International Journal of Engineering and Management Research
e-ISSN: 2250-0758 | p-ISSN: 2394-6962
https://ijemr.vandanapublications.com

Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

Prevention Service for Fraudulent and Non Fraudulent Payments using Online Payment

Adwani Vaishali Tulsi[1] and Prof. Dinesh D. Patil[2]
[1]Department of Computer Science & Engineering, Shri Sant Gadge Baba College of Engineering and Technology, Bhusawal, INDIA
[2]Guide, Department of Computer Science & Engineering, Shri Sant Gadge Baba College of Engineering and Technology, Bhusawal, INDIA

[1]Corresponding Author: vaishaliadvani98@gmail.com

Received: 13-11-2023 Revised: 28-11-2023 Accepted: 19-12-2023

ABSTRACT

In the era of rapid Internet technological advancement, the scale of online transactions is incessantly expanding. Concurrently, the issue of network transaction fraud has attained heightened significance. In contrast to credit card transactions, online transactions exhibit characteristics such as low cost, extensive coverage, and high frequency, rendering fraud detection a notably intricate challenge. This paper addresses the complexities associated with fraud detection in online transactions by proposing two distinct algorithms: one based on a Fully Connected Neural Network and the other utilizing XGBoost. These algorithms demonstrate commendable performance, with AUC values reaching 0.912 and 0.969, respectively.

Furthermore, to operationalize these advancements, an interactive online transaction fraud detection system has been meticulously designed based on the XGBoost model. This system autonomously analyzes uploaded transaction data and promptly delivers fraud detection results to users. The integration of advanced algorithms and the development of a user-friendly system underscore the commitment to addressing the nuanced challenges posed by online transaction fraud in an efficient and effective manner.

Keywords-- Fraud Detection, Fully Connected Neural Network, Xgboost

I. INTRODUCTION

1.1 Introduction

In the digital age, where online transactions have become an integral part of daily life, the surge in e-commerce and electronic payment systems has undeniably transformed the landscape of commerce. However, this unprecedented growth in online transactions has brought forth an intricate challenge - the escalating threat of online payment fraud. The convenience and efficiency of online payment methods have inadvertently created an environment ripe for malicious activities, compelling the need for robust and intelligent mechanisms to detect and prevent fraudulent transactions.

Online payment fraud is a sophisticated and dynamic problem that poses a significant risk to consumers, merchants, and financial institutions alike. The intricacies involved in discerning legitimate transactions from fraudulent ones demand advanced technological solutions. This project is dedicated to addressing this critical issue through the application of cutting-edge machine learning techniques, specifically tailored for the detection of fraudulent activities within the realm of online payments.

The distinctive characteristics of online transactions, including their low cost, wide coverage, and high frequency, make fraud detection a complex and evolving challenge. Traditional methods of fraud prevention are proving insufficient in the face of increasingly sophisticated fraud tactics. This project aims to contribute to the evolving field of cybersecurity by designing and implementing effective fraud detection algorithms based on Fully Connected Neural Networks and XGBoost.

As we delve into the intricacies of this project, we will explore the unique features of online payment transactions, the existing challenges in fraud detection, and the role of machine learning in providing adaptive and proactive solutions. By leveraging the capabilities of machine learning algorithms, we aim not only to enhance the security of online payment systems but also to contribute to the broader understanding of combating fraud in the ever-evolving landscape of digital transactions.

This project is positioned at the intersection of cybersecurity, machine learning, and financial technology, seeking to provide innovative insights and practical solutions to the pervasive issue of online payment fraud. Through a comprehensive exploration of methodologies, algorithms, and the development of a user-friendly online fraud detection system, this endeavor aspires to contribute to the ongoing efforts to secure the digital transactions that have become an integral part of modern life.

1.2 Necessity

The increasing prevalence of online transactions and the pervasive nature of e-commerce have undeniably streamlined financial processes and brought unparalleled

Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

convenience to consumers globally. However, this surge in digital transactions has also given rise to a pressing concern — the escalating threat of online payment fraud. The necessity of this project stems from the critical need to safeguard the integrity of online financial transactions and protect both consumers and businesses from the ever-evolving tactics employed by fraudsters.

1. Rising Frequency and Sophistication of Fraud:

The frequency and sophistication of online payment fraud incidents have witnessed a significant surge in recent years. Cybercriminals continuously adapt and refine their methods, necessitating advanced and adaptive countermeasures.

2. Financial Impact on Individuals and Businesses:

Online payment fraud not only imposes financial losses on individual consumers but also poses a severe threat to the financial stability of businesses. The repercussions include lost revenue, damage to brand reputation, and potential legal liabilities.

3. Inadequacy of Traditional Fraud Prevention Measures:

Traditional fraud prevention measures, while essential, are proving increasingly inadequate in the face of evolving fraud tactics. Machine learning, with its ability to analyze patterns and detect anomalies in real-time, offers a promising avenue to bolster existing fraud prevention strategies.

4. Economic and Social Implications:

The economic and social implications of online payment fraud extend beyond individual transactions. Fraudulent activities can lead to a loss of trust in online payment systems, hindering the growth of e-commerce and digital financial services.

5. Need for Proactive and Adaptive Solutions:

As fraudsters continually devise new methods, there is a pressing need for proactive and adaptive solutions. Machine learning algorithms, by learning from historical data and adapting to emerging patterns, offer a dynamic approach to fraud detection that can evolve with the changing nature of cyber threats.

6. Protection of Consumer Privacy and Data:

Online payment fraud often involves the compromise of sensitive personal and financial information. Implementing robust fraud detection mechanisms is imperative for protecting consumer privacy and preventing unauthorized access to sensitive data.

7. Compliance and Regulatory Requirements:

Regulatory bodies and industry standards increasingly mandate stringent measures for the detection and prevention of online payment fraud. Adhering to these requirements is crucial for businesses to operate ethically and legally.

1.3 Project Objectives

1. Develop and implement fraud detection algorithms based on Fully Connected Neural Network and XGBoost.
2. Evaluate algorithm performance using key metrics.
3. Integrate the optimal algorithm into a user-friendly online fraud detection system.
4. Enable real-time analysis and automate the detection process.
5. Ensure adaptability to changing fraud patterns and scalability.
6. Create a user-friendly interface and provide comprehensive documentation.

