Journal of Information Systems Engineering and Management
2025, 10(53s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Efficient Framework Based on Optimized CNN-RNN for Online Transaction Fraud Detection in Financial Transactions

T. Madhavappa*[1] ,Bachala Sathyanarayana[2]
[1]*Research Scholar, Department of Computer Science and Technology, Sri Krishnadevaraya University, Ananthapuramu, Andhra Pradesh-515003, India.*Corresponding author E-mail id: madhava998@gmail.com*
[2]*Professor, Department of Computer science and Technology, Sri Krishnadevaraya University, Ananthapuramu, Andhra Predesh-515003, India, E-mail id : bachalasatya@gmail.com*

ARTICLE INFO

Received: 31 Dec 2024

Revised: 20 Feb 2025

Accepted: 28 Feb 2025

ABSTRACT

Financial fraud is becoming a serious worry in a world where wireless communications are essential, aimed at transmitting enormous amounts of information while guarding against interference. This paper presents an efficient conceptual architecture (ECA) to detect and flag fraudulent transactions with consideration of various financial transactions. An accurate and robust model has been developed for detecting fraud in online payments. In addition to addressing the dynamic nature of fraudulent behavior and the intolerability requirements that are essential for financial institutions, the suggested system model aims to improve the accuracy of online fraud detection. Initially, the online data is collected and considered as the feature set. After that, the pre-processing step is considered, which consists of normalization and filling in missing parameters. The collected database may contain the label imbalance issue. To solve this issue, the Synthetic Minority Over-Sampling Technique (SIvIOTt) is utilized. To extract the features and identify fraudulent payments, the Optimized Convolutional Neural Network-Recurrent Neural Network (OCNN-RNN) is developed. In this architecture, the CNN layer is utilized to select the features from the input data, and BiLSTM is utilized for sequence detection to obtain the required outcomes. Additionally, the DNN is utilized to optimize the error and loss by using the Enhanced Gazelle Optimization Algorithm (EGOA). Finally, the proposed architecture is used for online fraud detection. The proposed method is implemented in Python, and performances are evaluated by performance measures.

Keywords: Efficient conceptual architecture, Synthetic Minority Over-Sampling Technique (SIvIOTt), convolutional neural network (CNN), recurrent neural network(RNN), online payment fraud detection, Enhanced Gazelle Optimization Algorithm (EGOA).

1. INTRODUCTION

These days, most transactions are done online, and online transaction fraud is becoming a significant risk to financial security [Vanini,et al,2023]. In the event of online transaction fraud, banks will sustain damages in terms of money and reputation. For certain clients, it not only results in financial loss but also casts a psychological shadow [Wei, et al,2023]. Nevertheless, it is challenging to quantify the damage caused by online transaction fraud. While many banks and businesses are reluctant to share the cases [Long,et al,2023], fraud detection takes longer than expected, which makes it more difficult to estimate losses. These days, one of the biggest risks to Internet security is fraudulent transactions. In a cloud setting, artificial intelligence is essential for controlling financial risk. Numerous research endeavours [Yu,Wangyang,et al, 2023] have endeavoured to investigate techniques for identifying online transaction fraud, yet the current approaches seem inadequate for achieving highly accurate detection [Boulieris, Petros,et al,2023].

Financial institutions that issue credit cards or conduct online transactions need automated fraud detection [Kannagi, A., J. Gori Mohammed,et al,2023]. This decreases losses and boosts client confidence. Big data and AI make advanced machine learning models more effective in detecting fraud [Karthika,et al,2023]. Current machine learning, deep learning, and advanced data mining-based fraud detection technologies work incredibly well [Karthikeyan, T., M. Govindaraj,et al,2023]. A binary classification node that can differentiate between fraudulent and legitimate transactions is constructed using a data set that contains labeled transactions (both regular and fraudulent) [Abd El-Naby,et al,2023]. The model is used to determine if incoming transactions are fraudulent or normal. There are a number of difficulties in applying classification algorithms to identify fraudulent transactions. Some of the factors that need to be considered are: class imbalance (a very small ratio of fraudulent to normal transactions)[Bakhtiari,et al,2023], cost sensitivity (different costs for correctly and incorrectly classifying fraudulent old normal transactions), temporal dependence between transactions [Hajek, Petr,et al,2023], concept drift (class conditional distributions changing over time and necessitating classifier updates), and dimensionality of search space feature pre-processing [Thimonier, Hugo et al,2023].

There is new and exciting potential in the domains of big data and artificial intelligence, particularly when it comes to using powerful machine learning algorithms to combat financial crime. The Modern machine learning and deep learning algorithms, along with data analysis, have made fraud detection systems highly successful. Such algorithms are typically trained on enormous labelled transaction datasets, allowing them to distinguish fraudulent from legal activities [Ghaleb, Fuad A., et al,2023]. In the end, binary classification models that identify genuine vs. fraudulent activities are created. Using classification algorithms to detect fraudulent activities is a difficult and well-known challenge for which novelty detection hybrid solutions have been invented [Banirostam,et al,2023]. The financial sector needs to develop constantly to remain ahead of financial crime in the same manner that invention ensures security, data availability, reliability, and resilience in the face of cyberwarfare attacks in the fight against wired communication interference [Al-Sayyed, Rizik,et al,2024]. Researchers develop convolutional neural networks, support vector machines, deep neural networks, and artificial neural networks for classification and detection. After that, CNN has several benefits, but choosing the right hyperparameters can improve it. Through the use of an optimization technique, the hyperparameter selection is determined.

The main contribution and organization of the research

- This paper presents an efficient ECA to detect and flag fraudulent transactions, taking into consideration various financial transactions. An accurate and robust model has been developed for detecting fraud in online payments.
- In addition to addressing the dynamic nature of fraudulent behavior and the interoperability requirements that are essential for financial institutions, the suggested system model aims to improve the accuracy of fraud detection.
- Initially, the online data is collected and considered a feature set. After that, the pre-processing step is considered, which consists of normalization and filling in missing parameters. The collected database may contain a label imbalance problem. To solve this issue, the Synthetic Minority Over-Sampling Technique (SMOTE) is utilized.
- To extract the features and identify fraud payments, the Optimized Convolutional Neural Network-Recurrent Neural Network (OCNN-RNN) is developed. In this architecture, the CNN layer is utilized to select the features from the input data, and BiLSTM is utilized for sequence detection to obtain the required outcomes.
- Additionally, the DNN is used to optimize error and loss using the Enhanced Gazelle Optimization Algorithm (EGOA).

The schedule for the remaining section of the paper is as follows: Section 2 contains the relevant recent papers on online fraud detection. Section 3 shows the full proposed architecture. Section 4 provides the results and a description of the recommended course of action. Section 5 provides a conclusion about the current paper.

2. LITERATURE REVIEW

Systems for detecting fraud are used to spot oddities in electronic payment transactions. The analyse a number of fraud detection systems in this study, which may be broadly classified into two types: systems that aggregate the original data and those that exploit the original features. An outline of the review is given in the end of this section.

The rapid growth of the internet under globalization across the world has heightened businesses in various industries and particularly in the financial sectors to provide their important services through online platforms. This has intensified financial fraud creating a huge revenue collection due to constant financial loss. Recognizing threats such as frequent attacks and unwanted access is essential to combating this problem. In recent years, there has been significant use of machine learning and data mining approaches to address this problem. ANFIS (Adaptive Neuro-Fuzzy Inference System) for fraud detection and FW-GWO (Fire Work-Grey Wolf Optimization) for feature selection have been presented by N. Krishnavardhan et al. [2024]. This study focuses on three types of fraud datasets: loan, credit card, and insurance fraud datasets.

A machine-learning approach for detecting payment card fraud has been presented by Manjeevan Seera et al. [2024], using both real and publicly available transaction information. Using a payment card to make a transaction is easy and convenient. Fraud incidents are increasing as a result of the growing use of payment cards, particularly for online purchases. Due to the annual billions in losses in the commercial sector, the rise entails financial risk and uncertainty. Nevertheless, authentic transaction records that can aid in the creation of efficient prediction models for fraud identification are challenging to locate, mostly due to concerns about client data privacy.

ResNeXt-embedded Gated Recurrent Unit (GRU) model (RXT) is an AI method for real-time financial transaction data processing Almazroi et al. [2023]. It reduces data imbalance and reveals crucial data patterns to combat financial fraud. Hyperparameters optimise the model's classification task. After through testing on three real financial transaction datasets, the model routinely outperforms existing algorithms by 10% to 18%, proving its potential to improve cyberwarfare security, data availability, reliability, and stability. Though feature engineering improves the design discriminative abilities, feature extraction employs an artificial intelligence ensemble strategy which syndicates ResNet (EARN) and autoencoders to uncover important data designs. This Al classification challenge is centered around the RXT model, which has been optimized using hyperparameters through the use of the Jaya optimization algorithm (RXT-J).

A multi-layer system based on machine learning has been presented by A. Asad Arfeen et al. [2023] to identify and categorize abnormal financial transactions. A financial service provider can prevent occurrences such as invasions and online fraud by using the suggested structure. In order to increase the legitimacy of these online financial platforms or gateways, it also offers a safe tool to identify network irregularities in financial transactions.

A two-stage approach for detecting fraudulent transactions has been presented by Hosein Fanai et al. [2023], which syndicates supervised deep learning techniques with a deep autoencoder as a depiction learning method. The outcomes of the investigational assessments presented that the deep learning-based classifiers performed better when using the suggested approach. In specific, the used deep learning classifiers perform much better across all performance measures than their baseline classifiers learned on the original information after being learned on the changed database produced by the deep autoencoder.

Lizhi Wanget al. [2021], Fraudulent transactions pose a significant threat to online security, and Artificial Intelligence is crucial for financial risk control in cloud environments. This chapter proposes a Deep-forest-based approach for online transaction fraud detection, integrating differentiation feature generation and deep-forest based models. The scheme uses transaction time-based differentiation features, such as Individual Credibility Degree (ICD) and Group Anomaly Degree (GAD), to distinguish between legal and fraudulent transactions. The Deep-forest algorithm is applied to detect extreme imbalances in online transactions, enhancing precision by focusing on outliers. Tests on one bank's transaction data show a 15% improvement in precision rate and a 20% recall rate.

Karim Zkik et al. [2024]. Online retail platforms face increasing challenges like cyber-attacks, data breaches, and operational disruptions. Conventional cybersecurity methods are insufficient against sophisticated cybercrime tactics. This paper proposes a novel resilience strategy using Explainable Deep Learning technologies and a Blockchainin-based consensus protocol. This strategy enables rapid incident detection, explains vulnerabilities, and enhances decision-making during cyber incidents. Experiments using NAB datasets and real online retail architectures validate the effectiveness of the proposed framework in supporting business continuity and creating efficient cyber resilience strategies.

Elberri, et al. [2024]. M.APhishing attacks pose a significant threat to online security, and traditional CNN-based image classification methods struggle to identify fake pages. A CNN-LSTM hybrid model is proposed to address this issue. This approach combines SMOTE, an enhanced GAN, and swarm intelligence algorithms to balance the dataset and generate grayscale images. Experiments show superior accuracy, precision, and sensitivity, enhancing online security.

