Towards Explainable and Balanced Federated Learning: A Neural Network Approach for Multi-Client Fraud Detection

Nurafni Damanik[1], Chuan-Ming Liu[2]*
College of Electrical Engineering and Computer Science, National Taipei University of Technology, Taipei, Taiwan[1]
Department of Computer Science and Information Science, National Taipei University of Technology, Taipei, Taiwan[2]

***Abstract*—The growing demand for secure and privacy-preserving machine learning frameworks has resulted in the implementation of federated learning (FL), especially in critical areas like Credit card fraud detection. This study presents a comprehensive federated learning architecture that incorporates Neural Networks as local models, in conjunction with KMeans-SMOTEENN to address class imbalance in distributed datasets. The system utilises the Flower framework, employing the FedAvg algorithm across ten decentralised clients to collectively train the global model while preserving raw data confidentiality. To improve model transparency and cultivate stakeholder trust, Local Interpretable Model-Agnostic Explanations (LIME) is utilized, offering localised, comprehensible insights into model decisions. The experimental results indicate that the suggested method effectively achieves high predictive accuracy and explainability, rendering it appropriate for real-world fraud detection contexts that necessitate data confidentiality and model accountability.**

Keywords*—Component federated learning; K-Means SMOTEENN; credit card fraud detection; LIME

I. Introduction

In recent years, the proliferation of global communication and advancements in computing technology have significantly contributed to the widespread use of credit card transactions. However, this growth has been accompanied by a surge in fraudulent credit card activities. According to data reported by the European Central Bank, Europe experiences annual financial losses amounting to billions of Euros as a result of credit cards [1].

Traditional fraud detection methods, which typically rely on rule-based systems or manual monitoring, have become increasingly inadequate due to their limited ability to dynamically adapt to evolving fraudulent patterns and their reliance on centralized data storage, raising significant privacy and regulatory concerns [2].

Consequently, researchers and practitioners have progressively shifted towards machine learning (ML) methods, which offer superior predictive capabilities by analyzing vast datasets and recognizing complex fraud patterns. While these ML-based approaches, including deep learning models such as CNN, LSTM, and Autoencoders, have shown promising results [3], the centralized nature of data processing inherent in these models poses significant challenges, particularly regarding data privacy, security, and regulatory compliance in sensitive sectors such as finance [4].

To address these critical limitations, federated learning (FL) has emerged as an innovative solution, enabling decentralized model training without exposing sensitive raw data. FL allows multiple institutions to collaboratively train global models by aggregating locally trained model parameters, significantly enhancing data privacy and security [5]. Despite its advantages, current FL implementations often face issues such as data imbalance and lack of model interpretability, critical aspects that affect real-world applicability, particularly in fraud detection scenarios [6]. How can we design a privacy-preserving and interpretable federated fraud detection system that remains effective under severe class imbalance and cross-client data heterogeneity, while satisfying regulatory constraints and enabling real-world deployment?

This study aims to: 1) develop a multi-client FL pipeline (FedAvg) that trains a global model without centralizing sensitive data; 2) address class imbalance at the client level by implementing a client-side hybrid resampling technique (KMeans-SMOTEENN) on each client's training subset prior to local training; validation and test datasets remain unaltered, no raw samples are transmitted from the client, and solely updated model parameters are dispatched for FedAvg aggregation; 3) compare six deep learning architectures—FNN, DNN, CNN, LSTM, Autoencoder, and a stacked DL model within the same FL pipeline to identify the best performer for fraud detection; 4) enhance interpretability by applying LIME to the top-performing model (by AUPRC) and analyzing feature attributions to support audit and domain validation; and 5) compare the performance of our proposed method with state-of-the-art approaches.

Regardless of its benefits, current research on FL-based fraud detection encounters two principal challenges: i) significant class imbalance and non-IID data distributions among clients, which obstruct the identification of minority-class fraud cases, and ii) restricted interpretability, which impedes auditing, regulatory compliance, and stakeholder confidence. Moreover, there is scant information concerning the efficacy of hybrid resampling techniques (e.g., KMeans-SMOTEENN) in federated environments, as well as the comparative performance of various deep learning architectures for rare-event fraud detection. These shortcomings present substantial obstacles to the dependable and regulatory-

*Corresponding Author.

compliant use of FL in practical financial settings.

The proposed framework advances regulation-aligned fraud detection by enabling cross-institution collaboration without data sharing, improving minority-class detection under severe imbalance and non-IID partitions, and closing the explainability gap via LIME. The result is a deployable recipe FL with client-side hybrid resampling, neural models, and post-hoc explanations that: i) raises recall/AUPRC for rare fraud events, ii) preserves privacy and supports compliance, iii) provides decision transparency for risk and audit teams, and iv) generalizes across heterogeneous clients. The approach is transferable to other sensitive domains (e.g., finance and healthcare), where privacy, imbalance, and interpretability are critical.

Building on these objectives, this work makes the following contributions:

1) We offer an innovative federated learning framework that integrates the FedAvg algorithm with KMeans-SMOTEENN, effectively tackling significant class imbalance and heterogeneity in distant client datasets.

2) Thorough Evaluation of Deep Learning Models: We rigorously assess and compare the efficacy of six different deep learning models (FNN, DNN, CNN, LSTM, Autoencoder, and Stacked-DL) in our federated learning framework, yielding critical insights for optimal model selection in fraud detection applications.

3) We utilize Local Interpretable Model-agnostic Explanations (LIME) on the top-performing model (highest AUPRC) to improve the interpretability and transparency of its predictions. This focused strategy guarantees that stakeholders comprehend feature contributions in essential projections.

4) Empirical Validation: Through thorough experimentation, we experimentally confirm that our proposed federated learning technique regularly outperforms standard centralized approaches in accuracy, recall, F1-score, AUC, and AUPRC.

This is how the rest of the study is structured: Section II reviews related work on centralized and federated credit card fraud detection, including class-imbalance remedies and model explainability. Section III details the materials and methods: the overall workflow, dataset, six neural architectures (FNN, DNN, CNN, LSTM, Autoencoder, and a stacked model), the KMeans-SMOTEENN resampling strategy, and the federated learning setup based on FedAvg and the system architecture. Section IV presents experimental results and discussion, including global evaluations, LIME-based explanations, and a comparison with state-of-the-art methods. Section V concludes the study and outlines directions for future work.

