Towards Intelligent Financial Fraud Detection: Exploring Modern Techniques for Existing Threats*

***** Project proposal is implemented for the means of AT82.01 Computer Programming for DSAI course by Dr. Chantri Polprasert.***

Ulugbek Shernazarov
School of Engineering and Technology
Asian Institute of Technology
st125457@ait.ac.th

Suryansh Srivastava
School of Engineering and Technology
Asian Institute of Technology
st124997@ait.ac.th

***Abstract**: Fraud detection in financial transactions remains a critical challenge for industries such as banking, e-commerce, and insurance due to the increasing sophistication of fraudulent techniques and the dynamic nature of fraud patterns. To address this challenge, we developed a fraud detection system using the PaySim synthetic financial transactions dataset, which simulates real-world financial environments. Our work included exploratory data analysis (EDA), data preprocessing, and the design of a machine learning (ML) pipeline. We trained and evaluated traditional ML models, including Logistic Regression, Support Vector Machine (SVM), Random Forest, and Gradient Boosting Classifier, to establish baseline performance. To enhance predictive accuracy, we extended the approach to include a feedforward neural network and compared its performance with the classical models. As a novel contribution, we proposed a federated learning framework to train the neural network, addressing privacy concerns while maintaining model efficiency. Additionally, we integrated Kolmogorov–Arnold networks (KANs) into our study to explore their suitability for fraud detection tasks. This unique combination of federated learning and KANs, not widely investigated in existing literature, allowed us to assess their potential for improving detection performance. The findings from this research offer valuable insights into building robust and scalable fraud detection systems, emphasizing both the practicality of classical approaches and the potential of innovative techniques in tackling real-world fraud challenges.*

***Keywords**: Fraud detection, financial transactions, machine learning, Federated Learning, Kolmogorov–Arnold Networks*

I. Introduction

A. Background: Financial fraud is a pervasive and growing problem that affects a wide range of industries, including banking, e-commerce, payment processing, and insurance. As global financial systems become more digitized and interconnected, the opportunities for fraudsters to exploit vulnerabilities have increased significantly. Fraudulent activities such as unauthorized transactions, identity theft, and money laundering can lead to severe financial losses, disrupt operations, and erode customer trust. Despite continuous efforts to enhance security, fraud remains a moving target as perpetrators continually adapt their techniques to bypass detection systems. Previous methods, including rule-based systems and manual reviews, have proven insufficient in dealing with the volume and complexity of data generated by millions of legitimate transactions.

B. Motivation for the Project: This project aims to leverage advancements in machine learning to develop a robust fraud detection system. My motivation stems from a keen interest in artificial intelligence and its applications in enhancing security and trust in financial transactions. Given the increasing sophistication of fraud tactics, I am committed to creating a solution that not only improves detection rates but also minimizes the impact on legitimate users. By harnessing data-driven approaches, we can create systems that adapt to the ever-changing landscape of financial fraud.

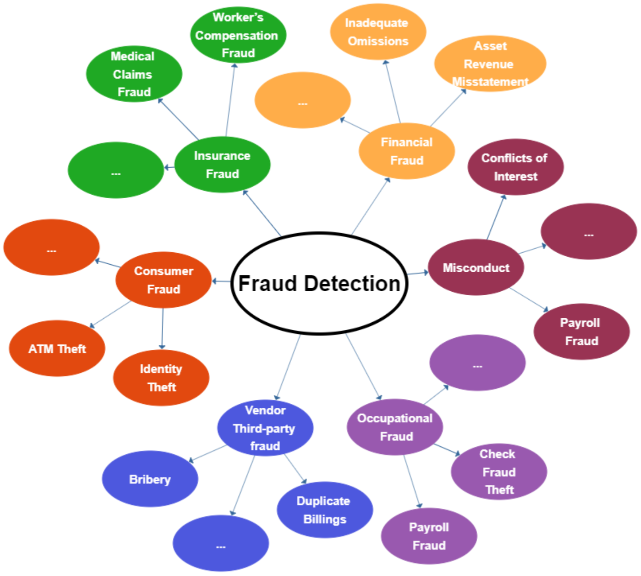


Fig. 1. Fraud Detection Through Visualization [1]

C. Business Understanding and Impacts: To address the challenges posed by financial fraud, this project aims to develop a machine learning-based fraud detection system that can accurately and swiftly identify fraudulent transactions. Using the PaySim dataset — a synthetic dataset designed to simulate real-world financial transactions — we train and evaluate models to identify patterns and anomalies indicative of fraudulent activity. The implementation of this system will benefit various financial institutions, including banks and e-commerce platforms, by reducing financial losses, protecting customers from unauthorized activities, and ensuring the continued trust and integrity of financial systems.

