legitimate outcomes, resulting in high overall accuracy but poor detection of fraudulent cases, especially subtle, low-frequency attacks. Techniques like oversampling, undersampling, and cost-sensitive learning have been widely adopted in centralized systems, but applying them in federated environments requires additional care, as synthetic oversampled data and class weights must remain private and locally managed. Therefore, handling imbalance in FL is a key methodological concern within this report.

1.2 Federated Learning in Financial Crime Analytics

Federated Learning enables a shared model to be trained across distinct financial entities banks, credit unions, merchants without necessarily sharing raw transaction records. Each participant computes intermediate model updates using its own data and contributes only the learned parameters to a global aggregator. This approach supports secure, scalable fraud detection, allowing for pattern recognition across currency systems, geographic regions, and customer segments.

Because fraud tactics evolve rapidly, FL also improves adaptability by continuously learning from fresh distributed data. As digital payment channels grow (online, ATM, mobile, PoS), federated learning helps institutions detect cross-channel fraud trends that would otherwise remain isolated within individual systems.

1.3 Dealing with Data Heterogeneity in Federated Systems

Another challenge in financial crime detection is **data heterogeneity**. Transaction records differ widely across institutions and regions due to variations in:

- Currency and exchange value representation
- Risk policies and transaction authorization patterns
- Customer spending behavior and account types
- Fraud prevalence and attack strategies
- Volume and frequency of transactions

This results in **Non-IID (non-independent and non-identically distributed)** data, which can slow model convergence and lowers predictive performance in standard federated algorithms. Techniques such as personalized federated learning, robust aggregation, and client clustering support better learning across diverse datasets and form an essential component of this study.

Chapter 2:
Literature Survey

2.1 Overview of Federated Learning Research

Federated Learning (FL) has emerged as a promising paradigm for privacy-preserving machine learning, enabling multiple organizations or devices to collaboratively train shared models without directly exchanging raw data. This approach has proven particularly useful in regulated domains such as healthcare, banking, and finance, where data confidentiality and compliance requirements restrict centralized data pooling. However, the effectiveness of FL continues to be challenged by two persistent issues: **unbalanced datasets** and **heterogeneous data distributions** across participating clients. Recent scholarly work highlights that these challenges significantly degrade model convergence and prediction reliability in federated financial fraud detection systems.

2.2 Previous Work and Research Gap

Multiple studies have investigated fraud detection using centralized machine learning, achieving strong performance with techniques ranging from random forests to deep neural networks. Additionally, several researchers have implemented federated learning in financial domains to enhance privacy and regulatory compliance. However, most existing research assumes relatively uniform data distributions and often focuses on a single currency or region, which is not always reflective of real-world financial settings. Prior studies also acknowledge unbalanced fraud datasets but typically address them through traditional resampling in centralized settings rather than in federated settings.

In contrast, the present work extends this line of research by exploring fraud detection for **multi-currency, cross-border, and multi-client financial transactions**, where fraud occurs at extremely low frequency and data distributions differ across institutions. This approach fills a gap by demonstrating how federated learning can maintain privacy while adapting to **inter-client imbalance and heterogeneity**, thereby offering a more realistic deployment context for global financial crime detection.

2.3 Handling Unbalanced Datasets in Federated Learning

Financial fraud datasets are characteristically unbalanced, with the proportion of fraudulent transactions frequently below 1% (Carcillo et al., 2021). Traditional centralized approaches rely on oversampling (SMOTE), undersampling, or cost-sensitive learning, but these techniques are not directly transferable to FL due to privacy, communication, and aggregation constraints.

Recent federated strategies propose algorithm-level solutions such as:

Method (Google Scholar Supported)	Contribution to Handling Imbalance
Class-weighted loss / Cost-sensitive FL (Li et al., 2020)	Penalizes misclassification of minority class more strongly without altering client data.
Federated SMOTE / partition-wise oversampling (Díaz et al., 2022)	Oversamples minority transactions **locally** to avoid sharing of synthetic fraud samples.
Focal Loss in FL (Lin et al., 2017; applied in FL by Karimireddy et al., 2020)	Refocuses model learning on hard-to-classify fraud cases.
Robust Aggregation (FedProx, FedNova) (Li et al., 2020; Wang et al., 2021)	Reduces the negative effect of uneven class proportions across clients.

These methods help preserve fraud patterns without increasing privacy risk. They enable local model updates to reflect the importance of minority fraud cases, hence improving recall without excessively harming precision which is an important metric in banking systems where false alarms increase operational costs.

