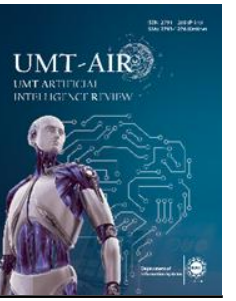
UMT Artificial Intelligence Review (UMT-AIR)
Volume 6 Issue 1, Fall 2026
ISSN$_{(P)}$: 2791-1276, ISSN$_{(E)}$: 2791-1268
Homepage: https://journals.umt.edu.pk/index.php/UMT-AIR



Title:	**Credit Card Fraud Detection Using an Ensemble Deep Learning Approach: CNN, LSTM, and DNN**
Author (s):	Abdullah Mehboob
Affiliation (s):	Department of Computer Science, University of Sargodha, Pakistan
DOI:	https://doi.org/10.32350.umt-air.61.04
History:	Received: January 22, 2026, Revised: February 14, 2026, Accepted: March 20, 2026, Published: June 08, 2026
Citation:	Mehboob, A., "Credit card fraud detection using an ensemble deep learning approach: CNN, LSTM, and DNN," UMT Artificial Intelligence Review, vol. 6, no. 1, pp. 57–73, Jun. 2026, doi: https://doi.org/10.32350.umt-air.61.04.
Copyright:	© The Authors
Licensing:	This article is open access and is distributed under the terms of Creative Commons Attribution 4.0 International License
Conflict of Interest:	Author(s) declared no conflict of interest

A publication of
Department of Information System, Dr. Hasan Murad School of Management
University of Management and Technology, Lahore, Pakistan

Credit Card Fraud Detection Using an Ensemble Deep Learning Approach: CNN, LSTM, and DNN

Abdullah Mehboob*

Department of Computer Science, University of Sargodha, Sargodha, Pakistan

ABSTRACT The prevalence of credit card fraud, resulting in substantial financial losses for both banks and customers, poses a significant challenge to the financial sector. This issue has been exacerbated by the surge in internet transactions, which complicates the task of fraud detection. Numerous machine learning methods, such as the extreme learning method, decision tree, Random Forest, Support Vector Machine, and Logistic Regression, have been employed in recent research. However, the current landscape now embraces deep learning algorithms, which enhance accuracy and performance. This paper introduces a credit card fraud detection system that harnesses Deep Learning Algorithms to scrutinize transactional data and identify potential instances of fraudulent activity, thereby addressing this concern. The system employs a range of features to construct a real-time fraud detection model. A comprehensive empirical analysis has been conducted, encompassing the latest model, variations in the number of epochs, and hidden layer configurations. The system's evaluation includes parameters such as accuracy, precision, recall, and F1 score. The results substantiate the system's efficacy in detecting credit card fraud, outperforming existing systems. The proposed method offers a reliable and effective means of real-time credit card fraud identification, boasting an exceptional accuracy rate of 99.95%.

INDEX TERMS Credit card fraud detection (CCFD), deep learning, convolutional neural network (CNN), long short-term memory (LSTM), dense neural network (DNN), voting ensemble classifier, machine learning.

I.INTRODUCTION

CCF (credit card fraud) is the term for using someone else's credit card number without their permission to make transactions or withdraw money from their account. Both physical transactions where the card itself is required for the transaction, much like an ATM and digital transactions which can be completed over the phone or online can result in this kind of fraud. Users are required to enter their credit card information, including the card number, expiration date, and CVC, while making transactions online. Simple rule-based systems were used as the foundation for early CCFD (credit card fraud detection) approaches, which are not effective enough to detect new fraud types.

Machine learning and data mining techniques were first used in the 2000s to identify CCF. These methods include decision trees and clustering algorithms which try to identify the complex patterns of fraud. In the early 2010s, real-time monitoring and anomaly detection methods were introduced which allow the detection in real time. This has been possible due to the increasing power of computing. With the ability to process enormous amounts of data, businesses have begun to concentrate on preventing fraud rather than detecting it by implementing systems such as two-factor authentication, tokenization, and biometrics as of the year 2010. However, background fraud detection from the 2000s to 2015 era provides a lot of security

*Corresponding Author: iam.abdullah113@gmail.com

measures related to CCFD and prevention, but still credit card frauds increase day by day. We will discuss this in the figures.

There are a lot of advantages related to CCFD. CCFD prevents the customer from losing his money by preventing them from CCF and also prevents them from unauthorized access to the customer information. It can decrease the loss of business which occurs due to CCF. It also helps to protect customer data which increases customer satisfaction. It can secure the transaction which customers perform on the internet too. The most widespread usage of CCFD techniques is in the banking and finance industries, which employ them to thwart fraud and shield their clients from financial losses. Another frequent application of CCFD is in e-commerce, where it helps businesses and customers avoid financial losses.

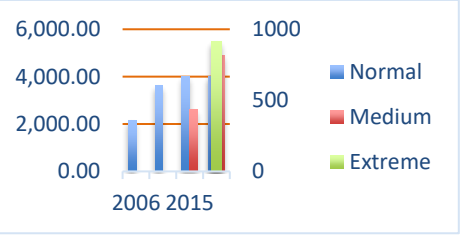


FIGURE 1. Annual global payment volume via credit card.

The Federal Trade Commission estimates that there were 133,100 fraud cases overall in 2017; this figure is expected to rise to 4 lac cases in the USA alone by 2021. The number of frauds is increasing day by day. Here is a graph of the total number of frauds shown in Figure 1.