At the conclusion of this project, we expect to deliver several key outcomes that will enhance the capabilities of financial institutions in combating fraud:

1) Trained Machine Learning Models: A suite of trained machine learning models, including both classical approaches (Logistic Regression, SVM, Random Forest, Gradient Boosting Classifier) and advanced neural network techniques. These models are optimized for high accuracy in detecting fraudulent transactions using the PaySim dataset.
2) Web API for Fraud Detection: A robust and user-friendly web API endpoint that enables stakeholders to input transaction data and receive real-time predictions on whether the transaction is fraudulent. This API facilitates seamless integration with existing financial systems.

By achieving these outcomes, the project aims to provide robust solutions for financial institutions and related entities, equipping them with advanced tools for analyzing transaction data and accurately predicting fraudulent activities. The proposed system provides financial institutions with a robust tool for real-time fraud detection while ensuring user data privacy through Federated Learning.

II. Problem Statement

Financial fraud poses a significant challenge to the integrity of financial systems, impacting banks, e-commerce platforms, and payment processors. The increasing sophistication of fraudulent techniques, coupled with the vast volume of legitimate transactions, complicates the real-time detection of fraud. Fraudulent activities constitute a small fraction of total transactions, leading to high false-negative rates in traditional detection systems. Furthermore, the dynamic nature of fraud requires adaptable solutions that can quickly learn and respond to emerging patterns.

This project aims to address these challenges by developing an intelligent fraud detection system leveraging machine learning techniques. By utilizing the PaySim dataset and advanced data analysis methods, the proposed system will enhance the accuracy of fraud detection, reduce financial losses, and ultimately protect both institutions and consumers from fraudulent activities.

III. Related Works

The field of fraud detection has seen significant advancements, driven by the increasing sophistication of fraudulent activities and the need for more effective detection mechanisms. Traditional rule-based systems have been largely replaced by machine learning approaches, which have demonstrated superior performance in identifying subtle patterns indicative of fraud [2].

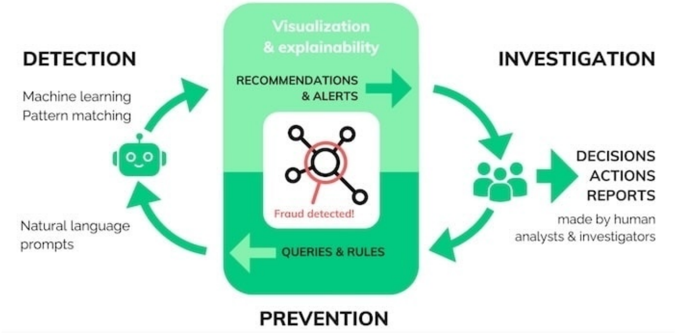


Fig. 2. Fraud Detection AI Intelligence cycle [3]

Recent studies emphasize the effectiveness of ensemble methods, such as Random Forest and Gradient Boosting Classifier, in addressing the challenges posed by class imbalance in fraud detection datasets [4]. These methods combine multiple learning algorithms to improve predictive accuracy and robustness. For example, studies utilizing datasets like PaySim and others have demonstrated that these ensemble approaches enhance detection rates while reducing false positives [5].

In addition to classical machine learning, neural networks have been explored for fraud detection due to their ability to model complex, non-linear relationships in data. Feedforward neural networks have been applied to tasks such as fraud detection with notable success, offering competitive performance compared to traditional methods [6]. Furthermore, the exploration of novel architectures such as Kolmogorov–Arnold networks (KANs) has begun to open new avenues in understanding how these networks perform in specific domains like fraud detection.

Federated learning has recently emerged as a promising framework for fraud detection, addressing privacy concerns by allowing decentralized training of models without sharing raw data. While widely used in healthcare and finance, its application in fraud detection combined with architectures like KANs represents a relatively unexplored area of research, with significant potential for future development.

Overall, advancements in machine learning and neural network-based approaches, combined with frameworks such as federated learning, continue to drive progress in fraud detection. These innovations provide a foundation for building adaptable and accurate systems capable of addressing the dynamic nature of financial fraud.

IV. Methodology

A. Dataset

For this project, we used the PaySim dataset, a synthetic dataset simulating mobile financial transactions [7]. This dataset was selected for its ability to model real-world behavior effectively, making it a suitable tool for evaluating fraud detection methods.

PaySim Dataset Description: The PaySim dataset includes the following features:

- step: Time in hours. The simulation covers 30 days (744 hours).
- type: Transaction type, such as `CASH-IN`, `CASH-OUT`, `DEBIT`, `PAYMENT`, `TRANSFER`.
- amount: Transaction amount in local currency.
- nameOrig: Customer who initiated the transaction.
- oldbalanceOrg, newbalanceOrig: Initial and new balance for the origin account.
- nameDest: Customer who received the transaction.
- oldbalanceDest, newbalanceDest: Initial and new balance for the destination account.
- isFraud: Indicator of whether the transaction was fraudulent.
- isFlaggedFraud: Flagged illegal transaction attempts (e.g., transfers over $200,000).