II. Related Work

Fraud detection algorithms employ machine learning to effectively identify fraudulent transactions. The majority of suggested CCFDS utilize centralized learning models, whereas a few academics are developing federated learning models to address fraud detection. The supervised, unsupervised, and semi-supervised learning models employ centralized learning algorithms [6].

Credit card fraud detection has been addressed through a variety of machine learning and deep learning techniques, encompassing both supervised and unsupervised methods. Khalid et al. [7] utilized a combination of supervised machine learning models, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), Bagging, and Boosting. Alrashdi et al. [8] concentrated on supervised machine learning techniques, employing Random Forest (RF), Extreme Gradient Boosting (XGBoost), and Light Gradient Boosting Machine (LightGBM) to identify fraudulent transactions. Feng and Kim [9] employed a combination of machine learning models, including Random Forest with AdaBoost (RF + AB), Gradient Boosted Decision Trees (GBDT), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Convolutional Neural Network (CNN).

Desai and Hase [3] undertook a review of deep learning-based CCFD algorithms, encompassing architectures such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Autoencoders (AE), Deep Belief Networks (DBN), and hybrid models like CNN-RNN.

Recent advancements in fraud detection through machine learning techniques have markedly improved detection efficacy; nevertheless, centralized methodologies frequently provide severe privacy and regulatory problems, particularly in financial institutions where the safeguarding of sensitive data is crucial.

Federated learning (FL) has emerged as a significant option, enabling collaborative learning among distant financial institutions while preserving the confidentiality of private data [10] [11].

Recent research in federated learning for credit card fraud detection has investigated diverse deep learning architectures and privacy-preserving methodologies. Tang and Liu [12] introduced a Structured Data Transformer (SDT) model that incorporates federated learning, utilizing the self-attention mechanism of Transformers to adeptly capture intricate feature correlations in serialized transaction data while preserving data privacy among several banks. Their adaptive federated aggregation technique mitigates client heterogeneity and improves model convergence. Nonetheless, their methodology fails to include explicit oversampling approaches for addressing class imbalance, nor does it emphasize model explainability. Complementary to this, Liu et al. [13] employed a federated hybrid oversampling method using K-Means SMOTEENN, which combines K-Means clustering, Synthetic Minority Oversampling Technique (SMOTE), and Edited Nearest Neighbors (ENN) cleaning to effectively tackle class imbalance and noise in fraud detection datasets. Their approach partitions the data into clusters to better preserve local minority class distributions before applying oversampling and noise filtering, thereby enhancing model robustness and generalization. Integrated with a stacking ensemble of diverse classifiers, this method significantly improved detection performance on credit card fraud data, achieving superior precision, recall, and F1-score compared to traditional resampling techniques.

Meanwhile, A. M. Salih et al. [14] highlighted the importance of Explainable AI techniques such as LIME and SHAP to provide transparency for deep learning models in regulated financial environments.

Our study builds upon these advances by integrating hybrid data balancing, federated learning, and explainability methods to achieve robust, interpretable fraud detection in distributed settings

III. Material and Method

A. Proposed Method

The workflow of the proposed method is illustrated in Fig. 1. Initially, raw data is collected and subjected to data preprocessing, which includes the application of the KMeans-SMOTEENN technique to handle imbalanced data effectively by generating synthetic samples and editing noisy instances. Afterwards, the federated learning (FL) framework is implemented, enabling collaborative training of deep learning models across distributed client environments without compromising data privacy. The deep learning models are then locally trained on each client's dataset and aggregated into a global model through the FL mechanism. Subsequently, evaluation metrics—including accuracy, precision, recall, F1-score, AUROC, and AUPRC—are computed to assess model performance. To enhance transparency and interpretability of the model decisions, the Explainable Artificial Intelligence (XAI) approach, specifically Local Interpretable Model-Agnostic Explanations (LIME), is integrated to explain the predictions and gain deeper insights into feature importance.

B. Dataset Description

This study employs a publicly accessible dataset of credit card transactions conducted by European cardholders. The European credit card fraud dataset was selected because it has become a widely accepted benchmark in fraud detection research, enabling direct and fair comparison with existing state-of-the-art methods. Its highly imbalanced distribution, with fraud cases accounting for only 0.172% of the total transactions, closely reflects real-world scenarios where fraudulent activities are rare but critical to detect.

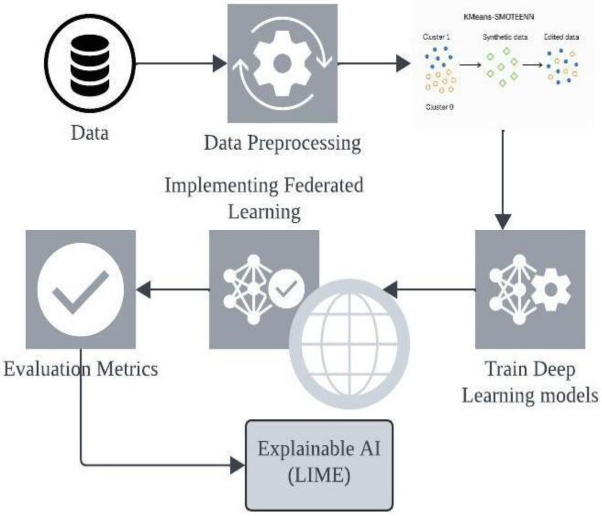

Fig. 1. Proposed method workflow.

The dataset's anonymized features, derived through PCA transformation, ensure compliance with data privacy and confidentiality requirements, aligning well with the regulatory considerations of financial applications. Furthermore, the dataset is publicly accessible, facilitating reproducibility and transparency in research. These characteristics make it an appropriate and representative choice for evaluating the effectiveness of federated learning in addressing fraud detection under severe class imbalance and privacy constraints.

The transactions encompass a two-day duration in September 2013, totaling 284,807 entries, of which merely 492 are classified as fraudulent. This indicates a significantly imbalanced class distribution, with fraudulent instances constituting merely 0.172% of the dataset. The dataset comprises 31 attributes: 'Time', 'Amount', 'Class', and the anonymized variables 'V1' to 'V28', which have been subjected to Principal Component Analysis (PCA) to maintain the confidentiality of sensitive data [1]. The dataset is suitable for machine learning operations because of its exclusively numeric properties. Moreover, its prevalent application in current research enables efficient benchmarking and performance evaluation against proven fraud detection algorithms.