1.4 Project Objectives

The aim of the Online Payment Fraud Detection project is to enhance the security of online transactions by developing and implementing effective machine learning algorithms. These algorithms will be designed to detect and prevent fraudulent activities in real-time, ensuring the reliability and trustworthiness of digital financial transactions. The project strives to contribute to the ongoing efforts in combating online payment fraud, ultimately safeguarding the interests of both consumers and businesses in the digital ecosystem.

1.5 Organization

Organizing an effective online fraud detection system involves several key components, including people, processes, and technology. Here's an overview of how you might structure an organization for online fraud detection:

1. Leadership and Management:

- Chief Information Security Officer (CISO): The CISO oversees the overall security strategy, including fraud detection, within the organization.
- Fraud Detection Manager: Responsible for the day-to-day operations of the fraud detection team, managing resources, and coordinating efforts with other departments.

2. Fraud Detection Team:

- Fraud Analysts: Investigate and analyze suspicious activities, patterns, and transactions to identify potential fraud.
- Data Scientists and Analysts: Develop and maintain machine learning models for fraud detection, leveraging historical data and patterns.
- Security Engineers: Implement and manage the technical infrastructure for fraud detection, including firewalls, intrusion detection systems, and encryption protocols.
- Incident Response Team: Respond to and mitigate fraud incidents in real-time, coordinating with law enforcement if necessary.

3. Collaboration with Other Departments:

- IT Department: Work closely with the IT team to ensure the security of networks, databases, and other infrastructure components.
- Customer Support: Collaborate to gather information and reports from customers, as they may be the first to notice unusual activities.
- Legal and Compliance: Ensure that fraud detection practices comply with legal requirements and industry regulations.

Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

4. Technology Infrastructure:

- Fraud Detection Systems: Implement advanced fraud detection systems that leverage machine learning, artificial intelligence, and data analytics.
- Identity Verification Tools: Use tools for identity verification, such as biometric authentication, two-factor authentication, and identity validation services.
- Transaction Monitoring Systems: Employ systems that continuously monitor transactions in real-time, flagging suspicious activities for further investigation.

5. Training and Awareness:

- Employee Training: Train employees across different departments on security best practices, recognizing and reporting potential fraud.
- Customer Education: Educate customers about security measures, phishing awareness, and how to report suspicious activities.

6. Continuous Improvement:

- Regular Assessments: Conduct regular assessments of the fraud detection system's effectiveness and make improvements based on lessons learned.
- Adaptability: Stay informed about new fraud tactics and continually update detection strategies to counter evolving threats.

7. Reporting and Documentation:

- Incident Reporting: Establish a clear process for reporting and documenting fraud incidents, ensuring that all relevant information is captured for analysis and future prevention efforts.

8. Legal and Regulatory Compliance:

- Compliance Officer: Ensure that the organization complies with all relevant laws, regulations, and industry standards related to fraud detection and prevention.

By establishing a well-organized and collaborative structure, an organization can effectively detect and prevent online fraud while continuously improving its capabilities to adapt to emerging threats.

II. LITERATURE SURVEY

2.1 Review of Papers

The literature review encompasses a thorough examination of research papers and studies related to online payment fraud detection. The following summarizes key insights gleaned from the reviewed literature:

Evolution of Fraud Detection Techniques:

The papers highlight the evolution of fraud detection techniques in response to the dynamic nature of online payment fraud. Traditional methods are deemed inadequate, paving the way for advanced technologies, particularly machine learning.

Machine Learning in Fraud Detection:

A consensus emerges on the efficacy of machine learning algorithms in fraud detection. Papers underscore the adaptability of algorithms, such as neural networks and ensemble methods, to discern patterns indicative of fraudulent activities.

Feature Engineering and Selection:

Feature engineering is identified as a critical aspect in enhancing the accuracy of fraud detection models. Papers emphasize the importance of selecting relevant features to improve the performance of machine learning algorithms.

Real-time Detection Challenges:

Challenges related to real-time fraud detection are recurrent themes. The need for algorithms capable of swift and accurate detection in online transactions is evident, with several papers proposing novel approaches to address this.

Imbalanced Data Challenges:

Addressing imbalanced datasets remains a significant concern. Papers discuss various techniques, including oversampling and undersampling, to tackle the challenges posed by a disproportionate distribution of fraud and non-fraud instances.

2.2 Gap Identification

Limited Exploration of Hybrid Models:

While the literature extensively discusses machine learning approaches, there's a gap in the exploration of hybrid models that integrate rule-based systems with machine learning algorithms. Investigating the synergies between these approaches may yield more robust fraud detection systems.

Insufficient Focus on Adaptive Models:

The evolving nature of online payment fraud calls for adaptive and self-learning models. Existing studies provide limited insights into the development and implementation of models that can dynamically adapt to emerging fraud patterns.

Inadequate Addressing of Imbalanced Data:

Although imbalanced data is recognized as a challenge, the depth of solutions proposed remains insufficient. There is a gap in comprehensive exploration of techniques to handle imbalanced datasets effectively, particularly in the context of online payment fraud.

Limited Integration with User Authentication:

While a few papers touch on the integration of fraud detection with user authentication, there is a gap in the depth of exploration. Further investigation into the seamless integration of these two elements is necessary to enhance overall security in online payment systems.

Ethical and Privacy Considerations:

Papers recognize ethical considerations and data privacy concerns, but there is a need for more in-depth exploration. A gap exists in providing detailed frameworks or guidelines for ensuring ethical practices and safeguarding user privacy in the context of fraud detection.

III. PROBLEM DEFINITION

3.1 Problem Statement

The rapid growth of online transactions has led to an alarming increase in online payment fraud, posing significant threats to consumers, businesses, and

Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

financial institutions. Current fraud detection methods, often reliant on rule-based systems, struggle to keep pace with the evolving tactics employed by sophisticated cybercriminals. As a result, there is a pressing need for advanced and adaptive fraud detection systems that can effectively mitigate the risks associated with online payment fraud.

3.2 Solution

The proposed solution for the spam detection system involves the implementation of an advanced and adaptive model leveraging machine learning algorithms for real-time text analysis. This solution integrates features such as feature extraction to capture relevant text characteristics, a user feedback mechanism for continuous learning and improvement, and adaptive algorithms to swiftly recognize evolving spam patterns. The system will cover multiple communication channels, including emails, messages, comments, and social media, ensuring comprehensive spam protection. Specific algorithms for phishing detection, identity theft prevention, and malware detection will be incorporated, enhancing security measures. Crossplatform integration, scalability, and optimal performance are prioritized, and the system will adhere to privacy regulations and include educational features to empower users in recognizing and handling potential spam threats. The goal is to create a robust, user-friendly, and privacy-conscious spam detection system that effectively mitigates the adverse impacts of spam across diverse digital environments.

