Transaction Fraud Detection System using DL & ML

P. P. Gadekar[1], Harde Gayatri Laxman[2], Kale Pratiksha Devidas[3], Shinde Shraddha Vikas[4], Thorat Anjali Dnyandev[5]

[1]Professor, Dept. of Computer Technology, P.Dr.V.V.P. Institute of Technology and Engineering, Loni, Maharashtra, India

[2,3,4,5] Final year Diploma Student, P.Dr.V.V.P. Institute of Technology and Engineering, Loni, Maharashtra, India

---***---

ABSTRACT - Over the last several years, there has been a constant rise in the number of transactions. Credit cards, electronic money transfers, and the United Payments Interface (UPI) have all contributed to the explosion in the volume of online financial transactions. There has also been a steady rise in the number of con artists that conduct fraudulent financial transactions. Financial institutions have implemented a plethora of fraud detection measures, but even the most cunning scammers manage to evade these systems. Consequently, this study lays out a method for detecting fraudulent transactions using Linear Clustering, Long Short Term Neural Networks, and Decision Making. In order to quantify the strategy that produced the desired results, a thorough study was conducted.

Keywords: Transaction Fraud, Decision Making, Linear clustering.

1.INTRODUCTION

With the startup ecosystem evolving at a quick pace and investment decision-making become more complex, machine learning-based startup investment suggestion systems are becoming more significant in the current context. It is becoming increasingly difficult for investors to sift through the thousands of new companies appearing in fields as varied as sustainability, healthcare, fintech, and technology in search of opportunities with both high growth potential and reasonable risk. Conventional approaches to evaluating investments largely depend on labor-intensive, subjective, and bias-prone manual analysis, expert intuition, and sparse historical comparisons. Financial metrics, market trends, business models, founder histories, consumer traction, funding history, and machine learning-based startup investment suggestion systems all contribute to data-driven decision-making. This aids in the early detection of promising businesses in extremely competitive and unpredictable markets, helps investors make consistent, well-informed investment decisions, and lowers financial risk.

To find promising new businesses to back, this approach employs a number of machine learning algorithms. First, data is gathered from a variety of sources, including startup databases, financial documents, platforms for market research, and records of past investments. To make sure the data is accurate and reliable, it is preprocessed through stages like data cleansing, normalization, outlier elimination, and addressing missing information. Revenue growth rate, burn rate, scalability, industry sector, funding stage, and management team experience are among the main indicators that are identified using feature extraction and selection methodologies. Investment results, success probabilities, and expected returns can be predicted using supervised learning methods like classification and regression. On the other hand, clustering and other unsupervised learning approaches can be used to group startups with similar traits and risk profiles. Prioritized investment recommendations based on investor tastes and comfort level with risk are then generated by applying ranking and recommendation algorithms.

Because of its capacity to represent complicated and non-linear correlations between input characteristics and investment results, the LSTM model is pivotal to the startup investment recommendation system. A normalized set of starting qualities is sent into the LSTM's input layer, which then uses activation functions to conduct weighted computations. Finally, the output layer generates predictions, including investment appropriateness scores or success likelihood. In order to train the model to make accurate predictions, it is fed past investment data and its weights are adjusted using backpropagation. In contrast to more conventional forms of analysis, LSTMs are able to learn and discover previously unseen patterns and complex interplay between operational, financial, and market variables. A strong instrument for contemporary startup investment analysis, the LSTM-based model offers accurate, dependable, and dynamic investment recommendations because to its adaptability, scalability, and high predictive capability.

[1] According to Jaber Jemai et al., financial institutions can save a significant amount of money by detecting fraudulent credit card transactions. Despite the best efforts of financial stakeholders, that loss is only going up. This paper presents the results of a comparative research on the performance of ensemble approaches in identifying fraudulent or legitimate credit card transactions. Basic classification algorithms, such as naive Bayes, are beaten out by XGBoost and bagging methods. On actual datasets, nevertheless, they fail to account for overfitting. All classifiers perform worse on simulated data sets because to the lack of a priori script or distribution for data generation. Since bankers adhere to a rigid script that can be easily detected through transfer learning, this explains why ensemble approaches work so well on real data. This discovery raises concerns about a potential script vulnerability in the processing of credit card transactions. During the authentication phase, future works should use factors that are multiplied, varied, or randomly generated. Looking at the algorithms' decision-making process and how to make them more explainable and interpretable is an intriguing technological angle for future research.

[2] The detection of financial fraud is an ongoing difficulty with far-reaching consequences for the financial industry. Abdulwahab Ali Almazroi et al. suggested a major improvement in this area. Financial fraud is still quite complicated, even though technology is always becoming better. To address this critical issue, the author presents a novel RXT-J-based financial fraud detection model that is specifically tailored for analysis of real-time transaction data. Particularly impressive is the author's model's ability to deal with large datasets while maintaining its outstanding performance in addressing the intricacies of contemporary financial fraud. One notable feature is the model's superior performance compared to other solutions. It significantly improves detection accuracy and quickly identifies complicated fraudulent patterns that were previously unnoticed. In addition, the model proposed by the author successfully tackles the fundamental shortcomings of conventional methods. The author has compared their model to both traditional machine learning methods and other deep learning techniques using data on fraudulent financial transactions that occurred in real-time. Although the author's model has shown great promise, when more data becomes accessible, future research could incorporate elements like fraud location and timing analysis to make it even more effective. This research offers improved security and efficiency in financial transactions and is a major step forward in the fight against financial fraud. The author's work is crucial in protecting financial transactions from fraud within the larger framework of wireless communications defense, where new algorithms improve security, data availability, and resistance to interference.

