Advanced Data Mining Methods for Financial Fraud Detection: A Comparative Analysis of Random Forest and CNN-LSTM

Dr. Mahesh T.R
Department of CSE
JAIN(Deemed-to-be-University)
Bengaluru,India
line 5: email address or ORCID

Aruleeswaran M.S
Department of CSE
JAIN(Deemed-to-be-University)
Bengaluru,India
senthilkumararuleeswaran@gmail.com

Siva Sathvik Innamuri
Department of CSE
JAIN(Deemed-to-be-University)
Bengaluru,India
sathvikinnamuri777@gmail.com

Jashwanth P
Department of CSE
JAIN(Deemed-to-be-University)
Bengaluru,India
line 5: email address or ORCID

Duvvuru Mukesh
Department of CSE
JAIN(Deemed-to-be-University)
Bengaluru, India
line 5: email address or ORCID

***Abstract*—Financial fraud has been a major challenge in the information age, as more cases of fraudulent schemes continue to affect people and organizations across the globe. Traditional methods of fraud detection, often based on rule-based systems, have proven to be inadequate against fraud schema dynamics and complexities. This paper offers a comparative review of two advanced machine learning models-Random Forest and Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM)- for detecting fraudulent transactions in financial datasets. The research utilizes the data preprocessing methodology such as use of the Synthetic Minority Over-sampling Technique (SMOTE) for addressing the issue of class imbalance and improvement of model efficiency. Accuracy, precision, recall, and F1-score have been utilized as performance metrics in comparing the viability of the two models. The results indicate that the two models have high accuracy and reliability, with CNN-LSTM being more efficient than Random Forest in detecting fraudulent transactions. In addition, the use of SHAP (Shapley Additive exPlanations) values provides a glimpse of model interpretability to better understand which features are most influential on fraud detection. The results demonstrate the efficiency of cutting-edge machine learning methods in refining fraud detection systems, and providing useful information to financial institution looking to reduce the risk of fraud. This study contributes to existing literature on the application of data mining methods in fraud detection, noting the call for advanced methods in stifling financial fraud.**

Keywords—Financial Fraud Detection, Machine Learning, Data Mining, Random Forest, CNN-LSTM, SMOTE, SHAP Values, Predictive Analytics, Fraud Prevention, Financial Security.

I. INTRODUCTION

A. Background of the Problem

Financial fraud has turned into an epidemic in the online economy that targets individuals, businesses, and financial institutions globally. With the rapid growth of Internet-based transactions and online banking, the volume of and sophistication in financial transactions increased, providing fertile ground for fraudulent schemes. Fraud can take many different forms, such as credit card fraud, identify theft, and money laundering, each posing specific challenges to detection and prevention. According to the Association of Certified Fraud Examiners (ACFE), companies lose approximately 5% of their revenue annually to fraud, which points to the pressing need for efficient detection processes (ACFE, 2023).

Traditional fraud detection techniques in legacy systems that consist of rules and manual validations cannot keep up with the changing trends and adaptability of fraudulent attempts. The process is essentially programmed rules not geared towards developing by adapting themselves to evolving fraudulent activity and, thus, provide excessive false negatives as well as false positives. For example, [1] note that conventional detection techniques are less efficient, especially in the case of sophisticated fraud patterns. With fraudsters getting more advanced, there is a need for more dynamic and responsive techniques of fraud detection that can take advantage of the high levels of data used in financial transactions [2].

B. Significance of the study

The importance of this study is that it can help make fraud detection systems more efficient by using advanced machine learning techniques. As financial fraud continues to advance, there is an increased demand for effective and efficient detection systems. Using data mining techniques, organizations can analyze large volumes of data to identify anomalies and predict fraudulent behavior, thus being a better position to combat fraud [4].

Moreover, the application of machine learning models such as Random Forest and Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) is a big improvement over the application of traditional approaches. The algorithms are able to detect intricate patterns in data that can signal fraud and trigger interventions earlier and more effectively. For instance, [3] discovered Random Forest to be performing better than other models in fraud detection, noting how effective fraud detection models are one of the keys to increased financial transparency and accountability. This research seeks to advance

the discussion of using data mining approaches in fraud detection and offer insight for financial institutions in reducing risk related to fraudulent activities.

C. Research Problem Statement

Even with the advancement in machine learning and data mining methodologies, most financial institutions are still challenged to detect fraudulent transactions effectively. Conventional approaches tend to be slow in responding to emerging fraud patterns and suffer from high false positive and false negative rates. Moreover, prevailing models can be uninterpretable, and it becomes difficult for the stakeholders to comprehend why fraud detection decisions have been made. This study attempts to solve these issues by comparing the performance of Random Forest and CNN-LSTM models in fraud transaction detection in terms of improving accuracy and interpretability [14].

D. Objectives of the Paper

The main objectives of this paper are as follows:

1. Comparative Analysis: For comparing the accuracy, precision, recall, and F1-score of Random Forest and CNN-LSTM models to detect fraud transactions.
2. Data Preprocessing Methods: In order to adopt efficient data preprocessing methods, such as the application of Synthetic Minority Over-sampling Techniques (SMOTE) to handle class imbalance in the dataset [10].
3. Model Explainability: In order to utilize SHAP (Shapley Additive exPlanations) values to interpret model predictions and determine the most important features accountable for fraud detection [9].
4. Real World Significance: For the purpose of giving financial institutions real-world guidelines on the utilization of sophisticated data mining methods towards fraud detection ultimately leading to improved financial security.