IV. SYSTEM DESIGN

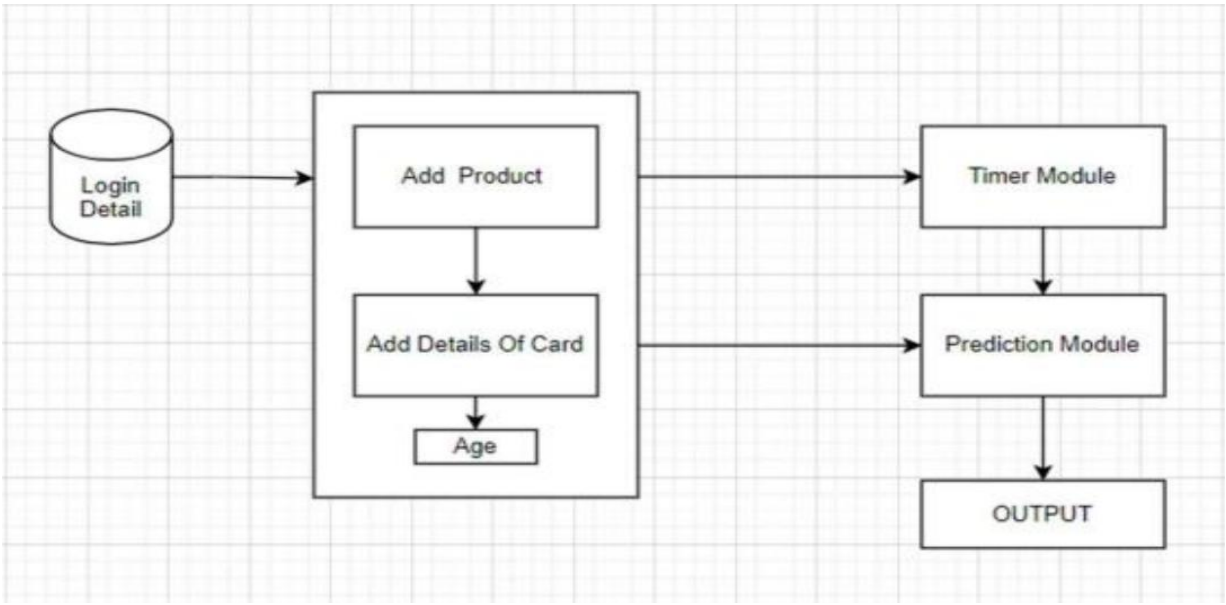
4.1 Advantages of the Proposed System:

Advantages:

1. High Accuracy
2. Real-time Detection
3. Adaptive Learning
4. Multi-Channel Protection

4.2 System Architecture

The system architecture for Online Payment Fraud Detection is designed to provide a comprehensive and secure framework for identifying and mitigating fraudulent activities in online transactions. This section outlines the key components, modules, and their interactions within the architecture.



4.3 Model of Proposed System

The proposed Online Payment Fraud Detection system incorporates a sophisticated model that integrates machine learning algorithms, real-time analysis, and adaptive learning mechanisms. This section provides an in-depth exploration of the key models shaping the system's functionality.

4.3.1 Machine Learning Algorithms:

The core of the proposed system lies in the integration of advanced machine learning algorithms tailored for fraud detection:

1. Fully Connected Neural Network (FCNN):

- A deep learning algorithm designed to identify intricate patterns and anomalies within transaction data. The FCNN excels in capturing complex relationships, making it well-suited for the dynamic nature of online payment fraud.

2. XGBoost Algorithm:

- An ensemble learning algorithm chosen for its efficiency in handling imbalanced datasets. XGBoost contributes to accurate predictions, enhancing the system's ability to differentiate between legitimate and fraudulent transactions.

4.3.2 Real-time Fraud Detection Engine:

The real-time fraud detection engine operates at the heart of the system, providing swift analysis of incoming transaction data. This module:

- Processes transactions in real-time to identify potential

Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

fraudulent activities.
- Utilizes the FCNN and XGBoost algorithms for pattern recognition and classification.
- Ensures immediate responses to mitigate fraud risks in online transactions.

4.3.3 Adaptive Learning Module:

The adaptive learning module enhances the system's resilience to evolving fraud patterns:
- Continuously learns from new data to adapt to emerging threats.
- Updates the knowledge base to stay current with the dynamic nature of online payment fraud.
- Utilizes insights gained from adaptive learning to improve the accuracy of future fraud identifications.

4.3.4 Hybrid Model Integration:

The hybrid model combines the strengths of rule-based systems with machine learning algorithms:
- Integrates predefined rules based on known fraud patterns.
- Maximizes fraud detection capabilities through a synergistic approach.
- Enhances the system's flexibility by leveraging both rule-based and algorithmic methodologies.

4.3.5 User Authentication Integration:

Seamless integration with user authentication mechanisms augments fraud detection accuracy:
- Considers user authentication data in conjunction with transaction patterns.
- Strengthens fraud detection by analyzing both transactional and user-specific attributes.
- Ensures a comprehensive approach to securing online transactions.

4.3.6 Blockchain Integration:

The optional integration of blockchain technology fortifies the security and integrity of transaction data:
- Utilizes decentralized and immutable ledger technology.
- Enhances transparency and trust in the stored transaction records.
- Provides an additional layer of security to safeguard against data tampering.

4.3.7 User Interface Components:

The user interface components facilitate user interaction and feedback:
- Implements user-friendly interfaces for inputting transaction data.
- Enables users to submit feedback on identified transactions.
- Displays results and alerts through an intuitive interface.

4.3.8 Compliance and Logging Module:

The compliance and logging module ensures regulatory adherence and comprehensive auditing:
- Enforces adherence to financial regulatory standards.
- Maintains detailed logs for auditing purposes.
- Generates comprehensive reports on system activities and identified fraud instances.

