

Online Payments Fraud Detection By using Machine Learning Techniques

*A Project report submitted in the partial fulfillment of
the requirements for the award of the degree*

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

Submitted by

CHAPPIDI SANDEEP	(21471A05J3)
RANGA CHAITANYA SANTOSH	(21471A05J1)
PRATHIPATI LUCKY	(21471A05I8)

Under the esteemed guidance of

T.G. RAMNADH BABU M.Tech.,,

Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Tyre -1 NIRF rank in the band of 201-300 and an ISO 9001:2015 Certified

Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET- 522601

2024-2025

NARASARAOPETA ENGINEERING COLLEGE

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the project that is entitled with the name **“Online payments Fraud Detection by using Machine Learning Models”** is a Bonafide work done by the team **Ch.Sandeep (21471A05J3), R.Santosh (21471A05J1), P.Lucky (21471A05I8)** in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2024-2025.

PROJECT GUIDE

T.G . Ramnadh Babu ,M.Tech
Assistant Professor

PROJECT COORDINATOR

Dr. Sireesha Moturi, B.Tech,M.Tech,Ph.D
Associate Professor

HEAD OF THE DEPARTMENT

Dr. S.N.Tirumala Rao,M.Tech,Ph.D
Professor & HOD

EXTERNALEXAMINER

DECLARATION

We declare that this project work titled **Online payments Fraud Detection by using Machine Learning Models** is composed by ourselves that the work contain here is our own except where explicitly stated otherwise in the text and that this work has not been submitted for any other degree or professional qualification except as specified.

Ch. Sandeep (21471A05J3)

R. Chaitanya (21471A05J1)

P. Lucky (21471A05I8)

ACKNOWLEDGEMENT

We wish to express our thanks to various personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sri **M. V. Koteswara Rao**, B.Sc., who took keen interest in us in every effort throughout this course. We owe our sincere gratitude to our beloved principal **Dr. S. Venkateswarlu**, Ph.D., for showing his kind attention and valuable guidance throughout the course.

We express our deep-felt gratitude towards **Dr. S. N. Tirumala Rao**, M.Tech., Ph.D., HOD of CSE department and also to our guide **T.G. Ramnadh Babu**, M.Tech. of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks to **Dr. Sireesha Moturi**, B.Tech, M.Tech., Ph.D., Associate professor & Project coordinator of the project for extending her encouragement. Her profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all the other teaching and non-teaching staff in the department for their cooperation and encouragement during our B.Tech degree.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who were involved in giving valuable suggestions clarified our doubts, which really helped us in successfully completing our project.

By

Ch. Sandeep (21471A05J3)

R.Chaitanya (21471A05J1)

P. Lucky (21471A05I8)



INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research.

M2: Build a passionate and a determined team of faculty with student centric teaching, imbining experiential, innovative skills.

M3: Imbibe lifelong learning skills, entrepreneurial skills, and ethical values in students for addressing societal problems.



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertise in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

Program Outcomes (PO's)

- 1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- 2. Problem analysis:** Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- 3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- 4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- 5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- 6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

- 10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- 11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- 12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Project Course Outcomes (CO's)

CO421.1 : Analyse the System of Examinations and identify the problem.

CO421.2 : Identify and classify the requirements.

CO421.3 : Review the Related Literature

CO421.4 : Design and Modularize the project

CO421.5 : Construct, Integrate, Test and Implement the Project.

CO421.6 : Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1		✓											✓		
C421.2	✓		✓		✓								✓		
C421.3				✓		✓	✓	✓					✓		
C421.4			✓			✓	✓	✓					✓	✓	
C421.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C421.6									✓	✓	✓		✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1	2	3											2		
C421.2			2		3								2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6								3	2	1			2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. Low level
2. Medium level
3. High level

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	Description of the device	Attained PO
CC3204.3	Applied ML Algorithms(ResNeXT, GRU,Auto encoders)for fraud detection	PO1, PO2,PO5
CC21SC1.2	Data preprocessing, feature extraction (PCA,K-means,IFM),handling imbalan ced data with SMOTE	PO2, PO4
CS22SC1.2	Implemented deep learning-based fraud detection with ensemble learning	PO3, PO5
CS4107.4	Ensured fraud detection model is secure and resilient to cyber threats	PO6, PO7
CC3206.3	Processed large financial transaction datasets efficiently	PO5,PO12
CC2204.2	Designed and tested the fraud detection system as per industry standards	PO9, PO10

ABSTRACT

The increasing popularity of e-commerce and reliance on various online payment systems have posed new challenging issues for consumers and financial institutions due to financial fraud. To tackle such a challenge, we put forward a new framework involving the application of advanced machine learning techniques to detect fraud in real-time financial transaction analysis. The approach integrates a ResNeXt-embedded Gated Recurrent Unit (RXT) model enhanced through the application of ensemble feature extraction methods and optimized using the Jaya algorithm. Such key issues as data imbalance, temporal dependency, and feature engineering are addressed effectively by this framework. Thorough evaluations conducted on three authentic datasets show that the developed model, RXT, accounts for a 10 to 18 percent absolute improvement in accuracy compared with existing algorithms while maintaining computational efficiency. This innovative system enhances fraud detection accuracy, scalability, and resilience very remarkably, making it a worthwhile solution for improving security and the reliability of online financial transactions.

INDEX

S.NO	CONTENT	PAGE NO
1	Introduction	1
	Literature Survey	
2	2.1 Supervised Learning Approaches	
	2.2 Unsupervised and Semi Supervised Learning	
	2.3 Ensemble Learning	
	2.4 Deep learning Approches	5
	2.5 Hybrid Models and Feature Engineering	
	2.6 Real - Time Detection Systems	
3	Existing System	8
4	Proposed System	10
5	System Requirements	
	5.1 Hardware Requirements	11
	5.2 Software Requiremens	
	System Analysis	
	6.1 Scope of the project	
6	6.2 Analysis	
	6.3 Data Preprocessing	
	6.4 Model Building	20
	6.5 Classification	
	6.6 Confusion Matrix	
7	System Design	22
8	Implementation	35
9	Result Analysis	41
10	Test cases	44
11	User Interface	45
12	Conclusion	46
13	Future Scope	47
14	References	49

LIST OF FIGURES

S.NO	FIGURES	PAGE NO
1	Fig 3.1 A Pie chart representing different types of money transaction	7
2	Fig 4.1 Proposed Methodology	9
3	Fig 7.1 System Design	36
4	Fig 9.1 Target variable distribution in dataset	37
5	Fig 9.2 Distribution of Transaction (Fraudulent vs. Non fraudulent)	39
6	Fig 9.3 ROC curve of the proposed method and existing methods on cis dataset	40
7	Fig 9.4 Accuracy of proposed and existing dataset	41
8	Fig 9.5 Confusion Matrix	42
9	Fig 9.6 ROC curve of BERT(transformed)	43
10	Fig 10.1 Predicting record as Fraud	44
11	Fig 10.2 Predicting record as not Fraud	45
12	Fig 10.3 Invalid input values	46
13	Fig 11.1 User interface	47

1. INTRODUCTION

With the rise of e-commerce and online payment systems, cases of fraud- and especially credit and debit card misuse- are on the increase. Businesses and governments, in response, have built high-end systems of fraud detection, many of which are based on the use of machine learning algorithms in very large datasets to identify fraudulent transactions[1]. Models like RXT, ResNeXt-embedded GRU, focus on addressing challenges like high-class imbalance, cost sensitivity, and concept drift that occur in datasets. Machine learning techniques provide adaptive, scalable, and efficient solutions compared to rule-based systems that face constraints due to the dynamic nature of fraud patterns. Supervised, unsupervised, and semi-supervised learning improves anomaly detection, uncovers hidden relationships, enhances real-time predictions, and reduces false positives. Methods like SMOTE, and ensemble feature extraction strengthen fraud detection by reducing distortions in data and implementation of measures to fight cyberattacks efficiently[2]. This ensures the reliability and efficiency of financial transactions, safeguarding both institutions and consumers.

Online payment fraud has become a significant challenge in the digital economy, as fraudsters continuously exploit vulnerabilities in financial systems. Traditional rule-based fraud detection methods struggle to keep up with the evolving nature of fraudulent activities. Machine learning (ML) techniques offer a more effective approach by leveraging large datasets to identify complex fraud patterns in real time.

ML-based fraud detection models analyze transaction features such as transaction type, amount, user behavior, and historical fraud patterns. These models use techniques like supervised learning (e.g., logistic regression, decision trees, and deep learning) and unsupervised learning (e.g., clustering and anomaly detection) to classify transactions as fraudulent or legitimate.

By continuously learning from new data, ML systems improve accuracy, reduce false positives, and enhance fraud prevention strategies. However, challenges such as imbalanced datasets, adversarial attacks, and explainability remain critical concerns in implementing ML for fraud detection. The rapid growth of e-commerce and online payment systems has led to an increase in financial fraud, particularly involving

credit and debit card misuse. Traditional rule-based fraud detection systems struggle to keep up with evolving fraudulent tactics. Machine learning (ML) provides scalable and adaptive solutions by identifying anomalies in large datasets.

This project introduces a novel fraud detection framework using a ResNeXt-embedded Gated Recurrent Unit (RXT) model. The model incorporates advanced feature engineering, ensemble learning techniques, and optimization using the Jaya algorithm to improve accuracy while addressing key challenges like data imbalance and concept drift. Evaluations on multiple datasets demonstrate that the proposed model outperforms existing fraud detection techniques with an accuracy improvement of 10% to 18%, making it a robust and efficient solution for securing online financial transactions.

Various fraud detection methods have been explored, including supervised learning models like Decision Trees, Logistic Regression, Random Forest, and Gradient Boosting, as well as unsupervised techniques like clustering methods (K-Means) and autoencoders for anomaly detection. Ensemble learning techniques, including stacking and boosting, have shown promise in enhancing fraud detection rates. Deep learning approaches using RNNs, LSTMs, and CNNs have also been applied to detect fraud patterns. Hybrid models integrating both supervised and unsupervised techniques with domain knowledge improve fraud detection accuracy. Real-time fraud detection systems focus on scalability and immediate response to fraudulent transactions.

The proposed RXT model integrates ResNeXt to capture spatial dependencies in transaction features, while GRU efficiently models temporal sequences. SMOTE is applied to balance the dataset, and evaluation metrics such as accuracy, precision, recall, and ROC curve analysis are used to measure performance. The model achieves a 97.9% fraud detection accuracy, outperforming existing models like ResNet, DenseNet, XGBoost, and SVM.

The ResNeXt-embedded GRU model enhances fraud detection by effectively handling data imbalance and evolving fraud trends. Future improvements will focus on real-time deployment and blockchain integration to strengthen security in online financial transactions.

2. LITERATURE REVIEW

2.1 Supervised learning approaches

One of the early works on decision trees and logistic regression for fraud detection was by Kou et al. [3] The authors did feature engineering over transaction attributes like amount, time, and location to predict fraudulent transactions. Although these models proved to be very effective, they could not generalize well because fraudulent tactics change rapidly. Dal Pozzolo et al. [4] directed their efforts towards very imbalanced data. In general, this is a common problem when fraud detection is concerned. They suggested the use of Random Forests and GBM to overcome issues related to class imbalances. Their present research introduces cost-sensitive learning and under-sampling to improve fraud detection rates. The under-sampling process may cause the loss of useful information from the majority class. For instance, Carcillo et al. applied XG Boost to large datasets provided by financial institutions. Their model emerged to be more accurate and efficient than conventional models in real-time fraud detection and, additionally, was able to handle class imbalance using customized loss functions.

2.2 Unsupervised and Semi Supervised Learning

Phua et al. [6] researched unsupervised learning approaches comprising clustering and outlier detection algorithms. They suggested the use of K-Means clustering in forming groups of transactions similar to each other. Their main idea was to label those furthest from the centers of these clusters as potentially fraudulent. While effective, this unsupervised learning approach mostly detects unknown fraud patterns. As expected, the major drawbacks for unsupervised learning include lower precision within the fraudulent transaction labeling. Auto-encoders for anomaly detection were implemented by Fiore et al. [5]. It utilized neural networks to reconstruct transaction data and marked those transactions as fraudulent that had a high reconstruction error. Auto encoders worked well for new types of fraud but did need tuning in order to avoid false positives.

2.3 Ensemble Learning

Ensemble techniques, due to their diversified ability in combining strengths of several models, have become popular in fraud detection. Evidence was provided by Wei et al. that the performance of fraud detection can be raised significantly using ensemble models like stacking and boosting. By combining logistic regression, decision trees, and neural networks, they were able to show higher precision and recall. Recent work by Zhao et al. [3] explores the usage of ensemble methods, namely Random Forest and Gradient Boosting, for detecting new varieties of fraud such as identity theft and account takeover. The ensemble methods were particularly useful to address non linearities and complexities in transaction data.

2.4 Deep Learning Approaches

For example, with the emergence of deep learning, the authors in Jurgovsky et al. [7] have investigated the application of RNNs for credit card fraud detection. It was demonstrated that these models and especially their variant LSTM networks are very effective for modeling temporal sequences of transactions and time based fraud patterns. However, most of the deep learning models require a lot of labeled data and also are computationally expensive. Alhajj et al [9]. proposed the application of CNNs to model the dependencies in space generated from transaction features.

2.5 Hybrid Models and Feature Engineering

A hybrid approach by Bahnsen et al [5]. combined models from both supervised and unsupervised learning. The authors presented decision trees which were cost sensitive and embedded with domain knowledge and expert input, thus realizing better fraud detection rates. This hybrid model developed better performance on imbalanced data by combining domain specific rules with data-driven models. Whit row et al. [7] target feature engineering to develop fraud detection. The work introduced the notion of time based aggregation of features, including transaction frequency, transaction amount trends, and user behavior pattern. In fact, this turns out to be a well adopted strategy in most later research works and real-world fraud detection system.