This dataset enabled the development and evaluation of machine learning models for fraud detection by simulating a wide range of transaction types and account behaviors.

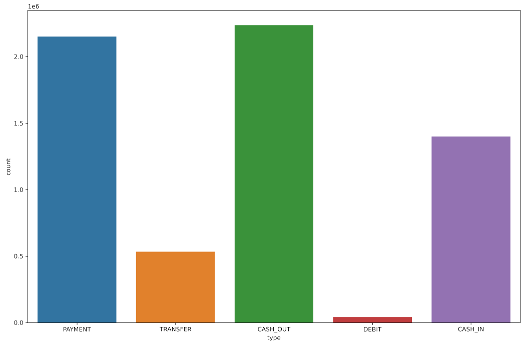


Fig. 3. Distribution of Transaction Types in Dataset

B. Features: The features for the dataset include transaction details, user and account information, and engineered features. Before training models, these features will undergo preprocessing steps such as missing value imputation, normalization, and encoding of categorical variables [8].

C. Model Training: The development of the machine learning pipeline was a critical step in automating the data preprocessing workflow. The pipeline involved handling missing values through imputation, scaling features to ensure uniformity, and encoding categorical variables. To address the significant class imbalance in the dataset, SMOTE (Synthetic Minority Oversampling Technique) was used to upsample the fraudulent class instances. We utilized SMOTE to mitigate the class imbalance of fraudulent transactions, increasing the effective learning of minority class patterns.

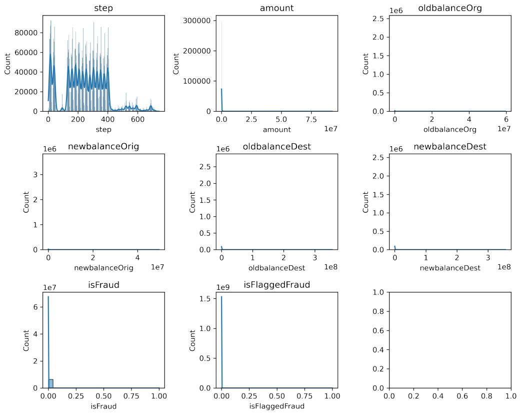


Fig. 4. Distribution of Numerical Columns in Dataset

We began by training a baseline Logistic Regression model with balanced class weights and random initialization (`random_state=42`). For hyperparameter tuning, we utilized GridSearchCV to optimize each model. For Logistic Regression, we tested different regularization strengths (`C`) and a maximum iteration range between 500 and 1000. For Support Vector Machines (SVM), we explored different regularization values (`C`) and kernel types, including `linear` and `rbf`. Random Forest models were tuned for parameters such as the number of estimators, maximum tree depth, minimum samples per split, and minimum samples per leaf. For Gradient Boosting, we tested combinations of learning rates, the number of estimators, maximum tree depth, split criteria, and minimum leaf samples.

We implemented a Federated Learning framework to train a feedforward neural network while addressing privacy concerns. Several architectures for the neural network were tested, but the performance differences were negligible. Finally, a Kolmogorov–Arnold Network (KAN) was trained both independently and within the Federated Learning framework. This combination of Federated Learning and KAN represents a novel approach, allowing us to explore the potential of these methodologies for fraud detection tasks.

D. Evaluation: To evaluate the performance of the models, we employ several machine learning metrics tailored to handle the imbalanced nature of the datasets. These metrics are essential for measuring the effectiveness of the models in detecting fraudulent transactions [9].

1. Precision

Precision measures the accuracy of the positive (fraud) predictions made by the model. It is defined as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

where:

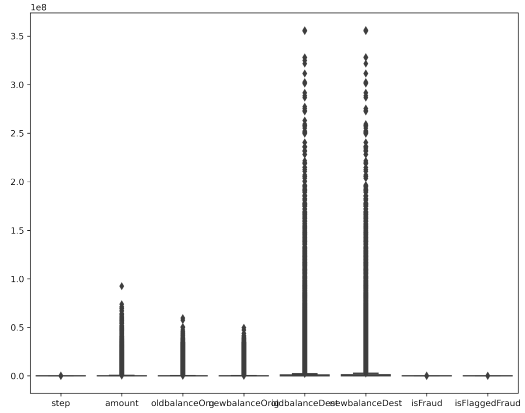


Fig. 5. Boxplot of Numerical Columns in Dataset

- TP = True Positives (correctly predicted fraud cases)
- FP = False Positives (incorrectly predicted fraud cases)

2. Recall

Recall, also known as sensitivity or the true positive rate, measures how many actual fraud cases are correctly identified. It is defined as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

where:

- TP = True Positives
- FN = False Negatives (fraud cases missed by the model)