A common type of CCF is a new account, and the other type is an existing account. The overall number of new account fraud cases in 2021 is 363,092, while the total number of current account fraud cases is 32,204. New account fraud means when someone stole your information and then makes a new credit card account with it. An existing account means that someone stole your card and uses it. By utilizing risk assessment and data enrichment, SEON offers quick, effective, and seamless CCFD. It enables you to compile a thorough profile of every visitor to your website, even those who leave before checking out. Mobile apps now have the extraordinary ability to identify the precise moment a user starts acting maliciously because SHIELD Sentinel is the only technology in the world that continuously assesses risk during a user's session. As pioneers in behavioral biometrics, we use cutting-edge technology that taps into the potential of human digital behavior to produce insightful data and advance a society based on simplicity and trust.

Our cutting-edge technologies are being adopted by financial institutions all around the world to strengthen their security protocols. To protect the veracity of their users' identities, BioCatch uses behavioral biometric identification and authentication. As a result, fraud is better defended against, digital transformation is made easier, and corporate growth is accelerated. They make use of a dataset that consists of two CSV files, one used to train the system and the other to test it. The training dataset contains 1 million rows of data and the second file contains 0.5 million of data. In this dataset, we extract important features which we are going to use in our system; some important features are Transaction_date_time, merchant, category, gender, street, city, state, longitude, latitude, merchant longitude, and merchant latitude.

How do we use these features? Suppose we have features like category in which we have a list of a customer where we will see the list of his buying items on daily basis; we will make a pattern according to this list

whether his future buying meets these patterns or not.

We employ a CNN-LSTM (Convolutional Neural Network-Long Short Term Memory) deep learning model for our approach. If we talk only about CNN and LSTM: CNN is used for image and video analysis and is good at identifying patterns and features in data, and LSTM is used to process sequential data such as time series or text. It remembers previous inputs and uses them to inform predictions about future inputs. In CNN-LSTM, the CNN is utilized to extract features from data, which are then passed as input values to LSTM, which makes predictions based on the data. To understand more, we made some questions we want to answer: Can we come up with more time efficiency? Can we come up with an algorithm that can show better accuracy than the existing algorithms? Can we have a strong feature in the datasets that can give more security-related CCFD? How does the performance of CCFD change depending on how features are represented? Combining CNN and LSTM may increase the model's efficiency compared to machine learning models because CNN is effective at extracting features from image and time series data and LSTM is efficient at working with time series data, which is useful for identifying patterns in the data.

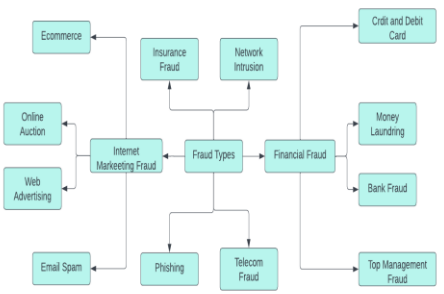


FIGURE 2. Fraud types.

The CNN-LSTM, combining these two, allows the model to analyze both dimensional and time sequence data features because the CNN-LSTM first extracts the features and then makes the pattern of that data; this will help to improve accuracy as well as time efficiency. We have a dataset first that has 1.5 million records related to datasets and also, we have very strong features from which we can make strong patterns that will improve accuracy and decrease CCF.

Strong features have a large impact on performance like category: when a customer buys a list of items on daily basis the algorithms take those data and detect the pattern; if the customer buys an unusual thing that does not meet that pattern, the system will block that transaction.

II. RELATED WORK

Baris Can et al. [1] This research presumably digs into a thorough examination of the distinctive characteristics and features that set valid card transactions apart from fraudulent ones. It could entail spotting trends, such as odd transaction timings, places, sums, and other characteristics that might help distinguish fraudulent operations. Nghia Nguyen et al. [2] This article proposes a unique method for fraud detection by merging two potent methods, deep neural networks and CatBoost (a gradient boosting algorithm known for handling categorical features well). The model might take advantage of the advantages of both approaches to improve the reliability and accuracy of detecting fraudulent card transactions. Amlan Kundu et al. [3] This paper may adapt methods like BLAST (Basic Local Alignment Search Tool) and SSAHA (Sequence Search and Alignment by Hashing Algorithm) for spotting similarities or trends in credit card

transaction data from bioinformatics. A novel fraud detection method might be developed by the hybridization of these algorithms.

YU-Yen Hsin et al. [4] This research may suggest innovative strategies to design pertinent features and build synthetic data through resampling techniques to address issues posed by limited data and shifting fraud trends in fund transfer scenarios. It aims to more effectively capture the dynamic nature of fund transfer fraud.

Seyedeh Khadijeh Hashemi et al. [5] This study probably offers an overview of different machine learning methods that can be used to identify fraud in banking data. The main point of emphasis can be on describing how decision trees, support vector machines, neural networks, or other techniques might be utilized to spot possibly fraudulent behavior in financial transactions. Lutao Zheng et al. [6] The TrAdaBoost algorithm may be improved upon or changed as a result of the work being done to improve fraud detection scenarios. TrAdaBoost is made to handle circumstances when there is a dearth of labeled data in the target field (in this example, fraud detection). Haibo Wang et al. [7] This study examines a cutting-edge idea that combines quantum annealing solvers with conventional machine learning techniques. To solve optimization issues more quickly, quantum annealing makes use of quantum computing concepts. Online fraud could be quickly detected by using this integration. Fuat Ogme et al. [8] This research focuses on identifying potential locations in a transaction sequence where a security breach or vulnerability could happen. It makes use of temporal transaction scraping, a technique for tracking historical transaction timestamps and trends. Autoencoders are probably used to compress representations of transaction data in order to identify significant patterns.