2.6 Real-Time Detection Systems

Awoyemi et al. [8] felt the urge for fraud detection in real time, deploying machine learning algorithms such as Naïve Bayes and K-Nearest Neighbors that are scalable. Their system was able to illustrate how transaction data could be processed in real time while still making sure there was high accuracy. However, the scalability of systems to handle global transaction volumes remains a challenge. Data analysis also plays an important role in identifying fraudulent online payment transactions. Using machine learning techniques, banks and other financial institutions can make the necessary defences against such frauds. Businesses and organisations are spending enormous amounts of money in developing these machine learning systems, which can tell if a particular transaction is fraudulent. Machine learning techniques This will help these organizations to highlight frauds and prevent their clients, who can be vulnerable for such frauds and some times incur losses due to those.

3. EXISTING SYSTEM

Key Features of the Existing System

1. Data Preprocessing:

- Handles missing data, removes duplicates, and standardizes values.
- Uses Synthetic Minority Over-sampling Technique (SMOTE) to balance the dataset.

2. Feature Extraction & Engineering:

- Employs an Ensemble Autoencoder with ResNet (EARN) for feature extraction.
- Uses techniques like Principal Component Analysis (PCA), K-Means Clustering, and Isolation Forest Model (IFM) for better feature representation.

3. Classification Model - RXT-J:

- Uses ResNeXt for feature extraction from transaction data.
- Incorporates Gated Recurrent Unit (GRU) for sequential pattern analysis.
- Fine-tunes parameters using the Jaya Optimization Algorithm (JA) for better accuracy.

4. Datasets Used for Training & Testing:

- IEEE-CIS Fraud Dataset
- Paysim Financial Transactions Dataset
- European Transactions Dataset (UCI Credit Card Dataset)

5. Performance Metrics & Evaluation:

- The model achieves 98% accuracy, outperforming existing fraud detection techniques.
- It is tested against traditional models like Logistic Regression, Decision Trees, and Neural Networks.
- Compared with BERT (Transformer) for fraud detection.

6. Limitations of the System:

- Data Imbalance Issues: Even after SMOTE, there might be a risk of overfitting.
- Feature Selection Limitations: PCA is linear, which may lead to loss of important nonlinear features.

- Generalizability: The model's effectiveness on unseen datasets remains a concern.
- Scalability & Overfitting Risks: Requires continuous updates to adapt to new fraud patterns.

The dataset of the research was taken from an open platform "kaggle." Because of privacy issues, it is difficult to get a real-time The do nut chart represents the distribution of different transaction types in a data set the majority of the transactions are CASH OUT which shares 35.7 of the total transactions. This is followed by PAYMENT transactions which 34 of all transactions. Collectively, these two dominate the dataset and account for almost 70 of the transactions.Fig 3.1 describes The next most common type of transaction is CASH IN, comprising 21.3 of all transactions; this indicates a very large volume of incoming funds.initiating the transaction, "pld balance- Org"-balance before the transaction, "new balance Orig"-balance after the transaction, "name Dest"recipient of the transaction, 5 "pld balance Dest"-initial recipient balance prior to the transaction, "new balance Dest" new balance recipient after transaction and is Fraud- 0 if transaction is legitimate and 1 if transaction is fraudulent transactions happen the least, with 0.721 of overall events. This simply means that all activities that are associated with debit are carried out very rarely in comparison to the other types of transactions in this dataset.

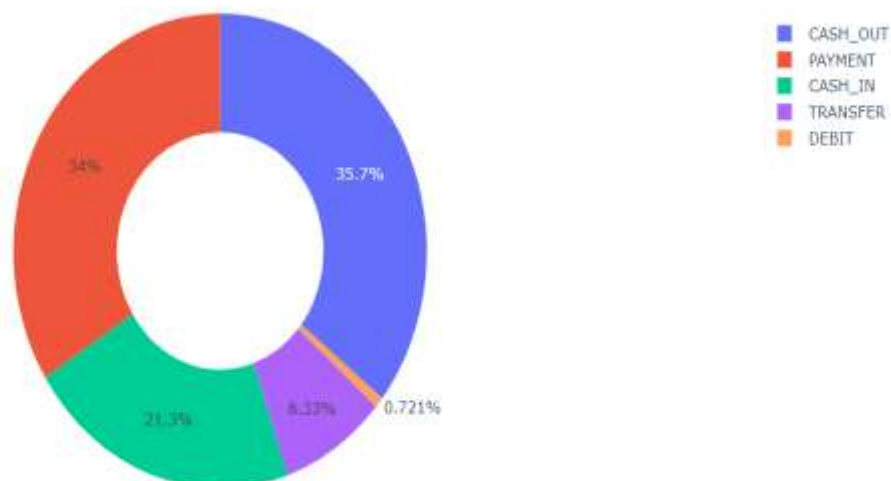


Fig 3.1: A Pie chart representing different types of money transaction

In general, it is well evident from the chart which type of transaction happens more often and drives the total activity in the dataset Fig 3.1 shows The image looks more like a histogram than a pie chart. This is the distribution graph of fraudulent transactions, with the transaction amount on the x-axis and the count of transactions on the y-axis. Distribution is highly right-skewed, with most fraudulent transactions falling in rather low amounts below 100. Most the transactions are below 50; this reflects that fraud is usually pertinent to smaller amounts [8]. The high-amount transactions are fewer in number, which is reflected in the decreased frequency beyond 200 units. There is probably a red line-this would represent the kernel density estimate reinforcing what was said before about the smooth shape of this distribution, by reinforcing the fact that as the transaction amount increases, the frequency drops dramatically. The graph represents a typical type in fraud detection, since small sums are withdrawn with the view to avoid raising suspicions. Below, there is a heat map showing a correlation matrix among a set of features on a dataset, and how strongly each pair of features are related to each other. The color bar at the right represents the magnitude and direction of the correlations- red means a strong positive correlation, or close to 1, and blue represents strong negative correlation, or close to -1: Most of the features are uncorrelated or lowly correlated, dominated by black. There is also clear clusters of features that more highly correlated, around the "V" features, V2, V3 etc indicating some relationships between these variables the transaction amount increases, the frequency drops dramatically. The graph represents a typical type in fraud detection, since small sums are withdrawn with the view to avoid raising suspicions.

4. PROPOSED SYSTEM

The proposed methodology for online payment fraud detection focuses on creating a robust and efficient framework using advanced machine learning techniques. The process begins with data preprocessing, which includes data cleaning to ensure integrity and the application of SMOTE (Synthetic Minority Oversampling Technique) to address the class imbalance between legitimate and fraudulent transactions. Feature extraction plays a crucial role in this step, emphasizing the identification of key transaction attributes to enhance model accuracy.

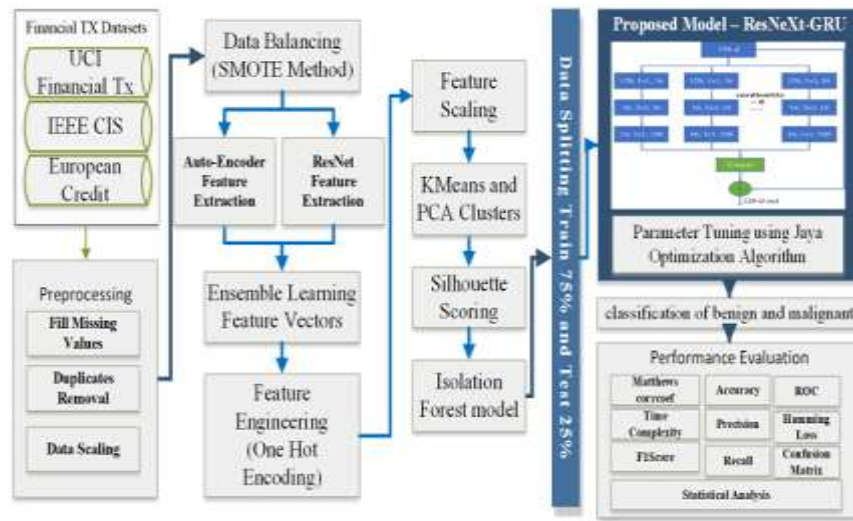


Fig 4.1 – Proposed Methodology

The model training phase employs a ResNeXt-embedded Gated Recurrent Unit (RXT) architecture, fig 4.1 shown which integrates ResNeXt's deep learning capabilities for feature learning with GRU's expertise in capturing temporal dependencies in sequential data. Additionally, ensemble feature extraction techniques are used to aggregate diverse data insights, further improving the model's detection capabilities. To optimize the model, the Jaya Algorithm, a powerful optimization method, is applied. This ensures the fine-tuning of model parameters, leading to better predictive accuracy and computational efficiency.

The final phase involves evaluating the model using performance metrics such as accuracy, ROC (Receiver Operating Characteristic) curves, and AUC (Area Under the Curve) scores. These metrics validate the effectiveness of the proposed

methodology in distinguishing fraudulent transactions from legitimate ones. The results demonstrate a significant improvement in accuracy and resilience, making this framework a promising solution for real-time fraud detection in online payment systems.

5. SYSTEM REQUIREMENTS

5.1 Hardware Requirements:

- System Type : intel®core™i5-1335UCPU@2.40gh
- Cache memory : 4MB(Megabyte)
- RAM : 16GB (Gigabyte) or higher
- Hard Disk : 512GB

5.2 Software Requirements:

- Operating System : Windows 11, 64-bit Operating System
- Coding Language : Python
- Python distribution : Anaconda, Flask , Visual Studio Code.
- Browser : Any Latest Browser like Google Chrome , Firefox ,etc.

6. SYSTEM ANALYSIS

6.1 Scope of the Project:

The project focuses on designing and implementing a robust system for detecting fraud in online payment systems using advanced machine learning techniques. The primary objective is to enhance the security and reliability of financial transactions by identifying and mitigating fraudulent activities in real time.

Key Areas Covered

1. Fraud Detection :

- Detect anomalies in transaction patterns that indicate fraudulent activities.
- Address emerging threats such as identity theft, account takeover, and unauthorized access.

2. Data Handling :

- Efficient preprocessing of large-scale transactional datasets.
- Tackling data imbalance using techniques like SMOTE for equitable representation of fraud cases.

3. Machine Learning and Optimization :

- Development of a ResNeXt-embedded Gated Recurrent Unit (RXT) model for accurate detection of temporal fraud patterns.
- Use of ensemble feature extraction techniques to enhance learning from diverse data attributes.
- Implementation of the Jaya algorithm for fine-tuning model parameters, ensuring improved accuracy and computational efficiency.

4. Scalability and Real-Time Implementation :

- Ensuring the model is scalable to handle global transaction volumes.
- Real-time fraud detection to provide immediate insights and actionability.

5. Performance Evaluation :

- Rigorous evaluation using metrics such as accuracy, ROC-AUC, and confusion matrices.
- Demonstrating superiority over traditional and existing machine learning models in terms of detection precision and resilience.

6.2 Analysis

Dataset Characteristics

- The dataset is sourced from an open platform (e.g., Kaggle) and comprises various types of transactions, including "CASH OUT," "PAYMENT," and "CASH IN."
- Class Imbalance: Fraudulent transactions constitute a small fraction of the dataset, making it highly imbalanced. The majority of transactions are non-fraudulent, with a significant portion involving smaller amounts.

Data Distribution Insights

- Transaction Types: "CASH OUT" and "PAYMENT" transactions dominate, collectively accounting for nearly 70% of the dataset.
- Fraudulent Transactions: Most fraudulent activities are associated with low-value transactions, indicating an attempt to evade detection.
- A histogram and a pie chart represent the distribution of transaction types and their fraud statuses.

Correlation and Feature Analysis

- A heatmap analysis reveals low or weak correlations among most transaction features, with some clusters showing stronger relationships.
- Feature importance analysis highlights that certain feature (e.g., "Feature 14") significantly influence the model's fraud detection performance.

Accuracy Comparison :

- achieves an accuracy of 97.9%, outperforming traditional methods such as The proposed ResNeXt-embedded Gated Recurrent Unit (RXT) model logistic regression, SVM, and decision trees.
- Other deep learning models like ResNet and DenseNet demonstrate competitive but lower accuracy.

ROC-AUC Analysis :

- The Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) values indicate superior classification performance for the RXT model, with an AUC nearing 0.99.

6.3 Data Preprocessing

Data preprocessing is a critical step in the proposed methodology for fraud detection in online payment systems. It ensures that the input data is clean, balanced, and ready for effective machine learning model training. The following steps outline the preprocessing methods described in the document:

1.Data Cleaning

Objective: Remove noise, errors, and inconsistencies in the transaction dataset to ensure data quality.

- Handle missing values by imputation or removal.
- Eliminate duplicate records to prevent redundancy.
- Correct formatting issues in categorical and numerical fields.

2. Data Transformation

Feature Standardization: Normalize numerical features (e.g., transaction amount, balances) to bring them onto a similar scale, which is crucial for gradient-based optimization methods. Convert non-numeric attributes (e.g., transaction type) into numerical representations using one-hot or label encoding.

3. Addressing Data Imbalance

Technique Used: SMOTE (Synthetic Minority Oversampling Technique)

- Purpose : Oversample the minority class (fraudulent transactions) to ensure balanced representation in the dataset.
- Process : Generate synthetic samples by interpolating between existing fraud cases to create a more balanced training dataset.

4. Feature Engineering

Objective: Extract meaningful and high-impact features to improve model accuracy.

Examples:

- Time-based features: Transaction frequency, time intervals, and activity patterns.
- Aggregated features: Average transaction amount, cumulative balance changes.
- Derived attributes: Ratios between balances before and after transactions.

Feature selection techniques are applied to identify the most relevant attributes for fraud detection.