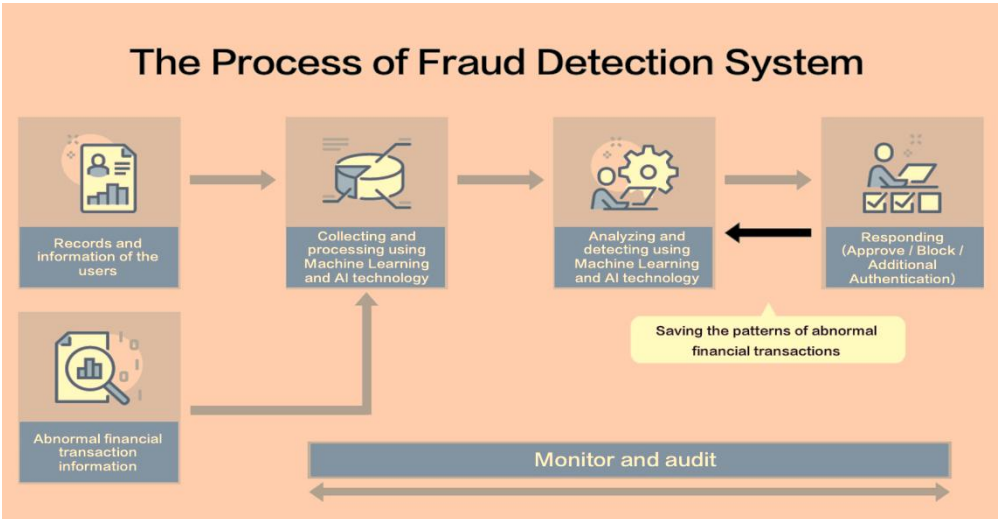
4.3.9 Continuous Monitoring and Maintenance:

Continuous monitoring and proactive maintenance processes contribute to the system's reliability:
- Monitors system performance for anomalies and irregularities.
- Implements a proactive maintenance plan for regular updates and improvements.
- Ensures the ongoing effectiveness of the fraud detection system.

4.4 System Modeling

In the dynamic realm of software development, system modeling emerges as a pivotal phase within the software development lifecycle. This process revolves around the meticulous creation of abstract representations or models, serving as architectural blueprints that play a multifaceted role in understanding, designing, and communicating complex systems.

These models transcend the abstract, providing a tangible framework that proves indispensable for stakeholders across the spectrum – from developers and designers to end-users. They offer a visual and conceptual scaffolding that facilitates a profound comprehension of a system's intricacies well before the commencement of the implementation phase.



Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

4.5 Requirement Specification

4.5.1 Introduction to Requirement Specification

Requirement Specification is a pivotal phase in the software development process, where the detailed and explicit needs of the system are documented. This chapter outlines the key requirements that the Online Payment Fraud Detection system must fulfill, providing a comprehensive overview of functional, non-functional, and technical specifications.

4.5.2 Functional Requirements

1. User Authentication:

- The system must authenticate users securely, incorporating multi-factor authentication for enhanced security.

2. Real-time Fraud Detection:

- The system should analyze incoming transaction data in real-time using machine learning algorithms for swift fraud detection.

3. Adaptive Learning:

- The system must include an adaptive learning module to continuously update its knowledge base and adapt to evolving fraud patterns.

4. Hybrid Model Integration:

- Integration of both rule-based systems and machine learning algorithms for a hybrid model, maximizing fraud detection accuracy.

5. User Interface Components:

- Implementation of user-friendly interfaces for seamless user interaction, feedback submission, and result display.

6. Blockchain Integration:

- Optional integration with blockchain technology to enhance the security and integrity of stored transaction data.

4.5.3 Non-functional Requirements

1. Performance:

- The system should be capable of handling a high volume of transactions with minimal latency, ensuring real-time fraud detection.

2. Scalability:

- It should be scalable to accommodate the growth in transaction volume and user base.

3. Security:

- Stringent security measures must be implemented to safeguard sensitive transaction data and user information.

4. Reliability:

- The system should be reliable, with minimal downtime and robust error handling mechanisms.

5. Compliance:

- Adherence to financial regulatory standards and guidelines to ensure the legitimacy and ethical standing of the system.

4.5.4 Technical Requirements

1. Programming Language:

- Implementation using a programming language suitable for machine learning algorithms, such as Python, with relevant libraries (e.g., scikit-learn, TensorFlow).

2. Database Management System (DBMS):

- Use of a reliable DBMS (e.g., MySQL, PostgreSQL) for storing and managing training data, user feedback, and system logs.

3. Web Framework (if applicable):

- If the system involves a web-based interface, choose a suitable web framework (e.g., Django, Flask for Python) for UI and backend development.

4. Development IDE:

- Selection of an Integrated Development Environment (IDE) for coding, debugging, and testing (e.g., PyCharm, Visual Studio Code).

5. Version Control System:

- Implementation of a version control system (e.g., Git) for source code management, collaboration, and tracking changes.

6. Testing Frameworks:

- Utilization of testing frameworks (e.g., PyTest) for the development of unit tests to ensure reliability and functionality.

V. IMPLEMENTATION

5.1 Algorithm

5.1.1 Data Collection:

Gather historical transaction data, including both legitimate and fraudulent transactions. Extract relevant features from the transaction data, such as transaction amount, device type, location, time of day, user behavior patterns, and more.

5.1.2 Data Preprocessing:

Clean and handle missing values in the data.

Normalize or standardize numerical features to ensure they are on a similar scale.

Encode categorical features into numerical representations.

5.1.3 Feature Engineering:

Create new features that capture more complex relationships between existing features.

Use domain knowledge to identify features that are particularly relevant for fraud detection.

5.1.4 Model Selection and Training:

Choose appropriate machine learning algorithms for fraud detection, such as logistic regression, decision trees, random forests, or support vector machines.

Split the data into training and testing sets. Train the selected machine learning models on the training data.

5.1.5 Model Evaluation:

Evaluate the performance of the trained models on the testing data using metrics like accuracy, precision, recall, and F1-score.

Select the model with the best performance for real-time fraud detection.

5.1.6 Real-time Fraud Detection:

Implement the selected machine learning model

Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

into the online payment system.

For each new transaction, extract relevant features and feed them into the model.

Evaluate the output of the model to determine whether the transaction is likely to be fraudulent.