Lutao Zheng, Guanjun Liu et al. [9] The entire order relation of transactions and the variety of transaction behaviors are taken into account in this paper to offer a novel viewpoint on fraud detection. Analyzing the order of transactions, their connections, and the differences in behaviors displayed by various transactions are all possible parts of this process. Yathartha Singh et al. [10] The variety of methods used to identify fraud, specifically in credit card transactions, are probably explored in this study. It might include conventional rule-based procedures, machine learning ideas, anomaly detection, and perhaps even more sophisticated techniques. S P Maniraj et al. [11] Although the paper's title seems to be lacking information, it may offer a model or method for detecting credit card fraud that makes use of other methods or algorithms. The IJERT framework could be used to develop a revolutionary fraud detection strategy. Lutao Zheng et al. [12] This study is probably a development of the TrAdaBoost method, which focuses on knowledge transfer from a source domain to a target domain where labeled data is scarce. The study might demonstrate how this upgraded TrAdaBoost is used specifically to identify fraud in transaction data.

Haibo Wang et al. [12] In order to improve online fraud detection, this research investigates the combination of conventional machine learning methods with quantum annealing solvers. Real-time fraud detection could be improved with the use of a combination of classical and quantum computing techniques. Ms. S.V. Juno Bella Gracia et al. [13] This document probably offers a summary of various machine learning methods used to identify payment fraud. It might go over algorithms

like decision trees, random forests, neural networks, and others, describing how they are designed to spot fraudulent payment practices. Vinod Jain et al. [14] The performance of several machine learning methods in the context of CCFD is the main objective of this article. When several algorithms are used to solve this particular problem, it may compare and contrast the accuracy, precision, recall, and other metrics of those algorithms.

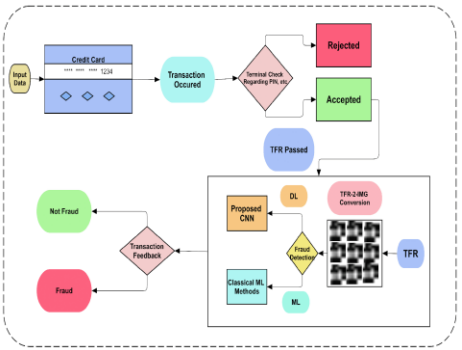


FIGURE 3. CCFD method.

Emmanuel Ileberi et al. [15] This work primarily focuses on applying methods like SMOTE (Synthetic Minority Over-sampling Technique) and AdaBoost to address class imbalance and enhance detection accuracy, with a specific emphasis on analyzing the efficacy of machine learning algorithms for CCFD. Several studies have been conducted on CCFD using machine learning and deep learning techniques. Sumanth et al. conducted an analysis of CCFD using machine learning techniques [16]. In order to improve prediction accuracy, Krishna and Praveen Chandar et al. [17] conducted a comparative analysis of CCFD utilizing logistic regression and random forest. A credit CCFD system based on machine and deep learning was proposed by Najadat et al. [18]. AdaBoost and majority voting were used by Randhawa et al. [19]. Babu and Pratap et al. [20] most likely review and build upon prior research in the areas of CCFD and deep learning applications in their work. In order to improve the accuracy of fraud detection, prior studies may have investigated feature engineering methods, anomaly detection techniques, ensemble models, and data augmentation techniques.

Additionally, the researchers may have looked at the use of deep neural network topologies for fraud detection, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Jonnalagadda et al. [21] created a CCFD system. Using the random forest method, Kumar et al. suggested a CCFD system. Machine learning and deep learning methods were used by Alarfaj et al. [22] to construct a CCFD system. A deep learning strategy for CCFD was proposed by El Naby et al. [23]. Deep representation learning combined with full center loss was used by Li et al. [24] to create a CCFD system. Deep neural networks were put to the test for the reduction of CCF alerts by San Miguel Carrasco and Sicilia-Urban et al. [25]. A fraud detection system for credit card transactions utilizing SVM and random forest techniques was proposed by Saddam Hussain et al. [26]. A strategy for detecting payment fraud using machine learning methods was proposed by Gracia et al. [27]. Proposed a secure federated learning framework combining CNN and ANN models for privacy-preserving classification. The study demonstrates the potential of federated deep learning for secure and scalable data analysis [28].

III. METHODOLOGY

Deep learning is generally used in a multi-stage research technique for CCFD. First, the dataset---which contains a substantial number of credit card transactions, both fraudulent and legitimate---is gathered.

This dataset is preprocessed and cleaned to ensure that the data is suitable for training and testing machine learning models. A deep learning model is selected and configured, such as a neural network with multiple layers, which is capable of learning complex patterns and relationships in the data. A suitable training procedure, such as stochastic gradient descent, is then used to train the model on a portion of the dataset. Following training, the model is assessed on a different subset of the dataset, referred to as the validation set, to make sure it is not overfitting the training data and can generalize well to new data. The model is then fine-tuned based on the results of the evaluation, and the process of training and validation is repeated until the model reaches a satisfactory level of performance.

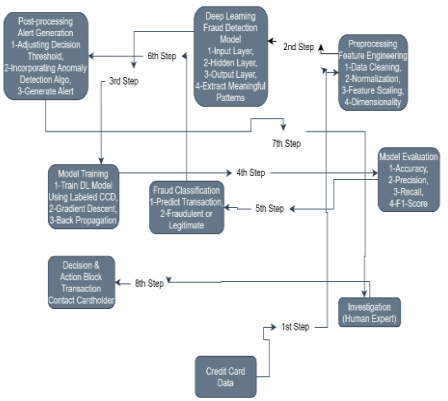


FIGURE 4. CCFD flow.

A. DATASET 1

We analyze credit card transactions with 21 attributes using the KAGGLE CCFD dataset, including time, card details, merchant information, amount, cardholder's data, and fraud indicator. Trend spotting is aided by transaction time. Cardholders' histories are tracked using credit card numbers. Information about the merchant aids in identifying potential fraud. Quantity and merchant information reveal suspicious activity. Demographics identify at-risk populations. Fraud flag develops predictive detection methods.