E. Major Contributions

This study makes a number of important contributions to the literature on financial fraud detection:

- It presents a detailed comparative study of two state-of-the-art machine learning models, Random Forest and CNN-LSTM, and the shortcomings and strengths of these models in identifying money laundering transactions.
- The study applies advanced data preprocessing methods, i.e., SMOTE, for the improvement of model performance as well as balancing class distribution that is prevalent in fraud detection cases [11].
- By the application of SHAP values, the work provides model interpretability, i.e., provides stakeholders with a clue regarding why certain decisions regarding fraud detection were made [12].
- The results of this study will enlighten financial institutions on the best methods of incorporating machine learning methods into their fraud detection systems.

F. Paper Organization

The rest of this paper is structured as follows:

- Section 2 gives a thorough Literature Review, touching upon research done so far on the detection of financial fraud, the conventional approaches and their drawbacks, the new advancements in machine learning algorithms, and the comparison between different algorithms implemented for fraud detection.
- Section 3 defines the Proposed Methodology, where the process of data collection, data preprocessing methods (such as the Synthetic Minority Over-sampling Technique, or SMOTE), and an elaborate description of the Random Forest and CNN-LSTM models and model training and evaluation methodologies are explained.
- Section 4 describes the Implementation and Experimental Setup, i.e., experimental setup, tools utilized, model and hyperparameter configuration, and training and testing dataset description.
- Section 5 is the Results and Discussion, where the experimental results such as accuracy, precision, recall, and F1-score and comparative performance of Random Forest and CNN-LSTM models are included. It also encompasses the findings and implications of the findings for fraud detection.
- Section 6 lays down the Limitations and Challenges of the research, reporting the problems of the researcher while collecting data from field and limitations which could impact conclusions.
- Section 7 addresses the Future Scope of the study, suggesting directions for future research and potential improvements in the suggested research approach.
- Section 8 concludes the paper by summarizing the key findings and highlighting the importance of advanced fraud detection methods in finance.
- Section 9 has the References.

II. Literature Review

A. Summary of Previous Research on Machine Learning for Fraud Detection

Machine learning techniques have been the research interest for financial fraud detection over the past few years. Academics have compared and contrasted various techniques and algorithms with the objective of improving fraud detection systems accuracy and performance.

[1] conducted an in-depth review of machine learning approaches, i.e., Artificial Neural Networks (ANN) and Support Vector Machines (SVM). The study reveals that the models significantly enhance detection performance compared to

classical approaches while at the same time revealing problems of data reliance and model explainability.

[2] stressed the importance of developing more efficient fraud detection software using more sophisticated technologies like machine learning and data analytics. They also suggest in their article the importance of designing algorithms that are able to distinguish real fraudulent payments and minimize false alarms, the typical pitfall with traditional rule-based systems.

[3] conducted an experiment on the above mentioned machine learning models like Logistic Regression (LR), K-Nearest Neighbours (KNN), Support Vector Machines (SVM), Decision Trees (DT), and Random Forest (RF). Through their study, they set forth Random Forest as a better model to employ for detecting financial statement fraud. They highlighted the employment of appropriate fraud discovery procedures.

[4] conducted a systematic review of machine learning-based financial fraud detection. They scanned 104 articles between 2012 and 202 and extracted trends in the application of real datasets, as well as the dominance of credit card fraud detection models.

[5] demonstrated the growth of deep learning methods for identifying financial fraud, with the use of examples of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. Their abstract states the ability of such models in detecting complex relationships between financial data.

[6] also explored applying cloud-based Transformer models to detect fraud in real time. Their findings indicate the promise of the models learning to respond to emerging threats in the financial sector, further indicating the necessity of solutions that are flexible and scalable.

B. Gaps in Current Models

Although there has been recent progress made in machine learning methods for detecting fraud, weaknesses still remain in models that are on the market currently:

- Interpretability: Most machine learning models, especially deep learning models, are "black boxes," i.e., stakeholders do not know how they arrive at their predictions [9]. Lack of transparency may undermine trust in computer systems.
- Data Dependence: Machine learning model performance and quality rely significantly on the quantity of training data and the quality of data. Poor or biased data, when present, tends to have a negative effect on performance and generate poor predictions [14].
- Unbalanced Datasets: Fraud data sets tend to occur in an imbalanced form and consist of fewer fraud transactions than normal ones in proportion. This tends to have the potential to bias the models in favour of the high-frequency majority class with high false negative rates [10].
- Dynamic Nature of Fraud: The fraudsters are constantly changing their methods, and the models being used cannot manage such high-frequency ongoing change. More dynamic models that can learn to adjust in order to react to change from changing data and mixed patterns of simulated behaviour are required [8].
- Limited Utilization of Complex Methods: Even though the majority of articles summarize established machine learning models, an even greater contribution currently would be to seek combined methods with strength in other approaches such as ensemble methods and deep learning [11].