2.4 Dealing with Data Heterogeneity in Federated Learning

In real financial systems, transaction data differ across banks, regions, currencies, customer segments, and usage channels. Such **Non-IID (Non-Independent and Identically Distributed)** data significantly degrade FL model convergence (Zhao et al., 2018). Factors contributing to heterogeneity include:

- Different **account types** (savings, business, credit)
- Different **transaction channels** (ATM, POS, online, mobile)
- Varied **currency values and exchange behaviors**
- Geographical transaction locations and regional fraud patterns
- Unequal transaction volumes per client

To address these challenges, recent work proposes:

Method (Google Scholar Supported)	Contribution to Non-IID FL
FedProx (Li et al., 2020)	Adds a proximal term to reduce divergence across models.
Clustered FL (Sattler et al., 2020)	Groups clients with similar data distributions and trains separate models.
Personalized FL / Meta-learning FL (Fallah et al., 2020)	Allows customized fraud detection models for each institution.
Knowledge distillation FL (Chang et al., 2019)	Shares logits or model insights instead of raw parameters, improving generalization.

These approaches help maintain higher performance when data reflect real-world irregularities, such as users transacting in multiple regions or businesses interacting internationally.

2.5 Synthesis and Implications for This Study

The literature review confirms that both **unbalanced fraud data** and **heterogeneous client distributions** continue to hinder federated optimization in financial domains. While prior work proposes promising techniques, few studies have jointly explored these challenges within a complex, multi-currency, distributed fraud detection context.

Therefore, the present system builds upon earlier findings but extends them by:

1. Applying federated learning to **multi-regional, multi-currency, multi-client fraud detection**.
2. Incorporating **class-sensitive optimization** to address extreme imbalance.
3. Using **heterogeneity-aware aggregation** to improve model stability in cross-institution training.

This provides a more realistic foundation for global financial crime prevention, with both privacy compliance and decentralized intelligence playing an ever-important role.

Chapter 3:

Model architecture and analysis

The implementation consists of two primary components :
A centralized server coordinating the federated learning process and multiple clients training local models on datasets.The neural network architecture remains consistent across both server and client implementations ensuring parameter compatibility during federated aggregation.

3.1 Neural Network Architecture

The architecture implements a feed forward neural network. The network employs a progressive dimensionality reduction strategy through three distinct transformation stages :

- First Hidden Layer (fc1) : This layer performs an expansion from the input features to 128 neurons.The increased width enables the network to extract rich, non linear features representations from the input transaction data. In the context of fraud detection, this expansion allows the model to capture subtle patterns and relationships between transaction attributes such as amount , merchant category , country and temporal features.
- Batch normalization layer (bn1) : Positioned after the first ReLU activation this 128 dimensional batch normalization layer serves multiple critical functions.I normalizes the distribution of activations across the mini batch, reducing internal covariate shift during training. This is particularly valuable in federated learning scenarios where different clients may have heterogeneous data distributions.The normalization helps maintain training stability across distributed nodes and allow more aggressive learning rates.
- Second Hidden Layer (fc2) : This layer compresses the 128 dimensional representation onto 64 neurons, creating an information bottleneck.This compression forces the network to learn a more compact, salient representation of fraud relevant features. The dimensionality reduction also serves as an implicit regularization mechanism, preventing overfitting on local client data.
- Output layer (fc3) : The final fully connected layer maps the 64-dimensional feature to 2 output neurons, corresponding to the binary classification task. The raw output logits are processed through a crossentropyloss function during training which internally applies software normalization.
- Activation Functions
  ReLU (Rectified Linear Unit) The architecture employs ReLU activations after both the first and second hidden layers.This choice offers several advantages :
  - Computational efficiency compared to sigmoid or tanh functions
  - Mitigation of vanishing gradient problems in backpropagation
  - Sparse activation patterns that can improve model interpretability
  - Faster convergence during distributed training

3.2 Federated Aggregation Strategies

The server supports multiple federated aggregation strategies, each with distinct characteristics :

- FedAvg (Federated Averaging) : Performs weighted averaging of client model parameters based on dataset sizes. This baseline approach treats all clients equally in proportion to their data.
  $w_{t+1} = \Sigma(n_k / n) \times w_k$
  Where :
  - **w_{t+1}**: Global model parameters at round t+1
  - **n_k**: Number of training samples on client k

- **n**: Total number of samples across all clients ($n = \Sigma n_k$)
- **w_k**: Local model parameters from client k after training