X. Jiang et al. [2019]. Artificial intelligence is enabling the insurance industry to create personalized solutions and services based on consumer knowledge. However, insurance data is heterogeneous and imbalanced, presenting challenges for machine learning. This paper proposes an efficient cost-sensitive parallel learning framework (CPLF) to enhance insurance operations without pre-processing. The framework uses a unified cost-sensitive parallel neural network to learn real-world heterogeneous data, with a cost-sensitive matrix automatically generating a robust model for minority classifications. The CPLF-based architecture is studied for real-world insurance intelligence operation systems, with fraud detection and policy renewal experiments demonstrating its effectiveness.

Terumalasettiet al. [2024].The rise of automated accounts on social networks has led to the need for advanced detecting systems. However, identifying malicious users is a challenging task due to the potential for unconstitutional actions or sensitive data. AIMDS (Artificial Intelligence-based malicious user Detection System) is an innovative method that combines RNN, CNN, and PSO to detect bot accounts. This method aims to identify inconsistent user patterns by considering vast amounts of data generated by digital devices, particularly in social networks. The research compared the effectiveness of AIMDS to traditional AI-based algorithms, revealing improved accuracy compared to traditional methods.

J. G. Sherwin Akshay et al. [2024]. This paper explores deep learning models for detecting and distinguishing fraudulent transactions from normal ones. It focuses on converting tabular data to graphs for better analysis. The methods use ensemble learning techniques and unsupervised deep neural networks, including Auto-Encoders and Recurrent Neural Networks. The study uses Louvain Modularity, Girvan-Newman, and Label Propagation Algorithm for community detection. The results show reliable results on credit card transaction datasets. Comparing the approach to traditional fraud detection methods demonstrates its superiority in identifying fraudulent transactions.

F. K. Alarfaj and S. Shahzadiet al. [2024].This study explores the use of deep learning for fraud detection in large organizations, focusing on Graph Neural Networks (GNNs) and Autoencoder. The research proposes a Graph neural network with lambda architecture for real-time fraud detection, while an autoencoder is used for credit card fraud detection. The findings show that these methods effectively detect fraud with a balance of precision and recall, improving the efficiency of banking systems. Python is used for analysis, highlighting the potential of deep learning in managing and preventing fraud in real-time on dynamic datasets.

D. Mienye and Y. Sun et al. [2023].Credit card usage in the digital economy has increased, leading to an increase in fraud. Machine learning algorithms have been used to detect this issue, but the dynamic shopping patterns and class imbalance problem make optimal performance challenging. This paper proposes a deep-learning approach using long short-term memory and gated recurrent unit neural networks, a multilayer perceptron as a meta-learner, and a hybrid synthetic minority oversampling technique and edited nearest neighbour method. The combination of these methods achieved a sensitivity and specificity of 1.000 and 0.997, respectively.

Chithanuru V, et al. [2023]. Block chaining technology, a cryptographic distributed transaction ledger, has become increasingly popular due to its tamper-proof transactions and robustness against cyber-attacks. However, adversaries still attempt to detect vulnerabilities in the infrastructure. This article aims to address these security breaches by detecting and mitigating anomalies using Artificial Intelligence Techniques. The review explores the security aspects of Block chaining, its infrastructure vulnerabilities, and various Block chaining-enabled use cases. It also analyses the use cases of AI in detecting anomalies with block chain and highlights the potential benefits, challenges, and future directions of integrating block chain with AI techniques.

Rani, Y.A., Reddyet al. [2024]. Cyber-attacks have increased the need for network intrusion detection systems to protect source data and individual privacy. Existing models struggle to efficiently protect target networks based on statistical features. A new meta-heuristic hybrid-based deep learning model is introduced to address these issues. The model uses pre-processing, auto-encoder, and IChOA to extract significant features from pre-processed data. The optimal features are then subjected to the hybrid deep learning model, DINet, which incorporates a deep temporal convolution network and gated recurrent unit. The model provides 97% accuracy and precision, enhancing data transmission significantly and securely. This enhanced model demonstrates the effectiveness of detecting malware, improving data transmission and enhancing overall security. The model is compared to previous detection approaches and is recommended for effective malware detection.

Table 1: Problem formulation for Research gap of the recent research

Authors	Methods	Research Gaps	Limitations	Advantages	Disadvantages
N. Krishnavardhan et al. [2024]	ANFIS for fraud detection, FW-GWO for feature selection	Focus on three datasets: loan, credit card, and insurance fraud	Limited dataset and lack of consideration for real-time fraud detection scenarios	Adaptive detection capabilities and effective feature selection	Scalability challenges and potential overfitting
Manjeevan Seera et al. [2024]	Machine learning for payment card fraud	Lack of publicly available authentic transaction records	Client privacy concerns limit dataset availability	Easy detection and convenience for users	Limited dataset diversity affecting prediction accuracy
Almazroi et al. [2023]	ResNeXt-embedded GRU with Jaya optimization (RXT-J)	Data imbalance and feature extraction challenges	Requires extensive hyperparameter optimization	Improved data processing, classification accuracy, and cyber security	High computational cost
A. Asad Arfeen et al. [2023]	Multi-layer ML system for abnormal transaction detection	Enhancement in legitimacy for online platforms	Potential false positives or negatives in detection	Secures financial transactions and improves platform credibility	Computational complexity in multi-layer implementation
Hosein Fanai et al. [2023]	Deep learning with deep autoencoder	Optimizing classifiers for better performance on imbalanced datasets	Dependency on data pre-processing	Enhanced fraud detection accuracy	Limited scalability for large-scale datasets
Lizhi Wang et al. [2021]	Deep-forest with differentiation feature generation	Handling extreme imbalance in online transaction datasets	Requires advanced feature engineering	Improved precision and recall rates	High computational requirements for deep-forest models
Karim Zkik et al. [2024]	Explainable DL with	Addressing sophisticated	Implementation complexity and	Rapid incident	Blockchain integration cost

	Blockchain-based consensus protocol	cybercrime tactics	high resource requirements	detection and enhanced decision-making	
Elberri et al. [2024]	CNN-LSTM hybrid with SMOTE, GAN, and swarm intelligence	Identifying fake pages effectively	May struggle with real-time adaptability	Enhanced accuracy, precision, and sensitivity	Dataset balancing challenges
X. Jiang et al. [2019]	Cost-sensitive parallel learning framework (CPLF)	Efficient handling of heterogeneous and imbalanced insurance data	Requires cost-sensitive matrix design	Robust model for minority classifications	Potentially resource-intensive
Terumalasetti et al. [2024]	AIMDS using RNN, CNN, and PSO for malicious user detection	Detecting malicious users effectively	Limited dataset diversity for training	Improved accuracy and detection of inconsistent user patterns	Computational overhead
J. G. Sherwin Akshay et al. [2024]	DL models with Auto-Encoders and RNN for fraudulent transaction detection	Analysing tabular data as graphs	Limited to specific datasets and scenarios	Reliable detection of fraud with superior analytical techniques	High processing time
F. K. Alarfaj and S.Shahzadi [2024]	Graph Neural Network with lambda architecture and autoencoder for fraud detection	Real-time fraud detection in dynamic datasets	Limited applicability for diverse datasets	Efficient detection with a balance of precision and recall	Scalability issues
D. Mienye and Y. Sun et al. [2023]	DL approach using LSTM, GRU, and hybrid SMOTE-ENN for credit card fraud detection	Addressing dynamic shopping patterns and class imbalance	Requires pre-processing for optimal results	High sensitivity and specificity	Resource-heavy for large datasets
Chithanuru V, et al. [2023]	AI techniques for	Addressing vulnerabilities	High implementation	Robust against	Lack of real-time adaptability

	Blockchain anomaly detection	in Blockchain infrastructure	complexity	cyber-attacks and secure transactions	
Rani, Y.A., Reddy et al. [2024]	Meta-heuristic hybrid DL model with autoencoder and IChOA for intrusion detection	Efficient protection of target networks	Requires advanced feature extraction methods	High accuracy and precision in malware detection	Implementation challenges in complex network environments

3. MATERIALS AND METHODS

The purpose of this section of the ECA is to identify and highlight financial transactions that contain fraudulent activity. The suggested architecture model was created to enhance the accuracy of fraud detection, handle the dynamic general of fraudulent dues, and meet the interpretability standards that are essential for financial institutions. Figure 1 displays the proposed system's architecture.

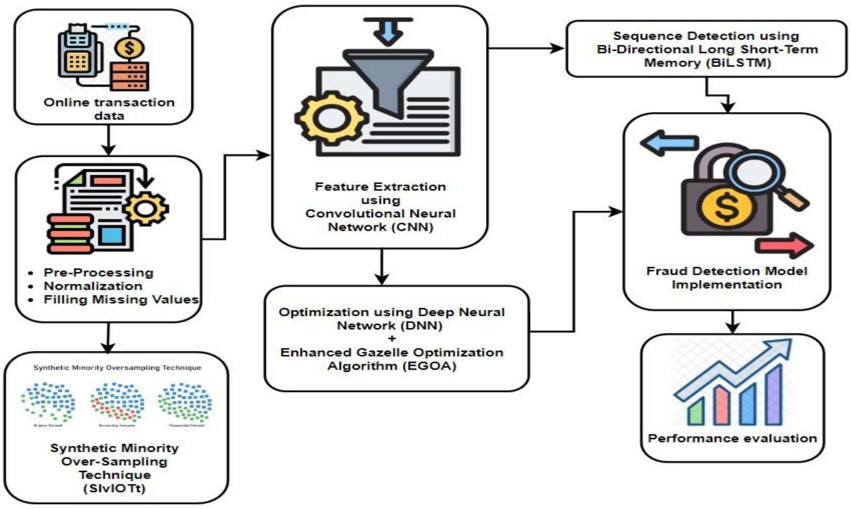
Online transaction data
Pre-Processing
Normalization
Filling Missing Values
Synthetic Minority Oversampling Technique
Synthetic Minority Over-Sampling Technique (SlvIOTt)
Feature Extraction using Convolutional Neural Network (CNN)
Optimization using Deep Neural Network (DNN) + Enhanced Gazelle Optimization Algorithm (EGOA)
Sequence Detection using Bi-Directional Long Short-Term Memory (BiLSTM)
Fraud Detection Model Implementation
Performance evaluation

Figure 1: Proposed Architecture

We first used the features of the database as input. The pre-processing stage is therefore regarded as missing value repair and normalization. The SMOTE algorithm is used to lessen the issue of class imbalance. It is used to generate synthetic models for sectional classes and enhance the complete database stability by balancing the class distribution in the database. Use the suggested classifier to go on to the classification phase following this crucial pre-processing stage.