This work involves data preprocessing in three primary stages: resampling to address class imbalance, standardization to equalize feature scales, and reshaping to prepare data for input into diverse machine learning architectures.

Fig. 2 demonstrates a significantly skewed class distribution within the dataset, comprising 283,253 normal transactions and merely 473 instances of fraud, or approximately 99.8% and 0.2% of the total data, respectively. This considerable disparity is a significant obstacle in the training of prediction models, as conventional classifiers often prioritize the majority class, hence risking the neglect of infrequent yet crucial fraudulent patterns. This discrepancy highlights the need for implementing resampling strategies to improve model sensitivity and guarantee effective fraud detection performance.

Considering that the minority class signifies fraudulent instances, employing the unaltered dataset without any balancing methods may considerably impair the model's capacity to identify fraud, resulting in suboptimal recall performance. An innovative hybrid resampling method, KMeans-SMOTEENN, is utilized to resolve this issue. This method integrates clustering-based oversampling with noise-filtering undersampling, facilitating a more efficient management of the dataset's class imbalance by producing synthetic minority samples while concurrently eliminating potential outliers and noisy majority samples.

Subsequent to resampling, feature standardization is executed utilizing StandardScaler, which is crucial owing to the diverse types and sizes of the dataset's features. Standardizing the data guarantees that each feature contributes uniformly to the learning process, hence improving model convergence and performance. Furthermore, the data is restructured by incorporating an additional dimension to align with the input specifications of the Neural Network model, which serves as the principal architecture for fraud detection in this research.

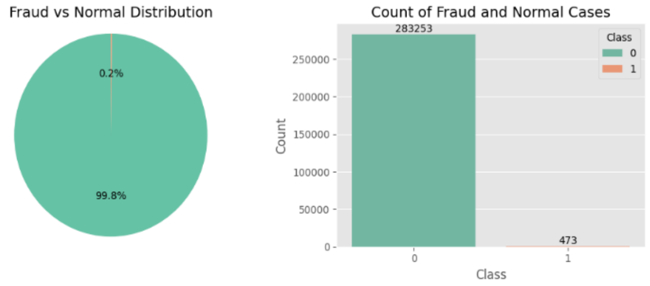

Fig. 2. Imbalance ratio of class fraud and non-fraud.

C. Deep Learning Model

This section succinctly elucidates the deep learning methodology employed in this work.

Since Hinton's breakthrough in 2006, deep learning has progressed swiftly, facilitating the development of robust architectures such as CNNs, RNNs, LSTMs, and Transformers. The breakthroughs, coupled with enhanced hardware capabilities and the accessibility of extensive open-source data, have established deep learning as a preeminent force in artificial intelligence [15].

The fundamental components of deep learning encompass backpropagation, stochastic gradient descent, and convolutional neural networks. It has proven useful in analyzing extensive datasets, especially in tasks such as semantic indexing, data annotation, and information retrieval. Nonetheless, obstacles persist, such as the management of high-dimensional and streaming data, in addition to guaranteeing model scalability. Notwithstanding these constraints, deep learning persists in its advancement, with continuous endeavors to create a more cohesive comprehension and methodology [16].

This research selected deep learning for its capacity to autonomously extract significant features from data, thus reducing the necessity for costly manual feature engineering. To examine the optimal architecture for resolving the research challenge, various deep learning models were deployed and assessed. These models typically consist of multi-layer neural networks, wherein each layer incrementally acquires data representations, spanning from fundamental properties to more intricate and abstract patterns.

In the studies, we employed six distinct variations of deep learning models to assess performance and identify the model yielding optimal results. The specifics of each model and the experimental findings will be comprehensively detailed in the experiment and analysis of results section. This methodology was employed to illustrate the generalizability of deep learning models and to identify the optimal solution depending on the dataset's specific attributes.

1) Autoencoder: Autoencoders are a particular category of neural networks intended to acquire efficient representations (encodings) of unlabeled input, primarily for dimensionality reduction or feature extraction [17].

2) FNN: A Feed-Forward Neural Network (FNN) is a category of artificial neural network in which information progresses unidirectionally from input, through hidden layers, to output without any cycles. Establishing the optimal network size, encompassing the quantity of hidden layers and neurons, is essential as it influences learning ability, generalization, and the likelihood of overfitting. The universal approximation theorem asserts that a single hidden-layer feedforward neural network can estimate any continuous function with a sufficient number of neurons; however, additional hidden layers frequently provide more efficient solutions [18].

3) CNN: A Convolutional Neural Network (CNN) is a prevalent deep learning method that has yielded favorable outcomes across several applications. Convolutional Networks can reveal hidden characteristics of fraudulent transactions and prevent model overfitting. The ConvNets algorithm comprises three primary layers: the convolution layer, the pooling layer, and the fully connected layer.

The convolution and pooling layers primarily execute feature extraction, but the fully connected layer subsequently maps the extracted features to the final output, such as classification [19].

4) LSTM: Long Short-Term Memory (LSTM) is a specialized architecture of artificial recurrent neural networks (RNN) employed to model time series data in deep learning. Unlike conventional feedforward neural networks, LSTM has feedback connections among hidden units linked to discrete time steps, enabling the learning of long-term sequence dependencies and the prediction of a transaction label based on the sequence of prior transactions [20].

5) DNN: A Deep Neural Network (DNN) is a kind of artificial neural network characterized by the presence of multiple hidden layers situated between the input and output layers, in contrast to shallow models. Like conventional neural networks, DNNs handle inputs by multiplying them with weights and transmitting the outcomes through hidden layers activated by nonlinear functions such as sigmoid, tanh, or ReLU. The model parameters are refined by minimizing an error function, typically by stochastic gradient descent, until convergence is achieved. DNN training has two primary phases utilizing the backpropagation algorithm: a forward pass, in which calculations flow from input to output, and a backward pass, during which incorrect gradients are propagated in reverse to adjust the weights[21].