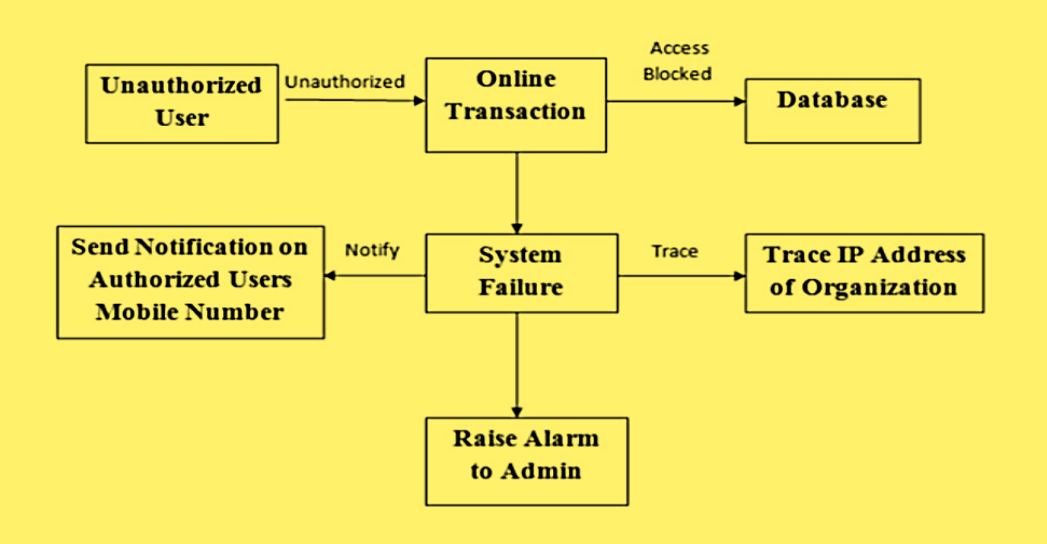
5.1.7 Fraud Analysis and Reporting:

Analyze fraudulent transactions to identify patterns and trends.

Generate reports summarizing fraud detection activities and trends.

Continuously monitor and update the fraud detection system as new fraud patterns emerge.

This algorithm provides a general framework for online payment fraud detection using machine learning. The specific implementation details will vary depending on the specific requirements and data availability of the payment system.



5.2 Project Implementation

Importing Libraries and Datasets

The libraries used are:

Pandas: This library helps to load the data frame in a 2D array format and has multiple functions to perform analysis tasks in one go.

Seaborn/Matplotlib: For data visualization.

Numpy: Numpy arrays are very fast and can perform large computations in a very short time.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

The dataset includes the features like type of payment, Old balance , amount paid, name of the destination, etc.

```
data = pd.read_csv('new_data.csv')
data.head()
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud
0	358	PAYMENT	2739.93	C1139154882	117230.00	114490.07	M2138213924	0.00	0.00	0
1	371	CASH_OUT	124119.87	C603024364	303076.00	178956.13	C1110031759	0.00	124119.87	0
2	129	CASH_IN	47715.14	C1297774858	21073457.15	21121172.29	C1790183137	838365.79	790650.65	0
3	329	CASH_IN	65739.60	C1830063487	16738153.92	16803893.52	C790107917	371889.30	306149.70	0
4	187	CASH_OUT	270253.77	C1879008092	0.00	0.00	C131873960	2167102.08	2437355.85	0

To print the information of the data we can use data.info() command.

```
data.info()
```

Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16000 entries, 0 to 15999
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   step            16000 non-null  int64
 1   type            16000 non-null  object
 2   amount          16000 non-null  float64
 3   nameOrig        16000 non-null  object
 4   oldbalanceOrg   16000 non-null  float64
 5   newbalanceOrig  16000 non-null  float64
 6   nameDest        16000 non-null  object
 7   oldbalanceDest  16000 non-null  float64
 8   newbalanceDest  16000 non-null  float64
 9   isFraud         16000 non-null  int64
dtypes: float64(5), int64(2), object(3)
```

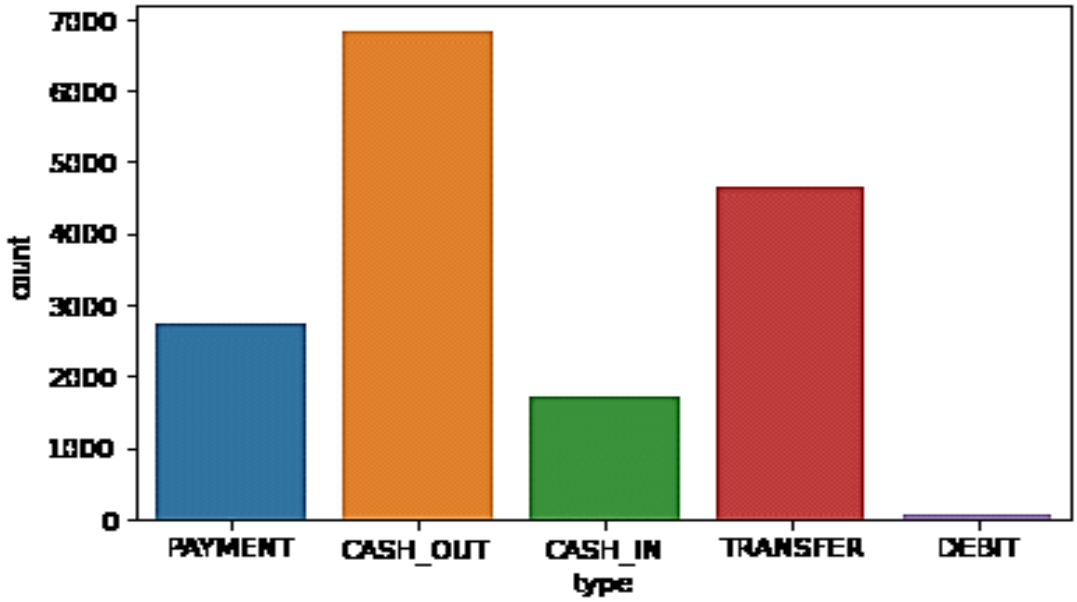
Let's see the mean, count, minimum and maximum values of the data.

```
data.describe()
```