3.1. Dataset Description

A machine learning model to distinguish between fraudulent and non-fraudulent payments in order to use it to detect online payment fraud. To do this, we need a dataset with data on online payment fraud so that we can identify the kinds of transactions that result in fraud. I gathered a dataset from Kaggle for this task that may be used to identify fraudulent online payments and contains historical data about fraudulent transactions

(https://www.kaggle.cotnidatasetsijainilcoderionline-payment-fraud-detection).

The data set consists of a variety of features involved in classifying whether a given transaction is fraudulent or not. The target variable Fraud is a binary indicator: 1 for a fraudulent transaction and 0 for a normal transaction. The other features are important to estimate fraud risk and study transaction dynamics. The New Baldest and OldBalDest represent the new and old balances of the destination account, which range from 0 [bad] to 1,000,000 [good]. These are more significant because they portray the financial health of the destination account pre- and post-transaction, which is especially important when it comes to identifying fraudulent activities. Similarly, both NewBalOrig and OldBalOrig indicate the current and past balances of the origin account, respectively, and both fall within the range of 0 to 1,000,000. These values are highly significant as they directly determine the performance of the origin side, taking into account any potential fraud risk. Amount is an important feature for fraud detection, with values between 0 and 100,000. This characteristic has a direct relationship with the fraud probability, as higher volume transactions can heighten suspicions of fraud, and it also has a connection with the account balance, which may undergo unexpected changes. Transaction Type (e.g., Cash Out, Transfer) is a categorical feature of medium importance. Different transaction types can also carry varying levels of risk, which can impact the likelihood of fraud. Then again, the step feature, indicating the time step of the transaction (from 1 to 744, corresponding to hours in a 31-day window), has low importance. While it might offer a summary of transaction trends, its significance for identifying fraudulent transactions is typically lower than that of other features. Success: This feature is a number in the range of 0 and 1 describing whether the transaction was successful, having a low, medium, or high value. This feature thus displays the low, medium, or high transaction success rate. A low success rate might indicate fraud attempts or processing issues. The final feature, "Risk from 0 to 100," holds significant importance as it conveys the risk assessment score. Such information is critical to understanding the likelihood a transaction is fraud, since it is an easy way of quantifying potential fraud. After that, we use machine learning-based classifiers to predict fraud, with some features having more importance than others and being useful in this scenario.

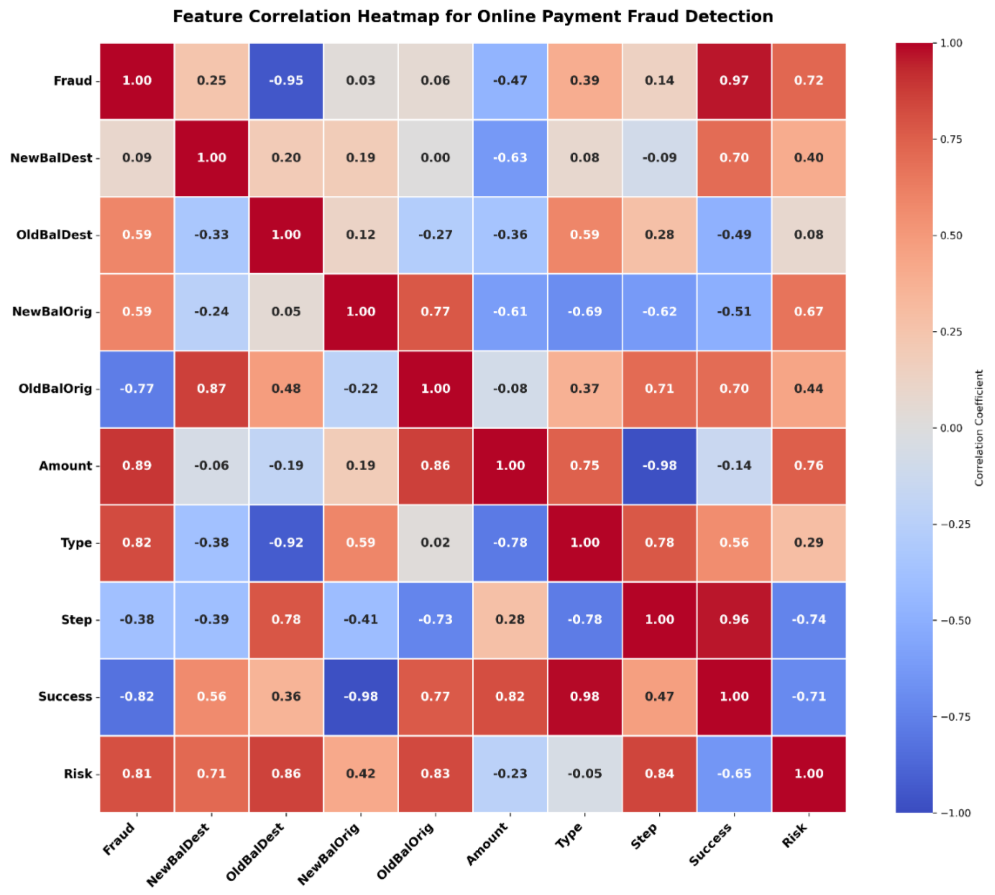

Figure 2 (a). Feature correction heat map for online payment fraud Detection

The following figure shows 2(b) the distribution of classes in the dataset, along with the amount of skew between fraudulent transactions and legitimate ones. This knowledge is critical to understanding class imbalance and the ramifications for training your model and its performance. To spotlight the dispersion and show the extent of the disparity, we provide a table-2. One major problem that appears in fraud detection tasks is class imbalance, where fraudulent transactions are much lower than legitimate transactions. When a model is trained, this imbalance can introduce bias, resulting in learning more about the majority class (legitimate transactions here), thereby leading to a model that is inaccurate when predicting fraudulent transactions. Our dataset is comprised of both fraudulent and non-fraudulent transactions, which break down as:

Table 2. Class imbalance: A dataset represents a valid transaction and a fraudulent transaction

Class	Number of Instances	Percentage
Legitimate	98,000	98%
Cycle Fraud	2,000	2%

The dataset has shown through the table above that there is a significant class imbalance: a legitimate transaction is 98% of the dataset, and a fraudulent transaction is 2%. This imbalance is problematic for training the machine learning model since the model could fatten over the predominant class and underperform on fraudulent transaction detection.

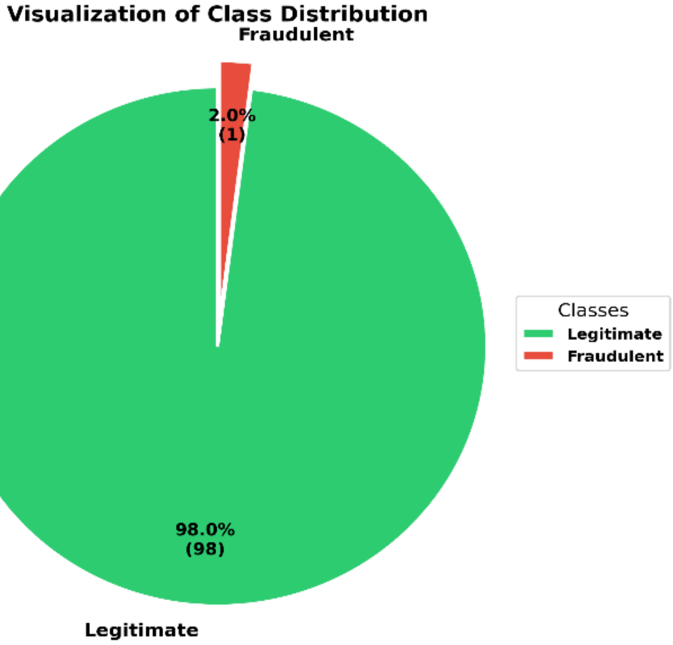

Figure 2 (b). visually reinforces the imbalance data of fraudulent transactions.

The pie chart serves as a good visual helper for the shocking difference; fraudulent transactions are only a small section of the dataset. Now, this is an important point about fraud detection: the models should be built in such a way that they can learn from minority classes and not be biased towards majority classes. To balance this ratio, we used techniques that generate synthetic samples for the minority class fraudulent transactions such as the Synthetic Minority Over-Sampling Technique (SMOTE). Decoding output positives obtained gives useful information for analysis. Overall, the class distribution ratio shown in the dataset is 98:2, meaning it is highly imbalanced between legitimate and fraudulent transactions, and techniques like SMOTE should be effectively applied to it to avoid the dirty model against fraud detection (if the model is trained and tested on this class distribution, there are no incentives in the credit card company, even though the transaction is fraudulent, but the system classifies the transaction as legitimate with 100% accuracy).

3.2. Pre-processing step

The preliminary and critical stage in making databases aimed at modeling and analyzing financial fraud includes the required pre-processing phases. This phase entails managing missing information, eliminating duplicate information, and normalizing data scaling. Finished these movements, we goal to design and cleanse the database, empowering a vigorous base for optimal analysis and optimal design to detect conceivable gears of financial fraud

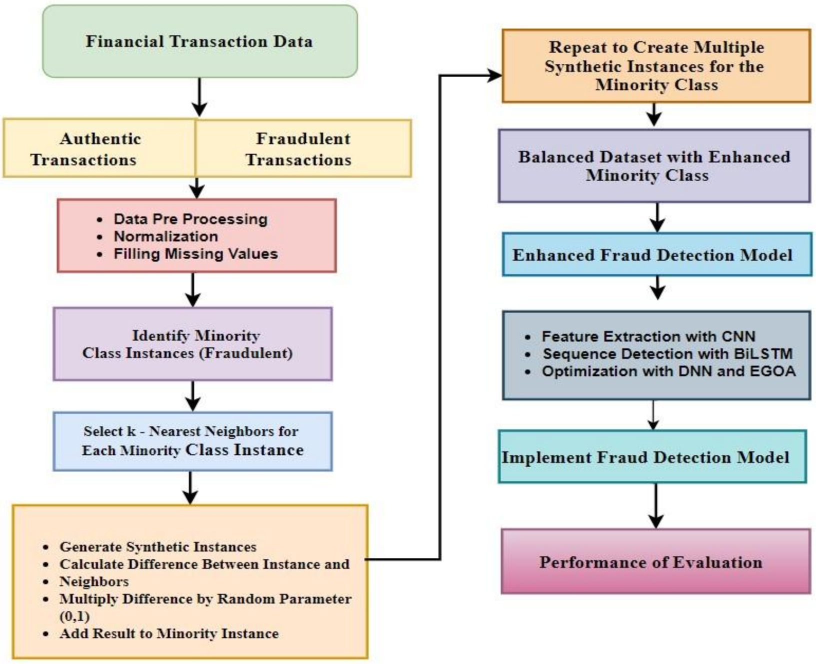
Financial Transaction Data
Authentic Transactions
Fraudulent Transactions
• Data Pre Processing
• Normalization
• Filling Missing Values
Identify Minority Class Instances (Fraudulent)
Select k - Nearest Neighbors for Each Minority Class Instance
• Generate Synthetic Instances
• Calculate Difference Between Instance and
• Neighbors
• Multiply Difference by Random Parameter (0,1)
• Add Result to Minority Instance
Repeat to Create Multiple Synthetic Instances for the Minority Class
Balanced Dataset with Enhanced Minority Class
Enhanced Fraud Detection Model
• Feature Extraction with CNN
• Sequence Detection with BiLSTM
• Optimization with DNN and EGOA
Implement Fraud Detection Model
Performance of Evaluation

Figure 3. Proposed Flow diagram

Information scaling: One of the primary steps in the pre-processing of data is data scaling, which aims to standardize and make easily compatple the scale of all full features [32]. To obtain data scales, two general techniques are used: min-max scaling and standardization. It is a procedure that changes a given characteristic, defined as T, so that the average parameter is 0 and the dispersion is constant, denoted by a standard deviation of 1. This modification empowers us to effectively computation feature T with the remaining features in a database, as demonstrated by the equation below.

$$T_{Scaled}=\frac{T-\mu}{\sigma} \qquad (1)$$

Here, σ^{-} is a standard deviation, μ is an average, T is an initial feature and T Scaled is an adjusted feature. Additionally, min-max scaling normally changes a feature T to fit inside a quantified period [0, 1].

$$T_{Scaled}=\frac{T-T_{min}}{T_{max}-T_{min}} \qquad (2)$$

Here ,T_{Max} is maximum value ,T_{Min} is minimum value and T_{Scale} is a feature that is scaled or normalization related to its size ,attractive into explanations the unique feature (T).