6) STACKING-DL: Stacking is a significant ensemble learning strategy that synthesizes an optimal model by amalgamating predictions from many foundational machine learning algorithms in the initial layer (Emmanuel et al. 2023). The primary concept is to employ the prediction results of the base learner as input features, thereafter training and predicting with the meta-learner. During the model's training phase, an optimum combination of various machine learning methods is employed to fully leverage their respective strengths, hence enhancing the accuracy of the ensemble model's predictions [22].

Table I presents the configuration details and hyperparameters of the neural network models implemented in this study for credit card fraud detection. The models include Feed-Forward Neural Network (FNN), Deep Neural Network (DNN), 1D Convolutional Neural Network (CNN 1D), Long Short-Term Memory (LSTM), Autoencoder, and a Hybrid model.

- Input Activation Function: All models utilize the Rectified Linear Unit (ReLU) activation function at the input or hidden layers to introduce non-linearity and facilitate efficient training.
- Output Activation Function: Sigmoid activation is applied in the output layer of all models except the Autoencoder, which uses a Sigmoid function suited for reconstruction tasks. This allows for output values between 0 and 1, suitable for binary classification problems like fraud detection.
- Optimizer: The Adam optimizer is consistently used across all models, with an initial learning rate primarily set at 0.001, except for CNN 1D, which uses a lower rate of 0.0001 to accommodate convolutional learning dynamics.
- Learning Rate Decay and Dropout: Learning rate decay and dropout regularization are employed selectively to improve generalization and prevent overfitting. DNN, CNN 1D, LSTM, and Hybrid models incorporate decay rates and dropout values ranging approximately between 0.2 and 0.5, while FNN and Autoencoder models omit these parameters.
- Communication Rounds and Federated Clients: All models are trained in a federated learning setup with 50 communication rounds and 2 to 3 federated clients participating in the training process, ensuring distributed learning and privacy preservation.
- Train-Test Split Ratio: A consistent 80%-10%-10% split is applied for training, validation, and testing datasets, respectively, to maintain fair performance evaluation across models.

This configuration setup balances model complexity and training efficiency, aiming to achieve optimal detection performance while maintaining computational feasibility in a federated environment.

TABLE I. MODELS HYPERPARAMETERS

Model Name	FNN	DNN	CNN 1D	LSTM	Autoencoder	Hybrid
Input Activation Function	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU
Output Activation Function	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
Initial Learning Rate	0.001	0.001	0.0001	0.001	0.001	0.001
Learning rate decay	-	0.3	0.2	0.2	-	-
Dropout rate	-	0.3	0.2, 0.4	0.5	-	0.3, 0.2
Communication round	50	50	50	50	50	50
Number of federated clients	3	3	3	2	3	3
Train test ratio	80%-10%-10%	80%-10%-10%	80%-10%-10%	80%-10%-10%	80%-10%-10%	80%-10%-10%

D. K-Means SMOTEENN Imbalanced Handling Dataset

This work employs a hybrid data-balancing methodology that integrates K-Means clustering with SMOTE (Synthetic Minority Oversampling Technique) and ENN (Edited Nearest Neighbors), generally referred to as K-SMOTEENN. This strategy seeks to resolve issues of class imbalance and overlap between classes frequently encountered in datasets, particularly in the fraud detection dataset utilized.

The K-SMOTEENN method we utilize consists of three primary phases:

1) K-Means clustering: The K-Means approach is employed to partition the minority class data into multiple clusters according to feature similarities. Each cluster signifies a distinct pattern within the minority class that exhibits greater homogeneity than the class as a whole.

2) Oversampling utilizing SMOTE: Subsequent to the segmentation of minority data into multiple clusters, we implement the SMOTE technique on each cluster independently to produce supplementary synthetic samples. This procedure is conducted just on clusters that satisfy the minimal sample size criteria for efficacy.

3) Sanitation utilizing ENN: Upon completion of the oversampling phase, the resultant dataset from the oversampling is amalgamated with the majority class. The final stage involves employing the ENN approach to eliminate unclear or potentially noisy samples, hence enhancing the quality of the final data.

The integration of the three methodologies yields a more balanced dataset, facilitating the classification model's ability to differentiate samples from other classes, particularly in overlapping areas. Practical applications of this methodology utilize Python libraries, including scikit-learn for K-Means clustering and imblearn for oversampling and undersampling techniques such as SMOTE and ENN. Optimal configurations, including the number of K-Means clusters and sample parameters, are determined during first experimentation [13].

Thus, this enables K-SMOTEENN to focus on sample data; this strategy is delineated in Algorithm 1.

Algorithm 1 K-SMOTEENN [13]

```
Input: training data S is a set of pairs $\{(x1, y1), \dots, (x2, y2), \dots, (xm, ym)\}$ where $x$ represents the input data and $y$ is the corresponding target vector.
n (the number of samples)
k (indicates the number of clusters)
irt (imbalance ratio threshold)
knn (quantity of nearest neighbors)
begin
// Step 1: Divide the input space into clusters clusters ← kmeans(X)
filtered clusters ← empty set for c in cluster:
    imbalance ratio ← $\frac{\text{majority count(c)+1}}{minority\ count\ (c)+1}$
    If the imbalance ratio is less than the irt, add the cluster "c" to the filtered cluster set. Repeat this process until all clusters have been checked.
 end
end
Step 2: Implement the SMOTE oversampling technique.
1) From the minority class, choose an instance $x_i$
 at random
2) Determine the kind of $x_i$ and denote the samples
 as $S_j$
3) Randomly create a synthetic data point p by then
 selecting a sample in Sj called z, then creating a
 line segment in the feature space by connecting p
 and z
4) Minority class label assigned to p.
5) Create a series of synthetic instances by combining p and z convexly.
Step 3: Employing ENN techniques
1) Choose the arbitrary instance $x_r$ from the set S
2) Determine the knn of $x_r$, with k being equal to 5
3) Remove the $x_r$ Element if it has a more significant number of neighbors from the other class.
4) Iterate 6 to 8 steps for the entire training dataset.
End
```

E. Federated learning

Federated learning mainly refers to a distributed machine learning method implemented among numerous clients. The procedure involves N clients $\{C_1, C_2, ..., C_N\}$ indexed by k, each possessing its own local dataset $\{D_1, D_2, ..., D_N\}$, which is maintained locally, and data cannot be transferred between clients or gathered by a third party. Typically, a server organizes various clients and their training. FL encompasses three essential steps:

1) Initialization: At communication round *t*, the clients get the newest model wt from the server for initialization.