B. DATASET 2

We identify credit card fraud using the CCFD dataset from KAGGLE. It records transaction information over time. Time since the initial transaction, V1-V28 PCA-transformed, and scaled numerical attributes are features. “Amount” denotes the value of the transaction. The target 'Class' indicates if there has been fraud (Class 1) or not (Class 0). This dataset combines time, amount, and transformed PCA features to help classification tasks to separate fraudulent from real transactions.

C. CNN-LSTM

The CNN-LSTM model for CCFD is an effective design that has two key components: a convolutional neural network (CNN) and an LSTM network. In the same way that a detective looks for clues, the CNN component looks for patterns in the transaction data that might point to fraud. It uses layers of convolutional filters to transform the input data and extract high-level features. These features are then passed on to the LSTM component, which is like a detective's memory. It remembers past transactions and sequences of features, looking for suspicious patterns and behavior over time. The LSTM component is composed of several layers of LSTM cells, which update their memory and hidden states as new input features are processed. Based on the patterns and sequences it has identified, the LSTM component's output predicts whether the input transaction is fraudulent or not. Overall, the CNN-LSTM model is a powerful tool for detecting CCF, capable of identifying complex patterns and sequences

of behavior that may be missed by traditional fraud detection methods.

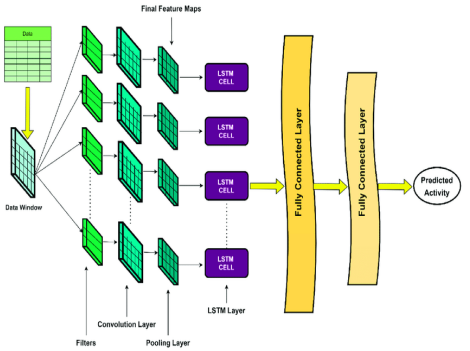


FIGURE 5. CNN-LSTM.

D. CNN

Convolutional neural networks (CNNs), a subclass of deep learning models, are extensively used for image and video recognition applications. They are made up of convolutional layers that automatically identify spatial patterns from input information like edges and textures. CNNs also employ pooling layers to reduce the size of the space and accelerate computation. To represent and classify high-level feature information, the architecture uses fully connected layers. The power of CNNs in computer vision applications comes from their capacity to automatically extract useful features from unprocessed data, enabling them to perform at the cutting edge in tasks like object detection, image classification, and other visual tasks.

Convolution Operation: Given an input feature map (picture) X and a convolutional kernel (filter) W, the convolution operation generates an output feature map Y as follows:

$$Y_{i,j} = \sum m \sum n \quad X_{i+m,j+n} \cdot W_{m,n}$$

In the output feature map, the element at position (i,j) is identified as Y(i,j). The element at location (i+m,j+n) in the input feature map is represented by the notation X(i+m,j+n). The element W(m,n) is located in the convolutional kernel at position (m,n). The sums are calculated for all m and n values that can fit inside the kernel's dimensions.

Pooling Operation: By choosing the maximum value in each pooling window, max pooling decreases the size of the feature map. The maximum pooling operation with a pooling window of size k × k is defined as follows for an input feature map X:

$$Y_{i,j} = max_{m,n}\{X_{i.k+m,j.k+n}\}$$

A direct link between the mathematical operations and the graphical representation of the CNN architecture is made possible by the visual representation of these equations within the corresponding layers.

E. LSTM

A recurrent neural network (RNN) architecture with the Long Short-Term Memory (LSTM) type can handle sequential data with long-distance dependencies in time series and natural language. The vanishing gradient issue affects conventional RNNs, whereas LSTM cells are unaffected. They instead employ a highly developed gating mechanism that enables them to selectively retain and forget information over time. This feature makes LSTMs very good at capturing temporal patterns and keeping important information for a long time. LSTMs have a wide range of uses in processes like sentiment analysis, language translation,

and speech recognition because of their ability to effectively model sequential data.

How much fresh data should be kept in the cell state is regulated by the input gate. The forget gate decides whether or not the prior cell state should be retained. The candidate cell state symbolizes the brand-new information that will be incorporated into the cell state. The cell state update uses the candidate cell state, the input gate, and the forget gate to update the cell state. The amount of the hidden state to be output from the cell is determined by the output gate. The hidden state, or output of the LSTM layer, is computed based on the output gate and cell state. Depending on the hidden state, the model generates the final prediction or output. These graphical representations show the data flow and transformations within the CNN and LSTM layers, giving a visual overview of the operations carried out in each layer.

To regulate the information flow via the network, LSTM units have three gates (input, forget, and output) and a cell state.

Input Gate:

$$i_t = \sigma(W_{xi}.x_t + W_{hf} \cdot h_{t-1} + b_i)$$

Forget Gate:

$$f_t = \sigma(W_{xf}.x_t + W_{hf} \cdot h_{t-1} + b_f)$$

Output Gate:

$$o_t = \sigma(W_{xo}.x_t + W_{ho} \cdot h_{t-1} + b_o)$$

Candidate Cell State:

$$\tilde{C}_t = tanh(W_{xc}.x_t + W_{hc} \cdot h_{t-1} + b_c)$$

Cell State Update:

$$C_t = f_t.C_{t-1} + i_t.\tilde{C}$$

Hidden State:

$$h_t = o_t.\tanh(C_t)$$

F. DENSE NN

In the field of deep learning, a dense neural network---also referred to as a fully connected neural network---plays a crucial role. This network architecture's key characteristic is how closely connected the neurons are across all of its layers. Every neuron within each layer forms complex connections with those in the layer below, creating a sophisticated lattice of information propagation. The network is given the ability to recognize and understand complex patterns, dependencies, and intrinsic representations embedded in the data thanks to this complicated design. A dense neural network often has an initial input layer that is in charge of taking in raw data. The data is then refined and reshaped progressively through one or more hidden levels, culminating in an output layer that outputs the final categorization or prediction made by the network.