Finding duplicate records: Finding duplicate records allows the database to hold unique data points while preventing bias brought about by repetitions and preserving the accuracy and dependability of the data [33]. This procedure entails reviewing every record and handling just those that are relevant by comparing them to complete records.

$$d_{unique}=\{d_j \epsilon d: \text{No iedntical information in d matches } d_1\} \qquad (3)$$

Here, d is an initial/ general database that may consist of duplicate information, d_i is defined as a single, distinct database within this database and d_{unique} is a database with complete duplication data removed.

Taking care of missing data: Managing mission parameters means taking care of the gaps in our database caused by missing or incomplete data [34]. The mean imputation technique is taken into consideration to handle this missing value. In this case, substitute the missing values. with the estimated mean, which is the average of the recognized data opinions inside a particular feature. The following is how the process is formulated.

$$T_{Imputed} = \frac{1}{n}\sum_{I=1}^{n} T_I \tag{4}$$

Here, n is the total count of parameters not missing, T_i is defined as the parameters initially observed and $T_{Imputed}$ is a parameter computed to fill the gaps.

3.3. Synthetic Minority Over-Sampling Technique (SMOTE)

Fraud detection results from the financial transactions. Due to the large difference between the count of authentic transactions and fraudulent transactions in collected database, it is a frequent problem. There is a particular challenge in creating efficient fraud detection models because of this class mismatch. During this procedure, the SMOTE can be used to create synthetic data points aimed at the marginal class while preserving the underlying patterns and correlations in the data. In order to achieve this, SMOTE [Agushaka,et al, Nour, Mohamed,et al,2023,2024] creates artificial examples that are situated in the feature space between an instance of a minority class and its closest neighbors. Add a random parameter, random (0,1), with a range of 0 to 1 to introduce some unpredictability and randomness into the process. Creating additional data points to link the underrepresented minority class with nearby data, improves and highlights the minority class throughout the entire dataset.

$$(S_1: S_2) = (S_1: S_2) + \text{Randoam }(0,1).(T_{01} - T_1);((T_{02} - T_2) \tag{5}$$

Make a random parameter with a range of 0 to 1 based on the random (0,1). Here, finds the difference between the instance's feature parameters and those of its nearest neighbors, denoted as (T01-T1; T02-T2). To create several fabricated instances for the underrepresented class, this process is repeated several times. Our fraud detection architecture's implementation of the SMOTE algorithm offers a fair distribution between the two classes that is, fraudulent and non-fraudulent transactions. This technique successfully addresses the problems of class imbalance, empowering the design ability to detect fraudulent activity deprived of conciliatory its presentation in legitimate communications.

3.4. Enhanced Gazelle Optimization Algorithm

With EGOA, the CNN hyper parameters are chosen. Numerous hyperparameters types are taken into consideration, including the stride rate, activation function, hidden layers, padding add layers, kernel type values, and kernel size. The error parameters are used to pick this parameter. Gazelles eat just plants [Chatterjee, Rajesh,et al,2024]. They only consume greenery, such as grass, leaves, and shoots. Humans, wild dogs, and other animals are the gazelles' main predators. The predicted daily danger rate resulting from human activity was found to be significantly greater than the hazard rate resulting from natural predation. The first step in this optimization is to initialize the gazelle (Z) candidate population according to the equation below. The population solution is generated with an oppositional function [PrabhakaraRao,etal,2024].

3.4.1. Initial Population

The population is generated stochastically between the upper and lower bounds of the given problem.

$$Z = \begin{bmatrix} z_{1,1}Z_{1,2} & \cdots & Z_{1D} \\ \vdots & \ddots & \vdots \\ Z_{N,1} & \cdots & Z_{N,D} \end{bmatrix} \tag{6}$$

Here, D is defined as the dimension of the issue, N is defined as the total number of candidate solutions and Z_{ij} is defined as the position of the population and dimension.

$$z_{i,j} = \text{RAND X}\left(UB_J\_LB_J\right) + LB_J \tag{7}$$

Here, LB_J is a lower bound, UB_J is an upper bound and RAND is a random number. The best option is suggested as the top gazelle for creating an Elite that is matrix-designed. The following phase of the gazelles is found and searched for using this matrix.

$$\text{Elite} = \begin{bmatrix} Z_{1,1} Z_{1,2} & \cdots & Z_{1D} \\ \vdots & \ddots & \vdots \\ Z_{N,1} & \cdots & Z_{N,D} \end{bmatrix} \quad (8)$$

Here, $Z_{i,j}$ defines thegazelle vector. N replications are made in order to create the Elite matrix. Here, think of the predator and the prey as search agents. The gazelles sprint into the sage refuge in the same direction when they see the predator pursuing them by time the gazelles flee, the predator will have had a chance to investigate the search area. If the best gazelle replaces the top gazelle at the end of each iteration, the elite will be enhanced.

3.4.2. Fitness function

The suggested algorithm is primarily used in connection with backpropagation. Various strategies to avoid becoming trapped in the local optimum are described. In this section, the backpropagation approach [Ye, Run Zhou,et al,2024] is not taken into consideration in favour of the suggested EGOA in order to reduce mistakes. The proposed classifier uses this metaheuristic approach to minimize the mean square error.

$$\text{Error} = \frac{1}{T}\sum_{j=1}^{m}\sum_{i=1}^{n}(A_j^I - B_j^I)^2 \quad (9)$$

Here, n is defined as the parameter for layers of the input, B_j^IBis the desired parameter of the CNN during the T period,$A_j^I$is a network output and in is defined as the data number in the formula. The proposed technique is very useful in the optimal detection of online fraudulent.

Algorithm 1: Pseudocode of the algorithm

```
Start
  Initialize the variables and CNN hyperparameter
  Initialize the gazelle populations with oppositional function While Iteration< Max iteration
      Calculate the fitness function
      Develop elite gazelle matrix
If r<2
      $\overrightarrow{Z_{T+1}} = \overrightarrow{Z_T} + s.\vec{r} * \overrightarrow{r_b} *.\overrightarrow{\text{Elite}} - \overrightarrow{r_b} * \overrightarrow{Z_T}$
If mode (iter,2 ==0
      $\mu == -1$
Else
      $\mu == 1$
For the gazelle populations
      Upgrade gazelles based on
$\overrightarrow{Z_{T+1}} = \overrightarrow{Z_T} + s.\mu\vec{r} * \overrightarrow{r_l} *.\overrightarrow{\text{Elite}} - \overrightarrow{r_l} * \overrightarrow{Z_T}$
$\overrightarrow{Z_{T+1}} = \overrightarrow{Z_T} + s.\mu.cf.\vec{r} * \overrightarrow{r_l} *.\overrightarrow{\text{Elite}} - \overrightarrow{r_l} * \overrightarrow{Z_T}$
End if
      Elite upgrade
Considering PSR effect and upgrade related on
```

$$\overrightarrow{Z_{T+1}} = \left\{ \begin{matrix} \overrightarrow{\overrightarrow{Z_{T+}}cf[\overrightarrow{LB} + \vec{r} * (\overrightarrow{UB} - \overrightarrow{LB} * \vec{U})]} \\ \overrightarrow{Z_{T+}}[PSRs)(1 - r) + r](\overrightarrow{Z_{r1}} - \overrightarrow{Z_{r2}}) \end{matrix} \right\}$$

```
if $r \leq PSRs$
  Else
    End while
  End
Save the optimal CNN parameters
```

3.4.3. Brownian Motion

The conventional Brownian motion is defined as follows in a stochastic technique where the step length is derived from the normal probability distribution function with unit variance and zero mean:

$$F_b(Z: \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \text{Exp}\left(-\frac{(z-\mu)^2}{2\sigma^2}\right) \tag{10}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} \text{Exp}\left(-\frac{(z)^2}{2}\right) \tag{11}$$

3.4.4. Model of Levy flight

It uses the levy distribution, which is given as follows, to execute a random walk.

$$L(z_l)_= |z_l|^{1-\alpha} \tag{12}$$

$$F_b(Z: \alpha, \gamma) = \frac{1}{\pi}\int_0^\infty \text{Exp}\,(-\gamma Q^\alpha)\cos\,(Qz)\delta Q \tag{13}$$

$$\text{Levy}\ (\alpha) = 0.05X\frac{Z}{|Y|^{\frac{1}{\alpha}}} \tag{14}$$

$$Z = \text{Normal}\ (0, \sigma_Z^2)\ \text{and}\ Y = \text{Normal}\ (0, \sigma_Y^2) \tag{15}$$

$$\sigma_Z = \left|\frac{\Gamma(1+\alpha)\text{Sin}(\frac{\pi\alpha}{2})}{\Gamma(\left(\frac{1+\alpha}{2}\right))\alpha_2(\frac{\alpha-1}{2})}\right|, \sigma_Y{=}1 \text{ and } \alpha{=}1.5 \tag{16}$$

Here, y is defined as the scale unit, α is defined as a distribution index that controls the scale properties of the motion. The survival strategies of gazelles, which include grazing in the absence of predators and fleeing from spotted predators to safe havens, are simulated by this algorithm. Two conditions are added to it: exploration and exploitation.

3.4.5. Exploitation

In this stage, it is assumed that the gazelles are either stalked by a predator or are grazing contentedly in the absence of one. The following is the mathematical model of these attributes:

$$\overrightarrow{Z_{T+1}} = \overrightarrow{Z_T} + s.\vec{r} * \overrightarrow{r_b} *.\overrightarrow{\text{Elıte}} - \overrightarrow{r_b} * \overrightarrow{Z_T} \tag{17}$$

Here, r is defined as a vector of uniform random numbers in [0,1]. $\overrightarrow{r_b}$ is defined as a vector containing relom numbers defining the Brownian motion, s defines the grazing speed of the gazelles, $\overrightarrow{Z_T}$ is the solution of the present iteration and $\overrightarrow{Z_{T+1}}$ is the solution of the next iteration.