2) Local training: Each client C_k conducts iterative training based on its own local dataset D_k and hyperparameter η. The local model weight ω_t^k is updated to ω_{t+1}^k after certain training epochs according to $\omega_{t+1}^k \leftarrow \omega_t^k$ (η, $D_{k),}$ leftarrow and subsequently transmitted to the server.

3) Model aggregation: the server does model aggregation on the received local models and updates the global model ω_{glob}^{t+1} ← Agg (ω_t^{k+1}; k ∈ [1,....,N]).

Thus, FL facilitates collaboration among different clients in training a model without the necessity of data exchange, which is particularly advantageous for privacy-sensitive applications [23].

This study utilizes the Federated Averaging (FedAvg) algorithm as the primary approach within our federated learning (FL) system. FedAvg is acknowledged as the predominant strategy in federated learning, enabling clients to collaboratively train a global model while refraining from sharing raw data. Each client develops a local model utilizing its own dataset, while the central server orchestrates parameter distribution, aggregation, and updates. The model parameters from all participants are averaged and redistributed until convergence is reached. This technique enhances data privacy, reduces communication overhead, and guarantees scalability for multiple customers [24]. The full procedure is depicted in Algorithm 2.

Algorithm 2 FedAvg [24]

```
Server performs:
k is indexed as the K clients,
the minibatch size is denoted by B,
the number of local epochs is E,
the learning rate is denoted as $\eta$
$\omega_0$ is initialized by the server
for $t$ =1, 2, … , $T$ do
    $m$←max($C.K$, 1)
    $S_t \leftarrow m$ clients (random set)
    for $k \in S_t$ do
        $\omega_{t+1}^{k}$← ClientUpdate($k$,$\omega_t$)
    $\sum_{k=1}^{K} \frac{nk}{n} \omega_{t+1}^{k}$
ClientUpdate($k$,$\omega$):
    $\beta \leftarrow$ split $P_k$ into batches of size $\boldsymbol{B}$
    for each local epoch $i$ from 1 to $E$ do
        for batch $b \in \beta$ do
            $\omega \leftarrow \omega - \eta\nabla l(\omega; b)$
        end for
    end for
end for
end for
return $\omega$ to server
```

Fig. 3 depicts the federated learning architecture utilized in this research. The system consists of a central server and numerous distributed clients (Client 1 to Client 10). Each client maintains a distinct local dataset, which is saved and processed privately without the dissemination of raw data. The central server first disseminates a global deep learning model to every client. Clients thereafter train their models locally with private datasets and transmit the updated model parameters to the server. The server consolidates these parameters through federated averaging or analogous aggregation techniques to enhance and update the global model. This iterative procedure persists until model convergence is attained, safeguarding data privacy, reducing communication overhead, and adeptly managing heterogeneous data distributions among clients.

IV. Result and Discussion

This research presents an innovative approach to credit card fraud detection by leveraging federated learning (FL) combined with deep neural network models. To address the prevalent data imbalance issue in fraud detection, we employed the KMeans-SMOTEENN resampling technique during preprocessing. The experimental results are structured into two main sections:

metrics comparison per training round and global (overall) metrics across all rounds. The federated learning framework incorporated six distinct deep learning architectures: Autoencoder, CNN, DNN, FNN, LSTM, and a stacked deep learning model. We implemented the FL process using the Flower federated learning framework, a widely recognized Python library tailored for distributed learning scenarios.

All experiments were conducted on a Windows-based workstation, powered by an Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz, and equipped with 16 GB RAM. Model performance was rigorously evaluated using multiple performance indicators: Accuracy, Precision, Recall, F1-score, AUROC (Area Under the ROC Curve), and AUPRC (Area Under the Precision-Recall Curve). The ROC curve graphically represents the classifier's capability in discriminating between fraudulent and non-fraudulent transactions by plotting the True Positive Rate against the False Positive Rate at varying thresholds. AUROC provides a succinct numerical summary of this capability, ranging from 0 to 1, where a value close to 1 signifies excellent predictive accuracy. Additionally, the Precision-Recall Curve focuses specifically on the performance of the classifier in imbalanced class scenarios, such as fraud detection tasks. The subsequent subsections provide detailed visualizations and analyses of the per-round and global performance metrics for each deep learning architecture utilized in the federated learning framework.

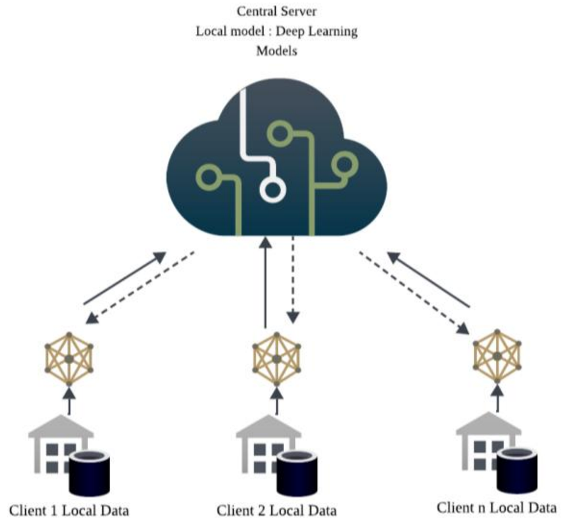

Fig. 3. Federated learning architecture.

A. Performance Evaluation

This section presents the assessment of our proposed federated learning framework for multi-client credit card fraud detection. To thoroughly evaluate the efficacy of various deep learning architectures employed in this federated framework, we analyzed their global performance indicators, consolidated over all clients and training iterations. The assessed designs are Autoencoder, Convolutional Neural Network (CNN), Deep Neural Network (DNN), Feedforward Neural Network (FNN), Long Short-Term Memory (LSTM), and Stacked Deep Learning model.

Table II delineates the comparative performance metrics—Accuracy, Precision, Recall, F1-score, AUROC (Area Under the Receiver Operating Characteristic Curve), and AUPRC (Area Under the Precision-Recall Curve)—for each deep learning model. All models demonstrated consistently high scores across the assessed metrics. The accuracy varied from 0.9992 to 0.9993, signifying an exceptionally high rate of right classifications between fraudulent and legitimate transactions. Precision scores were consistently outstanding, ranging from 0.9993 to 0.9994, indicating the models' exceptional proficiency in accurately recognizing actual positive cases, hence reducing false positive detections.