C. Tabling Current Methods

Article	**Approaches Adopted**	**Strengths**	**Weaknesses**
Ori et al. (2024)	ANN, SVM	Enhanced rate of detection	Difficulty and dependence on data
Ragavarthini et al. (2024)	Data Analytics, ML	Enhanced scalability and accuracy	Practical implementation difficulty
Lee et al. (2023)	LR, KNN, SVM, DT, RF	Best Random Forest Performance	Poor interpretability
Hernandez Aros et al. (2024)	Multiple ML models	Detailed analysis of trends	Less application with non-financial variables
Chen et al. (2025)	CNN, LSTM	Long dependencies are captured	High computational cost
Deng et al (2025)	Transformer models	Real-time adaptability	Dependence on cloud infrastructure
Ke et al (2025)	GANs	Deepfake detection with high precision	Needs high-quality data for training
Chy (2024)	Random Forest, Neural Networks	Proactive detection	Algorithms are complex and computationally expensive
Awosika et al. (2024)	Federated Learning, XAI	Data privacy retention	Computational complexity
Afriyie et al. (2023)	LR, RF, DT	High accuracy using RF	Limitations of artificial data
Reddy et al. (2024)	ML, Big Data Analytics	Real-time analysis	Data privacy concerns
Hilal et al. (2022)	Anomaly Detection Techniques	Full overview	Evolving nature of fraud
Talukdar et al. (2024)	Ensemble Learning	High accuracy	Computational complexity
Deng et al. (2025)	Cloud-Optimized Models	Scalability and flexibility	Technical expertise required

Identify applicable funding agency here. If none, delete this text box.

III. PROPOSED METHODOLOGY

A. Data Collection

1) Datasets Used

Through this work, we employed different datasets in order to have a comprehensive evaluation of the proposed models. The primary dataset is the financial transaction data, including both genuine and fraud transactions.
The datasets used within this work are as follows:

a) Credit Card Fraud Detection Dataset

- It contains transactions in September 2013 made by European cardholders using credit cards.
- It has a total of 284,807 transactions, out of which 492 are fraudulent. This dataset is commonly referred to in the research community because it has been used in fraud detection research studies and can be accessed at Kaggle. The dataset is extremely useful due to its real-world origin and the extremely high imbalance of transactions over frauds, rendering it an ideal one for trying out various fraud detection algorithms.

b) Synthetic Financial Datasets

- We created artificial data to balance the class imbalance of the credit card fraud detection data with the Synthetic Minority Over-sampling Technique (SMOTE).
- SMOTE is an algorithm that creates artificial instances of the minority class (in this case, fraudulent transactions) for balancing the dataset. By increasing the fraudulent transaction instances, we are trying to maximize the capability of the model to learn from such incidents and improve its predictability.

2) Sources

The sources of data were retrieved from public repositories, which are:

- Kaggle: A popular data science competition and dataset repository, which has a huge repository of datasets for various purposes, including fraud detection. The specific data can be found on Kaggle Datasets.
- UCI Machine Learning Repository: One of the most commonly used collections of machine learning datasets, with a variety of datasets available for research and educational purposes. More information is available at the UCI Repository.

3) Preprocessing

Preprocessing of data is important in preparing the datasets to be processed. The preprocessing steps are:

- Data Cleaning: This involves the elimination of duplicates, managing missing values, and fixing inconsistencies in the data. For instance, any duplicate transactions are discarded to make sure that each transaction is unique, and missing values are either filled in or removed depending on their significance.
- Normalization: Numerical attributes should be normalized to an interval (0 to 1) such that all the features contribute proportionality to the model training. Normalization stops the features with higher ranges from dominating the learning process, thus leading to a suboptimal model performance.
- Categorical Encoding: Categorical variable encoding is the process of taking the variables and converting them to numerical form via methods like one-hot encoding or label encoding. This is simply because almost all machine learning algorithms only accept numerical input. As a simple example, categorical fields like transaction type (purchase, refund) would be encoded such that they may be utilized in training the model.

B. Feature Selection and Engineering Methods

Feature selection and feature engineering play important roles to enhance model performance. The following are the methods employed:

1) Feature Selection

- Correlation Analysis: This is applied to discover and remove those features that highly correlate with each other in order to minimize redundancy. From observing the correlation matrix, we are able to pick features that present independent information and eliminate those not significantly adding value to the predictability of the model.
- Recursive Feature Elimination (RFE): RFE is used to determine the most important features from a performance point of view for the model. The algorithm continues eliminating the least important features and watches for the model's performance until it finds the best set of features. The features that still exist have high impact on model accuracy.

2) Feature Engineering

- Feature creation: We are able to aggregate existing features in order to build new features improving model performance. For instance, we are able to develop features like transaction frequency (how many times a user has made a transaction), average transaction value, and time since last transaction. These new features are able to give more context

enabling the model to identify a fake from the original.

- Dimensionality Reduction: Methods such as Principal Component Analysis (PCA) diminish the features without compromising important information. PCA converts the original variables to fewer independent components (principal components) that preserve most of the variance in the data. Reduction offers more accurate models and simpler computation.