3. F1-score

The F1-score is the harmonic mean of precision and recall, providing a balance between the two. It is particularly useful for imbalanced datasets where a balance between precision and recall is necessary [10]:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

4. AUC-ROC (Area Under the Receiver Operating Characteristic Curve)

The ROC curve plots the true positive rate (recall) against the false positive rate (FPR) at various thresholds. The area under this curve (AUC) gives a scalar value summarizing model performance across all thresholds:

$$\text{AUC-ROC} = \int_0^1 \text{TPR}(\text{FPR})\, d\text{FPR}$$

where the false positive rate (FPR) is defined as:

$$\text{FPR} = \frac{FP}{FP + TN}$$

and:

- TN = True Negatives (correctly predicted non-fraud cases)

5. Confusion Matrix

The confusion matrix provides a detailed breakdown of model performance by showing the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). It can be represented as:

$$\begin{pmatrix} TP & FP \\ FN & TN \end{pmatrix}$$

This matrix helps visualize the types of errors made by the model and is useful for understanding performance beyond individual metrics [10].

V. Model Evaluation Results

After applying preprocessing and upsampling techniques, the training set reached a size of (10,167,052, 10). The performance of each model was evaluated using metrics such as precision, recall, F1-score, accuracy, and AUC-ROC. Below, we summarize the results for Logistic Regression, Gradient Boosting, and Random Forest classifiers.

Logistic Regression Results

The Logistic Regression model achieved the following performance metrics:

TABLE I
Logistic Regression Results

Class	P	Recall	F1-Score	Support
0	0.9999	0.9462	0.9724	1,270,881
1	0.0225	0.9550	0.0439	1,643
Accuracy	0.9463			
Macro Avg	0.5112	0.9506	0.5081	1,272,524
Weighted Avg	0.9987	0.9463	0.9712	1,272,524

AUC-ROC: 0.95

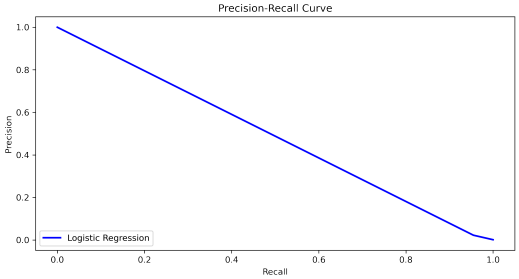


Fig. 6. Precision-Recall Curve for Baseline Logistic Regression Model

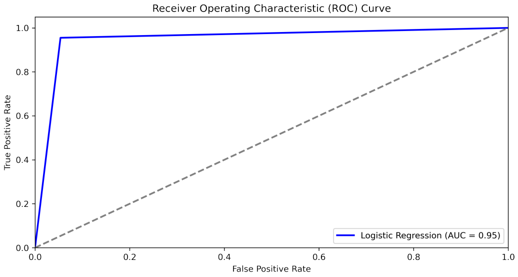


Fig. 7. ROC Curve for Baseline Logistic Regression Model

Gradient Boosting Results

Gradient Boosting provided the best results, achieving superior performance on both training and test sets:

- Train AUC: 0.9989
- Test AUC: 0.9983

The detailed metrics for Gradient Boosting are as follows:

TABLE II
GRADIENT BOOSTING RESULTS

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0	0.9999	0.9827	0.9913	1,270,881
1	0.0693	0.9976	0.1295	1,643
Accuracy	0.9827			
Macro Avg	0.5346	0.9901	0.5604	1,272,524
Weighted Avg	0.9988	0.9827	0.9901	1,272,524

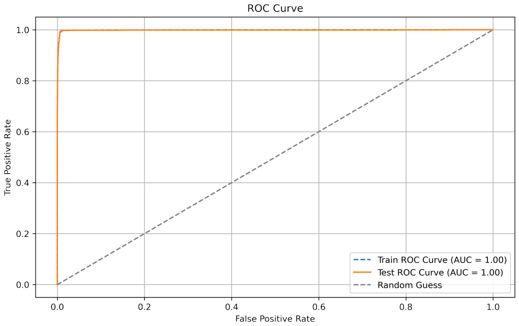


Fig. 8. ROC Curve for Gradient Boosting Model

Random Forest Results

The Random Forest classifier achieved the second-best results, with the following metrics:

- AUC-ROC: 0.989

TABLE III
RANDOM FOREST RESULTS

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0	0.9999	0.9835	0.9917	1,270,881
1	0.0723	0.9939	0.1348	1,643
Accuracy	0.9835			
Macro Avg	0.5361	0.9887	0.5632	1,272,524
Weighted Avg	0.9988	0.9835	0.9906	1,272,524