A general layer in a deep neural network can be represented as:

$$Z^{[l]} = W^{[l]}.A^{[l-1]} + b^{[l]}$$

$$A^{[l]} = \sigma(Z^{[l]})$$

The layer index is l. Z^[l] is the linear combination of the activations A^[l−1] from the previous layer using the weights W^[l] and bias b^[l]. The activation function is σ. The output (activations) of the current layer is A^[l].

The Adam optimizer, recognized for its effectiveness in modifying the network's weights during training, is used in the model's construction. Additionally, the loss metric is the binary cross-entropy function, which measures the discrepancy between expected and actual values in binary

classification scenarios. Accuracy is the chosen performance metric, which gives an indication of how well the network predicts precise binary outcomes. Using the provided dataset, this integration of the architectural design, activation mechanisms, and optimization parameters creates a comprehensive blueprint for a dense neural network.

G. VOTING ENSEMBLE CLASSIFIER (LSTM+DNN+CNN)

LSTM, DNN, and CNN classifiers for binary classification using Keras and TensorFlow are combined using the Scikit-Learn VotingClassifier to create an ensemble model. The probabilities predicted by each individual classifier are averaged by the ensemble model to produce predictions (soft voting). After the ensemble model is trained on the training set of data, its performance has been assessed using the accuracy, confusion matrix, precision-recall curve, and F1 score on the test data. The precision-recall curve for the ensemble model is then shown, and the evaluation metrics are printed. By utilizing each classifier's advantages, this strategy seeks to enhance classification performance overall.

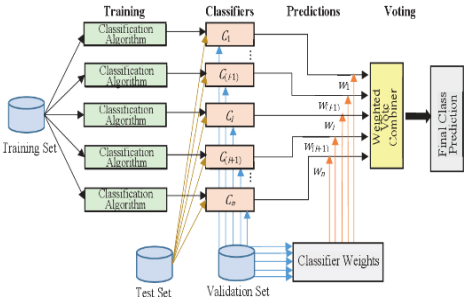


FIGURE 6. Voting ensemble classifier (CNN, LSTM, DNN).

IV. EXPERIMENTS & RESULTS

The banking industry faces a pressing issue of CCF, which makes it essential to detect and prevent fraudulent transactions. To tackle this challenge, we aim to develop a powerful machine learning model that can analyze customers' historical transactional data with various merchants and detect fraudulent activities accurately. Furthermore, we will provide a comprehensive cost-benefit analysis of our model to stakeholders and recommend practical measures to mitigate fraud risks. With this approach, we can empower banks to stay ahead of potential fraudulent transactions and minimize risks, providing a safe and secure environment for customers to conduct transactions.

A. UNDERSTANDING DATA

For our research, we make use of two datasets that contain different transactional information and consumer data. Using these variables, our goal is to forecast fraud ('is_fraud'). We'll change 'trans_date_trans_time' to datetime to improve analysis. The 'is_fraud' data balance evaluation is essential. We remove hours, days, months, and years from “trans_date_trans_time” and discard the original column because it has so many different unique values. Age is calculated for fraud analysis using “dob.” 'Dob,' 'first,' and 'last' are superfluous and are dropped due to privacy concerns. For the sake of efficiency and privacy, the terms “trans_date_trans_time,” “dob,” “first,” and “last” are eliminated.

B. DATA PREPROCESSING

Included in here are methods for dealing with missing values, encoding variables for usage with numbers, and scaling for uniformity. Accuracy is ensured by how you handle missing data. Categorical-numerical conversion is made possible by variable encoding, which helps our ML model. Analysis consistency is ensured by scaling. Our CCFD model is set up by these preparation processes.

Data cleaning: This process involves cleansing the data, dealing with missing numbers, and fixing errors. We correct sequence length differences for uniformity via padding (adding zeros) or truncation (removing items). After padding and tokenization, we numerically encode the data. Words are converted into high-dimensional vectors using methods like word embeddings and one-hot encoding. Splitting the data: The train, validation, and test sets are created from the preprocessed data. While training and validation sets hone hyperparameters, validation sets train the model. By applying changes like rotation or scaling, data augmentation broadens training data. Time series data are best suited by a sequential model, which layers linearly for learning hierarchical features. Sequential models are defined using classes in frameworks like TensorFlow or Keras, and additional layers are added with predefined parameters.

C. EXPLORATORY DATA ANALYSIS [EDA]

Dataset study reveals that only 0.57% of transactions are fraudulent, whereas 99.42% are not. For unbiased findings, balancing out the scales is essential. Equitable representation is ensured by balanced data, improving accuracy. The distribution of transaction amounts suggests a higher mean for fraud, highlighting the need to identify and stop high-loss fraudulent activity.

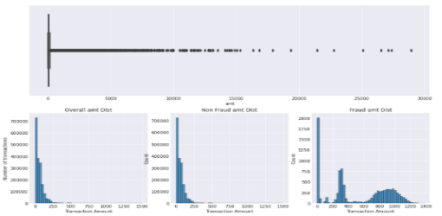


FIGURE 7. Amount distribution according to fraud and non-fraud.

A distinct pattern can be seen in Figure 7 when the total, fraudulent, and non-fraudulent transaction amounts are compared. The mean non-fraudulent amount is significantly lower than the mean fraudulent amount, indicating a greater average transaction value for fraud. Due to the significant financial losses associated with fraud cases, it is vital to identify and stop these behaviors in order to reduce risk.

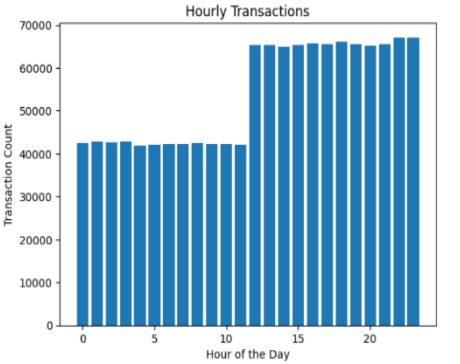


FIGURE 8. Hourly transaction.