[3] This paper, which was introduced by Hadi M. R. Al Lawati1 and Anazida Zainal et al., offers a structured framework for detecting drift and fraud, and its performance is assessed using multiple datasets, including the Credit Card, Spam, and Real datasets. Ultimately, a versatile and strong approach for ever-changing data environments can be achieved by combining Mutual Information with SelectKBest for feature selection, ADASYN for data balance, and improved drift detection using EDDM and ADWIN. With a perfect score of 1.000 and a drift detection rate of 1,000 on the Real dataset, the suggested framework proved to be very sensitive and specific for detecting fraud in real-world situations including mixed data and numerical data. While intri cate ensemble models are complex and difficult to understand, this method is straightforward and ideal for real-time applications. Future studies should focus on developing adaptation to shifting fraud patterns in the financial and non-financial sectors through the use of online learning.

An examination of earlier research that was deemed a Litereature Survey is presented in the second part of this publication. Section 3 provides a comprehensive description of the proposed methodology, outlining the path of action. The experimental evaluation is covered in Part 4, possible modifications are discussed in Section 5, and the essay concludes with a conclusion on the existing plan.

2. LITERATURE SURVEY

[4] Although many researchers remain skeptical that robots can be taught human-like skills, Fawaz Khaled Alarfaj et al. find that deep learning can replace humans. In an effort to curb money laundering, banks and other financial institutions throughout the world are integrating deep learning with BI. The author has conducted research on two topics: credit card anomaly detection using

autoencoders and graph neural network fraud detection. Many financial businesses, particularly banks, benefit from the author's study work since it helps them understand deep learning with business intelligence better. In the future, the author's study will also help Pakistani banks identify the difference between their present operational methods and new procedures that use AI and deep learning to identify fraud. The author presents credit card behaviors as a temporal transaction graph and defines the challenge as a semi-supervised node classification job in order to detect credit card fraud. Author is able to identify credit card fraud with the use of a novel attribute-driven temporal graph neural network. The author recommends a gated temporal attention network as a means of extracting attribute and temporal data. Attributes and risk information are passed on to the temporal transaction graph by making use of both labelled and unlabeled data. When used with other fraud detection systems, autoencoders can significantly enhance overall performance. To create a comprehensive fraud detection system, for example, autoencoder anomaly detection can be integrated with rule-based systems, machine learning classifiers, or algorithms for network analysis. With the right combination of supervised and unsupervised learning techniques, it is possible to build a deep learning-based fraud detection system and reduce the amount of human effort needed to manually classify datasets. One potential outcome of the model update is the incorporation of new fraud type detection capabilities. How the model is updated is dependent on the quantity of tagged data. To improve the company's operations, the author conducts research using Python for statistical analysis. Therefore, the author concludes that AI with deep learning in BI provides reasonable explanations for the different areas of business, and that the author can more effectively accomplish her goals by utilizing AI with deep learning in BI.

[5] To aid in the creation, evaluation, and interpretation of fraud-detection models, Nazerke Baisholan et al. presented FraudX SimS, a synthetic dataset driven by scenarios. In addition to providing interpretable features spanning spatial, temporal, behavioral, and contextual dimensions, FraudX SimS maintains the class imbalance typical of payment systems. For the purpose of simulating present-day attack behaviors, the dataset uses seven fraud typologies culled from current complaints in the industry. Some examples of these are behavioral drift, clustered terminal exploitation, time-based anomalies, and proximity-based compromise. Tests using a modular ensemble and many popular ML classifiers show that the dataset can handle both generalized and domain-specific evaluations of performance. Based on SHAP studies, it is possible to learn and explain embedded fraud signals, which allows for granular insights into model decisions. The FraudX SimS can be found on Zenodo and is available to the public to help with reproducible and transparent research [60]. Despite being artificial, the dataset encodes patterns of location, time, and behavior, including terminal breach, changes in risk over time, and unusual client behavior, making it a useful platform for testing ML-based fraud detection. Synthetic data has certain limitations that the author is aware of, including simplified user behaviors, the lack of complicated noise, and adaptable adversaries. In future editions, the author will make it a priority to include these patterns. Therefore, instead of being seen as an exact replica of production situations, FraudX SimS should be seen as a scalable benchmark. The author concludes that there is a need for more transparent, diversified, and scenario-aligned benchmarks in fraud detection and proposes scenario-specific supervised modeling and XAI-aligned evaluation as potential solutions. WORKS Cited [1] Finance Company Information on Fraud: Forecasts and Statistics for the Year 2024. Last updated: September 19, 2025. [Accessible Online]. Credit card fraud statistics can be found at https://merchantcostconsulting.com/lower credit-card-processing-fees.

[6] In their groundbreaking work, Yuhan Wang et al. presented IMHA, a new paradigm for predicting capital flows and detecting fraud in financial transactions. By representing the varied settings in which transactions take place and by clearly reflecting unequal connections between transaction participants, the author's approach tackles important issues in transaction modeling. The author's work primarily consists of the following: (1) an attention mechanism that adjusts the weights of sender and receiver information; (2) a feature encoder that learns the semantic relationships between account profiles and past behaviors; (3) a method for multi-task learning that uses signals that complement each other for fraud detection and capital flow prediction; (4) a battery of experiments showing better performance on the enhanced IEEE-CIS dataset and the author's own synthetic GPT-4o dataset; and (5) a battery of ablation studies that shed light on the relative importance of various model components. Across all evaluation metrics, the author's experimental data show that IMHA greatly surpasses existing techniques. The author's study reveals that in order to effectively detect fraud and predict capital flows, it is essential to model the asymmetric character of financial transactions and capture the intent behind them. In contrast to approaches that either fail to capture intent or

regard transactions as symmetric interactions, IMHA is able to detect subtle patterns and linkages that suggest fraudulent activity by explicitly including these features into the author's model architecture.