Recall metrics, which assess the models' efficacy in detecting genuine fraudulent cases (true positive rate), consistently exhibited elevated values (ranging from 0.9992 to 0.9993). Correspondingly, the F1-scores—which equilibrate precision and recall—were remarkably elevated (about 0.9992 to 0.9993), highlighting the models' equitable performance and formidable predictive capability in identifying fraudulent behaviors inside the federated learning framework.

To further assess the model's capacity to differentiate between classes, we used the AUROC and AUPRC measures. The AUROC scores for all models varied between 0.9748 and 0.9833, indicating that all models possess exceptional discriminative ability in differentiating fraudulent transactions from legitimate ones. Nevertheless, there was modest heterogeneity in the AUPRC ratings, indicating slight discrepancies in the efficacy of each model in managing the intrinsically imbalanced fraud detection data. The DNN got the highest AUPRC score (0.8420), closely succeeded by layered deep learning (0.8383) and CNN (0.8289), underscoring these models' specific capabilities in precision-recall equilibrium, a vital component in fraud detection contexts characterized by a scarcity of positive class cases.

TABLE II. PERFORMANCE COMPARISON OF EACH MODEL

Model Type	Accuracy	Precision	Recall	F1	Auroc	Auprc
Autoencoder	0.9993	0.9994	0.9993	0.9993	0.9748	0.8119
Cnn	0.9993	0.9993	0.9993	0.9993	0.9816	0.8289
Dnn	0.9993	0.9993	0.9993	0.9993	0.9833	0.8420
Fnn	0.9992	0.9993	0.9992	0.9992	0.9798	0.8117
Lstm	0.9993	0.9993	0.9993	0.9992	0.9708	0.8058
Stacked_dl	0.9993	0.9993	0.9993	0.9993	0.9855	0.8383

In addition to the tabular overview, Fig. 5 visually illustrates the comparative global metrics among the six architectures. The illustrated bar graphs validate the superior and closely comparable performance of each model across all measures, underscoring the resilience and dependability of our federated learning methodology. This consistency indicates that federated learning, when combined with suitable data resampling methods (namely KMeans-SMOTEENN) and diverse neural network topologies, may proficiently address data imbalance, resulting in highly accurate and balanced predicted outcomes.

In conclusion, both the table and graphical analyses unequivocally endorse the efficacy and dependability of our federated neural network methodology in identifying credit card fraud across various clients, yielding robust, equitable, and interpretable predicted results.

B. Explainable AI with LIME

We employed Explainable AI through the LIME (Local Interpretable Model-agnostic Explanations) technique to enhance the interpretability of the Deep Neural Network (DNN) model's predictions in fraud detection. Fig. 4 demonstrates LIME visualization for a transaction classified as non-fraudulent with complete certainty (probability = 1.00) that features V11, V2, and V17 significantly bolster the non-fraud forecast (shown in blue), but features V14, V3, and V15 marginally suggest a propensity for fraud (marked in orange).

Despite certain indicators indicating possible fraud, their total impact is negligible, therefore corroborating the model's determination as non-fraudulent. This visualization improves transparency in the model's decision-making process and elucidates the significance of various attributes in certain forecasts, which is essential for stakeholder comprehension in financial contexts.

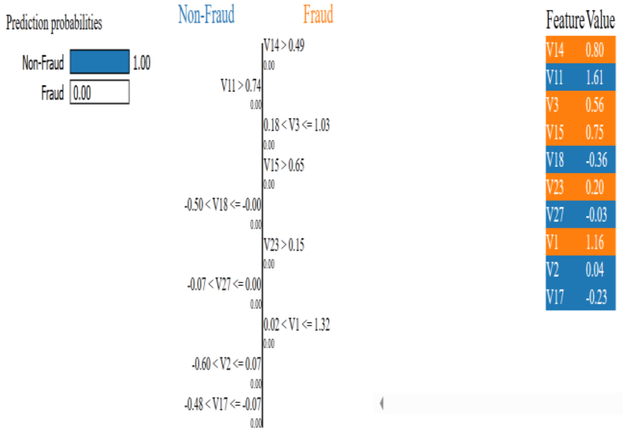

Fig. 4. Explainable AI.

C. Performance Comparison with State-of-the-Art Methods

To assess the efficacy and originality of our proposed approach, we performed a comparison analysis using other cutting-edge federated learning and unbalanced data management strategies documented in the literature. Table III demonstrates that our proposed model, which employs a Deep Neural Network (DNN) combined with the K-Means SMOTEENN method, exhibits exceptional performance, surpassing the majority of existing methodologies across various evaluation metrics, including Accuracy, Recall, Precision, F1-score, AUC, and AUPRC. Our method exhibited exceptional precision, recall, and F1-score, each at 99.93%, in addition to a high AUC value of 98.33% and a significantly raised AUPRC of 84.20%, outperforming comparable contemporary methodologies including those of Mustafa et al. [6], Liu et al. [13], and Saha et al. [29]. This thorough performance comparison highlights the efficacy and innovation of our federated learning methodology, especially in tackling the issues of imbalanced datasets in credit card fraud detection tasks.