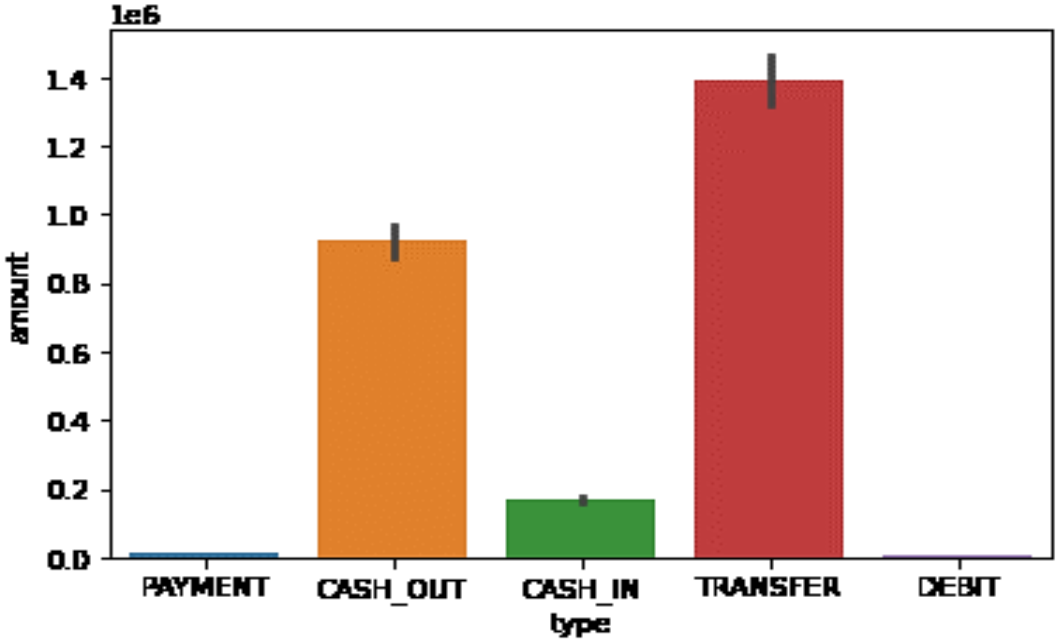
	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud
count	16000.000000	1.600000e+04	1.600000e+04	1.600000e+04	1.600000e+04	1.600000e+04	16000.000000
mean	306.068562	8.196301e+05	1.223819e+06	5.103682e+05	8.285281e+05	1.258598e+06	0.500000
std	194.036242	1.901944e+06	3.279212e+06	2.539758e+06	3.447489e+06	4.009254e+06	0.500016
min	1.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000
25%	161.000000	3.575912e+04	1.057991e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000
50%	282.000000	1.717888e+05	1.169403e+05	0.000000e+00	0.000000e+00	1.137627e+05	0.500000
75%	411.000000	5.362124e+05	7.643284e+05	0.000000e+00	4.922000e+05	1.077581e+06	1.000000
max	743.000000	5.778780e+07	5.958504e+07	4.958504e+07	2.362305e+08	2.367265e+08	1.000000

```
obj = (data.dtypes == 'object')
object_cols = list(obj[obj].index)
print("Categorical variables:", len(object_cols))

int_ = (data.dtypes == 'int')
num_cols = list(int_[int_].index)
print("Integer variables:", len(num_cols))

fl = (data.dtypes == 'float')
fl_cols = list(fl[fl].index)
print("Float variables:", len(fl_cols))

sns.countplot(x='type', data=data)
```

Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

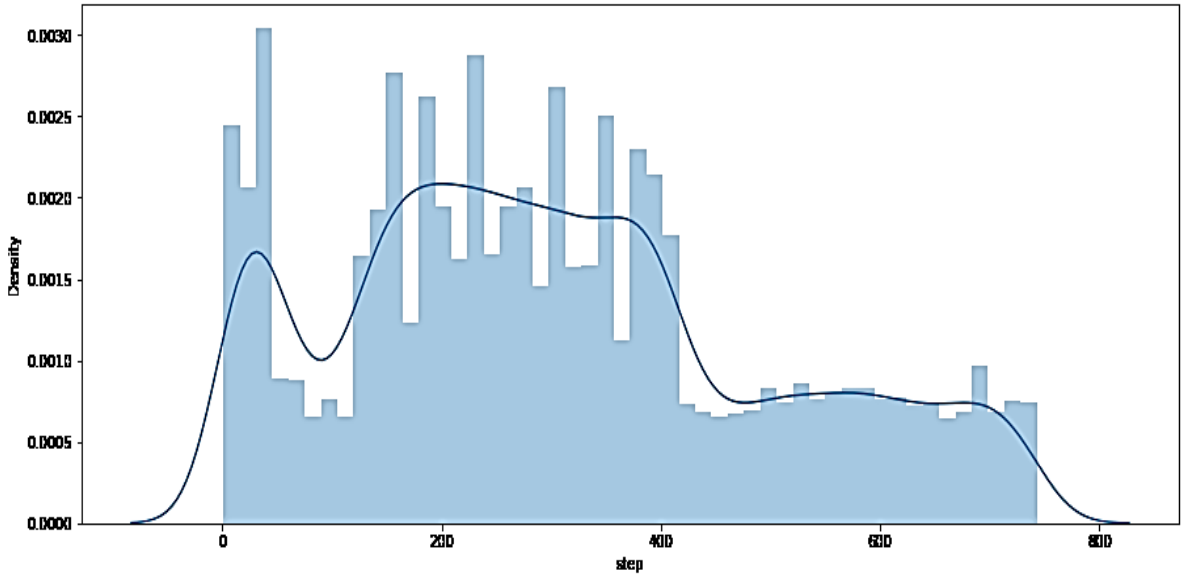


sns.barplot(x='type', y='amount', data=data)