- FedAvgM (Federated Averaging with Momentum) : Incorporated server-side momentum to smooth the aggregation process, potentially improving convergence speed and stability.
  $m_{t+1} = \beta \times m_t + \Delta w_t$ $w_{t+1} = w_t - m_{t+1}$
  Where:
  - **m_t**: Momentum buffer at round t
  - **β**: Momentum coefficient
  - **Δw_t**: Aggregated client update at round t
  - **w_t**: Global model parameters at round t
  - **w_{t+1}**: Updated global model parameters at round t+1
- FedProx (Federated Proximal) : Adds a proximal term that penalizes deviation from the global model, helping maintain consistency across heterogeneous client data distributions.
  $\min_w F_k(w) + (\mu/2) ||w - w_{global}||^2$
  Where :
  - **w**: Local model parameters being optimized
  - **$F_k(w)$**: Local loss function for client k
  - **μ**: Proximal coefficient
  - **w_{global}**: Current global model parameters
  - **$||w - w_{global}||^2$**: L2 squared distance between local and global parameters
- FedAdam (Federated Adam Optimizer) : The server maintains the first moment(mean) and the second moment (uncentered variance) estimated of the aggregated gradients using statistics to compute adaptive meaning rates for each parameter dimension.
  $m_t = \beta_1 \times m_{t-1} + (1 - \beta_1) \times \Delta w_t$
  $v_t = \beta_2 \times v_{t-1} + (1 - \beta_2) \times \Delta w_t^2$
  $\hat{m}_t = m_t / (1 - \beta_1^t)$
  $\hat{v}_t = v_t / (1 - \beta_2^t)$
  $w_{t+1} = w_t - \eta \times \hat{m}_t / (\sqrt{\hat{v}_t} + \tau)$
  Where :
  - **m_t**: First moment estimate (mean of gradients) at round t
  - **v_t**: Second moment estimate (uncentered variance of gradients) at round t
  - **β_1**: Exponential decay rate for first moment
  - **β_2**: Exponential decay rate for second moment
  - **Δw_t**: Aggregated client update (pseudo-gradient) at round t
  - **$\hat{m}_t$**: Bias-corrected first moment estimate
  - **$\hat{v}_t$**: Bias-corrected second moment estimate
  - **t**: Current round number
  - **η**: Server learning rate
  - **τ**: Small constant for numerical stability
  - **w_t**: Global model parameters at round t
- FedAdagrad (Federated Adaptive Gradient) : Accumulates squared gradients over all previous rounds to provide parameter specific learning rate adaptation.Unlike FedAdam it does not employ exponential moving averages but instead maintains a cumulative sum of squared gradients giving equal weight to all historical gradients.
  $v_t = v_{t-1} + \Delta w_t^2$
  $w_{t+1} = w_t - \eta \times \Delta w_t \ / \ (\sqrt{v_t} + \tau)$
  Where :
  - **v_t**: Cumulative sum of squared gradients up to round t
  - **v_{t-1}**: Cumulative sum of squared gradients up to round t-1
  - **Δw_t**: Aggregated client update at round t
  - **Δw_t^2**: Element-wise square of the aggregated update
  - **η**: Server learning rate

    - **τ**: Small constant for numerical stability
    - **w_t**: Global model parameters at round t
    - **√v_t**: Element-wise square root of cumulative squared gradients
- FedYogi (Federated Yogi Optimizer) : FedYogi represents a sophisticated variant of FedAdam that addresses its tendency to over adapt to recent gradients.It employs adaptive,sign-based updates that provide more controlled adaptation preventing the rapid increase of the second moment estimate and maintain more balanced learning rates throughout training
  $m_t = \beta_1 \times m_{t-1} + (1 - \beta_1) \times \Delta w_t$
  $v_t = v_{t-1} - (1 - \beta_2) \times \text{sign}(v_{t-1} - \Delta w_t^2) \times \Delta w_t^2$
  $w_{t+1} = w_t - \eta \times m_t / (\sqrt{v_t} + \tau)$
  Where :
    - **m_t**: First moment estimate at round t
    - **v_t**: Second moment estimate at round t (updated adaptively)
    - **β_1**: Exponential decay rate for first moment
    - **β_2**: Controls the adaptation rate of second moment
    - **Δw_t**: Aggregated client update at round t
    - **sign(x)**: Sign function returning -1, 0, or +1 based on whether x is negative, zero, or positive
    - **$v_{t-1} - \Delta w_t^2$**: Difference between previous second moment and current squared gradient
    - **η**: Server learning rate
    - **τ**: Small constant for numerical stability
    - **w_t**: Global model parameters at round t
    -

3.3 Performance Monitoring

The system tracks two primary metrics :

- Loss : Cross entropy loss is computed on each client's validation set and averaged across all participating clients.The loss provides insight into how well the model's predicted probability distribution aligns with the true label.Lower loss values indicate better model calibration and prediction confidence.
- Accuracy : Classification accuracy measures the proportion of correctly classified transactions on validation data. This metric offers an intuitive understanding of overall model performance.