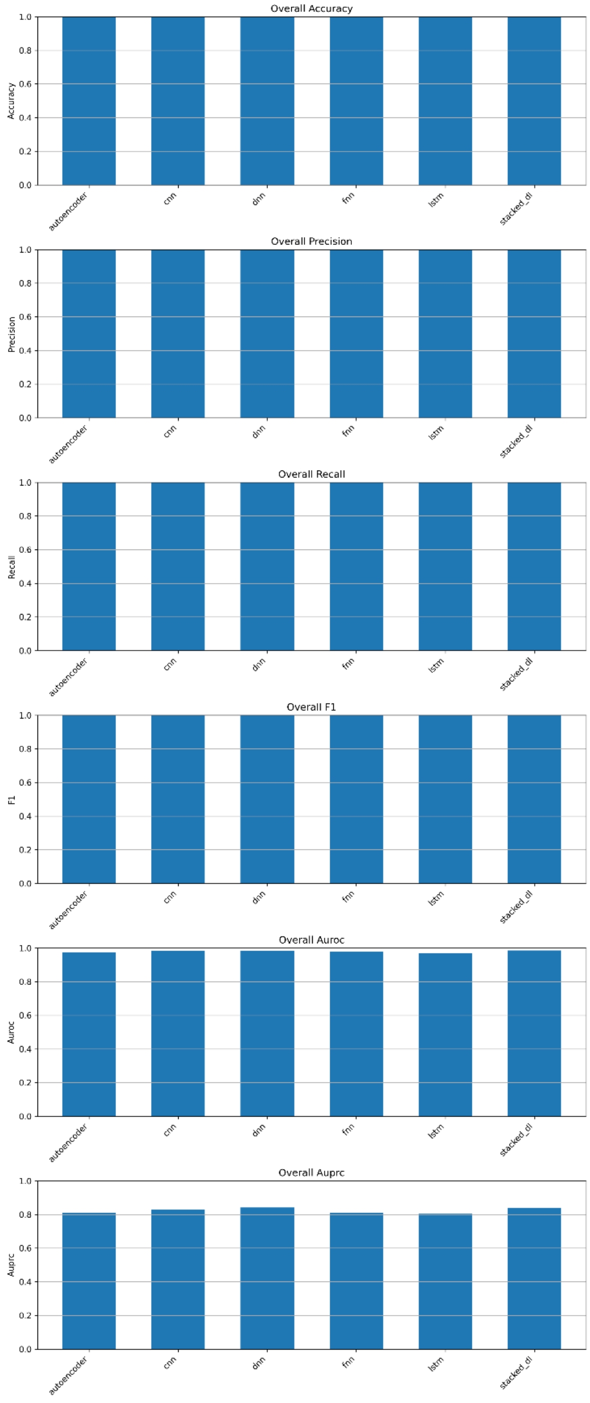

Fig. 5. Performance evaluation.

D. Discussion

Our findings illustrate the efficacy of integrating federated learning with the FedAvg algorithm, a K-Means-SMOTEENN hybrid methodology, and a Deep Neural Network (DNN) in tackling significant obstacles in fraud detection across several customers. The implementation of federated learning (FL) markedly improves data privacy by facilitating collaborative model training without the exchange of sensitive client information, consistent with prior study findings [1, 2]. Furthermore, the FedAvg technique effectively consolidates local models, yielding consistently superior prediction performance across diverse client datasets. The use of KMeans-SMOTEENN significantly enhanced the management of the pronounced class imbalance characteristic of fraud detection datasets. This hybrid approach substantially reduced the overlapping distribution between fraudulent and non-fraudulent occurrences by producing representative synthetic samples inside confined clusters. Our findings validate the results of other studies [3, 4], demonstrating that cluster-based oversampling strategies surpass conventional methods in intricate, imbalanced situations.

Additionally, we utilized Explainable AI (XAI) methodologies, specifically the LIME approach, to elucidate the predictions generated by our DNN model. The LIME display emphasized the most significant features impacting individual predictions, enhancing model transparency and interpretability. The interpretability of automated judgments is essential for stakeholders, particularly in financial services, since it enhances confidence and ensures regulatory compliance, as highlighted by Ribeiro et al. [5].

Notwithstanding the attainment of encouraging outcomes, our study possesses specific limitations. The existing solution predominantly assesses federated learning in simulated client environments, which may not accurately reflect real-world operational conditions characterized by network latency, data heterogeneity, and diverse compute resources among clients. Moreover, the utilization of synthetic data, although efficient, may generate artifacts that inadequately reflect authentic fraud trends.

TABLE III. PERFORMANCE COMPARISON WITH STATE-OF-THE-ART METHODS

Ref	Year	Method Use	Imbalanced Handling	Accuracy(%)	Recall	Precision	F1 Score	AUC	AUPPRC
Yang et al.[25]	2019	FL with CNN	-	-	-	-	-	-	-
Suvarna et al. [26]	2020	FL with RBF	-	94.00	-	-	-	-	-
Forough et al. [27]	2021	LSTM	-	-	74.08	95.6	78.1	83.3	-
Aurna et al. [28]	2023	FL with CNN	SMOTE	99.11	99.51	98.71	99.11	-	-
Aurna et al. [28]	2023	FL with MLP	SMOTE	98.28	98.77	97.78	98.28	-	-
Aurna et al. [28]	2023	FL with LSTM	SMOTE	95.78	98.20	80.23	88.31	-	-
Yuxuan et al. [12]	2024	Federated SDT	-	99.8	79.5	85.0	82.2	99.8	89.2
Mustafa et al. [6]	2024	Federated + CNN	SMOTE	-	80.9	8.26	81.7	93.7	-
Liu et al. [13]	2025	Stacking Ensemble without FL	K-Means SMOTEENN	1.00	0.88	0.95	0.92	1.00	0.96
SC Saha et al. [29]	2025	FinGraphFL	-	0.9780	-	-	-	0.9670	-
Proposed method	**2025**	**Our best model (DNN)**	**K-Means SMOTEENN**	**99.93**	**99.93**	**99.93**	**99.93**	**98.33**	**84.20**

V. CONCLUSION

This study introduces an innovative methodology that combines federated learning using FedAvg, KMeans-SMOTEENN to tackle data imbalance, and Explainable AI with a Deep Neural Network model to improve multi-client fraud detection systems. Our proposed method exhibited strong predictive performance, enhanced data privacy protection, effectively tackled class imbalance issues, and provided clear model interpretability.

The integration of these methodologies enhanced stakeholder trust and fostered a more profound comprehension of model-driven decisions in the financial sector.

Future research has found numerous promising avenues. Investigating sophisticated federated learning methodologies, including personalized FL (e.g., FedProx, FedBABU, or adaptive FL approaches), may enhance model efficacy and adaptability to client-specific data distributions. Moreover, the integration of Differential Privacy approaches would augment data privacy, providing enhanced safeguards against potential inference assaults.

Finally, applying this methodology to extensive real-world deployments, including dynamic and streaming data, could assess and enhance the practical usability and resilience of our suggested strategy.

ACKNOWLEDGMENT

This work was supported by the National Science and Technology Council (NSTC), Taiwan, with the grants 113-2221-E-027-051 and 114-2221-E-027-067.