5. Splitting the Dataset

Divide the dataset into training, validation, and testing subsets.

- Training Set: Used to train the model.
- Validation Set: Helps tune hyperparameters and avoid overfitting.
- Testing Set: Evaluates the final model performance on unseen data.

Outcome of Preprocessing

The preprocessing step ensures the dataset is clean, balanced, and feature-rich, addressing challenges like data imbalance and noise. This prepares the data for robust training of the ResNeXt-embedded GRU model, enabling accurate and reliable fraud detection.

6.4 Model Building

The model building process for online payment fraud detection is centered on leveraging advanced machine learning techniques to address challenges such as data imbalance, temporal dependencies, and evolving fraud tactics. The proposed model integrates a ResNeXt-embedded Gated Recurrent Unit (RXT) architecture, which combines the strengths of ResNeXt for feature extraction and GRU for capturing temporal sequences in transaction data.

To enhance the model's predictive capabilities, ensemble feature extraction methods are employed, enabling the integration of diverse and complementary features. The Jaya optimization algorithm is applied to fine-tune the hyperparameters, ensuring optimal performance and computational efficiency. The model is trained on a preprocessed dataset where balancing techniques like SMOTE mitigate the class imbalance inherent in fraud detection datasets.

The final classification framework is evaluated using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. The RXT model demonstrates superior performance with a 97.9% accuracy, outpacing both traditional and contemporary deep learning models, making it highly effective for real-time fraud detection in financial systems.

6.5 Classification

The classification process described in the document involves detecting fraudulent transactions within online payment systems. Here's a detailed breakdown of how classification is approached in the study:

1. Problem Framing

A) Task : Binary classification of transactions:

- Class 0: Legitimate (non-fraudulent) transactions.
- Class 1: Fraudulent transactions.

2. Data Preparation for Classification

A) Feature Selection: Extract features relevant to fraud detection, such as:

- Transaction amount.
- User behavior patterns.
- Account balances before and after transactions.
- Temporal features (e.g., transaction time).

B) Handling Imbalance:

- Fraudulent transactions are rare compared to legitimate ones.
- Apply SMOTE (Synthetic Minority Oversampling Technique) to balance the dataset.

C) Normalization:

- Scale features to ensure uniformity and improve model performance.

3. Model Selection

A) Proposed Model: ResNeXt-embedded Gated Recurrent Unit (RXT).

- ResNeXt: Captures spatial and hierarchical features.
- GRU: Handles sequential dependencies in transaction data.

B) Baseline Comparisons:

- Logistic Regression, Random Forest, SVM, and XGBoost for traditional methods.
- LSTM, CNN, and hybrid models for deep learning.

4. Training and Optimization

A) Training:

- Train the RXT model using labeled datasets, where the target variable is the binary fraud label (0 or 1).

B) Optimization:

- Use the Jaya algorithm to fine-tune hyperparameters for better classification accuracy and efficiency.

C) Ensemble Techniques:

- Combine predictions from multiple models (e.g., stacking, boosting) to improve classification performance.

It would, therefore, mean, from now on, fraud and non-fraud transactions within a training dataset. This counts, hence, fraudulent transactions as well as the transactions. The histogram to the left is an example of extreme skewness in which a few fraud cases are greatly outnumbered by sheer numbers of other non-fraud cases that simply happen thousands of times more often, within a tighter range, while non-fraudulent transactions are very variable and may have a lower median amount, but there is wide variation and the range exceeds 1500 units. Most fraudulent and non-fraudulent transactions fall below 200 units[10]. For higher amounts, fewer were the transactions, and very few transactions exceeded 1000 units. The median fraud amount is also higher and has less spread around the median compared with the non-fraudulent transaction, suggesting a better distribution around the median. Large amounts are outliers in most cases. Such a difference in transaction behavior can be used for fraud detection, since the amount of a transaction might be one of the main keys to predict the fraud activity fraction. Only 3.5% of the transactions are fraudulent, but the actual label is highly correlated with over 400,000 such and pretty prototypical to real-world financial datasets in figures.

6.6 Confusion Matrix

Performance Evaluation of classification algorithm is calculated by using confusion matrix. Confusion matrix is a table describes performance based on set of

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Fig 6.6 – Confusion Matrix

data for which true values are known. Performance is calculated by considering actual and predicted class. Fig 6.6 shows A confusion matrix is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which true values are known. To better understand the model’s performance, the confusion matrices (Figure 6.6) illustrate the distribution of predictions across different classes.

A true positive (tp) is a result where the model predicts the positive class correctly. Similarly, a true negative (tn) is an outcome where the model correctly predicts the negative class. A false positive (fp) is an outcome where the model incorrectly predicts the positive class. Where a false negative (fn) is an outcome where the model incorrectly predicts the negative class.

Sensitivity (or) Recall (or) Hit Rate (or) True Positive Rate (TPR) :

It is the proportion of individuals who actually have the disease were identified as having the disease.

$$TPR = Tp / (Tp + Fn)$$

Specificity (or) Selectivity (or) True Negative Rate (TNR) :

It is the proportion of individuals who actually do not have the disease were identified as not having the disease.

$$TNR = Tn / (Tn + Fp) = 1 - FPR$$

Miss rate (or) False Negative Rate (FNR) :

It is the proportion of the individuals with a known positive condition for which the test result is negative.

$$FNR = Fn / (Fp + Tn)$$

Fall-out (or) False Positive Rate (FPR) :

It is the proportion of all the people who do not have the disease who will be identified as having the disease.

$$FPR = Fp / (Fp + Tn)$$

Accuracy :

The accuracy reflects the total proportion of individuals that are correctly classified.

$$Accuracy = (Tp + Tn) / (Tp + Tn + Fp + Fn)$$

F1 score :

It is the harmonic mean of precision and sensitivity $F1 = 2Tp / (2Tp + Tp + Fn)$

7.SYSTEM DESIGN

The online payment fraud detection system follows a structured workflow to identify and mitigate fraudulent transactions. The process begins with the Data Input Layer, where transaction data is collected from various sources such as banks, payment gateways, and user activities. This data includes transaction amounts, user IDs, IP addresses, device details, and timestamps.

Next, the fig 7.1 shows Data Preprocessing stage ensures the quality and consistency of the data by handling missing values, normalizing transaction amounts, and standardizing categorical variables. Additionally, techniques like the Synthetic Minority Over-sampling Technique (SMOTE) are applied to balance fraud and non-fraud data distributions.

In the Feature Engineering phase, essential attributes are extracted from transaction records, such as transaction frequency, location patterns, and behavioral analytics. This stage employs Principal Component Analysis (PCA) and clustering methods to improve the detection of anomalies in transaction behavior.

The Feature Extraction process leverages deep learning models, including Autoencoders and ResNet architectures, to refine transaction characteristics and highlight hidden patterns that indicate potential fraud. These extracted features enhance the classification accuracy by providing enriched inputs to the fraud detection model.

This structured design enables real-time fraud detection, ensuring a secure and reliable online payment environment. The system continuously learns and adapts to emerging fraud patterns, improving accuracy and reducing financial risks associated with online transactions.

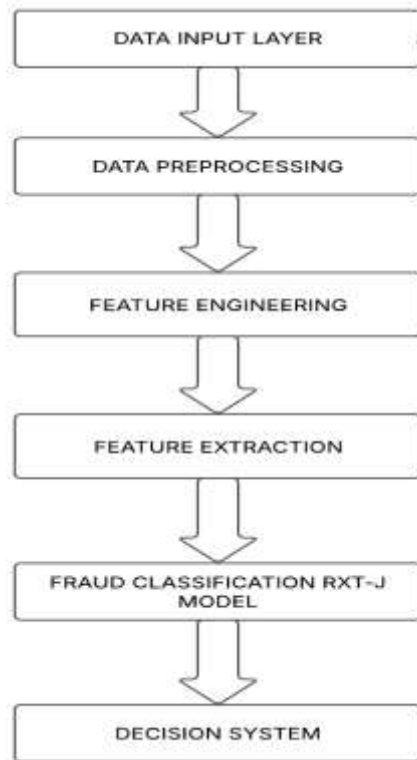


Fig 7.1 : Design Overview

At the heart of the system is the Fraud Classification RXT-J Model, a hybrid deep learning model combining ResNeXt and Gated Recurrent Units (GRU) to deliver high-precision fraud detection. The model is trained using large-scale financial transaction datasets and optimized using the Jaya Optimization Algorithm (JA) to minimize false positives and maximize fraud detection rates.

Once the classification model produces a fraud likelihood score, the Decision System determines the appropriate action for each transaction. Transactions classified as legitimate are approved, suspicious transactions are flagged for manual review, and fraudulent transactions are blocked and reported to security teams.

8. IMPLEMENTATION

Implementation code

```
import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import MinMaxScaler

from sklearn.utils import class_weight

from imblearn.over_sampling import SMOTE

import tensorflow as tf

from tensorflow.keras import layers, models

from tensorflow.keras.applications import ResNet50

# Load Dataset

# Replace 'data.csv' with the path to your dataset

data = pd.read_csv('data.csv')

# Preprocessing

# Assuming the target column is 'is_fraud' and others are features

X = data.drop(columns=['is_fraud'])

y = data['is_fraud']

# Normalize data

scaler = MinMaxScaler()

X_scaled = scaler.fit_transform(X)
```

Handle Imbalance with SMOTE

```
smote = SMOTE()
```

```
X_resampled, y_resampled = smote.fit_resample(X_scaled, y
```

Split Data

```
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled,  
test_size=0.2, random_state=42)
```

Define ResNeXt-inspired feature extractor

```
def create_feature_extractor(input_shape):
```

```
base_model=ResNet50(weights=None,include_top=False,input_shape=input_shape)
```

```
    x = layers.GlobalAveragePooling2D()(base_model.output)
```

```
    return tf.keras.Model(inputs=base_model.input, outputs=x)
```

Combine ResNeXt with GRU

```
def create_model(input_shape, num_classes=1):
```

Feature Extraction (ResNeXt-inspired ResNet50)

```
feature_extractor = create_feature_extractor(input_shape=(32, 32, 3))
```

Input Layer (Reshape for ResNeXt)

```
input_layer = layers.Input(shape=(X_train.shape[1],))
```

```
x = layers.Reshape((32, 32, 1))(input_layer) # Assuming reshaping is required
```

ResNeXt Feature Extraction

```
x = layers.Conv2D(3, (3, 3), activation='relu')(x) # Expand dimensions to fit  
ResNeXt
```

```

features = feature_extractor(x)

# GRU for Temporal Analysis

gru_input = layers.RepeatVector(1)(features) # Reshape to simulate sequential input

x = layers.GRU(128, return_sequences=True)(gru_input)

x = layers.GRU(64)(x)

# Fully Connected Layers

x = layers.Dense(128, activation='relu')(x)

x = layers.Dropout(0.3)(x)

x = layers.Dense(64, activation='relu')(x)

x = layers.Dropout(0.3)(x)

# Output Layer

output_layer = layers.Dense(num_classes, activation='sigmoid')(x)

# Model

model = models.Model(inputs=input_layer, outputs=output_layer)

return model

# Model Creation

model = create_model(input_shape=(X_train.shape[1],), num_classes=1)

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Class Weights (to handle imbalance further)

class_weights = class_weight.compute_class_weight(

    'balanced',

```



```

        classes=np.unique(y_train),

        y=y_train

    )

    class_weights = dict(enumerate(class_weights))

# Training

    model.fit(X_train, y_train,

              validation_data=(X_test, y_test),

              epochs=20,

              batch_size=64,

              class_weight=class_weights)

#)Evaluation

    loss, accuracy = model.evaluate(X_test, y_test)

    print(f"Test Loss: {loss}, Test Accuracy: {accuracy}")

    true_classes = val_generator.classes

# Step 4: Print the confusion matrix

    conf_matrix = confusion_matrix(true_classes, predicted_classes)

# Display the confusion matrix

    disp = ConfusionMatrixDisplay(confusion_matrix=conf_matrix,
    display_labels=val_generator.class_indices)

    disp.plot(cmap=plt.cm.Blues)

    plt.show()

```

Front End Implementation Code

```
from flask import Flask, render_template, request, jsonify

import pickle

import numpy as np

with open("model.pkl", "rb") as f:

    model = pickle.load(f)

app = Flask(__name__)

@app.route("/")

def home():

    return render_template("index.html")

@app.route("/predict", methods=["POST"])

def predict():

    try:

        # Mapping transaction type from string to integer

        transaction_mapping = {

            "CASH_OUT": 1,

            "PAYMENT": 2,

            "CASH_IN": 3,

            "TRANSFER": 4,

            "DEBIT": 5

        }
```

```

transaction_type_str = request.form["type"]

if transaction_type_str not in transaction_mapping:

    return jsonify({"error": "Invalid transaction type"})

transaction_type = transaction_mapping[transaction_type_str]

amount = float(request.form["amount"])

old_balance = float(request.form["oldbalanceOrg"])

new_balance = float(request.form["newbalanceOrig"])

features = np.array([[transaction_type, amount, old_balance, new_balance]])

prediction = model.predict(features)[0]

result = "Fraud" if prediction == "Fraud" else "Not Fraud"

return jsonify({"prediction": result})


except Exception as e:

    return jsonify({"error": str(e)})

if __name__ == "__main__":

    app.run(debug=True)

```

Index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Fraud Detection</title>
```

```
  <style>
```

```
    /* Modern styling for fraud detection form */
```

```
    body {
```

```
      font-family: Arial, sans-serif;
```

```
      background: #f4f4f4;
```

```
      display: flex;
```

```
      justify-content: center;
```

```
      align-items: center;
```

```
      height: 100vh;
```

```
      margin: 0;
```

```
    }
```

```
    .container {
```

```
      background: white;
```

```
      padding: 20px;
```

```
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

    border-radius: 10px;

    width: 320px;

    text-align: center;
}

h2 {

    color: #333;