[7] A study by Chansreynich Huot et al. provides evidence that QAEs have great promise for improving fraud detection in datasets that include skewed credit card transactions. The author's QAE FD model is a unique use of quantum computing principles that greatly accelerates analytical processing and greatly improves the detection rates of fraudulent transactions. Both of these enhancements deal with major problems with the way anomaly detection systems have always been implemented in the banking and insurance industries. At its heart, the author's method relies on quantum mechanical features to shrink high-dimensional transaction data into lower-dimensional quantum states, which improves the system's capacity to spot small irregularities that could be signs of fraud. The methodology is structured in a two-part operational model: first, it applies quantum classification algorithms to classical transaction data; and second, it transforms the data into quantum states. This innovative approach sets new standards for computing efficiency and accuracy in fraud detection systems. The practicality of quantum methods in real-time applications is demonstrated by empirical validation on a real-world dataset, which confirms the QAE-FD model's superior performance compared to traditional machine learning models. The author's research highlights the need to further investigate and incorporate quantum technologies into financial security frameworks, since these technologies show great promise in bolstering initial defenses against credit card theft. Theoretical and practical advances in QML are also aided by this work. In the future, researchers will be able to use simulators and actual quantum computers like IBM's Qiskit, Google's Cirq, and Rigetti's Forest to study how to integrate quantum computing into diverse areas of cybersecurity and anomaly detection. The suggested model has the ability to scale, which means it might change the way we spot data anomalies in a future where everything is becoming digital. Its practical uses aren't limited to the financial industry, either. In order to create a thorough performance benchmark, future studies will focus on systematically comparing the QAE-FD model to traditional machine learning methods including Support Vector Machines, Random Forest, and Logistic Regression. You may learn a lot about the model's operational efficiency, including its execution time, memory usage, and hardware dependencies, by comparing the computational resource requirements. Additionally, in order to thoroughly evaluate the model's resilience, we will simulate a true quantum noise model. To make the model even more useful for detecting fraud in the real world, we need to make it more noise resilient by using advanced error mitigation strategies and provide interpretability frameworks to explain latent space representations and circuit dynamics.

[8] An adaptive generative adversarial network-based method is developed in this article by Fahdah A. Almarshad et al., and it may be used with data from similar domains. While the dataset may have changed, the model still makes use of the suggested generative adversarial network-based approach, which has the benefit of reducing domain shifts when the domains are comparable. The model is evaluated on two datasets: one for credit card fraud and another for financial transaction fraud. Oversampling is performed using GAN and SMOTE to address the class imbalance issue in both datasets. The generated data is subsequently fed into the model. In addition, we compare the model's performance to that of LR, ANN, and RF, three popular classification methods. Additionally, this research suggests an inference technique that the receiving bank might utilize to spot fraudulent transactions. Adversarial machine learning forms the basis of the approach, which makes use of discriminator and genera tive models to reliably identify outlier samples from the normal distribution. The model's performance is significantly improved when loss-minimization learning is combined with denoising techniques.

[9] A new framework for detecting financial fraud, FraudGNN-RL, was introduced by Yiwen Cui et al. (1) TSSGC layers that successfully capture temporal-spatial-semantic patterns in transaction networks; (2) RL-based dynamic decision boundary adjustment for evolving fraud patterns; and (3) superior performance on three fraud detection datasets among different metrics–these are the main contributions of the author. When applied to real-world scenarios where fraudulent transactions are uncommon but expensive, FraudGNN-RL proves to be quite resilient to class imbalance while keeping computational overhead low.

[10] A new strategy for detecting credit card fraud has been proposed by Yuxuan Tang et al., which combines federated learning with neural networks and Transformer models. Deep learning-based neural networks do a good job of detecting credit card fraud, according to the author's evaluations, which are based on simulations using publicly available datasets. This is particularly true when dealing with large-scale data and intricate online relationships. The author successfully improves the

model's performance on time-series data and reduces the load of human feature engineering by adding the Transformer model, serializing the credit card transaction data, and adopting the self-attention technique for feature extraction. At last, the author accomplishes the goal of federated learning by making it possible for various financial institutions to share and update model parameters. With this, there is a workable way to ensure the confidentiality of student information during cross-institutional learning collaborations. Nevertheless, the author recognizes the constraints of their current method and stresses the importance of doing future studies with real-world subjects to confirm their findings.

[11] Phishing, malware, and social engineering are some of the threats that Iscan et al. identified as having the potential to compromise users' electronic wallets and the funds they hold. This is why financial platforms are implementing advanced fraud detection systems to mitigate the effects of these types of attacks. Using data acquired from Turkey's most widely used e-wallet service, this study aims to apply cutting-edge machine learning algorithms to detect fraudulent activity. Feature engineering and experimental analysis revealed that LightGBm was the most effective technique, with a 97% detection rate and ROC AUC score of 0.9857. Reducing the total number of alarms from 13,024 to 6,249 was the major aim of the study, and it was accomplished. These results show that there is potential for machine learning-based methods to augment scarce resources and identify fraudulent e-wallet activities. However, there are a few limitations to this study. Because this study only used data from one e-wallet platform in Turkey, we don't yet know how applicable the findings are to other platforms or regions. There should be additional investigation on the effectiveness of traditional tactics to identifying electronic wallet fraud since this study just considered machine learning based techniques. However, there is new information about how to detect fraud in cashless transactions, and the study's positive findings add to that. The effectiveness of machine learning-based technologies and classical methods in detecting fraud in electronic wallets and other cashless transactions should be further investigated in future research.