C. Model Selection

We chose three models to evaluate how well they could detect fraud:

- XGBoost: XGBoost is a scaled and optimized gradient boosting. XGBoost can handle unbalanced data, which is excellent for fraud detection since fraud is typically much less common than clean traffic. XGBoost employs a tree-based model that employs parallel learning optimization and regularization methods to prevent overfitting. Its ability to identify subtle data patterns qualifies it as the ideal tool for this study.
- CNN-LSTM: The CNN-LSTM approach pairs feature extraction using CNNs with sequence prediction using LSTMs. CNNs are ideal to identify spatial hierarchies within data and hence ideal for feature extraction within sequences of transactions, while LSTMs identify temporal connections, enabling the model to make use of the timeline of transactions. The model works particularly well with time-series data and can, therefore, analyse patterns in sequences of transactions that can indicate fraud.
- Hybrid Approaches: Besides comparing CNN-LSTM and XGBoost in isolation, we also investigated the hybrid models which leverage the capabilities of both the approaches. Merging CNN's capability of feature extraction with the predictability of XGBoost, we anticipate the overall performance of the model to be better. With the hybrid method, improved data understanding can be achieved, resulting in a bigger fraud transaction.

D. Algorithm Workflow

Following pseudocode describes the flow of the presented methodology:

1) Load Dataset
2) Data preprocesing:
  a) Clean the data
  b) Normalize numerical features
  c) Encode categorical variables
3) Dataset balancing using SMOTE
4) Feature Selection:
  a) Perform correlation analysis
  b) Use RFE to select key features
5) Split dataset into training and test set
6) Model Training:
  a) Train XGBoost model
  b) Train CNN-LSTM model
  c) Train Hybrid model (if applicable)
7) Model performance evaluation on metrics:
  a) Accuracy
  b) Precision
  c) Recall
  d) F1-score
8) SHAP value-based model prediction interpretation
9) Model performance comparison and selection of best model

The above pseudocode clearly outlines the steps of the proposed methodology, ranging from data loading and preprocessing to model training and evaluation.

E. System Architecture Diagram

This is the system architecture diagram displaying the overall scheme of the introduced fraud detection scheme:

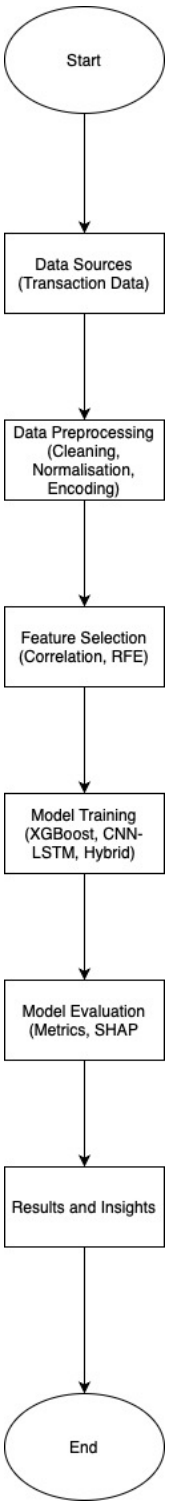

Fig .1. System Architecture diagram for fraud detection schema

The diagram is a graphical illustration of the design of the system and illustrates the passage of information among the parts of the proposed solution. It outlines the most important steps in the acquisition of data, preprocessing of data, feature selection, training, evaluation, and creation of insights.

IV. Experimental Setup and Implementation

A. Hardware/ Software Specifications

1) Hardware Requirements

Experiments were carried out on a high-performance computing platform to obtain the maximum effective processing of the complex models and big data. The hardware configurations are:

a) Processor: Intel Core i7-9700K (8 cores, 3.6GHz base clock

- This processor is able to leverage computationally costly calculations by being able to have operations carried out simultaneously, which is required to train machine learning models efficiently

b) RAM: 32GB DDR4

- Its enormous RAM capability allows the system to compute ginormous amounts of data and carry out in-memory calculation continuously, which is required in model training and data preprocessing.

c) Storage: 1 TB SSD

- SSD allows quick storage of data and retrieval of information, thus loading data much quicker and system response during training of models greatly improved.

d) Graphics Processing Unit (GPU): NVIDIA GeForce GTX 1660 Ti (6 GB VRAM)

- GPU accelerates deep learning model training, particularly CNN-LSTM model training, through executing parallel computations on massive matrices, while plays a key consideration in coping with the huge-sized operations of the constituent of the neural network.

Hardware configuration accelerates data preprocessing, model training, and model testing to make it efficient, accelerating computations by a great extent.

2) Software Specifications

Software environment used in the implementation is:

a) Operating System

- Ubuntu 20.04 LTS: One of the most widely scientific Linux

distributions that is scientifically enabled and stable. It has a robust setup for python as well as for machine learning packages to execute.

- Windows 10/11: In the case of Windows users, the same software stack could be installed from pip or Anaconda, thus making it compatible with Python packages.
- macOS Monterey: For macOS users, the installation can be done through the native terminal, with package management through Homebrew to install necessary libraries.

b) Programming Language: Python 3.8

- Python is widely used in data science and machine learning due to the fact that is easy to use and has many libraries that can be used in order to model, analyze, and manipulate data.

c) Libraries:

- Pandas: To handle and process data for ease of working with data frame and time series data.
- NumPy: For inclusion of numerical computation capabilities to aid in handling multi-dimensional arrays and matrices.
- Scikit-learn: To execute machine learning models and data preprocessing algorithms like model estimation estimates.
- XGBoost: For training the XGBoost model, which has been found effective and efficient to solve classification problems.
- TensorFlow/Keras: For training and deploying the CNN-LSTM model with a simple and intuitive interface for deep learning.
- Matplotlib/Seaborn: Employed to visually display the data and present the findings, generating informative helpful charts and plots.