3.4.6. Exploration

When a predator is nen, the exploring phase begins. The following formula represents the mathematical model of the gazelle's behavior after it detects the predator:

$$\overrightarrow{Z_{T+1}} = \overrightarrow{Z_T} + s.\mu\vec{r} * \overrightarrow{r_l} *.\overrightarrow{\text{Elıte}} - \overrightarrow{r_l} * \overrightarrow{Z_T} \tag{18}$$

Here,$\vec{r_l}$ is a vector of random numbers related to levy distributions, s is defined as the top speed. The predator chasing of the gazelle characteristics are presented as follows.

$$\overrightarrow{Z_{T+1}} = \overrightarrow{Z_T} + s.\mu.cf.\vec{r} * \vec{r_l} *.\overrightarrow{Elıte} - \vec{r_l} * \overrightarrow{Z_T} \qquad (19)$$

$$cf = (1 - \frac{\text{Iteration}}{\text{Max iteration}})^{2\frac{\text{Iteration}}{\text{Max iteration}}} \qquad (20)$$

In this case, cf is defined as the parameter that regulates the predator's motion. The success rate, which is determined by PSR, influences the gazelle's capacity to flee, preventing the algorithm from becoming stuck in a local minimum. The following is the design of the PSR effect.

$$\overrightarrow{Z_{T+1}} = \begin{Bmatrix} \overrightarrow{\overrightarrow{Z_{T+}}cf[\overrightarrow{LB} + \vec{r} * (\overrightarrow{UB} - \overrightarrow{LB} * \vec{U})]} \\ \overrightarrow{Z_{T+}}[PSRs)(1 - r) + r](\overrightarrow{Z_{r1}} - \overrightarrow{Z_{r2}}) \end{Bmatrix}$$

$\text{if } r \leq PSRs$

$$\text{Else} \qquad (21)$$

Here, r_1 and r_2 is random indexes of the gazelle matrix. Based on this algorithm, the optimal hyperparameters of the classifier is selected.

3.5. Convolutional Neural Network- Recurrent Neural Network (CNN-RNN)

An overview of the experimental findings is given in this section along with the suggested method for testing and categorizing online fraud detection. The suggested approach to designing the efficacy and usability of CNN-BiLSTM-based models is based on the ideal core architecture of the suggested method.

3.5.1. Convolutional Neural Network (CNN)

Convolutional neural networks handle data using many layers of arrays. The convolutional layer seeks to preserve sequential information while extracting features. In certain types of problems, such as online fraud detection, CNN [40] has produced outstanding results. Spatial correlations found in the input data are utilized by CNN. Input neuron connections make up each layer that comes after a neural network. The local receptive field is defined as this precise place. The focus of the resident interested arena is on invisible neurons. Deprived of being conscious of the variations taking place outdoor of the edge, the hidden neurons evaluate the inward data confidential the designated arena.

3.5.2. Deep Neural Network (DNN)

It is a better version of the traditional ANN, which has many layers. DNNs are neural networks with a certain level of complexity that is, pre than two layers in artificial intelligence. A DNN's main goal is to collect input results that can be used to address problems in the real world, such as regression and classification.

3.5.3. Long Short-Term Memory (LSTM)

An artificial neural network called LSTM makes use of deep learning and artificial intelligence techniques. Unlike traditional feed forward neural networks, LSTM has feedback connections. Recurrent neural networks of this kind are capable of analysing both single data points and entire data sequences. RNNs [Khayyat,et al,2023] are long-term dependents; they cannot recognize words that are kept in long-term memory, but they can identify words more accurately depending on the input that is being received at that moment. Data can be stored for a very long time in LSTM. It's related to time-series data and used for processing and classification. The following is how the LSTM architecture is shown:

$$F_T = \phi(\widehat{wf}.[H_{T-1}, X_T] + b_F \qquad (22)$$

$$I_T = \phi(\widehat{wf}.[H_{T-1}, X_T] + b_I \qquad (23)$$

$$\widehat{C_T} = \text{Tanh}(\widehat{wc}.[H_{T-1}, X_T] + b_c) \qquad (24)$$

$$C_T = F_T X C_T - 1 = I_T \, X C_T \quad (25)$$

$$O_T = \phi(\widehat{wO}.[H_{T-1}, X_T] + b_O \quad (26)$$

$$H_T = O_T \mathrm{Tanh}(\phi(C_T)) \quad (27)$$

Here, b_c, b_I and , b_F is a bias for the cell, gate, and drop as well as output and input, $\widehat{wc}$. and $\widehat{wf}$. and Of is the weight utilized as a memory cell and output gate. $\widehat{C_T}$ and $\widehat{wI}$. is the weight utilized on the drop and md input gate, C_T is the state candidate parameters and cell state, $\phi$is a sigmoid function, H_T is the output from the hidden layer and X_T is the network input. The cell logs the processing state, the input gate computes if the input information will be kept, the drop gate computes if the information will be misplaced, and the output gate outputs the result.

3.6. Proposed System- OCNN-RNN

The proposed design generates utilization of the CNN layer aimed at feature extraction on input information and BiLSTMs for order detection towards optimal the anticipated outcomes and the DNN to enhance enhancing loss and error. The combination of BiLSTM and CNN design was defined as the CNN BiLSTM design, that mounted for the Recurrent Convolutional Network of long-term and it was related to the first putting CNN layers on the front end, followed by the output layer. This layer is simulated with the ReLU function, batch normalization, and softmax function is formulated as follows,

$$\mathrm{ReLu}\ (X) = \mathrm{Max}\ (0, X) \quad (28)$$

$$\text{Batch normalization } (X) = \frac{X - \mathrm{Min}}{\mathrm{Max} - \mathrm{Min}} \quad (29)$$

$$\mathrm{Softmax}\ (X_I) = \frac{\mathrm{Exp}(X_I)}{\sum_j \mathrm{Exp}(X_j)} \quad (30)$$

Algorithm 2: Pseudocode of the proposed classifier

```
Input of online fraudulent dataset
Set features Here, Nno of chosen features
Divide features into testing and training model
ReLU: Activation function
BN: Batch normalization
BiLSTM: Bidirectional LSTM layer to model
RL: Reshape layer
BN: Batch normalization
DL: Dropout layer
DNN: DNN layer
ReLU: Activation function
OL: Output Layer
ReLU: softmax activation function
```

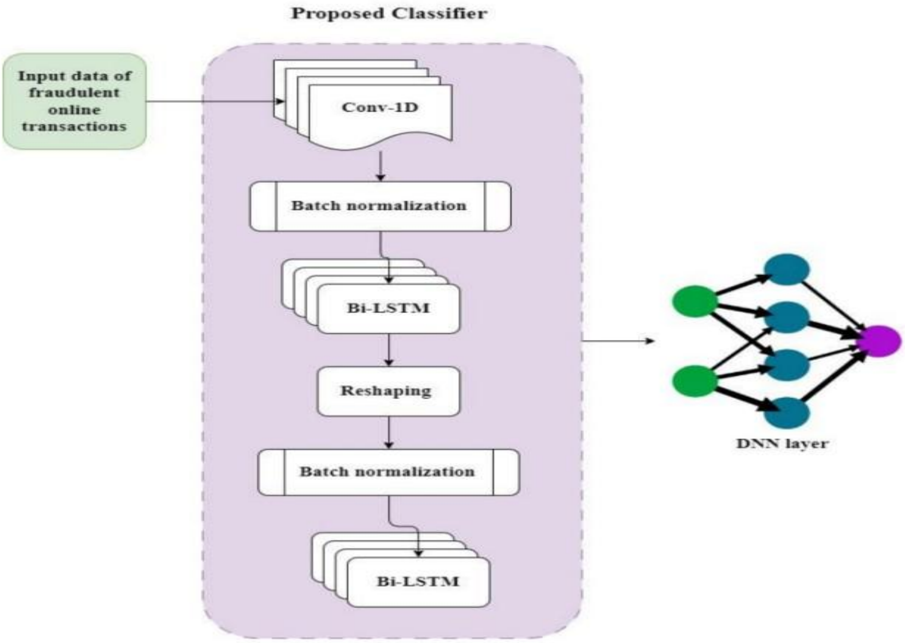

Figure 4: Proposed Architecture

BiLSTM requires the grouping of two self-governing LSTMs. This architecture empowers the networks to contain together forward and backward sequences. This proposed architecture has been designed with ReLU function and learning rate settings. In the learning phase, the EGOA is utilized.Details on Selecting EGOA Hyperparameters and the Effect on DNN Performance. Although the Enhanced Gazelles optimization Algorithm (EGOA) successfully selects the optimal hyperparameters (HPs) for Deep Neural Networks (DNNs), this HP selection provides us with significantly superior approaches when compared to traditional approaches such as Grid Search and Random Search. Optimizing hyperparameters is a time-consuming task in combinatorial learning; EGOA overcomes this by navigating through the hyperparameters search space, converging faster to the global optimum and avoiding falling into the local optima. This leads to fewer training epochs and faster model convergence. Moreover, EGOA-optimized hyperparameter configurations lead to better accuracy, better precision, better recall, and better AUC-ROC, which help to higher predictive power for all metrics. EGOA counteracts overfitting by choosing hyperparameter choices that guarantee good generalization during learning, which translates into higher test and validation performance. Nevertheless, EGOA features possible compromises. While its iterative nature adds computational complexity, demanding greater processing capabilities and memory compared to more straightforward methods. Moreover, although it helps prevent overfitting in the final model, the optimization process can itself lead to overfitting if we do not define a suitable search space or if we narrow our search too much. Setting the right boundary conditions at optimization can minimize this trade-off. Lastly, EGOA is resource-intensive to train with large-scale DNNs, which would require ample computational resources. Overall, EGOA offers well-known advantages, such as accelerating the convergence speed and achieving better performance metrics, but now on hyperparameter selection of DNNs. Though it offers better optimization than traditional approaches, the increase in computational complexity and resource consumption may outweigh the benefits for large models.

5. RESULTS AND DISCUSSION

This paper, online payments were classified as fraudulent or non-fraudulent in the experimental investigations. The proposed ECA applied to the identification process using different parameters. EGOA is used in the proposed classifier to choose the ideal hyperparameters. Finally, the fraud in the online payment is identified by using the suggested classifier. Specifically, in the current study, 20% of the data is used for the testing phase and 80% is used for the training process. Statistical measures of accuracy, precision, recall, F1-score,specificity, mean square errors, RMSE, and specificity are used to validate the suggested approach. Furthermore, Proposed OCNN-RNN, a comparison is made with traditional methods like CNN, RNN, and CNN-Whale Optimization Algorithm (WOA). The simulation parameters of the proposed model are given in Table 3.