Metrics are aggregated across clients using weighted averaging based on dataset sizes providing a global view of model performance while respecting data distribution differences.This weighting scheme ensures that clients with larger validation sets contribute proportionally more to the global performance estimate providing a representative view of model performance across the entire federated network while respecting data distribution differences.

Chapter 4:
Result Analysis

4.1 Comparative Result

Result	Random	Currency Wise	Target Label
FedAvg	0.958	0.912	0.681
FedAvg M	0.963	0.926	0.704
FedProx	0.960	0.933	0.742
FedAdam	0.972	0.948	0.781
FedAdagrad	0.949	0.901	0.660
FedYogi	0.968	0.940	0.764

4.2 Analysis

Case 1 : Random Equal Distribution

In this setting the dataset was randomly shuffled and evenly distributed among the three clients. Since all clients received a balanced and representative portion of the data, the global model trained smoothly and achieved high accuracy across all methods.

FedAdam performed the best with 97.2% accuracy , followed closely by **FedYogi , FedAvgM** and **FedProx**.

Although **FedAdagrad scored the lowest at 94.9%** it still delivered strong performance due to the consistency of the data seen by each client.

We can conclude that when clients receive similar and well balanced data all algorithms converge efficiently with adaptive optimizers providing a slight advantage.

Case 2 : Distribution based on currency

In the second setup each client received data corresponding to different transaction currencies.This introduced noticeable differences in feature patterns across clients.As a result, all aggregation methods experienced reduced accuracy compared to random split setting.

FedAvg dropped to **91.2%**, showing that it is sensitive to variations in client data.
Methods designed to handle client divergence such as **FedProx (93.3%)**, **FedAdam (94.8%)**, and **FedYogi (94.0%)** performed more reliably. These algorithms handled the variability between clients more effectively, maintaining higher accuracy.

When clients hold different types of data, approaches with adaptive updates or regularization are more resilient than basic averaging.

Case 3: Distribution Based on Fraud Label

The most challenging configuration assigned:

Only FraudLabel = 0 samples to Clients 1 and 2

Only FraudLabel = 1 samples to Client 3

This created a severe imbalance where no client possessed a complete view of both classes. As expected, model performance dropped sharply in this scenario.

FedAvg produced the lowest accuracy (68.1%), as client updates conflicted heavily.
FedAdagrad showed similar difficulties.
Methods designed to stabilize training in heterogeneous conditions performed better:
FedProx reached 74.2%, **FedYogi reached 76.4%**, and **FedAdam achieved the highest accuracy at 78.1%**, showing the strongest ability to handle extreme imbalance.

When clients hold fundamentally different types of data, algorithms such as FedAdam and FedProx show clear advantages due to their stability and resistance to client drift.

Overall Summary

Across all three data distribution strategies, several patterns emerged:

- FedAdam consistently achieved the highest accuracy, particularly in challenging conditions with uneven or imbalanced data.
- FedProx offered strong stability when clients had very different data, making it well-suited for real-world scenarios.
- FedAvg performed well only when client data was balanced, but dropped significantly when data varied across clients.
- FedYogi and FedAvgM provided solid, reliable performance in all cases.
- FedAdagrad was the weakest overall, especially when distributions differed between clients.

Chapter 5:
Future Work

1. Advance Imbalance Handling techniques : While the current system addresses class imbalance through basic cross-entropy loss , future iterations could incorporate more sophisticated techniques like Focal Loss, Class-Weighted Loss Function or Federated SMOTE.These approaches would enable better detection of rare fraud patterns by amplifying the learning signal from minority class samples without compromising client privacy.
2. Dynamic Client Clustering : The current system treats all clients uniformly during aggregation.Future implementations could incorporate dynamic client clustering based on data distribution similarity. Clients with comparable transaction patterns,currencies or fraud prevalence could cluster trained specialized sub models that better reflect the local characteristics.This approach would address the non IID challenge more effectively while maintaining the benefits of collaborative learning.
3. Real Time Fraud Detection : The current batch based training approach could extend to support online federated learning where models continuously adapt to emerging fraud patterns in real time.
4. Cross Borders and Multicurrency Optimization : The current implementation handles multiple currencies through one hot encoding but future system could develop currency aware embeddings or transfer learning approaches that explicitly model exchange rate dynamics and cross border patterns.This would improve fraud detection for international transactions where fraud patterns may differ from domestic operations.
5. Privacy Preserving Mechanism : Although federated learning inherently provides privacy by design, additional cryptographic techniques like differential privacy, secure multi-party computation(SMPC) , or homomorphic encryption could be integrated to provide formal privacy guarantees.