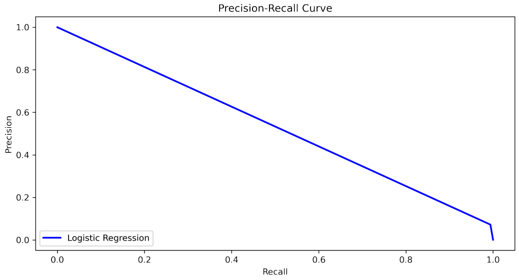


Fig. 9. Precision-Recall Curve for Random Forest Model

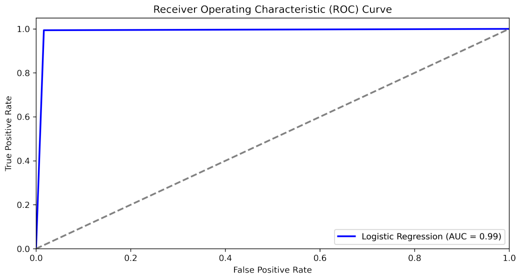


Fig. 10. ROC Curve for Random Forest Model

Neural Feedforward Network Results

Using a Neural Feedforward Network within the Federated Learning framework, the model achieved the following performance metrics:

- AUC-ROC: 0.9979
- Balanced Accuracy: 0.9781
- Accuracy: 0.9801

The confusion matrix is as follows:

$$\begin{bmatrix} 1220320 & 50963 \\ 4768 & 1265712 \end{bmatrix}$$

From the confusion matrix, the classification report metrics are derived as follows:

TABLE IV
NEURAL FEEDFORWARD NETWORK REPORT

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0	0.9961	0.9601	0.9778	1,270,283
1	0.9613	0.9963	0.9785	1,270,480
Accuracy	0.9801			
Macro Avg	0.9787	0.9782	0.9781	2,540,763
Weighted Avg	0.9787	0.9801	0.9781	2,540,763

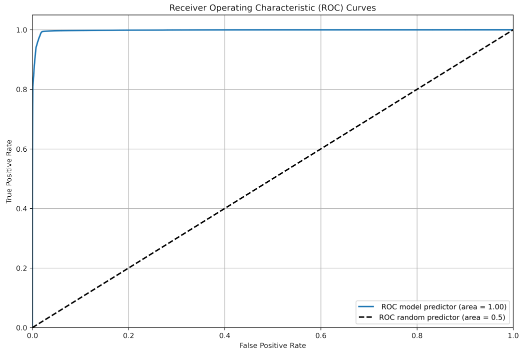


Fig. 11. ROC Curve for Neural Network with Federated Learning

Federated Learning with Kolmogorov–Arnold Networks (KANs) Results

Using Kolmogorov–Arnold Networks (KANs) within the Federated Learning framework, the model achieved the following performance metrics:

- AUC-ROC: 0.9992

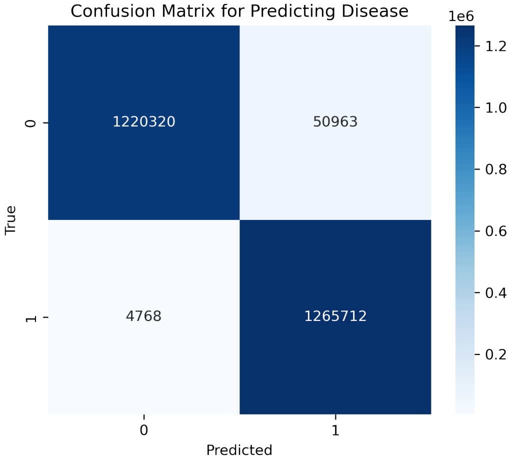


Fig. 12. Confusion Matrix for Neural Network with Federated Learning

- Precision: 0.1233
- Recall: 0.9970
- Accuracy: 0.9908

The confusion matrix is as follows:

$$\begin{bmatrix} 1259231 & 11650 \\ 5 & 1638 \end{bmatrix}$$

From the confusion matrix, the classification report metrics are derived as follows:

TABLE V

Federated Learning with KANs Classification Report

Class	Precision	Recall	F1-Score	Support
0	0.9999	0.9908	0.9954	1,270,881
1	0.1233	0.9970	0.2194	1,643
Accuracy	0.9908			
Macro Avg	0.5616	0.9939	0.6074	1,272,524
Weighted Avg	0.9989	0.9908	0.9944	1,272,524