Figure 8 presents the hourly transactions, which shows on which hours a large number of transactions occur. This gives us the point that if there is any other transaction which does not meet with that timeline or amount, then that transaction can be fraudulent.

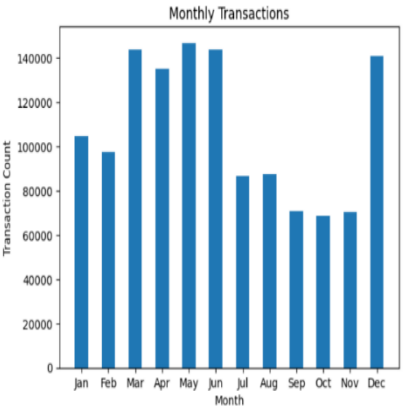


FIGURE 9. Monthly transaction.

Figure 9 presents the monthly transactions. It shows how many transactions occur in which month; the x-axis shows the count and the y-axis shows the name of the months.

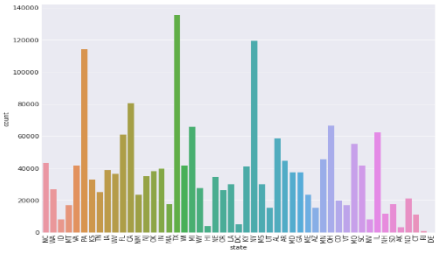


FIGURE 10. Year and monthly transaction.

Figure 10 describes the plot of year-month versus fraud transactions and fraud customers. Since there are a lot of categories in the 'state' variable of the data, we will now only look at the top 20 states wherein the transaction frequencies are the highest.

D. FEATURE ENCODING

Features like 'cc_num', 'first', 'last', and 'trans_num' appear to be unnecessary and can be removed. For effective encoding, 'Merchant' with several unique values may be deleted. The challenges of "street," "city," "state," and "job" are all related. To simplify the dataset for clear yet informative analysis, encoded variables like "age," "category," "gender," and "trans_day_of_week" can be thought to be unnecessary.

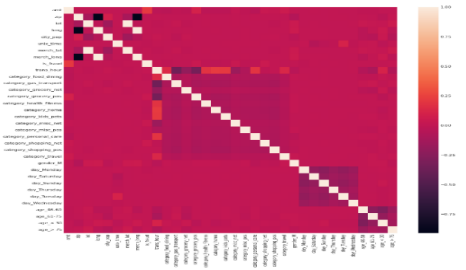


FIGURE 11. Confusion matrix.

Given the large number of variables in the dataset, it may be beneficial to identify the variables with high correlation using a function that outputs only those variables whose correlation is above a certain threshold. This approach can help to streamline the dataset and focus on only the most relevant features for subsequent analysis. Using such a function, we can easily identify the variables with high correlation and exclude those that are redundant or not relevant to our analysis. By doing so, we can reduce the dimensionality of the dataset while retaining only the most informative features, allowing us to make more accurate predictions and draw meaningful insights.

TABLE 1. Test results of four classifiers on dataset 1: credit card transactions fraud detection.

References	Models	Other papers Accuracy	Our Dataset 1 Accuracy	Our Dataset 2 Accuracy
3	CNN, LSTM	94.45	98.5%	97.12 %
5	Dense Net, RNN	95.30 %	98.09 %	98.32 %
8	CNN, Dense Net	93.50 %	98.70 %	98.77 %
	Purposed Voting Ensemble Classifier	Null	99.94	99.95 %

In the methodology part A section, we define the dataset 1. Here is the result of four classifiers in Table 1 which we use in this dataset; the voting ensemble classifier is the mixture of these three algorithms which is made by us.

TABLE 2. Test results of four classifiers on dataset 2: credit card fraud detection.

Dataset 2	Model	Accuracy	Previous Accuracy
CCFD	LSTM	98.5%	94.5%
CCFD	Densenet	99.13%	Null
CCFD	CNN	98.6%	93%
CCFD	Voting Ensemble Classifier	99.93%	Null

Table 2 shows the accuracy and test results of four classifiers which we use on dataset 2. In the previous accuracy section, null shows that nobody used that classifier on this dataset.

E. GRAPHICAL REPRESENTATION OF TEST RESULTS

A lower validation loss indicates higher generalization, and a lower training loss indicates learning, according to the validation and loss graph, which records model development. The ROC curve shows how sensitivity and specificity change at various classification criteria. The assessment of class separation and threshold selection are aided by the ROC curve. The precision-recall balance over thresholds may be seen in the F1 score graph. Better predictions are indicated by a higher F1 score, which is important for unbalanced datasets. In conclusion, the ROC measures binary performance, the F1 score measures effectiveness at thresholds, and the loss graph monitors training progress.

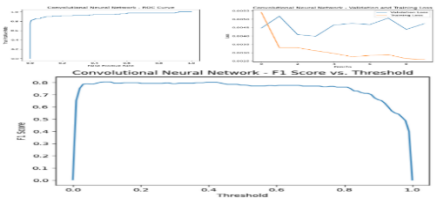


FIGURE 12. CNN (ROC, validation & training).

Figure 12 shows the ROC curve and validation and training loss graph of the CNN (Convolutional Neural Network). With these graphs we can clearly see how our model is working and how it is giving the output.

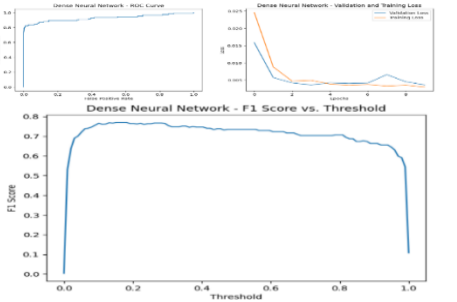


FIGURE 13. LSTM (ROC, validation & training, F1).