[12] According to Fawaz Khaled Alarfaj et al., CCF poses a growing risk to banks and other financial organizations. The best fraudsters are always thinking of new ways to con others. The ever-evolving fraud landscape is no match for a powerful classifier. The main goal of any fraud detection system should be to reduce false-positives and accurately anticipate fraud situations. For every given business scenario, the efficacy of ML approaches differs. One major aspect that influences many ML approaches is the type of incoming data. Key performance indicators for CCF detection models include feature count, transaction volume, and feature correlation. When it comes to text processing and the baseline model, DLmethods like CNN and its layers are what come to mind. When compared to more conventional algorithms, these techniques outperform them when it comes to credit card detection. When all the algorithms are compared side by side, the top approach with a 99.72% accuracy is the CNN with 20 layers and the baseline model. While several sampling approaches are employed to enhance the performance of pre-existing instances, their efficacy on unknown data is substantially diminished. As the disparity between the classes widened, the performance on hidden data improved. To further enhance the model's performance, future research may investigate using more cutting-edge deep learning techniques.

[13] The importance of machine learning models, and more especially gradient boosting approaches, in detecting online payment fraud was demonstrated by Nishant Upadhyay et al. In this paper, we will look at how businesses may keep their customers' confidence and money safe by using historical data to spot and stop fraudulent activity. Since these models, particularly XGBoost models, do a good job of detecting fraud, they are good options for improving fraud detection skills, and one of the key points in their favor is their high accuracy. Additional features and more sophisticated algorithms can be added to future research to further improve the identification capabilities, as demonstrated by 592 Nishant Upadhyay et al. To further improve the model's efficiency and transparency in real-world applications, understanding how it makes decisions is crucial.

[14] In order to identify Bitcoin transaction fraud in smart cities, Noor Nayyer et al. developed a model. By first setting a threshold value on the year and the transfer amount, 381464 examples are extracted from a total of 2916697 occurrences. If the quantity is outside the range, the data will not be included, and if the year is past 2016, the data will not be included either. Due to the dataset's extreme imbalance, the author uses ADASYN-TL to correct any outliers after data collection is complete. To find the specific value for the classifier parameters, hyperparameter tuning approaches including grid search, random search, and Bayesian optimization are employed. The stacking model is constructed for classification purposes by utilizing RF on the meta layer and integrating DT, KNN, and NB on the base layer. By contrasting it with other classifiers, we can verify the suggested model's efficacy. The information regarding the effect of

characteristics on the model's prediction is provided by SHAP. With a 97% accuracy rate, 99% area under the curve, 96% precision, 98% recall, 3% false positive rate, and 97% F1-score, the suggested stacking model utilizing ADASYN-TL outperforms all other algorithms in the simulation. The balancing methods SMOTE-ENN and ADASYN-TL are also contrasted. With an F1-score of 97%, ADASYN-TL surpasses SMOTE-ENN. Even without tweaking any parameters, the stacking model had a 95% success rate. However, after hyperparameter adjustment, the stacking model's accuracy went up 2%.

[15] A lot of the literature on financial fraud detection uses complex handcrafted features for machine learning model training rather than raw transaction data or automatically synthesized features, according to Yu-Yen Hsin et al., who state that this is because there is a lack of available transaction data and strong interpretability requirements. There are four types of handcrafted features seen in literature: recency, frequency, monetary, and anomaly (RFMA).In order to identify (legitimate) accounts, this research suggests using segmentation-type features and behavior-type features that describe non-RFMA attributes. The foundation of behavior-type features is often financial expertise, which may be seen as an expert system's knowledge base. Based on statistical summaries of the classifications of raw transaction data, segmentation-type features may be created, as shown in Tables 3 and 4. This provides a useful clue for future designs of automatic feature synthesis. In order to demonstrate the superiority of the author's proposed features, we compare their performance in training popular classifiers like SVM, random forests, XGBoost, and LGBM with features generated by automatic generation methods or suggested in previous fraud detection literature. This article delves into the characteristics that lead XGBoost and LGBM to yield unreliable detection outcomes. Using the Kolmogorov-Smirnov test, we may identify these noisy characteristics as time-inhomogeneous. Although SVM and random forest do not experience this unstable detection problem and generate stable predictions, XGBoostandLGB do, according to the experimental data.The removal of noisy characteristics allows for myieldbetter fraud detection findings with fair interpretability. The time-inhomogeneous nature of the modi operandi is further reflected by the existence of noisy elements. Due to the elimination of time inhomogeneity, deceptively good performance is produced when evaluating a machine learning model's resilience by creating training and testing sets using random sampling. The author investigates multiple resampling techniques including WGAN to rectify the data imbalance caused by the low number of fake accounts. The SMOTE-related approaches produce low-quality fraudulent data and reduce total fraud variability by applying inappropriate linear interpolations on different MO patterns. To get around these issues and increase detection results, though, you can use WGAN and complete (random) sampling. As stated in a premium financial journal, the author's recommended features (categories) are shown to rank highly according to the approach described in [28]. This verifies the quality of the author's work.