Table 3: Simulation parameters

S.No	Description	Values
1	PSR	0.34
2	Number of iteration	100
3	S	0.88
4	Lower bound	-100
5	Upper bound	100
6	CNN layer	1
7	RNN layer	2
8	Learning rate	0.0005
9	Loss	MSE loss
10	Batch size	100
11	Hidden size	64
12	Dropout layer	0.4

The definition of classification accuracy is an effective metric for calculating the effectiveness of the suggested approach when the test dataset has an equal number of class samples in it. The results show how effective the suggested method is in identifying fraudulent online payments. Confusion matrices have been used for performance validation of false positive detection in order to acquire more computations. A two-dimensional table known as a confusion matrix is typically taken into account while calculating the classification performance. Various indicators

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \qquad (31)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \qquad (32)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \qquad (33)$$

$$\text{F1} - \text{Score} = \frac{\text{2.Precision.Recall}}{\text{Precision +Recall}} \qquad (34)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \qquad (35)$$

Here, TN is a true negative, FP is a false positive, FN is a false negative and TP is a true positive. Based on the formulations, the statistical measurements are computed and validated.

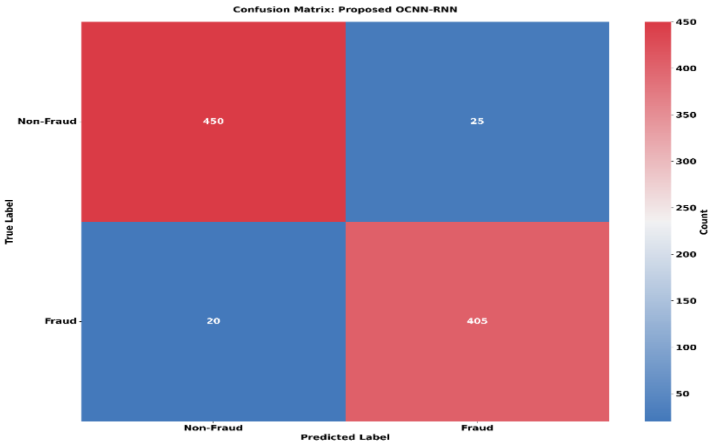

Figure 5. Confusion Matrix with proposed OCNN-RNN

Figure 5 displays a range of metrics that demonstrate the performance of the model. OCNN-RNN is trained to maximize the prediction accuracy, and as can be seen, the proposed OCNN-RNN architecture is also effective in the classification of the Confusion Matrix fraudulent and non-fraudulent transactions. The model showed a quite high level of accuracy and reliability given that the dataset includes 900 samples (425 addressed as fraudulent transactions and 475 as non-fraudulent transactions).

In this case, we have 450 True Negatives (TN), which means that out of the total of 500 transactions, we correctly identified 450 as non-fraudulent. The model is well capable of accurately classifying legitimate transactions, as indicated by these true positives. The model mistakenly classified 25 legitimate transactions as fraudulent, a process known as false positives (FP). Such errors are small, but they indicate where there is room for improvement in precision to avoid false alarms.

A total of 405 fraud transactions identified and labelled as True Positive (TP) shows that our model is efficient at catching real fraud. Conversely, it did not identify 20 fraudulent transactions leading to FN. This again stresses the need of fine-tuning the model more so we will not miss any fraud cases.

It demonstrates the overall accuracy of the model, which is about 95%, as shown to perform very well in classifying those fraudulent transactions. The model shows a high positive predictive value of fraud, meaning that with a confidence of 94.2%, this model is high yield in its positive fraud predictions. The recall (sensitivity) of 95.3% suggests that the model successfully captures a very large portion of fraud (exemplars). Enhancing detection, this specificity value of 94.7% indicates its capacity to accurately classify it as non-fraud by transaction.

The model achieves an F1 score of 94.7%, signifying balanced precision and recall along with its robustness and effectiveness in general fraud detection. These metrics validate the model's power to identify fraudulence and hold low error rates.

Therefore, the model achieves a careful trade-off between precision and recall, making it a robust and reliable detection tool that reduces both false positives and false negatives in fraud detection. Further optimization can enhance its performance and increase its accuracy in real-world applications.

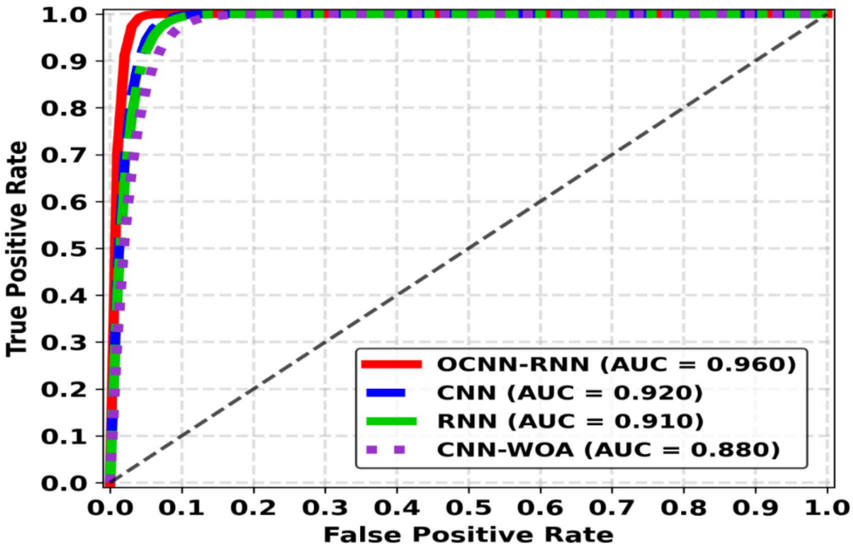

Figure 6. Roc Curve performance of compression methods

The ROC curve for OCNN-RNN, CNN, RNN and CNN-WOA is shown in Figure 6 for the operational framework for online transaction fraud detection in financial transactions. Next, we use three important evaluation metrics to synthesize these: True Positive Rate (TPR), False Positive Rate (FPR), and Area Under the Curve (AUC). All metrics give insight into how accurate and reliable each method is.

Of the four methods, OCNN-RNN shows the best performance. With a TPR of 0.96, it excels in accurately classifying fraudulent transactions. Moreover, OCNN-RNN achieves the best FPR value of 0.04, which confirms its accuracy in terms of false alarm. In the comparison test with these metrics, OCNN-RNN gets the highest AUC of 0.96. This shows that our algorithm is strong and works well at different decision thresholds for fraud detection tasks.

The CNN method comes next, exhibiting excellent performance (TPR=0.92, FPR=0.08). Although not as efficient as OCNN-RNN, CNN still proves to be a sturdy classifier with an AUC of 0.92. So CNN is a powerful approach, even on many of its characteristics inferior to OCNN-RNN.

The performance of RNNs is less than that of other networks which accounts for its lesser performance. It has 0.91 TPR and 0.09 FPR, making AUC = 0.91. Based on the index, the performance ranks third out of the four methods. Compared to CNN and OCNN-RNN, it has a higher FPR, indicating a higher probability of failure at each action level, thereby reducing the algorithm's reliability.

Meanwhile, the analysis reveals that CNN-WOA yields the poorest results. The TPR ranges from 0.88 to 0.12, with the highest FPR of 0.12 contributing to an AUC of 0.88. The results demonstrate profound limitations in being able to accurately classify transactions and, therefore, the least reliable method in this study.

The comparative analytical results demonstrate that OCNN-RNN is the most successful method, achieving the highest/satisfactory TPR and AUC, as well as the lowest FPR, making it the superior model for online transaction fraud detection. OCNN-RNN consistently outperforms CNN and RNN, despite their accuracy. CNN-WOA is the lowest performer in classification accuracy; this indicates that high margin changes are essential for the CNN-WOA classification method. In summary, the proposed framework that integrates with OCNN-RNN proves to be a viable method for complex fraud detection scenarios in finance.

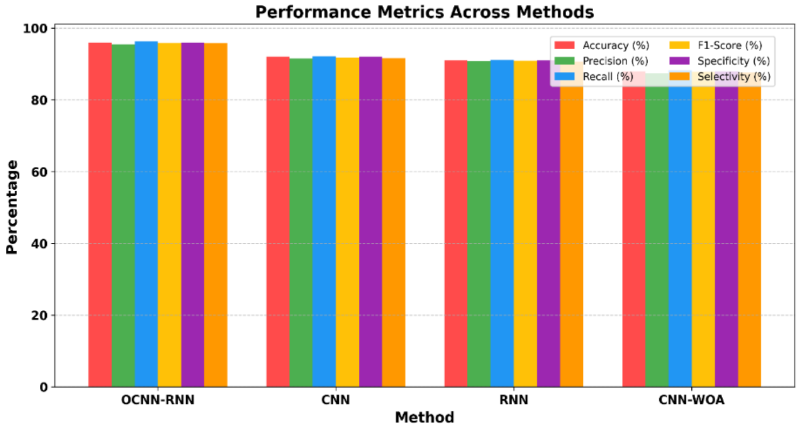

Figure 7(a). Performance Metrics validation of different methods for online fraudulently detection

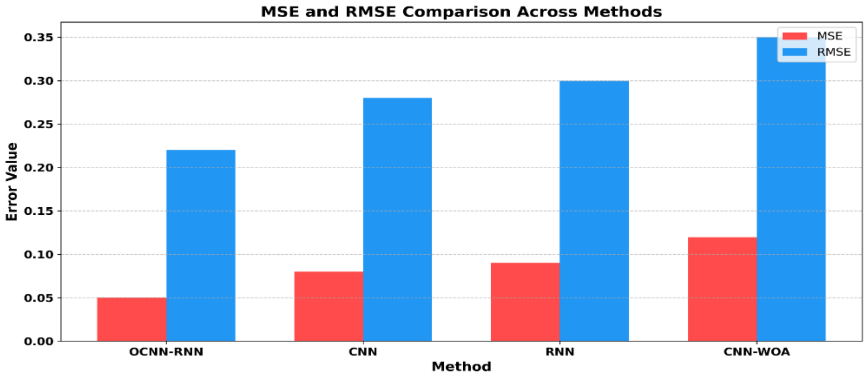

Figure 7(b). Performance Error Metrics validation of different methods for online fraudulently detection

Methods with Experimental Results Explained:

Figure 7. Performance Metrics of Various Methods for Online Fraudulent Detection in this research, four various to classify online payments as fraudulent or non-fraudulent. Multiple metrics (such as accuracy, precision, recall, F1-score, specificity, selectivity, and prediction errors) were used to evaluate the performance of these methods, thus providing an extensive overview of the capabilities of such methods in fraud detection.

OCNN-RNN (Optimized Convolutional Neural Network combined with Recurrent Neural Network):

OCNN-RNN is the best model with an accuracy of 96%, which means that it is the most successful in classifying fraudulent vs. non-fraudulent transactions correctly. With a precision metric of 95.5%, the model accurately identifies almost all transactions marked as fraudulent, thereby minimizing false positives. Moreover, the method achieves a high recall (96.3%), meaning it identifies a large majority of all fraudulent transactions, which is significant for not missing any fraud. With an F1-score of 95.9%, the method indicates a solid balance between precision and recall, with good robustness in the detection of fraud by the trained ensembles. The model has a specificity of 96%, indicating an ability to correctly classify legitimate transactions without unnecessarily flagging non-fraudulent transactions. Note that its selectiveness (95.8%) demonstrates the algorithm's effective distinction between illegal and legal transactions. Low mean square error (MSE) and root mean square error (RMSE) also emphasize that its prediction errors are little, thus making the algorithm overall reliable in the fraud detection.