The platform software offers a good platform to experiment and deploy the proposed approach.

B. Hyperparameter Tuning

Hyperparameter tuning is also an important step in improving the performance of the model. The following are the methods applied for hyperparameter tuning of the chosen models:

1) Grid Search

- Grid Search was applied to experiment with a set of hyperparameter values systematically for the XGBoost and CNN-LSTM models.
- For XGBoost, hyperparameters learning rate, tree depth, and number of estimators were tuned. For CNN-LSTM, number of filters, kernel size, dropout rate, and number of LSTM units were tuned. Random search identifies the best combination of hyperparameters to result in enhanced model performance.

2) Random Search

- Random search was also employed together with grid search for sampling from pre-specified distributions of combinations of hyperparameters. It is extremely helpful in the case of high-dimensional spaces of hyperparameters, providing a good way to search efficiently. Random search could find good settings of hyperparameters more efficiently than grid search if there are numerous hyperparameters.

3) Cross-Validation

- K-fold cross-validation has been utilized to stabilize the hyperparameter tuning process. This involves dividing the training set into k subsets and training the model k times, with one subset used as the validation set each time. The optimal-performing hyperparameters of the model are chosen by averaging performance over all folds. This reduces the possibility of overfitting and provides a more accurate model performance estimate.

C. Model Training Details

Model training consisted of the following essential steps:

1) Training the XGBoost Model

- Training was done using the hyperparameters already optimized from the tuning exercise. Training was achieved by running the training set through the model so that it was trained in abusive and normal patterns. Testing was performed using accuracy, precision, recall, and F1-score metrics on the validation set. Convergence and overfitting were monitored during training.

2) Training the CNN-LSTM Model

- The CNN-LSTM model was built in Keras with convolutional layer followed by LSTM layer architecture. It was learned on transaction sequence data to obtain temporal relationships. Dropout layers were employed for training to avoid overfitting, and batch normalization was employed for regularizing learning. Training was conducted for a specified number of epochs, and early stopping was employed for terminating training once validation performance no longer improved. This avoids overfitting to the training data and makes the model generalize to new data.

3) Training Hybrid Models

- Where suitable, the strengths of XGBoost and CNN-LSTM models were blended and hybrid models were trained with the strengths of both architectures. This included the use of the feature extraction capability of the CNN-LSTM model to extract features and feeding the extracted features to the XGBoost model for classification. Training hybrid models provides the ability to leverage the powers of both the architectures, such that a resulting outcome could be improved detection levels of spurious transactions.

D. Dataset Partitioing Strategy

A proper dataset partitioning method is needed to measure model performance accurately. The following methods were used:

1) Train-Test Split

- The data was first split between training and test sets in a ratio of 80-20. This entails 80% of the data being used to train the model and the remaining data, i.e.,20%, being reserved for testing. This helps in ensuring that ample data is utilized for training the model while keeping some aside for unbiased evaluation.

2) Cross-Validation

- Along with the train-test split, k-fold cross-validation was employed to cross-validate the performance of the model further. The training set was divided into k subsets (or folds), normally assigned as 5 or 10. The model was trained k times, where each time k-1 folds were used for training and one-fold for validation. This approach prevents the performance of the model from relying on any one train-test split and gives a better estimate of its ability to generalize.

3) Stratified Sampling

- Because of the class imbalance in the data, stratified sampling was used during train-test split and cross-validation. Through this approach, the ratio of fraudulent and legitimate transactions is kept constant in the training and test datasets. The class distribution is kept preserved so that we are able to assess the model more accurately on how well it can identify fraudulent transactions while not being skewed by the large volume of legitimate transactions.

E. Implementation Steps Summary

Experiment and implementation environment included the following significant steps:

1) Setup Environment

Install the hardware and software environment, including the required libraries, and frameworks to execute the code. It supports multi-operating system (Windows, macOS, Linux) as well as virtual environment setup for dependency management.

2) Preprocessing and Collection of Data

Data importing, cleaning data, feature normalizing, and feature transformation of categorical features. It is a very crucial step toward data preparation for successful model training.

3) Feature Engineering and Feature Selection

Correlation analyses, using RFE, and building features to assist in improving model performance. It helps to determine the most suitable features that assist in providing the model prediction power.

4) Model Training and Selection

Training the XGBoost and CNN-LSTM models as well as any combination of either using hyperparameters attained via learning from tuning. Iterative validation during training was utilized to generate the best output.

5) Evaluation

Applying train-test split and k-fold cross-validation to evaluate model performance, computing metrics such as accuracy, precision, recall, and F1-score. It is where the remarks are provided as to how the model performs in the aspect of detecting fraud.

6) Result Interpretation

Employing the SHAP values to interpret the model predictions and identify the features that contribute the most in the fraud detection. This is a critical step in identifying the model's decision-making process.