3. METHODOLOGY

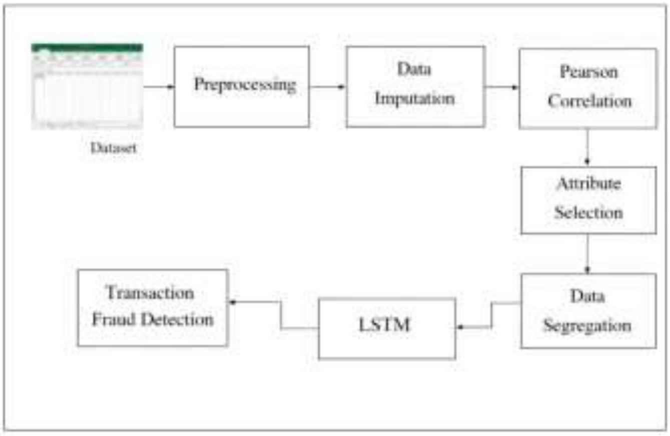


Figure 1: Proposed model System Overview

In figure 1 above, we can see a system overview of the proposed methodology for investment advice. Below, we have detailed the sequential procedures to implement the proposed approach.

Step 1: Dataset collection – The suggested model use a dataset that is not sourced from India in order to detect transaction fraud. This dataset may be found at: https://www.kaggle.com/code/llabhishekll/fraud-transaction-detection-notebook. You can tell if a financial transaction is legitimate or fraudulent by looking at the records it has in this dataset. A prominent use case for this dataset is the construction of machine learning models for fraud detection; it contains a high number of transactions collected over a particular period.

This dataset includes various attributes that are explained in detail below: "Transaction ID" denotes the unique identifier of each transaction, "Transaction Amount" the amount of money involved in the transaction, "Transaction Time" the time at which the transaction happened, "Merchant ID" the identification of the merchant where the transaction took place, "Customer ID" the unique identifier of the customer, "Transaction

Type" the type of transaction (e.g., payment, withdrawal, or transfer), "Location" the geographical location where the transaction was performed, and "Fraud Label" denotes whether the transaction is fraudulent or legitimate.

Step 2: Preprocessing – Once the dataset is obtained, it is applied to the pre-processing stage. In this step, the attributes are collected in a two-dimensional list after the dataset is read from the specified path. To read and manage the dataset efficiently, the Pandas library in Python is used.

This two-dimensional data structure is then used to estimate the initial statistical parameters of the attributes, such as mean and standard deviation, which help in understanding the overall characteristics and distribution of the dataset.

After this process, the dataset attribute information is analyzed based on different data types, such as string and numerical (float) values, to understand the structure and variability of the data. Categorical attributes related to fraud detection are converted into numerical labels. For example, the fraud transaction attribute is labeled as 1 for fraudulent transactions and 0 for legitimate transactions.

Once the dataset is labeled, the frequency of each class is calculated to examine whether the dataset is balanced or imbalanced. Since fraud datasets usually contain fewer fraudulent cases compared to legitimate transactions, the minority class is oversampled to balance the dataset. This balancing process improves the effectiveness of the pre-processing stage and helps in achieving better performance in the transaction fraud detection model.

Step 3: Data Imputation – If you want to see how various aspects are related, you can use the oversampled transaction dataset to make a heat map of all the properties. The total number of missing values and their related percentages in each attribute are determined by comparing the transaction data through a sorting and analysis procedure.

For attributes including customer information, merchant category, transaction type, amount, and location, the fillna() function is used to handle the discovered missing values. By doing so, we can make sure the dataset is ready for additional machine learning processing.

For every categorical attribute, the fit_transform() function is used in conjunction with a Label Encoder to do data imputation. This procedure makes numerical values out of previously described categories of data, such as location, merchant category, and type of transaction, so that machine learning algorithms can process them.

Once the attributes have been transformed, the IterativeImputer() function is employed to estimate the missing values by Multiple Imputation by Chained Equations (MICE). Using this method, we may fill in gaps in our dataset by making predictions about new data points based on existing, relevant attributes. The MICE technique utilizes the relationships among the available features to generate several estimations and anticipate the best acceptable values when there are missing values in the dataset for one or more variables.

In order to detect and deal with possible dataset outliers, the Interquartile Range (IQR) approach is used after the imputation phase, with ranges from 0.25 to 0.75. In the end, we get a dataset with corrected feature distributions and imputed missing values, which boosts the data quality overall and allows the transaction fraud detection model to produce more accurate findings.

Step 4: Pearson Correlation – After Mice imputation, a full dataset list is obtained and analyzed using Pearson Correlation. Using the Pearson Correlation values, we may find the qualities that have the lowest correlation. One way to determine if two qualities are related is to use Pearson's correlation. The end result is a correlation matrix that might be useful for picking the right combination of attributes. In light of this new information, we can calculate correlation values and ignore the attributes with lower correlation. To conduct the Pearson correlation, we use Equation 1, which can be found below.

$$r = \frac{\sum(x_i - \bar{y})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2}\sum(y_i - \bar{y})^2} \text{ -------- } (1)$$

Where,

x_i=values of x (independent) variable

y_i= values of y (Dependent) variable

x̄= mean of x variable values

ȳ= mean of y variable values

Following the compilation of the dataset in the double-dimensional list, after the correlation step, a few superfluous columns are eliminated from the list. The obtained list is then fed to the next step of data segmentation, as explained in the next step.