Chapter 6:

Conclusion

This project successfully demonstrates the application of federated learning to financial crime detection, addressing the critical challenges of enabling collaborative fraud pattern recognition across multiple institutions while preserving data privacy and regulatory compliance.

Through comprehensive experimentation with six distinct aggregation strategies such as FedAvg, FedAvgM, FedProx, FedAdam, FedAdagrad, and FedYogi we have systematically evaluated their performance under varying data distribution scenarios evaluated their performance under varying data distribution scenarios that reflect real world financial environments.

This work reinforces federated learning as a viable paradigm for privacy preserving financial crime detection. By enabling collaborative model training without raw data exchange, the system provides a practical pathway for financial institutions to comply with various data protection regulations while still benefitting from cross institutional intelligence.

This project demonstrates that sophisticated fraud detection models can be trained across distributed institutions without compromising customer privacy. As digital payment ecosystem expand globally and regulations become strict, federated learning offers a principled approach to balancing fraud detection with data protection obligations, providing fraud detection efficacy with data protection obligations, providing a foundation for secure, collaborative financial crime prevention.

References

1. H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. (2017). "Communication-Efficient Learning of Deep Networks from Decentralized Data." *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*.
2. D. Beutel, et al. (2020). "Flower: A Friendly Framework for Federated Learning." arXiv:2007.14390.

3. A. Paszke, et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library." *Advances in Neural Information Processing Systems 32 (NeurIPS)*.

4. W. McKinney. (2010). "Data Structures for Statistical Computing in Python." *Proceedings of the 9th Python in Science Conference (SciPy)*.

5. FedIIC: Towards Robust Federated Learning for Class-Imbalanced Medical Image Classification. (2022). arXiv:2206.13803. Also published in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, Lecture Notes in Computer Science, vol 14221. Springer, Cham.

6. Heterogeneity-Guided Client Sampling: Towards Fast and Efficient Non-IID Federated Learning. (2024). *Advances in Neural Information Processing Systems (NeurIPS)*. arXiv:2310.00198.

7. Fed-CBS: A Heterogeneity-Aware Client Sampling Mechanism for Federated Learning via Class-Imbalance Reduction. (2023). *Proceedings of Machine Learning Research (PMLR)*, vol. 202. arXiv:2209.15245.

8. HFL-GAN: Scalable Hierarchical Federated Learning GAN for High-Quantity Heterogeneous Clients. (2024). *Applied Intelligence*. https://doi.org/10.1007/s10489-024-05924-x

9. Class-Imbalanced Medical Image Classification Based on Semi-Supervised Federated Learning. (2023). *Applied Sciences*, vol. 13, no. 4, p. 2109. MDPI.

10. Shuffle-Diversity Collaborative Federated Learning for Imbalanced Medical Image Analysis. (2025). *Medical Image Computing and Computer Assisted Intervention – MICCAI 2025*.

Appendix-I

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF THE STUDENT	SREEJA SANYAL
ROLL NO	22051200
PROJECT TITLE	FINANCIAL CRIME DETECTION USING FEDERATED LEARNING
ABSTRACT OF THE PROJECT (WITHIN 80 WORDS)	This project implements a federated learning system for fraud detection using Flower and PyTorch. A central server coordinates training across three clients, aggregating their models without accessing their private data. The server saves the global model each round, but the system's evaluation method needs correction.

CONTRIBUTIONS:

- Designed and executed experimental evaluation across three data distribution scenarios
- Implemented data partitioning strategies (random, currency-based, fraud-label-based)
- Created comparative results tables and performed cross-strategy analysis
- Analyzed impact of data heterogeneity and class imbalance on model convergence and accuracy
- Wrote Chapter 4: Result Analysis , Section 3.3 and Conclusion

SIGNATURE OF GUIDE **SIGNATURE OF STUDENT**

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF THE STUDENT	AYANTIKA BARDHAN
ROLL NO	23057012
PROJECT TITLE	FINANCIAL CRIME DETECTION USING FEDERATED LEARNING
ABSTRACT OF THE PROJECT (WITHIN 80 WORDS)	This project implements a federated learning system for fraud detection using Flower and PyTorch. A central server coordinates training across three clients, aggregating their models without accessing their private data. The server saves the global model each round, but the system's evaluation method needs correction.