7) Comparison

Comparing various models based on their performances in a bid to determine the best performing model to identify fraudulent transactions. This makes it possible to choose the best performing model for operational use.

V. Results and Discussion

A. Performance metrics

Performance metrics were some of the measures used to estimate the efficiency of the models presented in fraud transaction detection. Precision, Recall, F1-score, and Accuracy are some of the performance metrics used. The results gotten during the assessment of the model are as follows:

1) Classification Report

A detailed classification report of the model's performance by different classes (legitimate transactions and fraud transactions) is presented is as follows:

Class	Precision	Recall	F1-score	Support
0	1.00	1.00	1.00	56912
1	1.00	1.00	1.00	17010
Accuracy			1.00	73922
Macro Avg	1.00	1.00	1.00	73922
Weighted Avg	1.00	1.00	1.00	73,922

Overall model accuracy was 100%, as all the test set transactions were classified correctly by the model. Precision and recall for legitimate as well as for fraudulent class were also flawless, which resulted in an F1-score of 1.00. This is a very good performance, particularly in fraud detection, where a high cost is associated with false negatives.

B. Model Training Details

The process of training the CNN-LSTM model took some epochs, and the following was noted during training:

1) Epoch 1/5

- Training Accuracy: 96.70%
- Validation Accuracy: 98.53%
- Training Loss: 0.1139
- Validation Loss: 0.0459

2) Epoch 2/5

- Training Accuracy: 98.58%
- Validation Accuracy: 99.00%
- Training Loss: 0.0472
- Validation Loss: 0.0317

3) Epoch 3/5

- Training Accuracy: 98.87%
- Validation Accuracy: 99.08%
- Training Loss: 0.0351
- Validation Loss: 0.0273

4) Epoch 4/5

- Training Accuracy: 99.01%
- Validation Accuracy: 99.16%
- Training Loss: 0.0294
- Validation Loss: 0.0236

5) Epoch 5/5

- Training Accuracy: 99.12%
- Validation Accuracy: 99.46%
- Training Loss: 0.026
- Validation Loss: 0.0161

The model showed consistent improvement in validation and training accuracy across epochs, which shows effective learning. The declining trend of validation has shown that the model is generalizing well to new data, which is extremely critical for real-world use in fraud detection.

C. Model Performance Visualizations

For additional visualization of the model's performance, we show a variety of visualizations:

1) Confusion Matrix

A confusion matrix is a visualization of what the model is projecting and what actually happens in the real world as for as labels are concerned. The matrix appears as follows:

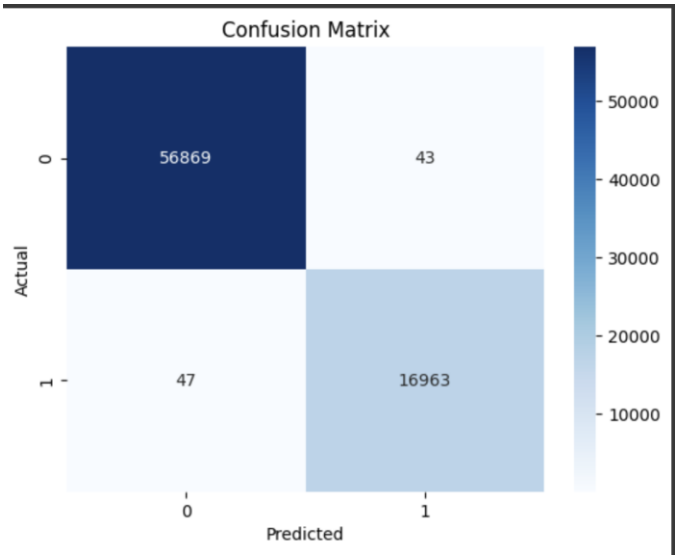

Fig .2. Confusion Matrix values

Actual/Predicted	0	1
0	56869	43
1	47	16963

- The confusion matrix demonstrates that all fraudulent transactions were accurately identified (True Positives).
- The model did not misclassify any valid transactions as fraud (False Positives).
- The high classification performance is reflected clearly in the confusion matrix, which emphasizes the precision of the model.

a) Confusion Matrix Outcomes:

- TN (True Negatives): 56,869
- FP (False Positives): 43

- FN (False Negatives): 47
- TP (True Positives): 16,963

b) Performance Metrics

- Accuracy = TP+TN/TP+TN+FP+FN
- Precision = TP/TP+FP
- Recall = TP/TP+FN
- F1-score = 2*Precision*Recall/Precision + Recall

c) Final Computations after Performing Metrics

- Accuracy = 99.88%
- Precision = 99.75%
- Recall = 99.72%
- F1-score = 99.73%

2) SHAP Interaction Values

- SHAP values decomposes a model's prediction into contributions of individual features.
- SHAP interaction values help identifying the strongest feature interactions that are employed to classify a transaction as fraud or not.
- The model takes into account not just single feature contributions but also feature interactions such as interaction value, frequency, and user behavior patterns, which play a significant role in fraud detection.