Step 5: Attribute Selection – Using the imputated and preprocessed, the minMaxscaler function is applied in this attribute selection procedure. To modify the features, minmaxscaler scales them to a specified range. This estimator applies separate scaling and translation to each feature in order to get it inside the specified range on the training set, for instance, between 0 and 1. While MinMaxScaler performs a good job at linearly scaling outliers into a predetermined range—where the biggest data point represents the maximum value and the smallest one represents the minimum value—it does nothing to reduce the effect of outliers. For an example of a visualization, see Compare MinMaxScaler with different scalers. Equations 2 provide the minmaxscaler transformation.

$$x_{std} = \frac{(x-x.min(axis=0))}{(x.max(axis=0)-x.min(axis=0))}$$

x_(scaled)=x_(std) *(max-min) +min— (2)

where min, max = feature_range.

In order to determine if a transaction is fraudulent or legitimate, two lists are created for the "Fraud" prediction attribute when the preprocessing stage is finished. The next step is to use a feature selection technique to determine which features are most significant and then pick the top ten features according to their ratings.

A model with a random state of 0 and a number of estimators of 100 is used to fit and alter these characteristics after they have been selected. By going through this procedure, we can zero in on the most important characteristics that aid in detecting fraud. Consequently, the model is trained in the following stages using the most important attributes that were gathered.

Amount, Time, Merchant Category, Type, Customer Location, Device Type, Frequency, Account Age, Prior Fraud History, and Payment Method are some of the key parameters used for the purpose of developing a model to detect fraudulent transactions. During the training and prediction process of the model, these features are crucial for spotting suspicious or fraudulent transactions.

Step 6: Data Segregation – X is a list that represents the input features; it contains all the attributes related to transactions, including TransactionAmount, TransactionTime, MerchantID, CustomerID, TransactionType, DeviceType, Location, PaymentMethod, TransactionFrequency, AccountAge, PreviousFraudHistory, and MerchantCategory. Conversely, a another list named Y stores the Fraud attribute, which shows if a transaction is fraudulent or lawful.

Using Scikit-test_split(), we partition the dataset into a training set and a testing set. The first step is to partition the dataset into two parts: features (X) and labels (Y). Partitioning the dataset into X_train, X_test, Y_train, and Y_test follows splitting. To train and fit the fraud detection model, we utilize the X_train and Y_train datasets. To evaluate the model and ensure that it correctly predicts the transaction labels, we use the X_test and Y_test datasets. As a rule of thumb, the training dataset should always be bigger than the testing dataset.

Data Used to Train the Model: This data set is known as the training dataset. The model uses this data to learn patterns and relationships

Final Performance Evaluation: The trained model is evaluated using the test dataset. Training uses 67% of the data in the suggested system, while testing makes use of 33%.

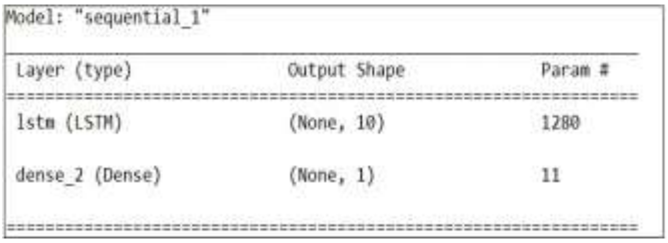
Before running the model through its paces in a final test, it is common practice to fine-tune its hyperparameters using a subset of the training data known as a validation dataset.

The feature values are normalized by using data scaling once the dataset is split. Each feature's minimum and maximum values are set between 0 and 1 during this process. For this, we employ Scikit-MinMaxScaler's() function. While preserving the original data distribution structure, the MinMax scaler transforms the data into a defined range, typically 0 to 1.

The MinMaxScaler() function is used to obtain the scaled datasets train_X, test_X, train_Y, and test_Y. Afterwards, the Artificial Neural Network (ANN) model for detecting transaction fraud is trained using these processed datasets.

Step 7: Long short term memory (LSTM) - You can feed a scalar normalization object, test_x, train_x, and test_y into an LSTM neural network. The LSTM model with parameters like train_X1.shape[1] and train_X1.shape[2] is introduced in a one-dimensional space with a single feature, 10 units of data, and a TRUE return sequence. Following that, a Dense layer is included, which has an activation function called "relu" and a kernel size of 1. The dense layer of a densely coupled neural network learns new information effectively by using activation functions on neurons. The basic LSTM neural network consists of two dense layers; however, with one-dimensional data, only one kernel of size 1 is used. A neural network is constructed with a batch size of 100, 100 epochs, and the shuffle parameter set to false.

Figures 4 and 5 show the built LSTM model overview and the training results, correspondingly.



```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 10)                1280

 dense_2 (Dense)             (None, 1)                 11

=================================================================
```

Figure 2: LSTM Model Summary

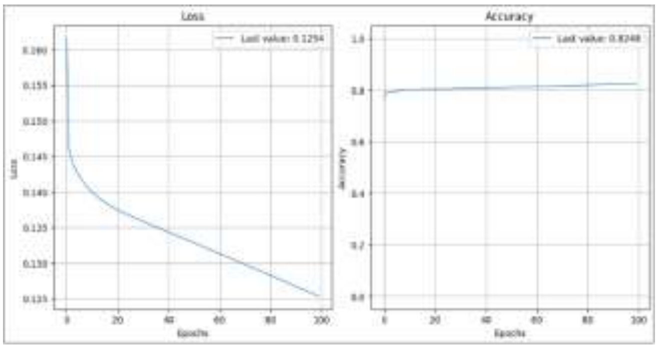


Figure 3: LSTM model training Results

4. RESULTS AND DISCUSSIONS

The suggested approach to detecting fraudulent transactions is built on top of the Python programming language and is compatible with Windows-based computer systems. Throughout the development and implementation phase, the Jupyter Notebook IDE is utilized. The development machine has the computational power to train and test the fraud detection models, thanks to its Intel Core i5 processor, 500 GB hard disk, and 8 GB RAM.