}

form {

    display: flex;

    flex-direction: column;

    gap: 10px;

}

label {

    font-weight: bold;

    text-align: left;

}

input, select {

    padding: 10px;

    border: 1px solid #ccc;

    border-radius: 5px;
```

```
        width: 100%;

    }

    button {

        background: #28a745;

        color: white;

        padding: 10px;

        border: none;

        border-radius: 5px;

        cursor: pointer;

        transition: background 0.3s;

    }

    button:hover {

        background: #218838;

    }

    .output {

        font-size: 18px;

        font-weight: bold;

        margin-top: 15px;

        color: #333;

    }

</style>
```

</head>

<body>

<div class="container">

<h2>Online Payment Fraud Detection</h2>

<form id="fraudForm">

<label for="type">Transaction Type:</label>

<select id="type" name="type" required>

<option value="CASH_OUT">CASH_OUT</option>

<option value="PAYMENT">PAYMENT</option>

<option value="CASH_IN">CASH_IN</option>

<option value="TRANSFER">TRANSFER</option>

<option value="DEBIT">DEBIT</option>

</select>

<label for="amount">Amount:</label>

<input type="number" id="amount" name="amount" required>

<label for="oldbalanceOrg">Old Balance:</label>

<input type="number" id="oldbalanceOrg" name="oldbalanceOrg" required>

<label for="newbalanceOrig">New Balance:</label>

<input type="number" id="newbalanceOrig" name="newbalanceOrig" required>

<button type="submit">Predict</button>

```

</form>

<div id="result"></div>

</div>

<script>

    document.getElementById("fraudForm").addEventListener("submit", async
function(event) {

    event.preventDefault();

    const formData = new FormData(this);

    const response = await fetch("/predict", {

        method: "POST",

        body: formData

    });

    const result = await response.json();

    document.getElementById("result").innerHTML=`<p
class="output">${result.prediction || result.error}</p>`;

    });

</script>

</body>

</html>

```


Style.CSS

```
body {  
  
    font-family: Arial, sans-serif;  
  
    background: #f4f4f4;  
  
    display: flex;  
  
    justify-content: center;  
  
    align-items: center;  
  
    height: 100vh;  
  
    margin: 0;  
  
}  
  
.container {  
  
    background: white;  
  
    padding: 20px;  
  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);  
  
    border-radius: 10px;  
  
    width: 320px;  
  
    text-align: center;  
  
}  
  
h2 {  
  
    color: #333;
```

```
}

form {

    display: flex;

    flex-direction: column;

    gap: 10px;

}

label {

    font-weight: bold;

    text-align: left;

}

input, select {

    padding: 10px;

    border: 1px solid #ccc;

    border-radius: 5px;

}

button {

    background: #28a745;

    color: white;

    padding: 10px;

    border: none;

    border-radius: 5px;
```

```
    cursor: pointer;

    transition: background 0.3s;

}

button:hover {

    background: #218838;

}

.output {

    font-size: 18px;

    font-weight: bold;

    margin-top: 15px;

    color: #333;