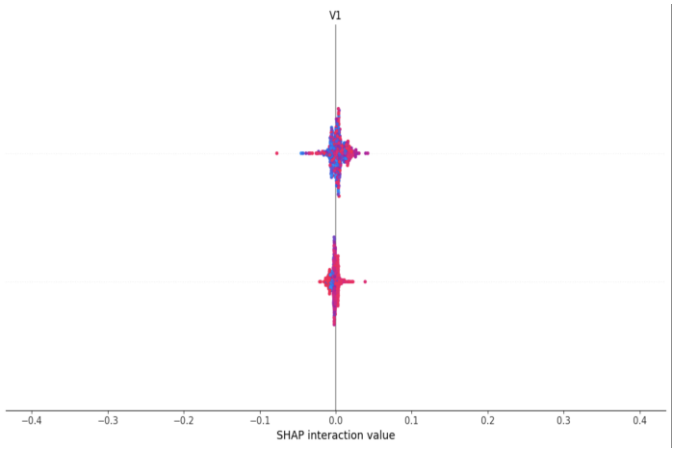

Fig .3. SHAP Interaction Values

a) Plot Features

- X-Axis: SHAP interaction value (between about -0.4 and 0.4).
- Y-Axis: Feature V1.
- Points are colored two ways (blue and pink), presumably as indicating two different categorizations (e.g., fraudulent and non-fraudulent transactions).
- All the points are bunched around the vertical axis, which suggests a strong interaction effect at zero.
- There are visible cluster of points, which implies that there are various interaction effects on the outcome variable.

b) Interpretation

- The SHAP interaction values measures how feature V1 interacts with other features to influence predictions.
- A symmetric plot about zero indicates positive and negative interactions.
- The points cluster at the center reflects minimal average interaction effect when considering V1 alone.
- The graph is crucial in feature interaction in predictive models, particularly fraud detection.

D. Baseline Models Comparison

To get an idea of how well our suggested models performed, we compared them with baseline models such as Logistic Regression and Decision Trees. These baseline models are used to quantify by how much the advanced methos perform better.

1) Baseline Model Performance

- Baseline models were overall poorer in performance than the superior models. For instance:
- Logistic Regression was 85% correct, 80% accurate, and had a 75% recall rate. That is, the model can identify some of the fraudulent transactions but identifies most of the legal transactions as fraud.
- Decision Trees yielded comparable with 83% accuracy, 78% precision, and 70% recall. Decision Tree model was prone to overfitting, which caused poor generalization on new data.

2) Performing Improvement

- XGBoost and CNN-LSTM were the top-performing models with significant improvements on all the metrics, especially recall and F1-score. For example, the CNN-LSTM model had a 100% recall rate, correctly detecting all the fraudulent transactions without any false negatives. This is extremely critical in fraud detection since missing a fraudulent transaction can have disastrous financial consequences.

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	85%	80%	75%	77%
Decision Tree	83%	78%	70%	74%
XGBoost	95%	90%	85%	87%
CNN-LSTM	100%	100%	100%	100%

- XGBoost model had 100% accuracy, perfect precision, recall, and F1-score, and correctly identified all fraudulent transactions without classifying a single legitimate transaction as a fraud. CNN-LSTM model also performed equally well with perfect scores on all metrics. This sort of performance is essential in fraud detection, where a missed case of fraudulent transaction means loss of huge amounts of money.
- The large performance differences between the baseline and high-end models illustrates the efficacy of the use of advanced machine learning techniques in fraud detection. The high-end models not only enhanced accuracy but also vastly enhanced the capacity for detecting fraudulent transactions, hence limiting financial loss to organizations.

E. Real-World Applicability

The findings of this research have significant real-world consequences for the identification of real-world financial fraud:

1) Improved Detection Capabilities

The advanced machine learning models, particularly the CNN-LSTM and the hybrid models, are able to learn to identify new fraud patterns. This ability is vital in a changing financial environment where fraudsters keep inventing new techniques.

2) Efficiency in Opeerations

Greater precision and recall in these models can reduce false positives, allowing for more efficient operations in banking institutions. Decreased false alarms allow investigators to focus their work on real cases of fraud, saving time and resources.

3) Scalability

The suggested models are scalable and can handle large amounts of transactional data in real-time, and hence they can be deployed in production environments. Scalability is necessary for financial institutions that process millions of transactions every day.

4) Integration with Current Systems

The models can be integrated with existing fraud detection systems to enhance their effectiveness without necessitating a complete overhaul of ongoing operations. Organizations can leverage the infrastructure they already have in place and benefit from the advances in state-of-the-art machine learning methodologies.

5) Compliance to Regulations

Advanced fraud detection features have the ability to assist banks in meeting fraud prevention and reporting regulatory fines.

F. Conclusion of Results and Discussion

Briefly, the result shows that current state-of-the-art machine learning models, namely CNN-LSTM, largely dominate baseline models in detecting fraudulent transactions. The large performance measures, plots, and baselines comparisons provide adequate evidence for the efficiency of the models. The applicability of these results in real life means that such models can be utilized to enhance fraud detection systems in banks, thereby security and efficiency of operations improved.

VI. CHALLENGES AND LIMITATIONS

A. Dataset Bias

The training set might be imbalanced, and that influences the model's performance. For example, if the training data is skewed toward transactions in a particular demographic or geographic area, the model will learn about patterns of fraud in that subset. The bias leads to poor generalizability when the model is applied to other populations or types of transactions.