Convolutional Neural Network (CNN):

The CNN model, on the other hand, although competitive, is in second place, achieving an accuracy of 92%, which is 4% lower than OCNN-RNN. Its accuracy score of 91.5% means it is reliable at identifying fraudulent transactions but is somewhat less accurate than OCNN-RNN.Its recall is at 92.2%, which means that the CNN identifies most of the fraudulent transactions but does not find all of them and therefore shows less sensitivity compared to the OCNN-RNN. Such accuracy translates to an F1 score of 91.8%, which reflects a balanced performance but is somewhat weaker than OCNN-RNN. CNN has a specificity of 92%, which is also quite successful in predicting non-fraudulent transactions, but results are slightly worse than OCNN-RNN. A selectivity score of 91.7% additionally reveals that CNN can discriminate a fraudulent transaction from a non-fraudulent transaction but not as well as OCNN-RNN. This hybrid has a higher MSE (0.08) and RMSE (0.28) than for OCNN-RNN, which could indicate that it is more prone to produce fraudulent predictions and therefore not very robust.

RNN (Recurrent Neural Network):

The RNN performs well with an accuracy score of 91% but is outperformed by CNN and OCNN-RNN. 90.8% accuracy suggests that RNN correctly predicts a large crispy of fraud transactions but at the same time has a higher error rate than CNN and OCNN-RNN. It's also somewhat sensitive, at 91.1%, suggesting that some attempted fraud wasn't detected. Overall, with an F1 score of 90.9%, it is an indicator of a balanced model at the expense of precision or recall. RNN has a specificity of 91%, meaning it detects the legitimate transactions quite well but is not as effective as other methods. A 90.7% selectivity equals a reasonable level of discrimination between transactions that are fraudulent as well as ones that are not fraudulent. The RNN model exhibits high errors and deviance in predictions, with a high Mean Squared Error (MSE) of 0.09 and an RMSE of 0.3, which is higher than that of CNN and OCNN-RNN.

CNN -WOA (Convolutional Neural Network- Whale Optimization Algorithm):

In summary, the CNN-WOA technique yields the lowest performance score of 88%, indicating its poor suitability for fraud detection. Out of all methods, this one has the lowest precision of 87.5%, resulting in a high number of false positives, thereby marking legitimate transactions as fraud. CNN-WOA has the lowest recall of 88.2%, which indicates that it fails to detect the greatest number of fraudulent transactions but detects the fewest true positives of all methods. With an F1 score of 87.8%, we observe a worse balance of precision and recall which highlights its shortcoming. The 88% specificity suggests that the model is worse than other at determining which transactions are genuine. This makes a selectivity of 87.6% a clear indication of the decline in the effectiveness of the model in differentiating fraudulent from non-fraudulent transactions. As shown in the table, the mean square error (MSE) (0.12) and root mean square error (RMSE) (0.35) are also the highest of all methods, suggesting higher prediction errors, as well as higher variability of the results. Emphasis should therefore be placed on choosing the right method according to the specific requirements of the fraud-setting, with OCNN-RNN being the most robust and reliable method for practical scenarios.

As shown in figure 8Comparison methods for accuracy Analysis, proposed OCNN-RNN (96%), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) work together to provide a strong way to find fraud in online financial transactions. This allows the model to capture sequential patterns in the temporal dimension through temporal features and investigate more complex features in the spatial dimension through convolution features. Using deep learning for more complex and nuanced fraud detection methods yields results with an accuracy of 96%, significantly higher than previous models.

In contrast to Lizhi Wang et al.[2020], they use a deep-forest ensemble learning method that builds multiple decision trees from transaction data to achieve 92.5% correct transaction classification. It optimizes for fraud detection generation through differentiation, which allows one to focus better on the patterns of these events. Despite their impressive performance, OCNN-RNN surpasses them by a slight margin, primarily due to its sequential processing and sequential pattern recognition capabilities.

Similarly, propose a framework to build Explainable Deep Learning (DL) models resulting from the process of training the models, along with a blockchain-based consensus protocol [Karim,Zkik,et al,2024] (91.8%). It also provides an inherently understandable method of fraud detection, which is a key consideration for banks. The indirect method prioritizes insight and trust, striking a balance between performance and interpretability. This approach is especially beneficial in application fields where the detection process, such as identifying malicious URLs, is more crucial than accuracy. However, its accuracy is still lower than that of the OCNN-RNN framework.

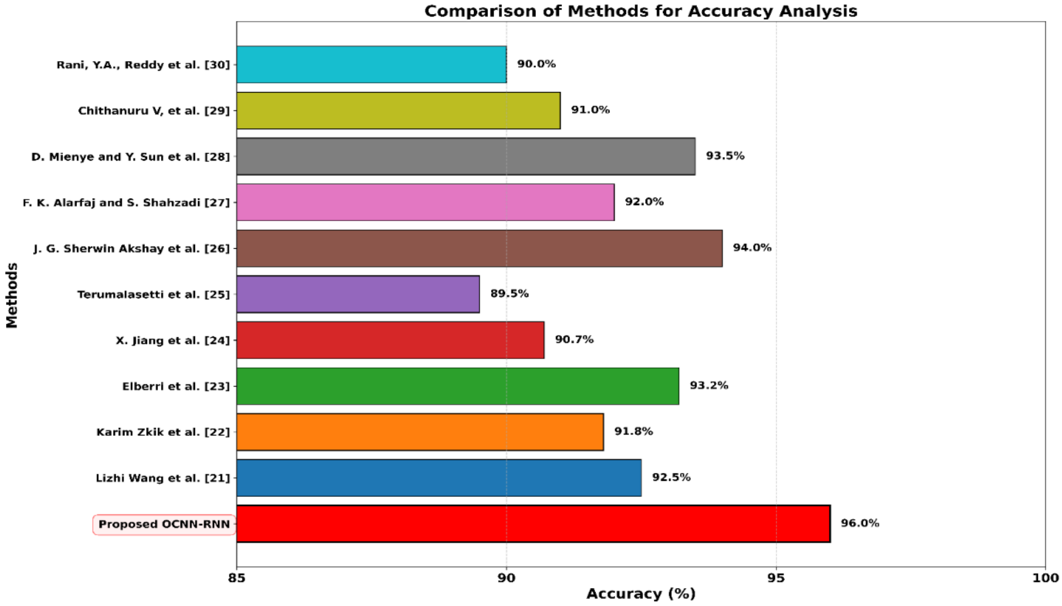

Figure 8. Comparison methods for accuracy Analysis

Elberri et al. [2024] developed hybrids based on CNN and Long Short-Term Memory (LSTM) networks and further integrated them with SMOTE, GANs, and swarm intelligence. By leveraging the capability of deep learning to approximate intricate transaction pattern sequences, this integration enables the method to manage imbalanced datasets better. While these techniques produce a stronger result than OCNN or RNN alone, the OCNN-RNN model is broadly more accurate.

X. Jiang et al. [2019] (90.7%) introduce a cost-sensitive parallel learning framework (CPLF) that optimizes for classification costs, including those associated with false positives and false negatives. This approach has proven to be efficient in identifying cost optimization for fraud detection systems, but its accuracy is quite low. This is because optimizing for cost, rather than maximizing predictive accuracy, incurs performance costs.

The Terumalasetti et al. [2024] (89.5%) This method employs an Adaptive Intelligent Multi-Stage Decision System (AIMDS) consisting of RNN, CNN, and PSO to detect malicious users. Although effective, this method primarily focuses on optimizing decision-making processes for anomaly detection, rather than directly addressing fraud detection, which accounts for its lower accuracy compared to methods that directly target fraud detection.

J. G. Sherwin Akshay et al. [2024] conducted the study. In the literature review, 94% of the approaches are based on deep learning models (autoencoder and RNNs). These methods compress the data with their autoencoder and analyze it using RNNs to determine if fraudulent activities have taken place. The OCNN model frames every reconstruction of data in each frame and combines it with the sequential capture ability of a 1D RNN to establish dependencies. This gives it a high level of accuracy and makes it a strong alternative to the OCNN-RNN method for finding fraud.

The approach uses a GNN based on Lambda Architecture by F. K. Alarfaj and S. Shahzadi [2024] (92%). Lambda Architecture is excellent for large-scale fraud detection, where GNNs generally perform very well, capturing complex relationships in data. However, for accuracy, it does less than the OCNN-RNN approach, probably due the complexity of the model, and difficulties of expansion of a model on the production and maintenance on real time.

D. Mienye and Y. Sun et al. [2023] comprised 93.5 percent of the study. The method LSTM, GRU (Gated Recurrent Units), and a combination of SMOTE ENN (Edited Nearest Neighbors) to address class imbalance, and to capture temporal patterns. These two sustainable countries, led by this temporal data class imbalance, represent a good initial step in improving the predictive performance of fraud detection in a deep learning-based model. Although it achieves a good result, it is not as good as the result from OCNN-RNN model.

The Chithanuru V, et al. namely [2023] applies techniques based on AI for blockchain anomaly detection with a 91% rate. Although the model can efficiently detect patterns of fraud associated with blockchain, it does not cover all types of online transactions, while the OCNN-RNN can identify all kinds of transactions, potentially leading to overall inaccuracy in the task.

Finally, Rani, Y.A., Reddy et al. [2024] use dimension reduction to reduce the dimensionality of data by 90%, utilizing a meta-heuristic hybrid deep learning model with an autoencoder. It incorporates optimization techniques to enhance the overall performance of deep learning models, enabling flexibility, but it achieves slightly lower accuracy in terms of fraud detection when compared with more targeted deep learning approaches.

In conclusion, despite the existence of several techniques targeted at fraud detection accuracy, the suggested OCNN-RNN surpasses earlier approaches, attaining a significantly enhanced accuracy of 96%, thanks to its optimized network that utilizes sequential data.

5. CONCLUSION

A reliable and accurate model has been developed to detect online payment fraud. The proposed system model was designed to improve fraud detection accuracy, account for the dynamic nature of fraudulent conduct, and meet interpretability standards required by financial institutions. The feature set was initially created by collecting and categorizing web data. The preprocessing stage was then explored., which included tasks such as normalization and parameter filling. In the collected database, the label imbalance problem could be present. To solve this issue, the SMOTE was utilized. The OCNN-RNN was built to extract features and detect fake parameters. In this design, BiLSTM was used to detect sequences, and CNN was utilized to extract features from the input data to provide the required results. Furthermore, the EGOA is combined with the DNN to reduce error and loss. Finally, the proposed architecture was applied to detect online fraud. Python was used to implement the proposed method, and performance metrics were employed to evaluate the outcomes. The proposed method has an accuracy level of 0.96 for detecting fraudulent online payments. In the future, the best approach will be created and tested using real-time data.