Figure 13 shows the validation and training loss graph and ROC curve of the LSTM (Long Short-Term Memory).

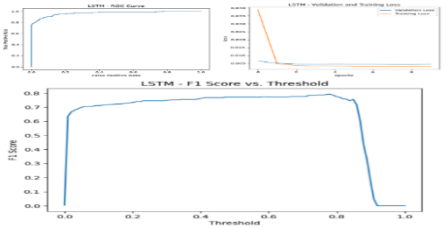


FIGURE 14. Dense neural network (ROC, validation & training, F1).

Figure 14 shows the ROC curve and validation and loss graph of the dense neural network.

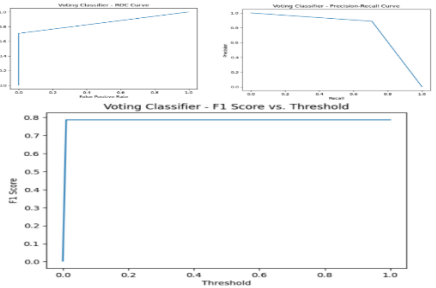


FIGURE 15. Ensemble voting classifier (ROC, validation & training, F1).

Figure 15 shows the ROC curve and precision-recall curve graph of the ensemble voting classifier. The ensemble voting classifier is made by three classifiers: LSTM, DenseNet, and CNN, in which the trained classifiers are used together to conduct the testing.

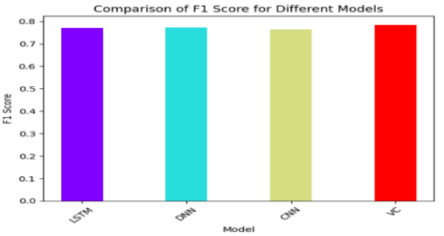


FIGURE 16. F1 score comparison.

Figure 16 shows the F1 score of three different models: LSTM (Long Short-Term Memory), DNN (Dense Neural Network), and CNN (Convolutional Neural Network), plus the voting classifier. A greater F1 score indicates that the model strikes a more effective equilibrium between precision and recall, demonstrating overall strong performance.

V.CONCLUSION & FUTURE WORK

Transactions increase after midday and on holidays, and fraud is also more prevalent. Over 75s are particularly vulnerable since they are less familiar with the fraudsters' shifting techniques. Targets tend to be more frequently 'female' people. Certain zip areas, employment profiles, and demographics like the “DE” state have 100% fraud rates, which calls for investigation.

High fraud rates in the 'gas_transport,' 'grocery_pos,' 'home,' 'shopping_pos,' and 'kids_pets' categories may indicate compromised POS or gateways. Treutel, Dickens, 'fraud_Kozey-Boehm,' 'fraud_Herman,' and other merchants exhibit irregularities that may indicate fraud. By using three methods and sampling from 12 balanced and unbalanced models, modeling eliminates unimportant information. The F1 score of 0.100, 0.98 precision, and 0.98 recall make the Tuned Random Forest Classifier stand out.

The process of one-hot encoding, which groups comparable features for model training, is the following phase. CCFD security is transformed by cutting-edge technology: AI examines user habits for quick anomaly notifications. Biometrics verify via fingerprint and facial recognition while blockchain guarantees immutable transactions. Rapid, precise fraud prediction is made possible by quantum computing and predictive analytics. Delay in detection is decreased by real-time processing via 5G and edge computing. Innovative cooperation in cybersecurity, neural networks, and anomaly detection actively battle fraud. Visualization is aided by AR, NLP monitors interactions, and international data sharing reduces cross-border fraud. Accounts are frozen by automation, and continual verification is guaranteed by zero trust architecture. Models that are adaptable promise a highly secure financial environment that is impervious to the effects of fraud.

Author Contribution

Abdullah Mehboob: conceptualization, methodology, data Curation, software, formal analysis, investigation, visualization, validation, writing – original draft, writing – review & editing, and project administration.

Conflict of Interest

The authors of the manuscript have no financial or non-financial conflict of interest in the subject matter or materials discussed in this manuscript.

Data Availability Statement

Data supporting the findings of this study will be made available by the corresponding author upon request.

Funding Details

No funding has been received for this research.

Generative AI Disclosure Statement

The authors did not use any type of generative artificial intelligence software for this research.

REFERENCES

[1] Sumanth, P. P. Kalyan, B. Ravi, and S. Balasubramani, "Analysis of CCFD using machine learning techniques," in 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, Jun. 2022, pp. 1140–1144, doi: 10.1109/ICCES54183.2022.9835751.

[2] M. V. Krishna and J. Praveenchandar, "Comparative analysis of CCFD using logistic regression with random forest towards an increase in accuracy of prediction," in 2022 International Conference on Edge Computing and Applications (ICECAA), Tamilnadu, India, Oct. 2022, pp. 1097–1101, doi: 10.1109/ICECAA55415.2022.9936488.

[3] H. Najadat, O. Altiti, A. A. Aqouleh, and M. Younes, "CCFD based on machine and deep learning," in 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, Apr. 2020, pp. 204–208, doi: 10.1109/ICICS49469.2020.239524.

[4] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, "CCFD using AdaBoost and majority voting," IEEE Access, vol. 6, pp. 14277–14284, 2018, doi: 10.1109/ACCESS.2018.2806420.

[5] A. M. Babu and A. Pratap, "CCFD using deep learning," in 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS), Thiruvananthapuram, India, Dec. 2020, pp. 32–36, doi: 10.1109/RAICS51191.2020.9332497.

[6] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed, "CCFD 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.