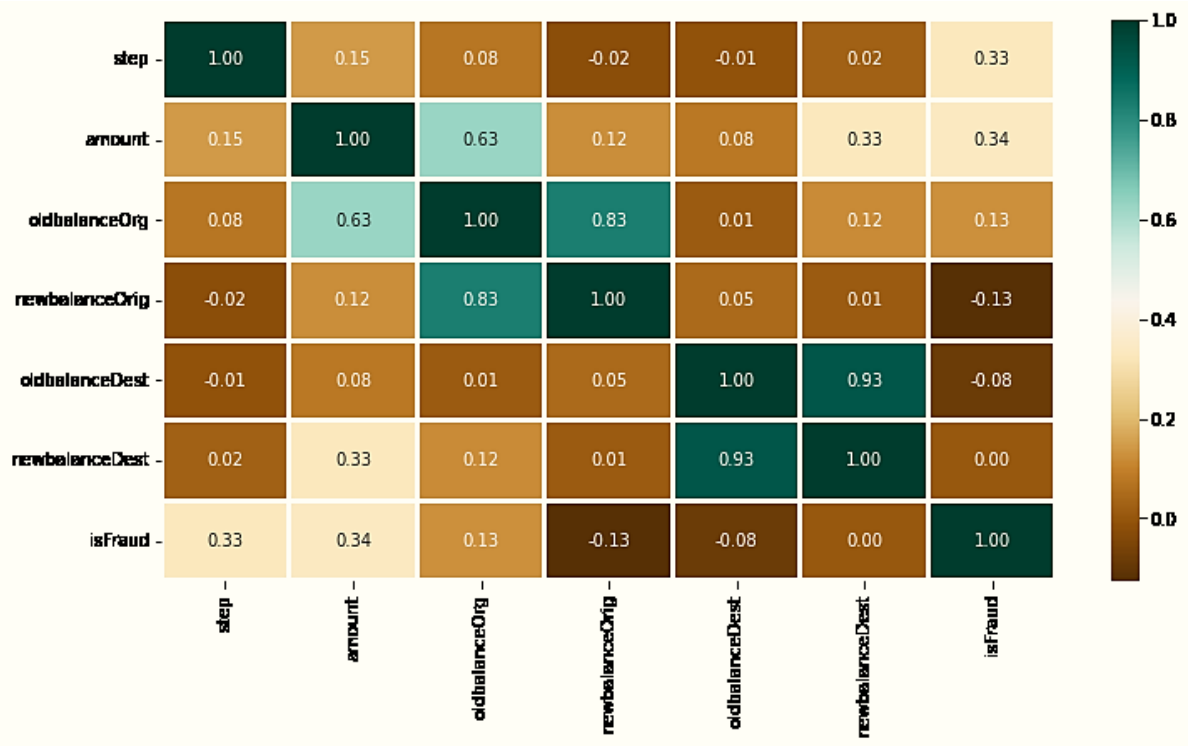


Both the graph clearly shows that mostly the type cash_out and transfer are maximum in count and as well as in amount.

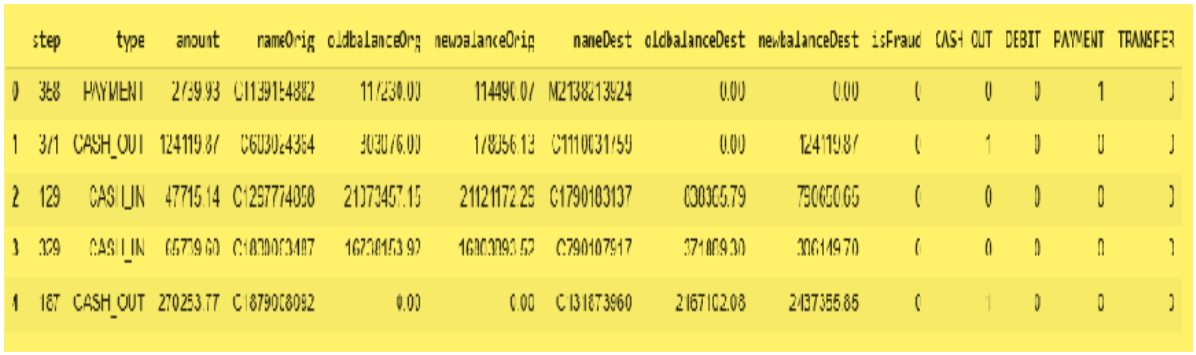
```
data['isFraud'].value_counts()
plt.figure(figsize=(15, 6))
sns.distplot(data['step'], bins=50)
```




```
plt.figure(figsize=(12, 6))
sns.heatmap(data.corr(),
                    cmap='BrBG',
                    fmt='.2f',
                    linewidths=2,
                    annot=True)
```




```
type_new = pd.get_dummies(data['type'], drop_first=True)
data_new = pd.concat([data, type_new], axis=1)
data_new.head()
```



	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	CASH_OUT	DEBIT	PAYMENT	TRANSFER
0	388	PAYMENT	27599.93	C1139154882	117230.00	144490.07	M2138213924	0.00	0.00	0	0	0	1	0
1	371	CASH_OUT	124119.87	C6030243454	303076.00	178956.13	C1110031750	0.00	124119.87	0	1	0	0	0
2	129	CASH_IN	47715.14	C1257774058	21373457.15	21421172.29	C1790103137	630305.79	750650.65	0	0	0	0	0
3	339	CASH_IN	65779.60	C1879003487	16730153.92	16803093.52	C790107917	374809.30	306149.70	0	0	0	0	0
4	167	CASH_OUT	270253.77	C1879068092	0.00	0.00	C1318173960	2167162.08	2437355.85	0	1	0	0	0

```
X = data_new.drop(['isFraud', 'type', 'nameOrig', 'nameDest'], axis=1)
y = data_new['isFraud']

X.shape, y.shape

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.3, random_state=42)
```

5.3 Model Training

Logistic Regression: It predicts that the probability of a given data belongs to the particular category or not.

XGB Classifier: It refers to Gradient Boosted decision trees. In this algorithm, decision trees are created in sequential form and weights are assigned to all the independent variables which are then fed into the decision tree which predicts results.

SVC: SVC is used to find a hyperplane in an N-dimensional space that distinctly classifies the data points. Then it gives the output according the most nearby element.