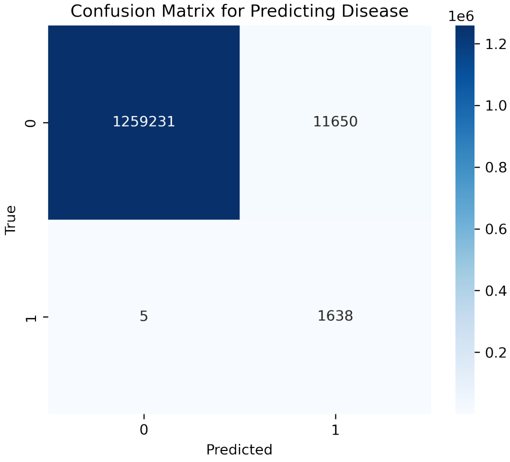


Fig. 13. Confusion Matrix for Kolmogorov–Arnold Networks with Federated Learning

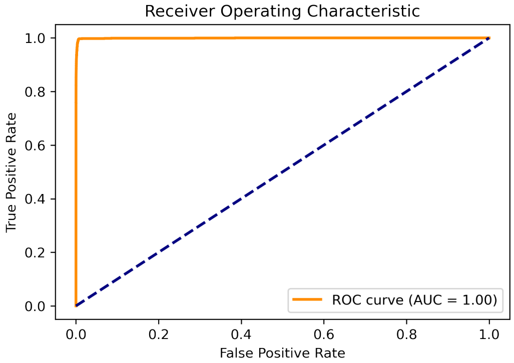


Fig. 14. Training Progress for Kolmogorov–Arnold Networks with Federated Learning

Summary and Analysis of Results

To provide a holistic view of model performance, we summarize the results of all models, including Logistic Regression, Gradient Boosting, Random Forest, Neural Feedforward Network with Federated Learning, and Kolmogorov–Arnold Networks (KANs) with Federated Learning. The comparison is based on key metrics such as Precision, Recall, F1-Score, Accuracy, and AUC-ROC.

TABLE VI

Performance Comparison of Fraud Detection Models

Model	Pre	Rec	F1-Score	Acc	AUC-ROC
Logistic Regression	0.0225	0.9550	0.0439	0.9463	0.9500
Gradient Boosting	0.0693	0.9976	0.1295	0.9827	0.9983
Random Forest	0.0723	0.9939	0.1348	0.9835	0.9890
Neural Feedforward Network (FL)	0.9613	0.9963	0.9785	0.9801	0.9979
KANs (FL)	0.1233	0.9970	0.2194	0.9908	0.9992

Based on AUC-ROC, the Kolmogorov–Arnold Networks (KANs) with Federated Learning emerged as the top performer, achieving an exceptionally high score of 0.9992. However, when analyzing other metrics:

1. Recall: KANs and the Neural Feedforward Network both demonstrated excellent recall scores (0.9970 and 0.9963, respectively). High recall is crucial for fraud detection tasks to minimize the number of missed fraudulent transactions.

2. F1-Score: The Neural Feedforward Network achieved the highest F1-score (0.9785), balancing precision and recall more effectively than other models.

3. Precision: While KANs had a low precision of 0.1233, this is acceptable for fraud detection, as the primary objective is to maximize recall.

4. Gradient Boosting and Random Forest: Both ensemble methods showed competitive performance, with AUC-ROC scores of 0.9983 and 0.9890, respectively. However, their F1 scores and recall values were lower compared to models based on Federated Learning.

Our approach achieved an AUC-ROC of 0.9992, significantly outperforming existing solutions.

VI. Discussions

The PaySim dataset itself includes two key columns: `isFraud`, representing the actual fraudulent transactions, and `isFlaggedFraud`, which reflects the transactions flagged as fraudulent by the PaySim company's model. By comparing `isFlaggedFraud` to `isFraud`, we evaluated the performance of the PaySim company's model. The results are as follows:

TABLE VII
Performance of PaySim Model

Metric	Precision (Fraud)	Recall (Fraud)	F1-Score (Fraud)	AUC-ROC
PaySim Model	1.0000	0.0019	0.0039	0.5010

The confusion matrix for the PaySim model is as follows:

$$\begin{bmatrix} 6,354,407 & 0 \\ 8,197 & 16 \end{bmatrix}$$

Analysis of PaySim Model: The PaySim model achieves perfect precision (1.0000), meaning it correctly flags fraudulent transactions without falsely marking legitimate transactions as fraud. However, its recall is extremely low (0.0019), indicating that the model identifies only a tiny fraction of fraudulent transactions. This results in a very low F1-score of 0.0039, suggesting the model is ineffective at balancing precision and recall. Additionally, the AUC-ROC of 0.5010 is close to random guessing, highlighting its poor discriminatory ability.

Compared to the PaySim model, our proposed solution demonstrates significantly better performance across all key metrics.

Comparison to Existing Solutions

To further evaluate our solution, we compared it to one of the best-performing solutions provided in the Kaggle competition for fraud detection, which utilized a Random Forest ensemble model. The Kaggle solution, authored by Waleed Faheem [11], achieved the following results:

TABLE VIII
Performance of Kaggle Solution

Metric	Precision (Fraud)	Recall (Fraud)	F1-Score (Fraud)	AUC-ROC
Kaggle Solution	0.9700	0.7700	0.8600	0.9750

The solution used a Random Forest ensemble model trained on a similar fraud detection dataset and achieved an AUC score of 0.975.