Proper evaluation of the model's performance is essential for a successful implementation of the transaction fraud detection system. Following the above steps, the suggested technique takes as input a transaction dataset that includes various financial transaction attributes. The model can learn to distinguish between valid and fraudulent transactions with the help of these attributes. Applying deep learning models like ANN and LSTM to the dataset allows us to detect fraudulent transactions, as we'll see in the next section. In order to assess how well the suggested system detects and prevents transaction fraud, these models produce prediction results.

Performance Evaluation based on Root Mean Square Error (RMSE)

By including the system's degree of inaccuracy in the testing process, the approach's efficacy could be precisely measured. By utilizing mean absolute error, this error can be calculated, revealing the methodology's inaccuracy. Use equation 3 to get a good grasp on how often continuous qualities are inaccurate. The scores for rainfall prediction that are generated by ANN and LSTM models are suitable for this evaluation. In order to apply Root Mean Square Error (RMSE) to score prediction. One way to find out how off the suggested method is is to calculate its root mean square error (RMSE). In this investigation, the root mean square error (RMSE) is used to calculate the margin of error between the actual and expected event detection performed by the ANN and LSTM models. The root-mean-squared error (RMSE) method is shown in equation 1 below.

$$\mathrm{RMSE}_{fo} = [\sum_{i=1}^{N} (z_{f_i} - z_{o_i})^2 / N]^{1/2} \qquad \_\_\_\_(3)$$

Where

∑ - Summation.

(Zfi - Zoi)2 - Differences Squared for the Rainfall Prediction.

N - Number of trails.

Before the Error Rate of the Approach can be evaluated using RMSE, the Mean Square Error, or MSE, must be obtained. The Mean Square Error (MSE) quantifies the discrepancy between the predicted and actual occurrence of transaction fraud. To measure the model's accuracy in determining the legitimacy or fraud of a transaction, MSE is employed. The testing step involves conducting several trials, or iterations, to evaluate the proposed system's performance. To ensure the accuracy and consistency of the fraud detection model, these trials are conducted multiple times. We document and analyze the outcomes of these experiments.

Models	RMSE
ANN	0.37631
LSTM	0.36339

Table 1: RMSE results for ANN and LSTM

Table 1 displays the effectiveness of the Long Short-Term Memory (LSTM) models trained on the non-Indian transaction dataset, as a consequence of these trials. Figure 4 graphically displays the comparative results, which aid in comprehending the disparities in performance between the two deep learning models for the transaction fraud detection system.

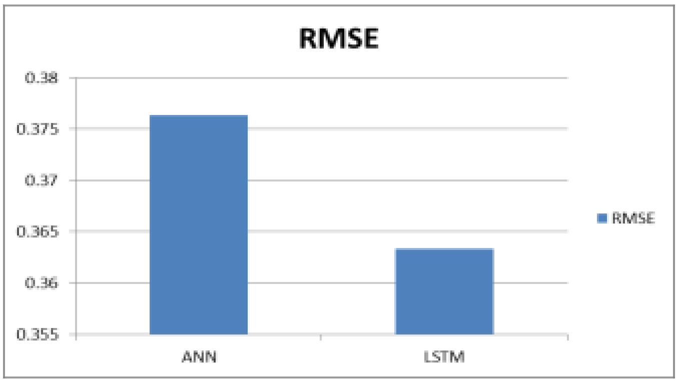


Figure 4: RMSE results for LSTM

5. CONCLUSION AND FUTURESCOPE

Finally, this article suggested a data-mining-based, all-encompassing method for detecting fraudulent transactions in order to tackle the growing problem of fraud in today's digital financial systems. The sophistication and volume of fraudulent operations have increased dramatically due to the fast adoption of e-commerce platforms, mobile payments, and online banking, rendering old rule-based detection systems ineffective. To get around these problems, the suggested method uses data mining and machine learning techniques to sift through mountains of transaction data, unearth hidden patterns, and spot suspicious activity quickly and accurately. Financial organizations and their clients can build trust through this data-driven strategy, which improves real-time monitoring and decreases financial losses.

Enhancing the reliability and robustness of the fraud detection process is achieved through the combination of numerous analytical methodologies. By providing analysts and decision-makers with clear and interpretable results, linear regression helps them comprehend typical transaction behavior and spot outliers through risk and trend analysis. Simultaneously, the system is able to grasp intricate, non-linear correlations between transactional characteristics like amount, frequency, location, and user behavior thanks to the utilization of Artificial Neural Networks. The ANN model takes into account the ever-changing nature of fraudulent operations by learning from past transaction data and adjusting to new fraud patterns, therefore increasing the accuracy of detection.

Securing digital financial transactions has never been easier than with a data mining-based transaction fraud detection system. The paper's results and discussion show that it's an effective and scalable solution. Financial institutions can benefit from better decision-making, quicker response times, and proactive fraud prevention with the help of the suggested method. The technology might greatly improve financial security and help build more robust digital payment ecosystems with future upgrades including real-time data integration, powerful ensemble models, and explainable AI techniques.