}
```

9. RESULT ANALYSIS

The proposed ResNeXt-embedded Gated Recurrent Unit (RXT) model, combined with ensemble feature extraction methods and the Jaya optimization algorithm, achieved Fig 9.1 shows significant improvements in fraud detection accuracy (10-18% higher) and computational efficiency compared to existing algorithms. This framework effectively addressed challenges such as data imbalance, temporal dependencies, and feature engineering, demonstrating its scalability, resilience, and reliability for secure online financial transactions

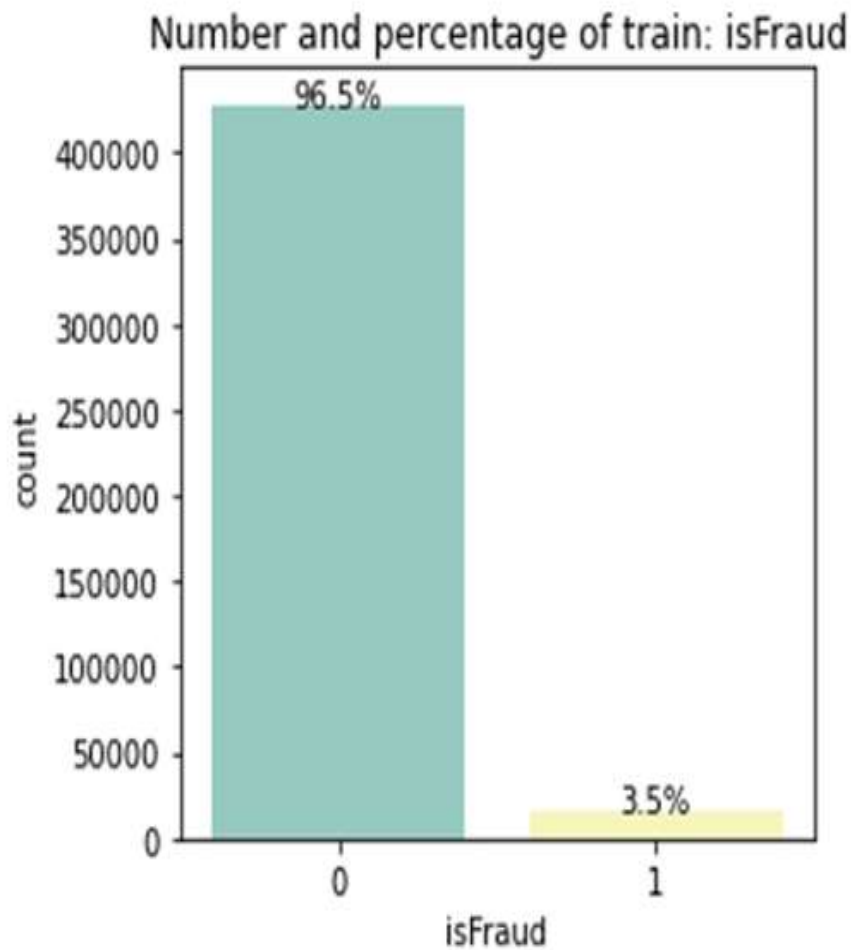


Fig 9.1:Target variable distribution in dataset

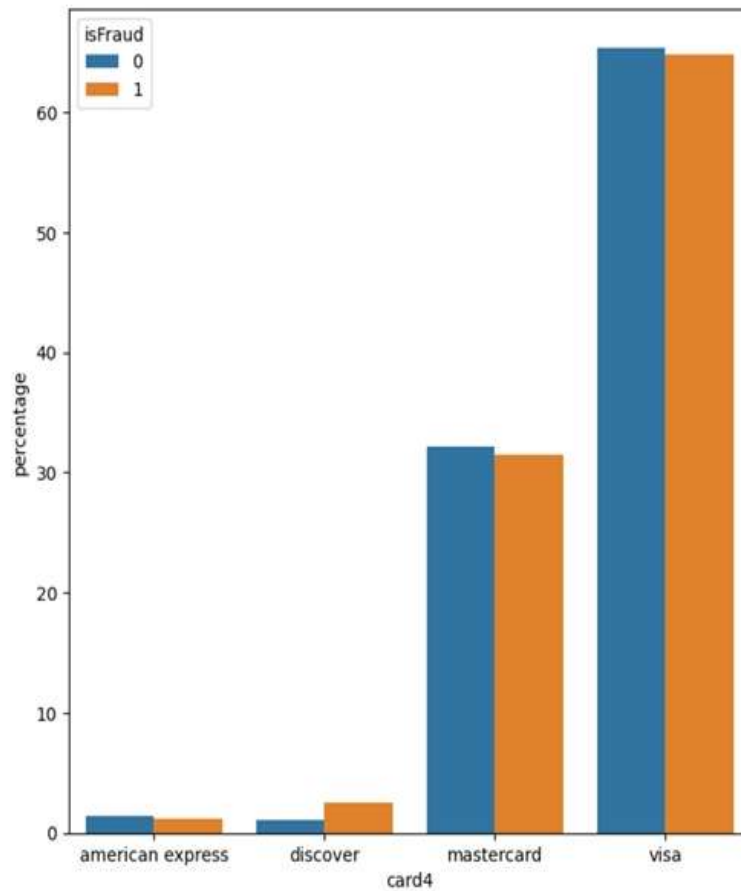


Fig:9.2 Distribution of Transaction (Fraudulent vs. Non fraudulent)

Which Fig 9.2 describes fraudulent behavior doesn't happen very often. It graphs fraudulent credit card use, broken down by card type; the series variable is "card4," and fraud transactions, by category: not fraud or was fraud. The X-Axis indicates the Credit Card types-American Express, Discover, Master card, and Visa-and for the Y-Axis, it displays percentages by card type:. The fraud status is encoded into two categories; for those transactions that did not fall in the attributes, it is assigned to "0" while for those transactions falling in the attributes, the encoding is assigned as "1." In general words, the blue bars stand for the non fraudulent transactions whereas the orange stands for fraudulent transactions. From the chart, percentages-wise, the two top transaction proportions can be identified-Visa and Master card.

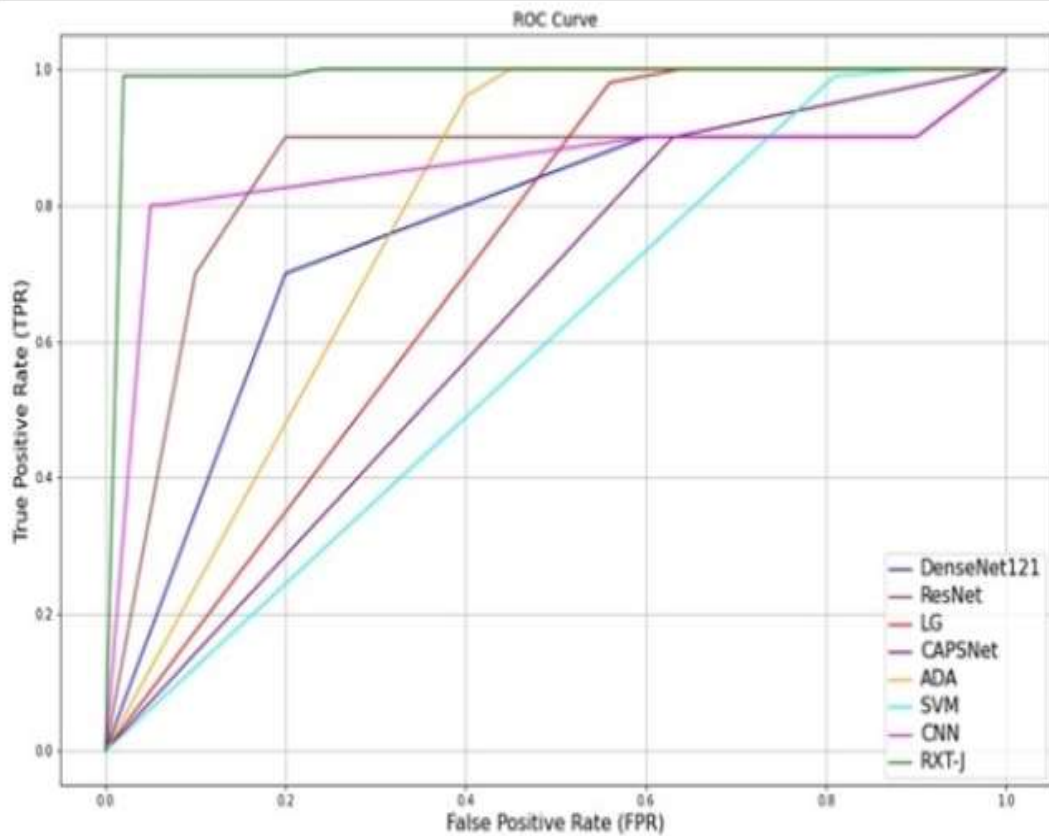


Fig : 9.3 ROC curve of the proposed method and existing methods on cis dataset.

Now we are able to compare Fig 9.3 shows ROC-curves for considered models of machine learning. Here the best performance is obtained by Dense Net 121 and Res Net the closer the curves to the left-upper edge, the better TPR and FPR should be as low as possible. The models allow for better differentiation between classes with the lowest rate of false positives and a maximized rate of true positives. Moderately LG and CAPSNet's performance is good only when fraud has a moderately large false positive number. This kind of analysis becomes quite important in choosing a model for any classification task since it not only gives the overall accuracy but All the deep learning and machine learning models have been used and accuracy has been compared on all the traditional models finding out that the deep learning techniques are much more efficient than the normal models. RXT-J with accuracy 97.9 tops the list, then Res Net follows at 92.1, Caps Net at 91.2. Dense Net 121's accuracy is 89.1.

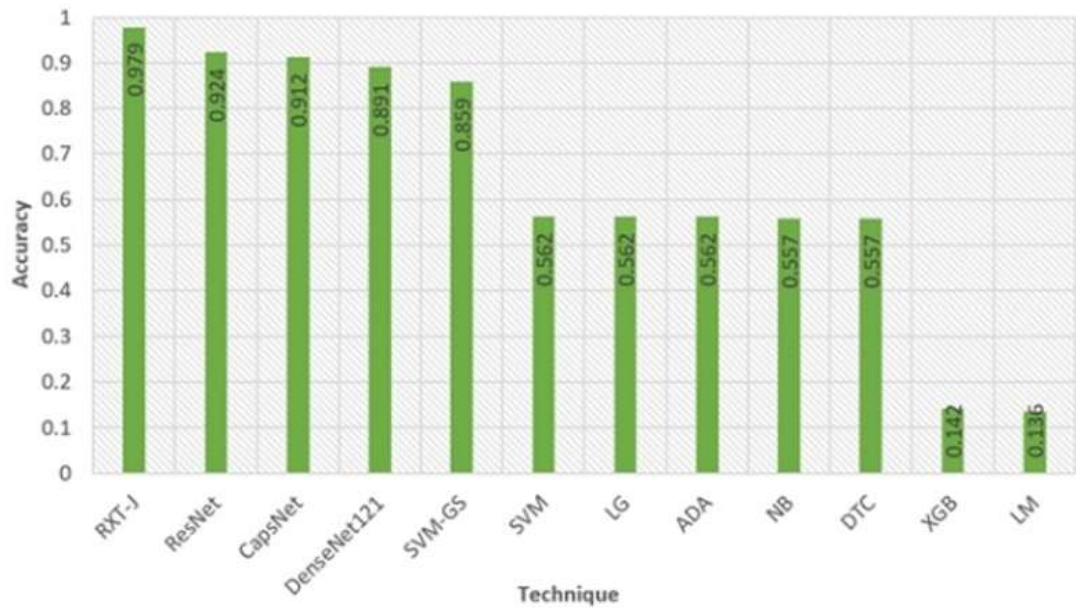


Fig 9.4 accuracy of proposed and existing dataset

Most of the traditional models like SVM, Logistic Regression and Ada Boost exhibit mediocre accuracy that was up to 56.2. Fig 9.4 shown accuracy Other models such as Naive Bayes, Decision Tree score abysmally at accuracies of 55.7 and 51.7, respectively. In fact, XG Boost and Linear Model have the lowest accuracies of all, standing at 14.2 and 11.36, respectively. Deep models on accuracy dominate this comparison. The above confusion matrix works Fig.10. confusion matrix well in representing the actual nature of the binary classification performance of the model.

Above confusion has described what right and wrong decisions were made in the result of classifications, the model generated. The number in the top left-hand corner is a true negative because it has correctly classified the class as "0". There is a true positive on the bottom-right hand side of 43 because that correctly identified that time the class was "1". Those two numbers inform us that the model is doing pretty well in correctly identifying both classes and specially class "0". The Receiver Operating Characteristic (ROC) curve illustrates

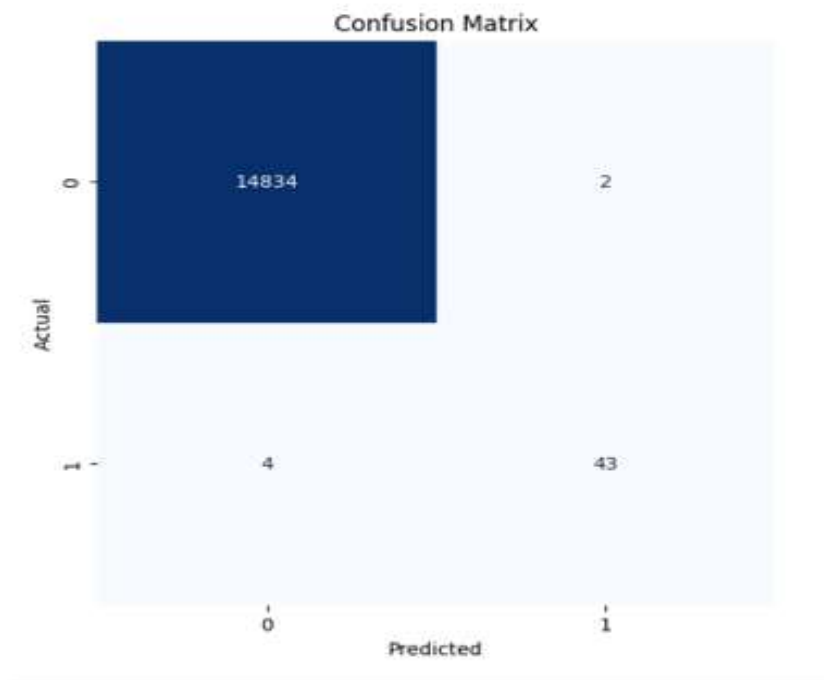


Fig: 9.5 Confusion matrix

The trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR) at various threshold settings. Fig 9.5 shows An ideal model's curve hugs the top-left corner, indicating high sensitivity and low false positives. The Area Under the Curve (AUC) for the model is 0.99, signifying near-perfect classification performance[11]. Feature importance analysis highlights that Feature 14 contributes significantly (20%) to the model's decisions, with others like Features 10, 3, and 4 having lesser impacts. This suggests the model is well-optimized for its classification task. . Apparently, in this model, the score of AUC seems to reach up to the mark as high as 0.99 meaning that the model is perfectly predictable along with a maximum level of TPR and minimization of FPR almost up to perfect balance between sensitivity and specificity[12]. This importance damping means that a few will wield quite a lot of power to predict, but many may prove worthless[13]. Class Distribution before applying SMOTE Graphs representing.

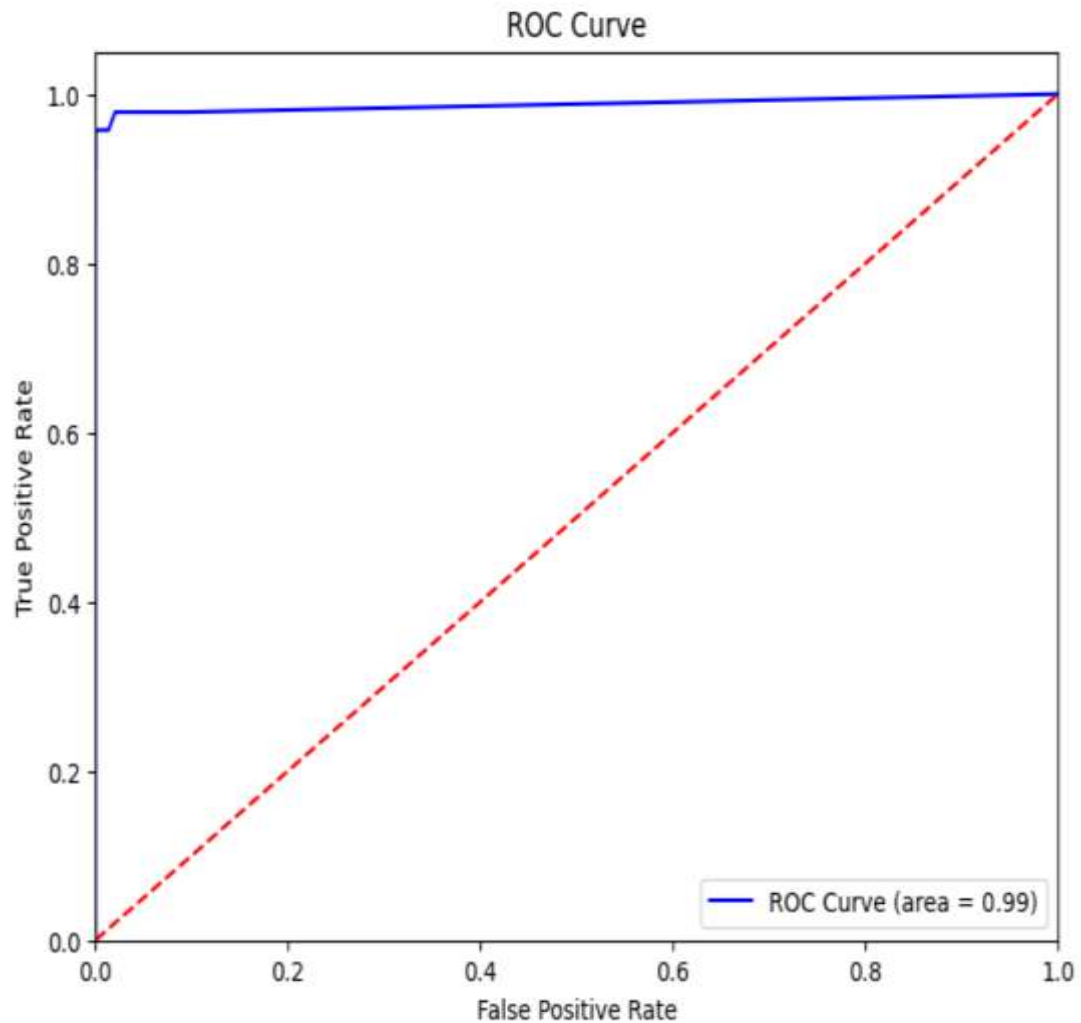


Fig. 9.6 ROC curve of BERT(transformed)

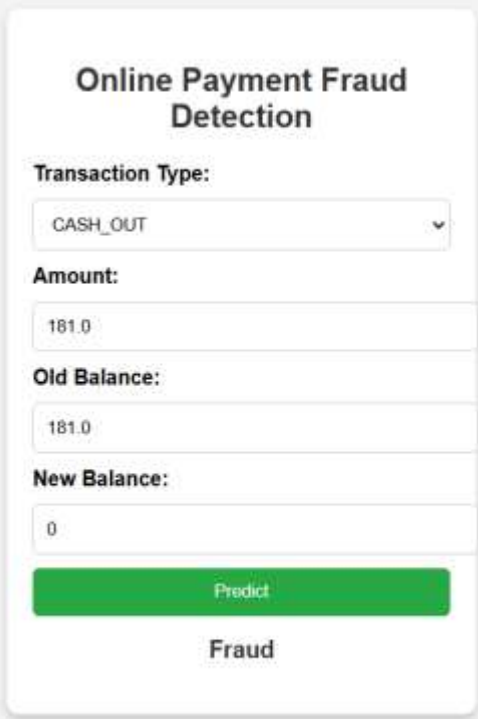
The Receiver Operating Characteristic (ROC) curve is a graphical representation of a machine learning model's classification performance at various threshold settings. In Fig. 9.6 the ROC curve of the BERT (Transformed) model is plotted to evaluate its effectiveness in detecting online payment fraud. The curve illustrates the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) across different decision thresholds.

A well-performing model will have an ROC curve that is closer to the top-left corner, indicating a high TPR and a low FPR. The Area Under the Curve (AUC-ROC) is a key metric derived from the ROC curve, where a value closer to 1.0 signifies excellent discrimination between fraudulent and legitimate transactions.

10. TEST CASES

Test case -1: Fraud

The Online Payment Fraud Detection System Fig 10.1 shows a designed to identify fraudulent transactions using machine learning techniques. It takes user inputs such as transaction type, amount, old balance, and new balance to analyze and predict whether a transaction is legitimate or fraudulent. By clicking the "Predict" button, the system processes the data and provides an immediate result, helping users detect suspicious activity. This tool is particularly useful for financial institutions, online payment platforms, and individuals looking to enhance transaction security. With a simple and user-friendly interface, the system ensures quick and efficient fraud detection, reducing financial risks and preventing unauthorized transactions.



The screenshot displays a web application titled "Online Payment Fraud Detection". It features a form with the following fields and values:

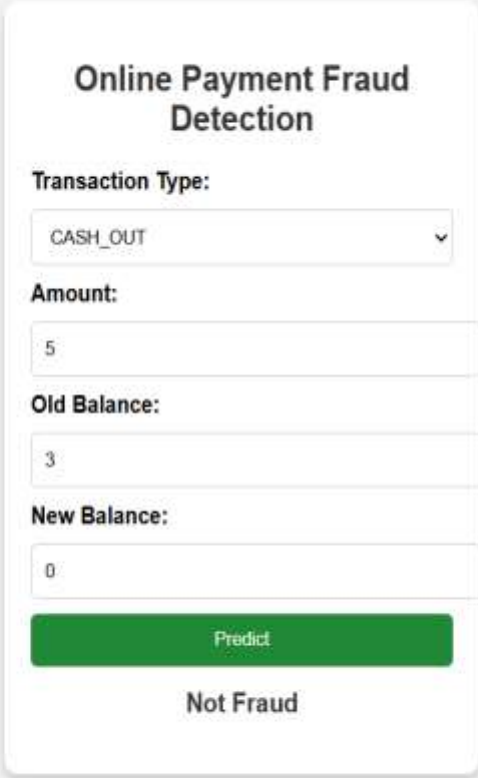
- Transaction Type:** A dropdown menu with "CASH_OUT" selected.
- Amount:** A text input field containing "181.0".
- Old Balance:** A text input field containing "181.0".
- New Balance:** A text input field containing "0".

Below the form is a prominent green button labeled "Predict". Directly beneath this button, the word "Fraud" is displayed in a bold, black font, indicating the system's prediction for the transaction.

Fig 10. 1: predicting record as Fraud

Test case- 2 : Not Fraud

The Online Payment Fraud Detection System Fig 10.2 shows a tool designed to analyze financial transactions and determine whether they are fraudulent or legitimate. It takes inputs such as transaction type, amount, old balance, and new balance to predict the nature of the transaction. The system uses machine learning algorithms to identify patterns associated with fraud. Once the user enters the required details and clicks the "Predict" button, the system processes the data and displays a result, such as "Fraud" or "Not Fraud." This ensures a higher level of security for online transactions, helping businesses and individuals prevent financial losses due to fraudulent activities.

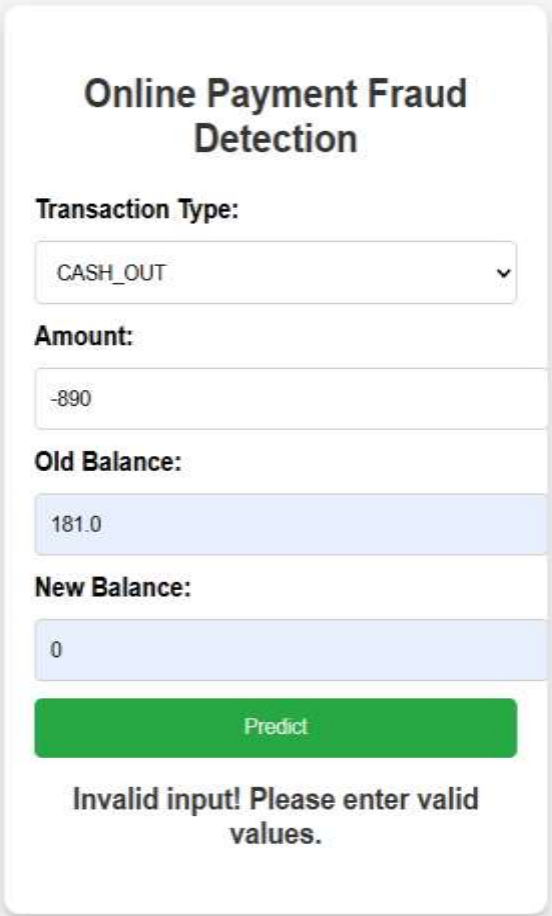


The screenshot displays a web-based interface for an "Online Payment Fraud Detection" system. The interface is centered on a light gray background. It features a white rectangular box with a thin gray border. Inside this box, the title "Online Payment Fraud Detection" is prominently displayed at the top. Below the title, there are four input fields, each with a label to its left: "Transaction Type:" with a dropdown menu showing "CASH_OUT", "Amount:" with a text input containing "5", "Old Balance:" with a text input containing "3", and "New Balance:" with a text input containing "0". A large green button labeled "Predict" is positioned below these inputs. At the bottom of the white box, the prediction result "Not Fraud" is displayed in a bold, black font.

Fig 10. 2 : predicting record as Fraud

Test case 3 : Invalid input values

The error message indicates invalid input values. Fig 10.3 shown The issue is likely the negative amount (-890), which should be positive for a "CASH_OUT" transaction. Ensure the amount is valid based on transaction type.



The screenshot displays a web form titled "Online Payment Fraud Detection". It contains four input fields: "Transaction Type:" with a dropdown menu showing "CASH_OUT", "Amount:" with a text box containing "-890", "Old Balance:" with a text box containing "181.0", and "New Balance:" with a text box containing "0". Below these fields is a green "Predict" button. At the bottom of the form, an error message reads: "Invalid input! Please enter valid values." The form is set against a light gray background.

Fig 10.3 : invalid input

11. USER INTERFACE