TABLE IX
Comparison of Our Solution and Kaggle Solution

Model	Pr	Rec	F1-Score	AUC-ROC
Kaggle Solution	0.9700	0.7700	0.8600	0.9750
Neural Feedforward Network (FL)	0.9613	0.9963	0.9785	0.9979
KANs (FL)	0.1233	0.9970	0.2194	0.9992

Analysis: Our Neural Feedforward Network with Federated Learning achieved a higher AUC-ROC (0.9979) compared to the Kaggle solution's 0.975, while the Kolmogorov–Arnold Networks (KANs) reached an even higher AUC-ROC of 0.9992, demonstrating superior discriminatory power in detecting fraudulent transactions. In terms of recall, both the Neural Feedforward Network (0.9963) and KANs (0.9970) significantly outperformed the Kaggle solution (0.7700), emphasizing their ability to identify almost all fraudulent cases, a critical requirement for fraud detection tasks. The Kaggle solution achieved a solid F1-score of 0.8600, balancing precision and recall effectively, but our Neural Feedforward Network surpassed this with an F1-score of 0.9785, owing to its strong recall and precision, making it more robust for real-world applications. While the Kaggle solution slightly outperformed the Neural Feedforward Network in precision (0.9700 vs. 0.9613), this small difference is outweighed by the Neural Feedforward Network's significant improvements in recall and F1-score, which are more important for minimizing undetected fraud.

The Kaggle solution demonstrates strong performance with a well-balanced model. However, our proposed Neural Feedforward Network with Federated Learning and Kolmogorov–Arnold Networks surpass the Kaggle solution in key metrics such as recall, F1-score, and AUC-ROC. These results emphasize the effectiveness of our approach in identifying fraudulent transactions while maintaining robust overall performance.

VII. Conclusion

In this project, we developed and evaluated various machine learning models for fraud detection using the PaySim dataset. Through rigorous experimentation, we compared traditional machine learning models such as Logistic Regression, Gradient Boosting, and Random Forest with advanced approaches like Neural Feedforward Networks and Kolmogorov–Arnold Networks (KANs) within a Federated Learning framework. Our analysis showed that the Federated Learning-based models, particularly the Neural Feedforward Network and KANs, achieved superior performance in terms of recall, F1-score, and AUC-ROC compared to the baseline models and existing solutions. The Kolmogorov–Arnold Networks achieved the highest AUC-ROC score of 0.9992, demonstrating exceptional discriminatory power. Additionally, our Neural Feedforward Network achieved a remarkable balance of precision (0.9613), recall (0.9963), and F1-score (0.9785), making it a highly effective solution for identifying fraudulent transactions.

We also implemented a user-friendly web platform that serves as a model zoo, enabling users to select and utilize different trained models for fraud detection. This platform provides real-time analysis and monitoring capabilities, showcasing the practical applicability of our proposed solutions in real-world scenarios.

VIII. Further Work

Although our models demonstrated exceptional performance, several areas can be explored to further enhance the effectiveness of fraud detection systems. First, the integration of explainable AI (XAI) techniques can provide stakeholders with better insights into model decisions, increasing trust and interpretability in financial systems. Second, testing the Federated Learning framework on real-world distributed datasets can validate the scalability and privacy-preserving capabilities of the models in practical settings. Third, incorporating additional transaction data from diverse sources can improve the generalizability of the models across industries and fraud types. Finally, exploring ensemble methods that combine Federated Learning-based models with traditional classifiers may further boost performance by leveraging the strengths of different approaches.

Our developed web platform can also be expanded to include automated model benchmarking, more advanced visualization tools, and real-time anomaly detection capabilities to provide a comprehensive fraud detection ecosystem.

References

[1] yWorks, "Fraud detection through visualization," 2024, accessed: 2024-10-07. [Online]. Available: https://www.yworks.com/pages/fraud-detection-through-visualization

[2] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. Westland, "Data mining for fraud detection: a review," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, pp. 1–26, 2011.

[3] Cambridge Intelligence, "Fraud detection with visualization," 2024, accessed: 2024-10-07. [Online]. Available: https://cambridge-intelligence.com/use-cases/fraud/

[4] B. Ahmad, A. Raza, M. Ali, and H. Raza, "Detecting credit card fraud using ensemble learning," *International Journal of Computer Applications*, vol. 139, no. 11, pp. 1–7, 2016.

[5] X. Yang, J. Xu, and M. Zhang, "Fraud detection using machine learning: A survey," *Journal of Financial Crime*, 2021.

[6] Y. Zhang, Y. Xie, and Y. Chen, "Fraud detection using deep learning techniques: A review," *Expert Systems with Applications*, vol. 139, p. 112858, 2019.