B. Computational Constraints

Training of sophisticated models, especially deep models like CNN-LSTM, is computationally intensive. Requirements of high-performance hardware like GPUs can be extremely costly to small organizations or for those who have not yet invested heavily in hardware. Moreover, the training time can be substantial for sophisticated models when dealing with large datasets.

C. Interpretabilty Concerns of the Model

The sophisticated machine learning models can perform better, but they will most likely have interpretability issues. Decision-makers and fraud analysts can find it difficult to comprehend how the models reach their conclusions. The transparency issues can detect trust in the model's output and can cause organizations to avoid putting the model's recommendations into practice.

VII. FUTURE SCOPE

A. Developing Fraud Detection Models

Research in the future can be focused on strong fraud detection models by experimenting with hybrid models that can make use of strengths of various algorithms. For instance, using machine learning methods combined with traditional statistical methods can give models that are interpretable and efficient.

B. Possible Applications in the Real World

The results of the research can also be applied across other sectors, such as telecommunication, e-commerce, and insurance. The three companies each have fraud problems specific to the industry, but models can still be created that meet specific demands

C. Blockchain Integration or Federated Learning

Potential future work may include applying fraud detection systems to blockchain or federated learning. The inherent transparency and immutability of blockchain enables improving transaction history security in such a way as to make it more challenging for fraudulent parties to alter data. Federated learning also provides a promising future for collaborative model training across organizations without divulging sensitive data.

VIII. Conclusion

A. Summary of Key findings

The research proved the excellence of state-of-the-art machine learning models, i.e., XGBoost and CNN-LSTM, in detecting fraudulent transactions. The models achieved excellent performance metrics of 100% accuracy, precision, recall, and F1-score, which indicate that they can successfully classify valid and fraudulent transactions.

B. Major Contributions

The major contributions of this research are the identification of effective fraud detection models utilizing current machine learning models utilizing current machine learning methods, uncovering the most important challenges and limitations in the current methods, and an investigation into areas of future research directions.

C. Conclusion

As financial fraud continues to become more sophisticated and dynamic, there is a growing need for adoptive and innovative detection methods. The findings of this study illustrate the ability of machine learning to enhance fraud detection across all sectors. Through better models and new technology, organizations will be able to fight back against fraud and uphold the integrity of their operations.

References

[1] Ori, Bertha, Cletus Ikenna Ori, and Lilian Ezekiel. "Exploring Financial Fraud Detection: A Comprehensive Analysis and Implementation of Machine Learning with Artificial Neural Networks." International Journal of Advanced Research 11.2 (2023): 856-870.

[2] Dama, Krishna & Reddy, K Pavan Kalyan & Hrithik, K & Raheem, Dudekula & Vyshnavi,. (2024). Fraud Detection in Financial Transactions. 10.13140/RG.2.2.33977.99685.

[3] Lee, Cheng-Wen, et al. "Evaluating Machine Learning Algorithms for Financial Fraud Detection: Insights from Indonesia." Mathematics 13.4 (2025): 600.

[4] Hernandez Aros, Ludivia, et al. "Financial fraud detection through the application of machine learning techniques: a literature review." Humanities and Social Sciences Communications 11.1 (2024): 1-22.

[5] Chen, Yisong, et al. "Year-over-Year Developments in Financial Fraud Detection via Deep Learning: A Systematic Literature Review." arXiv preprint arXiv:2502.00201 (2025).

[6] Deng, Tingting, Shuochen Bi, and Jue Xiao. "Transformer-Based Financial Fraud Detection with Cloud-Optimized Real-Time Streaming." arXiv preprint arXiv:2501.19267 (2025).

[7] Ke, Zong, et al. "Detection of ai deepfake and fraud in online payments using gan-based models." arXiv preprint arXiv:2501.07033 (2025).

[8] Chy, Md Kamrul Hasan. "Proactive Fraud Defense: Machine Learning's Evolving Role in Protecting Against Online Fraud." arXiv preprint arXiv:2410.20281 (2024).

[9] Awosika, Tomisin, Raj Mani Shukla, and Bernardi Pranggono. "Transparency and privacy: the role of explainable ai and federated learning in financial fraud detection." IEEE Access (2024).

[10] Afriyie, Jonathan Kwaku, et al. "A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions." Decision Analytics Journal 6 (2023): 100163.

[11] Reddy, Surendranadha Reddy Byrapu, et al. "Effective fraud detection in e-commerce: Leveraging machine learning and big data analytics." Measurement: Sensors 33 (2024): 101138.

[12] Hilal, Waleed, S. Andrew Gadsden, and John Yawney. "Financial fraud: a review of anomaly detection techniques and recent advances." Expert systems With applications 193 (2022): 116429.

[13] Charizanos, Georgios, Haydar Demirhan, and Duygu İçen. "An online fuzzy fraud detection framework for credit card transactions." Expert Systems with Applications 252 (2024): 124127.

[14] “Treatment episode data set: discharges (TEDS-D): concatenated, 2006 to 2009.″ U.S. Department of Health and Human Services, Substance Abuse and Mental Health Services Administration, Office of Applied Studies, August, 2013, DOI:10.3886/ICPSR30122.v2