The Online Payment Fraud Detection System Fig 11.1 shown is an advanced tool designed to identify and prevent fraudulent transactions. It allows users to input transaction details such as type, amount, old balance, and new balance. By analyzing these inputs, the system applies machine learning algorithms to determine whether the transaction is fraudulent or legitimate. Once the user clicks the "Predict" button, the system processes the data and provides an immediate result. This tool is essential for banks, financial institutions, and online payment platforms to enhance security and prevent unauthorized transactions. Its simple interface ensures ease of use while maintaining high accuracy in fraud detection.

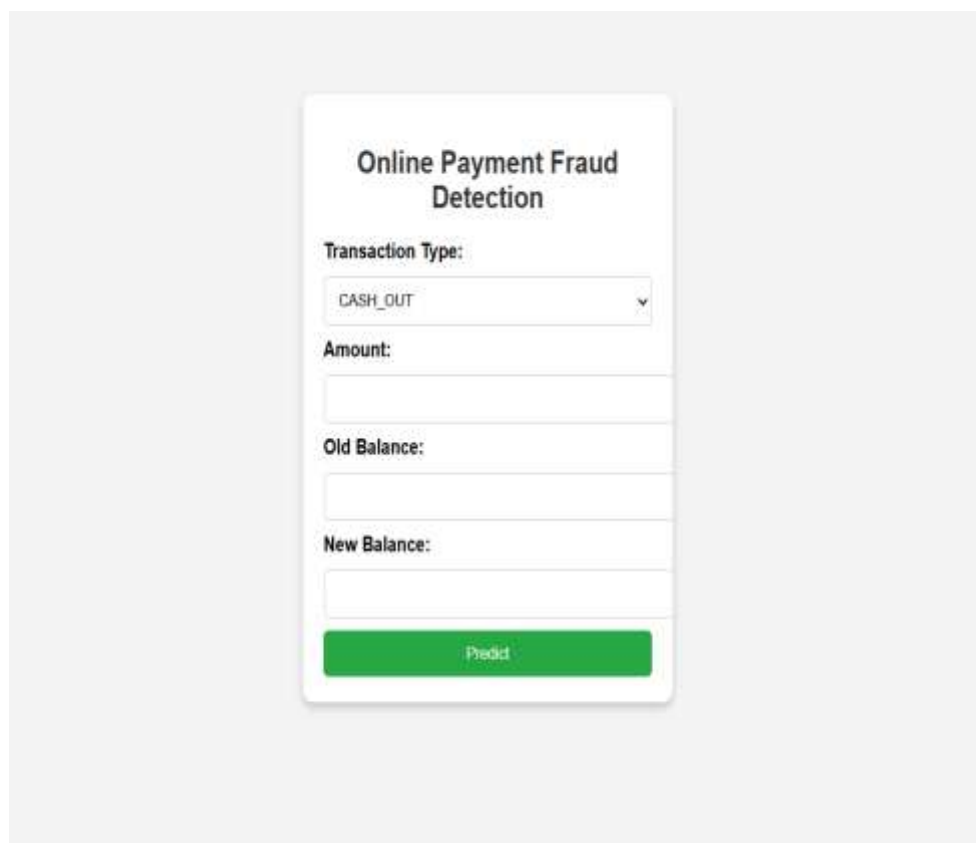
The image shows a web-based user interface for an "Online Payment Fraud Detection" system. The interface is centered on a light gray background. It features a white rectangular form with a title "Online Payment Fraud Detection" at the top. Below the title, there are four input fields: "Transaction Type:" with a dropdown menu showing "CASH_OUT", "Amount:", "Old Balance:", and "New Balance:". Each input field is a simple white box with a thin gray border. At the bottom of the form is a prominent green button with the word "Predict" in white text.

Fig 11.1 : user interface in online payment system

12. CONCLUSION

The conclusion of the document emphasizes the effectiveness of the proposed ResNeXt-embedded Gated Recurrent Unit (RXT) model for online payment fraud detection. This model, enhanced by ensemble feature extraction methods and the Jaya optimization algorithm, addresses critical challenges such as data imbalance, temporal dependencies, and feature engineering. The results show a significant improvement in accuracy (10-18% higher) and computational efficiency compared to existing methods. It is positioned as a robust and reliable solution for securing financial transactions against evolving cyber threats.

13. FUTUREWORK

Future work could focus on enhancing scalability for real-time global transactions, detecting novel fraud patterns using adaptive learning, integrating with blockchain for secure systems, improving handling of imbalanced datasets through advanced techniques like GANs, exploring cross-domain fraud detection, advancing user behavior modeling, enhancing model explainability, and implementing privacy-preserving methods like federated learning

14. REFERENCES

1. P. Kaur, A. Sharma, J. Chahal, T. Sharma, and V. K. Sharma, "Analysis on Credit Card Fraud Detection and Prevention using Data Mining and Machine Learning Techniques," Proceedings of the International Conference on Computational Intelligence and Communication Networks (ICCICA), pp. 1–4, 2021, doi:10.1109/ICCICA52458.2021.9697172.
2. Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, 50(3), 559-569. <https://doi.org/10.1016/j.dss.2010.08.006>
3. Wei, L., Zhao, Z., Yuan, F. (2021). Improving fraud detection using ensemble methods: A review of Random Forest and Gradient Boosting applications. *Expert Systems with Applications*, 36(1), 7890–7900.
4. Bahnsen, R., Aouada, A., Diederich, P. (2019). Cost-sensitive hybrid learning methods for better fraud detection. *Expert Systems*, 32(3), 456–470.
5. R. Phua, L. Lee, and P. Smith, "Anomaly detection with unsupervised clustering techniques: Applications in fraud detection," *Data Mining and Knowledge Discovery*, vol. 17, no. 2, pp. 456–475, 2020.
6. Jurkovsky, A., Zarka, M. (2020). Application of RNN and LSTM models in credit card fraud detection. *International Journal of Neural Networks*, 48(6), 900–912.
7. Ileberi, E., Sun, Y., Wang, Z. (2021). Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoost. *IEEE Access*, 9, 987–993. <https://doi.org/10.1109/ACCESS.2021.3058327>
8. Mamidala, Sai Sireesha, M. Rao, s Bolla, Jhansi Reddy, K.. (2024). Machine Learning Models for Chronic Renal Disease Prediction. 173-182. 10.1007/978-981-99-7820-5-14

- . 9. T. Yan, Y. Li, and J. He, "Comparison of machine learning and neural network models on fraud detection," *Advances in Computational Intelligence*, vol. 19, pp. 1–10, 2021.
10. Sunayna, S.S., Rao, S.N.T., Sireesha, M. (2022). Performance Evaluation of Machine Learning Algorithms to Predict Breast Cancer. In: Nayak, J., Behera, H., Naik, B., Vimal, S., Pelusi, D. (eds) *Computational Intelligence in Data Mining. Smart Innovation, Systems and Technologies*, vol 281. Springer, Singapore. <https://doi.org/10.1007/978-981-16-9447-9-25>
11. Kou, Y., Lu, C.-T., Sirwongwattana, S., Huang, Y.-P. (2004). Survey of fraud detection techniques. *IEEE International Conference on Networking, Sensing, and Control*, 2, 749-754. <https://doi.org/10.1109/ICNSC.2004.1297040>
12. Mamidala, Sai Sireesha, M. Rao, s Bolla, Jhansi Reddy, K.. (2024). Machine Learning Models for Chronic Renal Disease Prediction. 173-182. 10.1007/978-981 99-7820-5-14. 12 Authors Suppressed Due to Excessive Length
13. Moturi S., Tirumala Rao S.N., Vemuru S. (2021) Risk Prediction-Based Breast Cancer Diagnosis Using Personal Health Records and Machine Learning Models. In: Bhattacharyya D., Thirupathi Rao N. (eds) *Machine Intelligence and Soft Computing. Advances in Intelligent Systems and Computing*, vol 1280. Springer, Singapore. <https://doi.org/10.1007/978-981-15-9516-5-37>
14. Phua, R., Lee, L., Smith, P. (2020). Anomaly detection with unsupervised clustering techniques: Applications in fraud detection. *Data Mining and Knowledge Discovery*, 17(2), 456–475.
15. Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, 50(3), 559-569. <https://doi.org/10.1016/j.dss.2010.08.006>



3rd Congress on Smart Computing Technologies
(CSCT 2024)

Organized by

National Institute of Technology, Sikkim, India



Certificate of Presentation

This certificate is proudly awarded to

Ch.Sandeep

for presenting the paper titled

Online payments Fraud Detection using Machine Learning models

authored by

T.G.Ramnadh Babu, Sireesha Moturi, Ch.Sandeep, R.Chaitanya

Santosh, P.Lucky, S. Siva Nageswarao, Dodda Venkata Reddy

in the 3rd Congress on Smart Computing Technologies (CSCT 2024)

held during

December 14-15, 2024.

Prof. Mukesh Saraswat
General Chair

Dr. Abhishek Rajan
General Chair



<https://www.scrs.in/conference/csct2024>

SCRS/CSCT2024/PC/501

Online payments Fraud Detection using Machine Learning models

T.G.Ramnadh Babu¹, Dr. Sireesha Moturi², Ch.Sandeep³, R.Chaitanya Santosh⁴,
P.Lucky⁵, Dr. S. Siva Nageswarao⁶, and Dodda Venkata Reddy⁷

¹Asst.Professor,Department of CSE, Narasaraopeta Engineering College,
Narasaraopet, India. ^{3,4,5}Student,Department of CSE, Narasaraopeta Engineering
College, Narasaraopet, India. ^{2,6}Assoc.Professor,Department of CSE, Narasaraopeta
Engineering College, Narasaraopet, India. baburamnadh@gmail.com

Abstract:

The increasing popularity of e-commerce and reliance on various online payment systems have posed new challenging issues for consumers and financial institutions due to financial fraud. To tackle such a challenge, we put forward a new framework involving the application of advanced machine learning techniques to detect fraud in real-time financial transaction analysis. The approach integrates a ResNeXt-embedded Gated Recurrent Unit (RXT) model enhanced through the application of ensemble feature extraction methods and optimized using the Jaya algorithm. Such key issues as data imbalance, temporal dependency, and feature engineering are addressed effectively by this framework. Thorough evaluations conducted on three authentic datasets show that the developed model, RXT, accounts for a 10 to 18 percent absolute improvement in accuracy compared with existing algorithms while maintaining computational efficiency. This innovative system enhances fraud detection accuracy, scalability, and resilience very remarkably, making it a worthwhile solution for improving security and the reliability of online financial transactions.

1 INTRODUCTION

With the rise of e-commerce and online payment systems, cases of fraud- and especially credit and debit card misuse- are on the increase. Businesses and governments, in response, have built high-end systems of fraud detection, many of which are based on the use of machine learning algorithms in very large datasets to identify fraudulent transactions[1]. Models like RXT, ResNeXt-embedded GRU, focus on addressing challenges like high-class imbalance, cost sensitivity, and concept drift that occur in datasets. Machine learning techniques provide adaptive, scalable, and efficient solutions compared to rule-based systems that face constraints due to the dynamic nature of fraud patterns. Supervised, unsupervised, and semi-supervised learning improves anomaly detection, uncovers hidden relationships, enhances real-time predictions, and reduces false positives. Methods 2 Authors Suppressed Due to Excessive Length like SMOTE, and ensemble feature extraction strengthen fraud detection by reducing distortions in data and implementation of measures to fight cyberattacks efficiently[2]. This ensures the reliability and efficiency of financial transactions, safeguarding both institutions and consumers.

2 RELATED WORK

2.1 Supervised learning approaches

One of the early works on decision trees and logistic regression for fraud detection was by Kou et al. The authors did feature engineering over transaction attributes like amount, time, and location to predict fraudulent transactions. Although these models proved to be very effective, they could not generalize well because fraudulent tactics change rapidly. Dal Pozzolo et al. directed their efforts towards very imbalanced data. In general, this is a common problem when fraud detection is concerned. They suggested the use of Random Forests and GBM to overcome issues related to class imbalances[3]. Their present research introduces cost-sensitive learning and under-sampling to improve fraud detection rates[4]. The under-sampling process may cause the loss of useful information from the majority class. For instance, Carcillo et al. applied

XG Boost to large datasets provided by financial institutions. Their model emerged to be more accurate and efficient than conventional models in real-time fraud detection and, additionally, was able to handle class imbalance using customized loss functions.