Data mining's future applications in the identification of transaction fraud are vast and full of promise for advancement. Improving the system's detection accuracy is possible by the integration of behavioral biometrics, such as typing habits, device usage, and location dynamics, with real-time transaction streams. Incorporating advanced ML/DL models, such as ensemble and hybrid techniques, can help deal with complex and ever-changing fraud scenarios.

Furthermore, the system can be enhanced with the help of cloud-based architectures and big data in order to facilitate large-scale implementation across various financial platforms. By making it easier for stakeholders to comprehend the reasoning behind fraud detection choices, explainable AI techniques will increase openness and conformity with regulations. Future systems can better protect against new forms of financial fraud by integrating with global fraud intelligence networks and using continuous learning techniques.

REFERENCES

[1] J. Jemai, A. Zarrad and A. Daud, "Identifying Fraudulent Credit Card Transactions Using Ensemble Learning," in *IEEE Access*, vol. 12, pp. 54893-54900, 2024, doi: 10.1109/ACCESS.2024.3380823.

[2] A. A. Almazroi and N. Ayub, "Online Payment Fraud Detection Model Using Machine Learning Techniques," *IEEE Access*, vol. 11, pp. 137188–137203, 2023. doi: 10.1109/ACCESS.2023.3339226.

[3] H. M. R. Al Lawati, A. Zainal, B. A. S. Al-Rimy, M. Al-Azawi, M. N. Kassim, S. A. Almalki, and T. A. Alghamdi, "An Integrated Preprocessing and Drift Detection Approach With Adaptive Windowing for Fraud Detection in Payment Systems," *IEEE Access*, vol. 13, pp. 92040–92056, 2025. [Online]. Available: https://doi.org/10.1109/ACCESS.2025.3569609

[4] F. K. Alarfaj and S. Shahzadi, "Enhancing Fraud Detection in Banking With Deep Learning: Graph Neural Networks and Autoencoders for Real-Time Credit Card Fraud Prevention," *IEEE Access*, vol. 13, pp. 20632–20646, 2025. doi: 10.1109/ACCESS.2024.3466288.

[5] N. Baisholan, J. E. Dietz, S. Gnatyuk, M. Turdalyuly, E. T. Matson, and K. Baisholanova, "FraudX SimS: A Synthetic Dataset for Anomaly Detection in Payment-Card Transactions," *IEEE Access*, vol. 13, pp. 208549–208562, 2025. [Online]. Available: https://doi.org/10.1109/ACCESS.2025.3637828

[6] Y. Wang and W. Kang, "Intent-Aware Multi-Source Hybrid Attention for Financial Fraud Detection and Capital Flow Prediction," *IEEE Access*, vol. 14, pp. 601–615, 2026. [Online]. Available: https://doi.org/10.1109/ACCESS.2025.3642572

[7] C. Huot, S. Heng, T.-K. Kim, and Y. Han, "Quantum Autoencoder for Enhanced Fraud Detection in Imbalanced Credit Card Dataset," *IEEE Access*, vol. 12, pp. 168673–168685, 2024. doi: 10.1109/ACCESS.2024.3496901.

[8] F. A. Almarshad, G. A. Gashgari, and A. I. A. Alzahrani, "Generative Adversarial Networks-Based Novel Approach for Fraud Detection for the European Cardholders 2013 Dataset," *IEEE Access*, vol. 11, pp. 107353–107368, 2023. doi: 10.1109/ACCESS.2023.3320072.

[9] Y. Cui, X. Han, J. Chen, X. Zhang, J. Yang, and X. Zhang, "FraudGNN-RL: A Graph Neural Network With Reinforcement Learning for Adaptive Financial Fraud Detection," *IEEE Open Journal of the Computer Society*, vol. 6, pp. 426–437, 2025. [Online]. Available: https://doi.org/10.1109/OJCS.2025.3543450

[10] Y. Tang and Z. Liu, "A Credit Card Fraud Detection Algorithm Based on SDT and Federated Learning," *IEEE Access*, vol. 12, pp. 167389–167403, 2024. doi: 10.1109/ACCESS.2024.3491175.

[11] C. Iscan, O. Kumas, F. P. Akbulut, and A. Akbulut, "Wallet-Based Transaction Fraud Prevention Through LightGBM With the Focus on Minimizing False Alarms," *IEEE Access*, vol. 11, pp. 131460–137474, 2023. doi: 10.1109/ACCESS.2023.3321666.

[12] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed, "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms," *IEEE Access*, vol. 10, pp. 39700–39715, 2022. doi: 10.1109/ACCESS.2022.3166891.

[13] N. Upadhyay *et al.*, "Machine Learning Perspective: Fraud Payment Transaction Detection," *Journal of Mobile Multimedia*, vol. 21, no. 4, pp. 581–600, 2025.

[14] N. Nayyer, N. Javaid, M. Akbar, A. Aldegheishem, N. Alrajeh, and M. Jamil, "A New Framework for Fraud Detection in Bitcoin Transactions Through Ensemble Stacking Model in Smart Cities," *IEEE Access*, vol. 11, pp. 90924–90938, 2023. doi: 10.1109/ACCESS.2023.3308298.

[15] Y.-Y. Hsin, T.-S. Dai, Y.-W. Ti, M.-C. Huang, T.-H. Chiang, and L.-C. Liu, "Feature Engineering and Resampling Strategies for Fund Transfer Fraud With Limited Transaction Data and a Time-Inhomogeneous Modi Operandi," *IEEE Access*, vol. 10, pp. 86101–86116, 2022. doi: 10.1109/ACCESS.2022.3199425.