[7] A. A. El Naby, E. El-Din Hemdan, and A. El-Sayed, "Deep learning approach for CCFD," in 2021 International Conference on Electronic Engineering (ICEEM), Menouf, Egypt, Jul. 2021, pp. 1–5, doi: 10.1109/ICEEM52022.2021.9480639.

[8] Z. Li, G. Liu, and C. Jiang, "Deep representation learning with full center loss for CCFD," IEEE Trans. Comput. Soc. Syst., vol. 7, no. 2, pp. 569–579, Apr. 2020, doi: 10.1109/TCSS.2020.2970805.

[9] R. San Miguel Carrasco and M.-A. Sicilia-Urban, "Evaluation of deep neural networks for reduction of CCF alerts," IEEE Access, vol. 8, pp. 186421–186432, 2020, doi: 10.1109/ACCESS.2020.3026222.

[10] S. K. Saddam Hussain, E. Sai Charan Reddy, K. G. Akshay, and T. Akanksha, "Fraud detection in credit card transactions using SVM and random forest algorithms," in 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, Nov. 2021, pp. 1013–1017, doi: 10.1109/I-SMAC52330.2021.9640631.

[11] S. V. J. B. Gracia, J. G. Ponsam, S. Preetha, and J. Subhiksha, "Payment

fraud detection using machine learning techniques," in 2021 4th International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, Dec. 2021, pp. 623–626, doi: 10.1109/ICCCT53315.2021.9711887.

[12] A. Thennakoon, C. Bhagyani, S. Premadasa, S. Mihiranga, and N. Kuruwitaarachchi, "Real-time CCFD using machine learning," in 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2019, pp. 488–493, doi: 10.1109/CONFLUENCE.2019.8776942.

[13] B. Can, A. G. Yavuz, E. M. Karsligil, and M. A. Guvensan, "A closer look into the characteristics of fraudulent card transactions," IEEE Access, vol. 8, pp. 166095–166109, 2020, doi: 10.1109/ACCESS.2020.3022315.

[14] N. Nguyen, T. Duong, T. Chau, V.-H. Nguyen, T. Trinh, D. Tran, and T. Ho, "A proposed model for card fraud detection based on CatBoost and deep neural network," IEEE Access, vol. 10, pp. 96852–96861, 2022, doi: 10.1109/ACCESS.2022.3205416.

[15] A. Kundu, S. Panigrahi, S. Sural, and A. K. Majumdar, "BLAST-SSAHA hybridization for credit card fraud detection," IEEE Trans. Dependable Secure Comput., vol. 6, no. 4, pp. 309–315, 2009, doi: 10.1109/TDSC.2009.11.

[16] 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.

[17] S. K. Hashemi, S. L. Mirtaheri, and S. Greco, "Fraud detection in banking data by machine learning techniques," IEEE Access, vol. 11, pp. 3034–3043, 2023, doi: 10.1109/ACCESS.2022.3232287.

[18] L. Zheng, G. Liu, C. Yan, C. Jiang, M. Zhou, and M. Li, "Improved TrAdaBoost and its application to transaction fraud detection," IEEE Trans. Comput. Soc. Syst., vol. 7, no. 5, pp. 1304–1316, Oct. 2020, doi: 10.1109/TCSS.2020.3017013.

[19] H. Wang, W. Wang, Y. Liu, and B. Alidaee, "Integrating machine learning algorithms with quantum annealing solvers for online fraud detection," IEEE Access, vol. 10, pp. 75908–75917, 2022, doi: 10.1109/ACCESS.2022.3190897.

[20] F. Ögme, A. G. Yavuz, M. A. Guvensan, and M. E. Karslıgil Yavuz, "Temporal transaction scraping assisted point of compromise detection with autoencoder based feature engineering," IEEE Access, vol. 9, pp. 109536–109547, 2021, doi: 10.1109/ACCESS.2021.3101738.

[21] L. Zheng, G. Liu, C. Yan, and C. Jiang, "Transaction fraud detection based on total order relation and behavior diversity," IEEE Trans. Comput. Soc. Syst., vol. 5, no. 3, pp. 796–806, 2018, doi: 10.1109/TCSS.2018.2856910.

[22]A. Ali, S. Abd Razak, S. H. Othman, T. A. E. Eisa, A. Al-Dhaqm, M. Nasser, T. Elhassan, H. Elshafie, and A. Saif, "Financial fraud detection

based on machine learning: A systematic literature review," Appl. Sci., vol. 12, no. 19, Art. no. 9637, 2022, doi: 10.3390/app12199637.

[23] 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.

[24] I. D. Mienye and Y. Sun, "A deep learning ensemble with data resampling for credit card fraud detection," IEEE Access, vol. 11, pp. 30628–30638, 2023, doi: 10.1109/ACCESS.2023.3262020.

[25] E. Ileberi, Y. Sun, and Z. Wang, "Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoost," IEEE Access, vol. 9, pp. 165286–165294, 2021, doi: 10.1109/ACCESS.2021.3134330.

[26] T. A. Olowookere and O. S. Adewale, "A framework for detecting credit card fraud with cost-sensitive meta-learning ensemble approach," Sci. Afr., vol. 8, Art. no. e00464, 2020, doi: 10.1016/j.sciaf.2020.e00464.

[27] N. F. Aurna, M. D. Hossain, L. Khan, Y. Taenaka, and Y. Kadobayashi, "FedFusion: Adaptive model fusion for addressing feature discrepancies in federated credit card fraud detection," IEEE Access, vol. 12, pp. 136962–136978, 2024, doi: 10.1109/ACCESS.2024.3464333.

[28] A. Mehboob, M. S. Farooq, M. F. Manzoor, M. Shaheen, and J. N. Qureshi, "Secure and scalable federated learning for citrus disease detection with CNN-ANN and data privacy for agriculture," *Cogent Food & Agriculture*, vol. 12, no. 1, 2026, doi: 10.1080/23311932.2026.2690308.