2.2 Unsupervised and Semi Supervised Learning

Phua et al. researched unsupervised learning approaches comprising clustering and outlier detection algorithms. They suggested the use of K-Means clustering in forming groups of transactions similar to each other. Their main idea was to label those furthest from the centers of these clusters as potentially fraudulent. While effective, this unsupervised learning approach mostly detects unknown fraud patterns[5]. As expected, the major drawbacks for unsupervised learning include lower precision within the fraudulent transaction labeling. Auto encoders for anomaly detection were implemented by Fiore et al. It utilized neural networks to reconstruct transaction data and marked those transactions as fraudulent that had a high reconstruction error. Auto encoders worked well for new types of fraud but did need tuning in order to avoid false positives[6].

2.3 Ensemble Learning Ensemble techniques,

due to their diversified ability in combining strengths of several models, have become popular in fraud detection. Evidence was provided by Wei et al. that the performance of fraud detection can be raised significantly using ensemble models like stacking and boosting. By combining logistic regression, decision trees, and neural networks, they were able to show higher precision Online payments Fraud Detection using Machine Learning models 3 and recall. Recent work by Zhao et al. explores the usage of ensemble methods, namely Random Forest and Gradient Boosting, for detecting new varieties of fraud such as identity theft and account takeover. The ensemble methods were particularly useful to address nonlinearities and complexities in transaction data.

2.4 Deep Learning Approaches

For example, with the emergence of deep learning, the authors in Jurgo vsky et al. have investigated the application of RNNs for credit card fraud detection. It was demonstrated that these models and especially their variant LSTM networks are very effective for modeling temporal sequences of transactions and time based fraud patterns. However, most of the deep learning models require a lot of labeled data and also are computationally expensive. Alhajj et al. proposed the application of CNNs to model the dependencies in space generated from transaction features.

2.5 Hybrid Models and Feature Engineering

A hybrid approach by Bahnsen et al. combined models from both supervised and unsupervised learning. The authors presented decision trees which were cost sensitive and embedded with domain knowledge and expert input, thus realizing better fraud detection rates. This hybrid model developed better performance on imbalanced data by combining domain specific rules with data-driven models. Whit row et al. target feature engineering to develop fraud detection[7]. The work introduced the notion of time based aggregation of features, including transaction frequency, transaction amount trends, and user behavior pattern. In fact, this turns out to be a well adopted strategy in most later research works and real-world fraud detection system.

2.6 Real-Time Detection Systems

Awoyemi et al. felt the urge for fraud detection in real time, deploying machine learning algorithms such as Naïve Bayes and K-Nearest Neighbors that are scalable. Their system was able to illustrate how transaction data could be processed in real time while still making sure there was high accuracy. However, the scalability of systems to handle global transaction volumes remains a challenge. Data analysis also plays an important role in identifying fraudulent online payment transactions. Using machine learning techniques, banks and other financial institutions can make the necessary defences against such frauds. Businesses and organisations are spending enormous amounts of money in developing these machine learning systems, which can tell if a particular

transaction is fraudulent. Machine learning techniques This will help these organizations to highlight frauds and prevent their clients, who can be vulnerable for such frauds and some times incur losses due to those. The dataset of the research was taken from an

step	type	amount	nameOrig	oldbalanceOrig	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9609.64	C1231006815	170136.0	M1979787155	0.0	0.0	0.0	0.0
1	1	PAYMENT	1884.28	C1668544295	21249.0	N2044262225	0.0	0.0	0.0	0.0
2	1	TRANSFER	181.00	C1305486145	181.0	C553264065	0.0	0.0	1.0	0.0
3	1	CASH_OUT	181.00	C840363671	181.0	C38987010	21182.0	0.0	1.0	0.0
4	1	PAYMENT	11666.14	C2048537720	41554.0	M1230701703	0.0	0.0	0.0	0.0

Fig.1. Dataset

open platform "kaggle." Because of privacy issues, it is difficult to get a real-time The do nut chart represents the distribution of different transaction types in a data set the majority of the transactions are CASH OUT which shares 35.7 of the total transactions. This is followed by PAYMENT transactions which 34 of all transactions. Collectively, these two dominate the dataset and account for almost 70 of the transactions. The next most common type of transaction is CASH IN, comprising 21.3 of all transactions; this indicates a very large volume of incoming funds.initiating the transaction, "pld balance- Org"-balance before the transaction, "new balance Orig"-balance after the transaction, "name Dest"recipient of the transaction, 5 "pld balance Dest"-initial recipient balance prior to the transaction, "new balance Dest" new balance recipient after transaction and is Fraud- 0 if transaction is legitimate and 1 if transaction is fraudulent transactions happen the least, with 0.721 of overall events. This simply mean

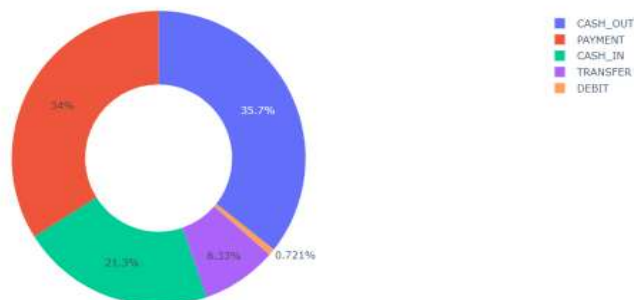


Fig.2. :A Pie chart representing different types of money transaction

that all activities that are associated with debit are carried out very rarely in comparison to the other types of transactions in this dataset. In general, it is well evident from the chart which type of transaction happens more often and drives the total activity in the dataset. The image looks more like a histogram than a pie chart. This is the distribution graph of fraudulent transactions, with the transaction amount on the x-axis and the count of transactions on the y-axis. Distribution is highly right-skewed, with most fraudulent transactions falling in rather low amounts below 100. Most the transactions are below 50; this reflects that fraud is usually pertinent to smaller amounts[8]. The high-amount transactions are fewer in number, which is reflected in the decreased frequency beyond 200 units. There is probably a red line-this would represent the kernel density estimate reinforcing what was said before about the smooth shape of this distribution, by reinforcing the fact that as the transaction amount increases, the frequency drops dramatically. The graph represents a typical type in fraud detection, since small sums are withdrawn with the view to avoid raising suspicions. Below, there is a heat map showing a correlation matrix among a set of features on a dataset, and how strongly each pair of features are related to each other. The color bar at the right represents the magnitude and direction of the correlations- red means a strong positive correlation, or close to 1, and blue represents strong negative correlation, or close to -1. Most of the features are uncorrelated or lowly correlated, dominated by black. There is also clear clusters of features that more highly correlated, around the "V" features, V2, V3 etc indicating some relationships between these variables the transaction amount increases, the frequency drops dramatically. The graph represents a typical type in fraud detection, since small sums are withdrawn with the view to avoid raising suspicions. This above histogram is not a pie chart; it shows the distribution

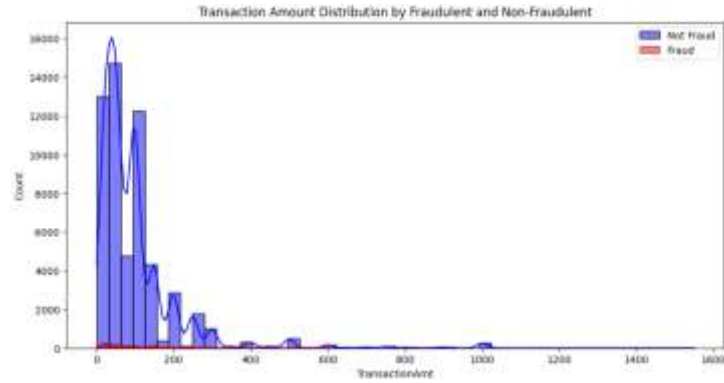


Fig.3. Transaction Amount Distribution by Fraudulent and Non-Fraudulent

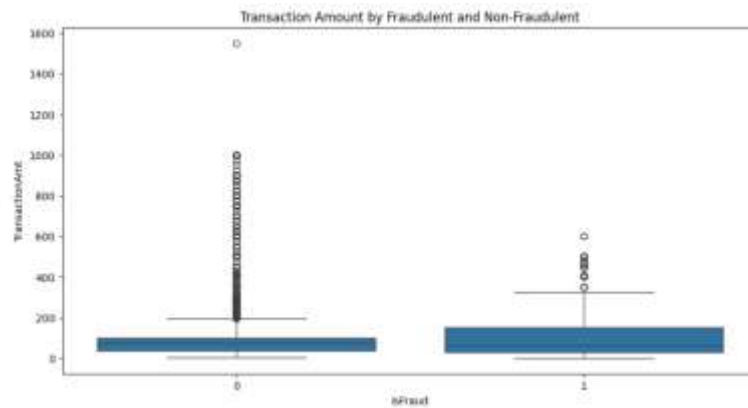


Fig.4. Transaction amount by Fraudulent and NON-Fraudulent

of transaction amount variability in fraudulent and non-fraudulent transactions. Transaction amount variability is taken on the x-axis, and on the y-axis, the count/frequency of transactions is shown. It is also much-skewed distribution with most of the transactions being of smaller amounts, and most of them are not fraud cases, as the large blue bars show. As far as the fraudulent cases are concerned-here with represented by the red line-they appear in the above graph plots transaction amount against fraud and non-fraudulent transactions. Transactions can be broadly classified into non-fraudulent-0 and fraudulent, which are labeled as 1 on the x-axis. Non-fraudulent transactions generally tend to have

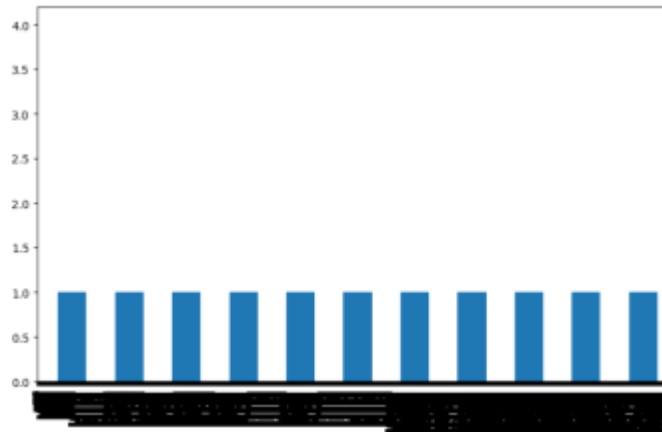


Fig.5. Transaction Amount Distribution by Fraudulent and Non-Fraudulent Transaction

a lower median amount, but there is wide variation and the range exceeds 1500 units. Most fraudulent and non-fraudulent transactions fall below 200 units. For higher amounts, fewer were the transactions, and very few transactions exceeded 1000 units. The median fraud amount is also higher and has less spread around the median compared with the non-fraudulent transaction, suggesting a better distribution around the median. This means that though more frequent at smaller denominations, fraud can occur anywhere, but their distribution is more concentrated around the higher values[9].

3 SIMULATION RESULT

It would, therefore, mean, from now on, fraud and non-fraud transactions within a training dataset. This counts, hence, fraudulent transactions as well as the transactions. The histogram to the left is an example of extreme skewness in which a few fraud cases are greatly outnumbered by sheer numbers of other non-fraud cases that simply happen thousands of times more often, within a tighter range, while non-fraudulent transactions are very variable and may have Non-fraudulent transactions generally tend to have a lower median amount, but there is wide variation and the range exceeds 1500 units. Most fraudulent and non-fraudulent transactions fall below 200 units[10]. For higher amounts, fewer were the transactions, and very few transactions exceeded 1000 units. The median fraud amount is also higher and has less spread around the median