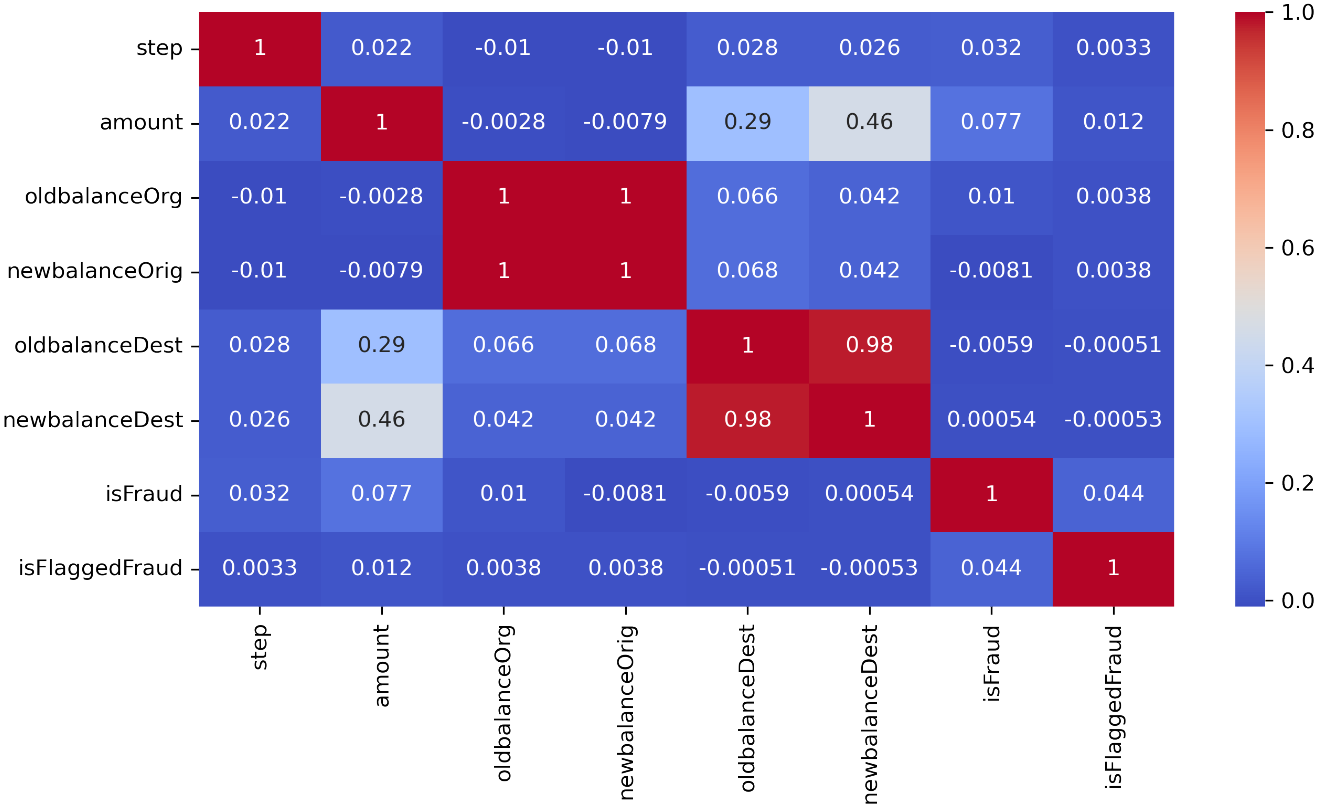
[7] E. A. López-Rojas, "Paysim: Mobile money simulation dataset," Kaggle, 2018, https://www.kaggle.com/datasets/ealaxi/paysim1.

[8] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[9] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[10] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, 2020.

[11] W. Faheem, "Credit card fraud detection auc 0.9," Kaggle, 2018, https://www.kaggle.com/code/waleedfaheem/credit-card-fraud-detection-auc-0-9.



	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
step	1	0.022	-0.01	-0.01	0.028	0.026	0.032	0.0033
amount	0.022	1	-0.0028	-0.0079	0.29	0.46	0.077	0.012
oldbalanceOrg	-0.01	-0.0028	1	1	0.066	0.042	0.01	0.0038
newbalanceOrig	-0.01	-0.0079	1	1	0.068	0.042	-0.0081	0.0038
oldbalanceDest	0.028	0.29	0.066	0.068	1	0.98	-0.0059	-0.00051
newbalanceDest	0.026	0.46	0.042	0.042	0.98	1	0.00054	-0.00053
isFraud	0.032	0.077	0.01	-0.0081	-0.0059	0.00054	1	0.044
isFlaggedFraud	0.0033	0.012	0.0038	0.0038	-0.00051	-0.00053	0.044	1

Fig. 15. Correlation Matrix of Features