REFERENCES

[1] V. V. K. Reddy, R. V. K. Reddy, M. S. K. Munaga, B. Karnam, S. K. Maddila, and C. S. Kolli, “Deep learning-based credit card fraud detection in federated learning,” Expert Syst. Appl., vol. 255, p. 124493, 2024.

[2] M. Adil, Z. Yinjun, M. M. Jamjoom, and Z. Ullah, “OptDevNet: A Optimized Deep Event-based Network Framework for Credit Card Fraud Detection,” IEEE Access, 2024.

[3] M. F. Ahamed, A. Salam, M. Nahiduzzaman, M. Abdullah-Al-Wadud, and S. M. R. Islam, “Streamlining plant disease diagnosis with convolutional neural networks and edge devices,” Neural Comput. Appl., vol. 36, no. 29, pp. 18445–18477, 2024.

[4] 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,” J. Risk Financ. Manag., vol. 18, no. 4, p. 179, 2025.

[5] S. Ji et al., “Emerging trends in federated learning: From model fusion to federated x learning,” Int. J. Mach. Learn. Cybern., vol. 15, no. 9, pp. 3769–3790, 2024.

[6] M. Abdul Salam, K. M. Fouad, D. L. Elbably, and S. M. Elsayed, “Federated learning model for credit card fraud detection with data balancing techniques,” Neural Comput. Appl., vol. 36, no. 11, pp. 6231–6256, 2024.

[7] A. R. Khalid, N. Owoh, O. Uthmani, M. Ashawa, J. Osamor, and J. Adejoh, “Enhancing credit card fraud detection: an ensemble machine learning approach,” Big Data Cogn. Comput., vol. 8, no. 1, p. 6, 2024.

[8] I. D. Mienye and N. Jere, “Deep learning for credit card fraud detection: A review of algorithms, challenges, and solutions,” IEEE Access, 2024.

[9] X. Feng and S.-K. Kim, “Novel machine learning based credit card fraud detection systems,” Mathematics, vol. 12, no. 12, p. 1869, 2024.

[10] M. A. Salam, D. L. El-Bably, K. M. Fouad, and M. Elsayed, “Enhancing Fraud Detection in Credit Card Transactions using Optimized Federated learning Model.,” Int. J. Adv. Comput. Sci. Appl., vol. 15, no. 5, 2024.

[11] T. Baabdullah, A. Alzahrani, D. B. Rawat, and C. Liu, “Efficiency of federated learning and blockchain in preserving privacy and enhancing the performance of credit card fraud detection (CCFD) systems,” Futur. Internet, vol. 16, no. 6, p. 196, 2024.

[12] Y. Tang and Z. Liu, “A Credit Card Fraud Detection Algorithm Based on SDT and Federated learning,” IEEE Access, vol. 12, pp. 182547–182560, 2024.

[13] N. Damanik and C.-M. Liu, “Advanced Fraud Detection: Leveraging K-SMOTEENN and Stacking Ensemble to Tackle Data Imbalance and Extract Insights,” IEEE Access, 2025.

[14] A. M. Salih et al., “A perspective on explainable artificial intelligence methods: SHAP and LIME,” Adv. Intell. Syst., vol. 7, no. 1, p. 2400304, 2025.

[15] X. Chen et al., “Deep learning-based software engineering: progress, challenges, and opportunities,” Sci. China Inf. Sci., vol. 68, no. 1, pp. 1–88, 2025.

[16] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[17] I. D. Mienye and T. G. Swart, “Deep autoencoder neural networks: A comprehensive review and new perspectives,” Arch. Comput. methods Eng., pp. 1–20, 2025.

[18] G. Bebis and M. Georgiopoulos, “Feed-forward neural networks,” Ieee Potentials, vol. 13, no. 4, pp. 27–31, 1994.

[19] M. L. Gambo, A. Zainal, and M. N. Kassim, “A convolutional neural network model for credit card fraud detection,” in 2022 International Conference on Data Science and Its Applications (ICoDSA), 2022, pp. 198–202.

[20] I. Benchaji, S. Douzi, B. El Ouahidi, and J. Jaafari, “Enhanced credit card fraud detection based on attention mechanism and LSTM deep model,” J. Big Data, vol. 8, no. 1, p. 151, 2021.

[21] N. Nguyen et al., “A proposed model for card fraud detection based on Catboost and deep neural network,” Ieee Access, vol. 10, pp. 96852–96861, 2022.

[22] W. Wang, M. Gu, Z. Li, Y. Hong, H. Zang, and D. Xu, “A stacking ensemble machine learning model for improving monthly runoff prediction,” Earth Sci. Informatics, vol. 18, no. 1, p. 120, 2025.

[23] P. Qi, D. Chiaro, A. Guzzo, M. Ianni, G. Fortino, and F. Piccialli, “Model aggregation techniques in federated learning: A comprehensive survey,” Futur. Gener. Comput. Syst., vol. 150, pp. 272–293, 2024.

[24] B. Yurdem, M. Kuzlu, M. K. Gullu, F. O. Catak, and M. Tabassum, “Federated learning: Overview, strategies, applications, tools and future directions,” Heliyon, vol. 10, no. 19, 2024.

[25] W. Yang, Y. Zhang, K. Ye, L. Li, and C.-Z. Xu, “Ffd: A federated learning based method for credit card fraud detection,” in International conference on big data, 2019, pp. 18–32.

[26] R. Suvarna and A. M. Kowshalya, “Credit card fraud detection using federated learning techniques,” Int. J. Sci. Res. Sci. Eng. Technol., vol. 7, no. 3, 2020.

[27] J. Forough and S. Momtazi, “Ensemble of deep sequential models for credit card fraud detection,” Appl. Soft Comput., vol. 99, p. 106883, 2021.

[28] N. F. Aurna, M. D. Hossain, Y. Taenaka, and Y. Kadobayashi, “Federated learning-based credit card fraud detection: Performance analysis with sampling methods and deep learning algorithms,” in 2023 IEEE International Conference on Cyber Security and Resilience (CSR), 2023, pp. 180–186.

[29] Z. Xia and S. C. Saha, “FinGraphFL: Financial Graph-Based Federated learning for Enhanced Credit Card Fraud Detection,” Mathematics, vol. 13, no. 9, p. 1396, 2025.