Acknowledgments

Authors would like to thank the anonymous reviewers for their useful comments and suggestions.

Conflict of Interest

The authors declared that there is no conflict of interest

REFERENCES

[1] Vanini, Paolo, Sebastiano Rossi, Ermin Zvizdic, and Thomas Domenig. "Online payment fraud: from anomaly detection to risk management." Financial Innovation 9, no. I (2023): 66.

[2] Wei, Yu-Chih, You-Xin Lai, and Mu-En Wu. "An evaluation of deep learning models for chargeback Fraud detection in online games." Cluster Computing 26, no. 2 (2023): 927¬943.

[3] Long, Ting, Fei Fang, Cuiting Luo, Yehua Wei, and Tien-Hsiung Weng. "MS HGNN: hybrid online fraud detection model to alleviate graph-based data imbalance." Connection Science 35, no. 1 (2023): 2191893.

[4] Yu, Wangyang, Yadi Wang, Lu Liu, Yisheng An, Bo Yuan, and John Panneerselvam. "A multiperspective fraud detection method for multiparticipant E-commerce transactions." IEEE Transactions on Computational Social Systems (2023).

[5] Boulieris, Petros, John Pavlopoulos, Alexandros Xenos, and Vasilis Vassalos. "Fraud detection with natural language processing." Machine Learning (2023): 1-22.

[6] Kannagi, A., J. Gori Mohammed, S. Sabari Gin Murugan, and M. Varsha. "Intelligent mechanical systems and its applications on online fraud detection analysis using pattern recognition K-nearest neighbor algorithm for cloud security applications." Materials Today: Proceedings 81 (2023): 745-749.

[7] Karthika, J., and A. Senthilselvi. "Smart credit card fraud detection system based on dilated convolutional neural network with sampling technique." Multimedia Tools and Applications 82, no. 20 (2023): 31691-31708.

[8] Karthikeyan, T., M. Govindaraj an, and V. Vijayakumar. "An effective fraud detection using competitive swarm optimization based deep neural network." Measurement: Sensors 27 (2023): 100793.

[9] Abd El-Naby, Aya, Ezz El-Din Hemdan, and Ayman El-Sayed. "An efficient fraud detection framework with credit card imbalanced data in financial services." Multimedia Tools and Applications 82, no. 3 (2023): 4139-4160.

[10] Bakhtiari, Saeid, Zahra Nasiri, and Javad Vahidi. "Credit card fraud detection using ensemble data mining methods." Multimedia Tools and Applications 82, no. 19 (2023): 29057-29075.

[11] Hajek, Petr, Mohammad Zoynul Abedin, and Uthayasankar Sivarajah. "Fraud detection in mobile payment systems using an XGBoost-based framework." Information Systems Frontiers 25, no. 5 (2023): 1985-2003.

[12] Thimonier, Hugo, Fabrice Popineau, Arpad Rimmel, Bich-Lien Doan, and Fabrice Daniel. "Comparative Evaluation of Anomaly Detection Methods for Fraud Detection in Online Credit Card Payments." arXiv preprint arXiv:2312.13896 (2023),

[13] Ghaleb, Fuad A., Faisal Saeed, Mohammed Al-Sarem, Sultan Noman Qasem, and Tawfik Al-Hadhrami. "Ensemble Synthesized Minority Oversampling based Generative Adversarial Networks and Random Forest Algorithm for Credit Card Fraud Detection." IEEE Access (2023).

[14] Banirostam, Hamid, Touraj Banirostam, Mir Mohsen Pedram, and Amir Masoud Rahmani. "A Model to Detect the Fraud of Electronic Payment Card Transactions Basedon Stream Processing in Big Data." Journal of Signal Processing Systems 95, no. 12 (2023): 1469-1484.

[15] Al-Sayyed, Rizik, Esra'A. Alhenawi, Hadeel Alazzam, Ala'A. Wrikat, and Dima Suleiman. "Mobile money fraud detection using data analysis and visualization techniques." Multimedia Tools and Applications 83, no. 6 (2024): 17093-17108.

[16] Krishnavardhan, N., M. Govindarajan, and S. V. Rao. "An intelligent credit card fraudulent activity detection using hybrid deep learning algorithm." Multimedia Tools and Applications (2024): 1-26.

[17] Seera, Manjeevan, Chee Peng Lim, Ajay Kumar, Lalitha Dhamotharan, and Kim Hua Tan. "An intelligent payment card fraud detection system." Annals of operations research 334, no. 1 (2024): 445-467.

[18] Almazroi, Abdulwahab Ali, and Nasir Ayub. "Online Payment Fraud Detection Model Using Machine Learning Techniques." IEEE Access 11 (2023): 137188-137203.

[19] Arfeen, A. Asad, and B. Muhammad Asim Khan. "Empirical analysis of machine learning algorithms on detection of fraudulent electronic fund transfer transactions." IETE Journal of Research 69, no. 11 (2023): 7920-7932.

[20] Fanai, Hosein, and Hossein Abbasimehr. "A novel combined approach based on deep Autoencoder and deep classifiers for credit card fraud detection." Expert Systems with Applications 217 (2023): 119562.

[21] Lizhi Wang, Zhaohui Zhang, Xiaobo Zhang, Xinxin Zhou, Pengwei Wang, Young'un Zheng, A Deep-forest based approach for detecting fraudulent online transaction, Advances in Computers, Elsevier, Volume 120,2021, Pages 1-38, ISSN 0065-2458, ISBN 9780128211472, https://doi.org/10.1016/bs.adcom.2020.10.001.

[22] 22.Karim Zkik, Amine Belhadi, Sachin Kamble, Mani Venkatesh, Mustapha Oudani, Anass Sebbar,Cyber resilience framework for online retail using explainable deep learning approaches and blockchainin-based consensus protocol, Decision Support Systems, Volume 182,2024,114253,ISSN 0167-9236, https://doi.org/10.1016/j.dss.2024.114253.

[23] Elberri, M.A., Tokeşer, Ü., Rahebi, J. et al. A cyber defense system against phishing attacks with deep learning game theory and LSTM-CNN with African vulture optimization algorithm (AVOA). Int. J. Inf. Secur. 23, 2583–2606 (2024). https://doi.org/10.1007/s10207-024-00851-x

[24] X. Jiang, S. Pan, G. Long, F. Xiong, J. Jiang and C. Zhang, "Cost-Sensitive Parallel Learning Framework for Insurance Intelligence Operation," in IEEE Transactions on Industrial Electronics, vol. 66, no. 12, pp. 9713-9723, Dec. 2019, doi: 10.1109/TIE.2018.2873526.

[25] Terumalasetti, S., S R, R. Artificial intelligence-based approach to detect malicious users using deep learning and optimization techniques. Multimed Tools Appl (2024). https://doi.org/10.1007/s11042-024-19872-8

[26] J. G. Sherwin Akshay, T. Vinusha, R. Sharon Bianca, C. K. Sarath Krishna and G. Radhika, "Enhancing Credit Card Fraud Detection with Deep Learning and Graph Neural Networks," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-6, doi: 10.1109/ICCCNT61001.2024.10725042.

[27] F. K. Alarfaj and S. Shahzadi, "Enhancing Fraud Detection in Banking with Deep Learning: Graph Neural Networks and Autoencoder for Real-Time Credit Card Fraud Prevention," in IEEE Access, doi: 1-1, 23 September 2024,10.1109/ACCESS.2024.3466288

[28] I. D. Mienye and Y. Sun, "A Deep Learning Ensemble with Data Resampling for Credit Card Fraud Detection," in IEEE Access, vol. 11, pp. 30628-30638, 2023, doi: 10.1109/ACCESS.2023.3262020.

[29] Chithanuru V, Ramaiah M. An anomaly detection on blockchain infrastructure using artificial intelligence techniques: Challenges and future directions – A review. Concurrency Computat Pract Exper. 2023; 35(22): e7724. doi: 10.1002/cpe.7724

[30] Rani, Y.A., Reddy, E.S. Deep intrusion net: an efficient framework for network intrusion detection using hybrid deep TCN and GRU with integral features. Wireless Netw (2024). https://doi.org/10.1007/s11276-024-03800-7

[31] Khayyat, Manal M. "Improved bacterial foraging optimization with deep learning based anomaly detection in smart cities." Alexandria Engineering Journal 75 (2023): 407-417.

[32] Vanini, Paolo, Sebastiano Rossi, Ermin Zvizdic, and Thomas Domenig. "Online payment fraud: from anomaly detection to risk management." Financial Innovation 9, no. 1 (2023): 66.

[33] Abd El-Naby, Aya, Ezz El-Din Hemdan, and Ayman El-Sayed. "An efficient fraud detection framework with credit card imbalanced data in financial services." Multimedia Tools and Applications 82, no. 3 (2023): 4139-4160.

[34] Ni, Lina, Jufeng Li, Huixin Xu, Xiangbo Wang, and Jinquan Zhang. "Fraud feature boosting mechanism and spiral oversampling balancing technique for credit card fraud detection." IEEE Transactions on Computational Social Systems (2023).

[35] Agushaka, Jeffrey o., Absalom E. Ezugwu, and Laith Abualigah. "Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer." Neural Computing and Applications 35, no. 5 (2023): 4099-4131.

[36] Nour, Mohamed K., Imene Issaoui, Alaa Edris, Ahmed Mahmud, Mohammed Assiri, and Sara Saadeldeen Ibrahim. "Computer Aided Cervical Cancer Diagnosis using Gazelle Optimization Algorithm with Deep Learning Model." IEEE Access (2024).

[37] Chatterjee, Rajesh, Md Amir Khusru Akhtar, Dinesh Kumar Pradhan, Falguni Chakraborty, Mohit Kumar, Sahil Verma, Ruba Abu Khurma, and Maribel Garcia-Arenas. "FNN for diabetic prediction using oppositional whale optimization algorithm." IEEE Access (2024).

[38] PrabhakaraRao, T., Satish Kumar Patnala, Ch V. Raghavendran, E. Laxmi Lydia, Yeonwoo Lee, Srijana Acharya, and Jae-Yong Hwang. "Oppositional Brain Storm Optimization with Deep Learning based Facial Emotion Recognition for Autonomous Intelligent Systems." IEEE Access (2024).

[39] Ye, Run Zhou, Kirill Lipatov, Daniel Diedrich, Anirban Bhattacharyya, Bradley J. Erickson, Brian W. Pickering, and Vitaly Herasevich. "Automatic ARDS surveillance with chest X-ray recognition using convolutional neural networks." Journal of Critical Care 82 (2024): 154794.

[40] Gozuoglu, Abdulkadir, Okan Ozgonenel, and Cenk Gezegin. "CNN-LSTM Based Deep Learning Application on Jetson Nano: Estimating Electrical Energy Consumption for Future Smart Homes." Internet of Things (2024): 101148.

[41] https://www.kaggle.cotnidatasetsijainilcoderionline-payment-fraud-detection