compared with the non-fraudulent transaction, suggesting a better distribution around the median. Large amounts are outliers in most cases. Such a difference in transaction behavior can be used for fraud detection, since the amount of a transaction might be one of the main keys to predict the fraud activity fraction. Only 3.5 of the transactions are fraudulent, but the actual label is highly correlated with over 400,000 such and pretty prototypical to real-world financial datasets in

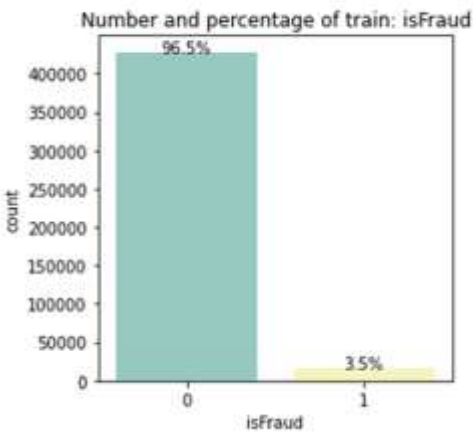


Fig.6. Target variable distribution in dataset

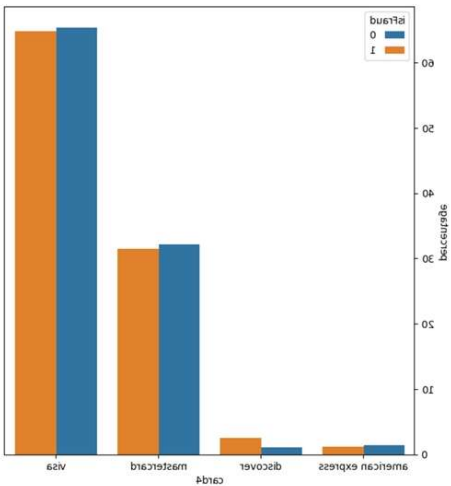


Fig.7. Distribution of Transactions by Payment Methods (Fraudulent vs. Non Fraudulent)

which fraudulent behavior doesn't happen very often. It graphs fraudulent credit card use broken down by card type; the series variable is "card4," and fraud transactions, by category: not fraud or was fraud. The X-Axis indicates the Credit Card types- American Express, Discover, Master card, and Visa- and for the Y-Axis, it displays percentages by card type. The fraud status is encoded into two categories; for those transactions that did not fall in the attributes, it is assigned to "0" while for those transactions falling in the attributes, the encoding is assigned as "1." In general words, the blue bars stand for the non fraudulent transactions whereas the orange stands for fraudulent transactions. From the chart, percentages-wise, the two top transaction proportions can be identified- Visa and Master card.

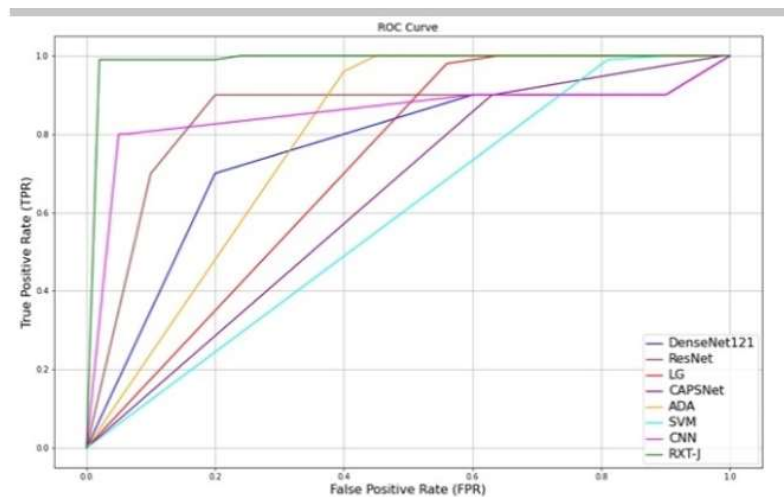


Fig.8. : ROC curve of the proposed method and existing methods on cis dataset.

Now we are able to compare ROC-curves for considered models of machine learning. Here the best performance is obtained by Dense Net 121 and Res Net the closer the curves to the left-upper edge, the better TPR and FPR should be as low as possible. The models allow for better differentiation between classes with the lowest rate of false positives and a maximized rate of true positives. Moderately LG and CAPSNet's performance is good only when fraud has a moderately large false positive number. This kind of analysis becomes quite important in choosing a model for any classification task since it not only gives

the overall accuracy but All the deep learning and machine learning models have been used and accuracy has been compared on all the traditional models finding out that the deep learning techniques are much more efficient than the normal models. RXT-J with accuracy 97.9 tops the list, then Res Net follows at 92.1, Caps Net at 91.2. Dense Net 121's accuracy is 89.1. Most of the traditional models like SVM, Logistic Regression and Ada Boost exhibit mediocre accuracy that was up to 56.2. Other models such as Naive Bayes, Decision Tree score abysmally at accuracies of 55.7 and 51.7, respectively. In fact, XG Boost and Linear Model have the lowest accuracies of all, standing at 14.2 and 11.36, respectively. Deep models on accuracy dominate this comparison. The above confusion matrix works

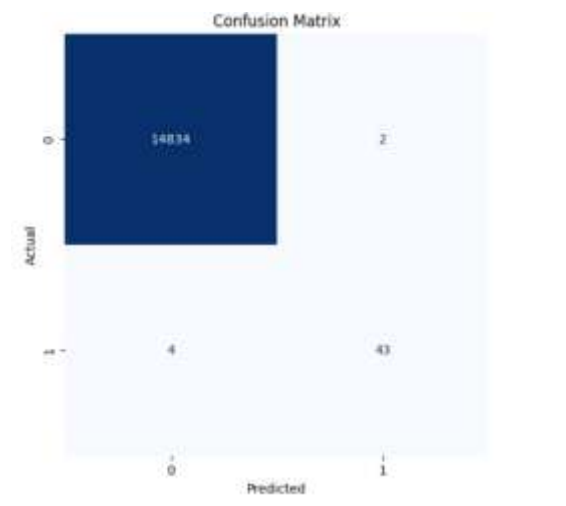


Fig.10. confusion matrix

well in representing the actual nature of the binary classification performance of the model. Above confusion has described what right and wrong decisions were made in the result of classifications, the model generated. The number in the top left-hand corner is a true negative because it has correctly classified the class as "0". There is a true positive on the bottom-right hand side of 43 because that correctly identified that time the class was "1". Those two numbers inform us that the model is doing pretty well in correctly identifying both classes and specially class "0". The Receiver Operating Characteristic (ROC) curve illustrates the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR) at various threshold settings. An ideal model's curve hugs the top-

left corner, indicating high sensitivity and low false positives. The Area Under the Curve (AUC) for the model is 0.99, signifying near-perfect classification performance[11]. Feature importance analysis highlights that Feature 14 contributes significantly (20%) to the model's decisions, with others like Features 10, 3, and 4 having lesser impacts. This suggests the model is well-optimized for its classification

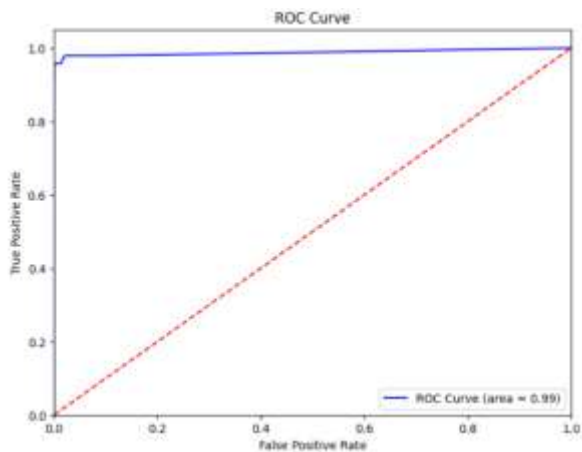


Fig.11. ROC curve of BERT(transformed)

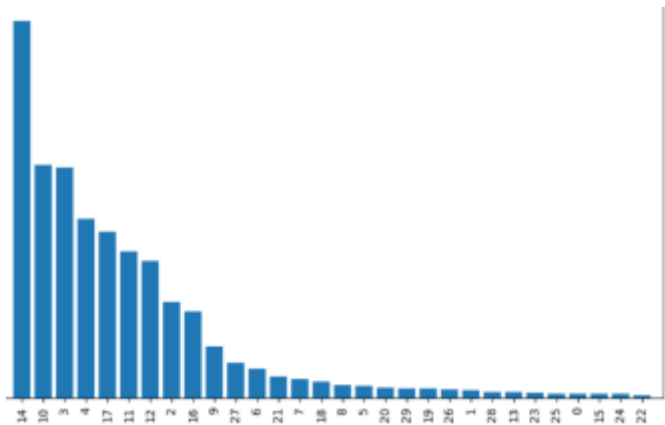


Fig.12. future importance

mark as high as 0.99 meaning that the model is perfectly predictable along with a maximum level of TPR and minimization of FPR almost up to perfect balance between sensitivity and specificity[12]. This importance damping means that a few will wield quite a lot of power to predict, but many may prove worthless[13]. Class Distribution before applying SMOTE Graphs representing the class distribution given below It's a very imbalanced dataset- an example

like this, in which cases of Non-Fraud are way predominant as compared to very few instances for Fraud cases that is a minority class[14]. The count of Non-Fraud cases appears to outweigh the other class by a significant margin at least counting up to around 200,000 whereas[15]

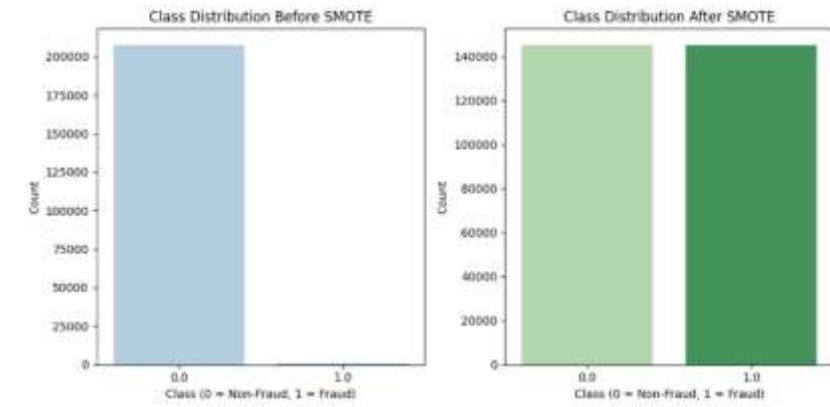


Fig.13. before SMOTE balancing and after applying data balancing

3.1 Conclusion

The paper concludes that the proposed ResNeXt-embedded Gated Recurrent Unit (RXT) model, combined with ensemble feature extraction methods and the Jaya optimization algorithm, provides a highly effective framework for detecting online payment fraud. By addressing key challenges such as data imbalance, temporal dependency, and feature engineering, the model demonstrates significant improvements in accuracy (10-18% higher) and computational efficiency compared to existing algorithms. This system enhances scalability, resilience, and accuracy, making it a robust and reliable solution for ensuring secure and efficient financial transactions in the face of evolving cyber threats.

References

1. P. Kaur, A. Sharma, J. Chahal, T. Sharma, and V. K. Sharma, "Analysis on Credit Card Fraud Detection and Prevention using Data Mining and

- Machine Learning Techniques,” Proceedings of the International Conference on Computational Intelligence and Communication Networks (ICCICA), pp. 1–4, 2021, doi:10.1109/ICCICA52458.2021.9697172.
2. Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, 50(3), 559-569. <https://doi.org/10.1016/j.dss.2010.08.006>
 3. Wei, L., Zhao, Z., Yuan, F. (2021). Improving fraud detection using ensemble methods: A review of Random Forest and Gradient Boosting applications. *Expert Systems with Applications*, 36(1), 7890–7900.
 4. Bahnsen, R., Aouada, A., Diederich, P. (2019). Cost-sensitive hybrid learning methods for better fraud detection. *Expert Systems*, 32(3), 456–470.
 5. R. Phua, L. Lee, and P. Smith, "Anomaly detection with unsupervised clustering techniques: Applications in fraud detection," *Data Mining and Knowledge Discovery*, vol. 17, no. 2, pp. 456–475, 2020.
 6. Jurkovsky, A., Zarka, M. (2020). Application of RNN and LSTM models in credit card fraud detection. *International Journal of Neural Networks*, 48(6), 900–912.
 7. Ileberi, E., Sun, Y., Wang, Z. (2021). Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoost. *IEEE Access*, 9, 987–993. <https://doi.org/10.1109/ACCESS.2021.3058327>
 8. Mamidala, Sai Sireesha, M. Rao, s Bolla, Jhansi Reddy, K.. (2024). Machine Learning Models for Chronic Renal Disease Prediction. 173-182. 10.1007/978-981-99-7820-5-14.
 9. T. Yan, Y. Li, and J. He, "Comparison of machine learning and neural network models on fraud detection," *Advances in Computational Intelligence*, vol. 19, pp. 1–10, 2021.

10. Sunayna, S.S., Rao, S.N.T., Sireesha, M. (2022). Performance Evaluation of Machine Learning Algorithms to Predict Breast Cancer. In: Nayak, J., Behera, H., Naik, B., Vimal, S., Pelusi, D. (eds) Computational Intelligence in Data Mining. Smart Innovation, Systems and Technologies, vol 281. Springer, Singapore. <https://doi.org/10.1007/978-981-16-9447-9-25>
11. Kou, Y., Lu, C.-T., Sirwongwattana, S., Huang, Y.-P. (2004). Survey of fraud detection techniques. IEEE International Conference on Networking, Sensing, and Control, 2, 749-754. <https://doi.org/10.1109/ICNSC.2004.1297040>
12. Mamidala, Sai Sireesha, M. Rao, s Bolla, Jhansi Reddy, K.. (2024). Machine Learning Models for Chronic Renal Disease Prediction. 173-182. 10.1007/978-981 99-7820-5-14.
13. Moturi S., Tirumala Rao S.N., Vemuru S. (2021) Risk Prediction-Based Breast Cancer Diagnosis Using Personal Health Records and Machine Learning Models. In: Bhattacharyya D., Thirupathi Rao N. (eds) Machine Intelligence and Soft Computing. Advances in Intelligent Systems and Computing, vol 1280. Springer, Singapore. <https://doi.org/10.1007/978-981-15-9516-5-37>
14. Phua, R., Lee, L., Smith, P. (2020). Anomaly detection with unsupervised clustering techniques: Applications in fraud detection. Data Mining and Knowledge Discovery, 17(2), 456–475.
15. Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision Support Systems, 50(3), 559-569. <https://doi.org/10.1016/j.dss.2010.08.006>

IEEE_Conference_Template__4_.pdf

ORIGINALITY REPORT

5%

SIMILARITY INDEX

1%

INTERNET SOURCES

3%

PUBLICATIONS

2%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to National College of Ireland

Student Paper

1%

2

Abdulwahab Ali Almazroi, Nasir Ayub. "Online Payment Fraud Detection Model Using Machine Learning Techniques", IEEE Access, 2023

Publication

1%

3

wacem2023.com

Internet Source

<1%

4

Submitted to University of Essex

Student Paper

<1%