CONTRIBUTIONS:

- Designed and implemented the data preprocessing pipeline
- Implemented one-hot encoding for categorical variables
- Developed feature standardization using StandardScaler for numerical normalization
- Researched best practices for handling categorical and numerical features in federated learning.
- Wrote Chapter 1: Introduction (Sections 1.1, 1.2, 1.3).

SIGNATURE OF GUIDE **SIGNATURE OF STUDENT**

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF THE STUDENT	SAKET KUMAR
ROLL NO	22054212
PROJECT TITLE	FINANCIAL CRIME DETECTION USING FEDERATED LEARNING
ABSTRACT OF THE PROJECT (WITHIN 80 WORDS)	This project implements a federated learning system for fraud detection using Flower and PyTorch. A central server coordinates training across three clients, aggregating their models without accessing their private data. The server saves the global model each round, but the system's evaluation method needs correction.

CONTRIBUTIONS:

- Designed and implemented the neural network architecture with batch normalization
- Developed the client-side federated learning logic
- Configured client-side training parameters including local epochs, batch size, and learning rate
- Researched and documented the neural network architecture design principles
- Wrote Chapter 3: Model Architecture and Analysis (Section 3.1)

SIGNATURE OF GUIDE **SIGNATURE OF STUDENT**

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF THE STUDENT	ADITYA MAJUMDER
ROLL NO	22054214
PROJECT TITLE	FINANCIAL CRIME DETECTION USING FEDERATED LEARNING
ABSTRACT OF THE PROJECT (WITHIN 80 WORDS)	This project implements a federated learning system for fraud detection using Flower and PyTorch. A central server coordinates training across three clients, aggregating their models without accessing their private data. The server saves the global model each round, but the system's evaluation method needs correction.

CONTRIBUTIONS:

- Designed and implemented the central server logic
- Developed custom strategy wrappers (SaveModelFedAvg, SaveModelFedAvgM, SaveModelFedProx) to enable model persistence after each round.
- Configured server-side aggregation strategies including number of rounds, client requirements (min_available_clients), and client selection fractions
- Researched and documented FedAvg and FedAvgM aggregation algorithms
- Wrote Chapter 3: Model Architecture and Analysis (Section 3.2)

SIGNATURE OF GUIDE **SIGNATURE OF STUDENT**

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF THE STUDENT	ANANYA BISWAL
ROLL NO	2205874
PROJECT TITLE	FINANCIAL CRIME DETECTION USING FEDERATED LEARNING
ABSTRACT OF THE PROJECT (WITHIN 80 WORDS)	This project implements a federated learning system for fraud detection using Flower and PyTorch. A central server coordinates training across three clients, aggregating their models without accessing their private data. The server saves the global model each round, but the system's evaluation method needs correction.

CONTRIBUTIONS:

- Conducted comprehensive literature survey on federated learning and fraud detection
- Analyzed previous work and identified research gaps in federated financial crime detection
- Compiled references and citations from Google Scholar supported research
- Researched techniques for handling unbalanced datasets in federated environments
- Wrote Chapter 2: Literature Survey .

\

SIGNATURE OF GUIDE **SIGNATURE OF STUDENT**

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF THE STUDENT	ANUSA DE
ROLL NO	22052969
PROJECT TITLE	FINANCIAL CRIME DETECTION USING FEDERATED LEARNING
ABSTRACT OF THE PROJECT (WITHIN 80 WORDS)	This project implements a federated learning system for fraud detection using Flower and PyTorch. A central server coordinates training across three clients, aggregating their models without accessing their private data. The server saves the global model each round, but the system's evaluation method needs correction.

CONTRIBUTIONS:

- Implemented data partitioning strategies (random, currency-based, fraud-label-based)
- Collected and analyzed performance metrics (accuracy, loss) for all six aggregation methods.
- Researched future directions for federated learning in financial crime detection
- Implemented the weighted average metric aggregation function for a centralized accuracy report.
- Wrote Chapter 5: Future Work

SIGNATURE OF GUIDE **SIGNATURE OF STUDENT**