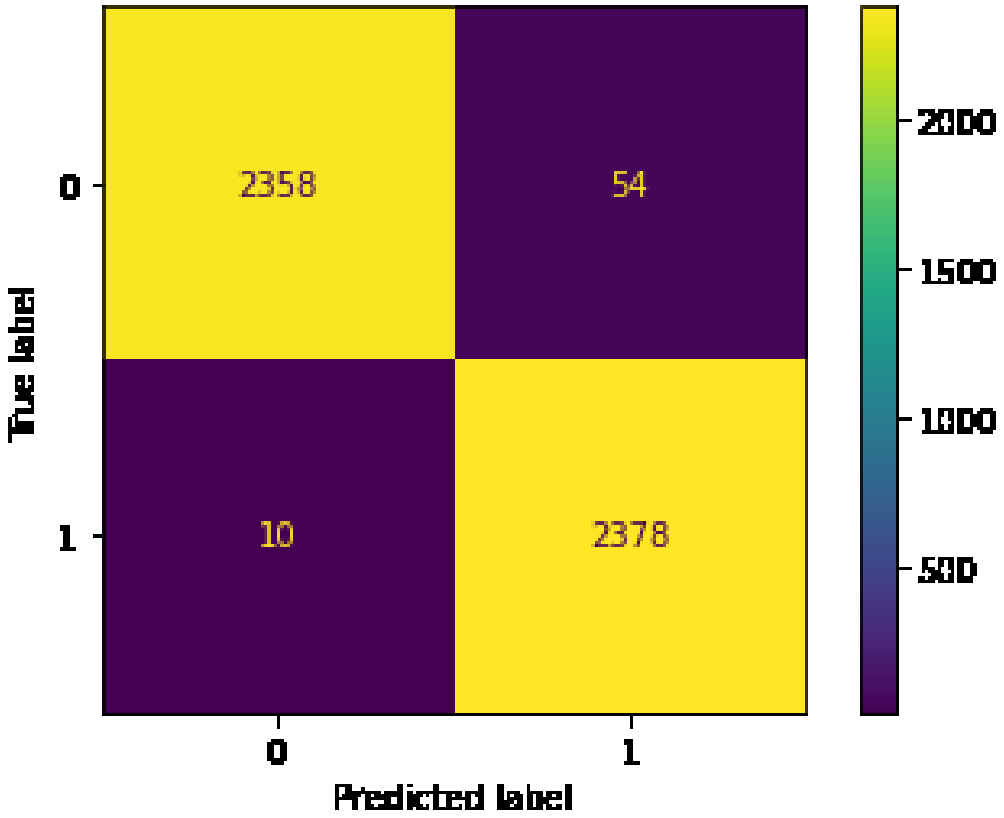
Random Forest Classifier: Random forest classifier creates a set of decision trees from a randomly selected subset of the training set. Then, it collects the votes from different decision trees to decide the final prediction.

```
    from xgboost import XGBClassifier
    from sklearn.metrics import roc_auc_score as ras
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier

models = [LogisticRegression(), XGBClassifier(),
          SVC(kernel='rbf', probability=True),

    RandomForestClassifier(n_estimators=7,

          criterion='entropy',

          random_state=7)]

for i in range(len(models)):
    models[i].fit(X_train, y_train)
    print(f'{models[i]} : ')

    train_preds = models[i].predict_proba(X_train)[:, 1]
    print('Training Accuracy : ', ras(y_train, train_preds))

    y_preds = models[i].predict_proba(X_test)[:, 1]
    print('Validation Accuracy : ', ras(y_test, y_preds))
    print()
```

```
LogisticRegression() :
Training Accuracy :  0.9610946236487818
Validation Accuracy :  0.9650647516187905

XGBClassifier() :
Training Accuracy :  0.9990647916240432
Validation Accuracy :  0.9988292242028274

SVC(probability=True) :
Training Accuracy :  0.9577130392435476
Validation Accuracy :  0.9610511096110737

RandomForestClassifier(criterion='entropy', n_estimators=7, random_state=7) :
Training Accuracy :  0.9999942442337746
Validation Accuracy :  0.9966858546463663
```

Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

```
from sklearn.metrics import plot_confusion_matrix

plot_confusion_matrix(models[1], X_test, y_test)
plt.show()
```



5.4 Project Limitations

1. Imbalanced Data:

- The dataset used for training the fraud detection model may be imbalanced, with a significantly higher number of legitimate transactions compared to fraudulent ones. This can impact the model's ability to accurately identify fraud.

2. Dynamic Fraud Patterns:

- Fraudsters constantly adapt their techniques, leading to evolving fraud patterns. A model trained on historical data may struggle to keep up with new and sophisticated fraud methods.

3. False Positives and Negatives:

- No model is perfect, and there will be instances of false positives (legitimate transactions flagged as fraud) and false negatives (fraudulent transactions not detected). Balancing these errors is a challenging task.

4. Data Quality and Variability:

- The quality of data used for training and testing the model is crucial. Incomplete or inaccurate data can lead to poor model performance. Moreover, the variability in transaction patterns may be high, making it challenging to capture all fraud scenarios accurately.

5. Adversarial Attacks:

- Fraudsters may attempt to manipulate the system by understanding its weaknesses and exploiting them. Adversarial attacks can include tactics to deceive the model and make it misclassify transactions.

6. Privacy Concerns:

- The need to collect and analyze sensitive customer data for fraud detection raises privacy concerns. Striking a balance between effective fraud detection and protecting user privacy is a challenge.

7. Regulatory Compliance:

- Adhering to regulations and compliance standards, such as GDPR, can be challenging. Balancing the need for data to improve fraud detection with regulatory requirements is a complex task.

8. Computational Resources:

- Implementing real-time fraud detection requires significant computational resources. Ensuring the system's scalability and performance under varying transaction loads can be a limitation, especially for resource-constrained environments.

9. Model Interpretability:

- Many advanced machine learning models, such as deep neural networks, are often considered "black boxes," making it difficult to interpret the reasoning behind a specific prediction. Understanding and explaining the model's decisions are crucial for building

Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

trust.

10. Integration Challenges:

- Integrating the fraud detection system with existing payment processing systems and workflows can be complex. Ensuring seamless integration and minimal disruption to existing processes is a limitation.

11. Cost and Resource Constraints:

- Developing and maintaining an effective fraud detection system can be resource-intensive and costly. Organizations may face budget constraints and need to prioritize their investments in fraud prevention.

VI. SYSTEM PERFORMANCE

6.1 Screenshot

6.1.1 Screenshot 1

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanc
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C38997010	21182.0	0.0
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0

6.1.2 Screenshot 2

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 11 columns):
 #   Column          Dtype
---  ------          -----
 0   step            int64
 1   type            object
 2   amount          float64
 3   nameOrig        object
 4   oldbalanceOrg   float64
 5   newbalanceOrig  float64
 6   nameDest        object
 7   oldbalanceDest  float64
 8   newbalanceDest  float64
 9   isFraud         int64
 10  isFlaggedFraud  int64
dtypes: float64(5), int64(3), object(3)
memory usage: 534.0+ MB
```

6.1.3 Screenshot 3

```
df.describe()
```

	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud	is
count	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6
mean	2.433972e+02	1.798619e+05	8.338831e+05	8.551137e+05	1.100702e+06	1.224996e+06	1.290820e-03	2
std	1.423320e+02	6.038582e+05	2.888243e+06	2.924049e+06	3.399180e+06	3.674129e+06	3.590480e-02	1
min	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0
25%	1.560000e+02	1.338957e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0
50%	2.390000e+02	7.487194e+04	1.420800e+04	0.000000e+00	1.327057e+05	2.146614e+05	0.000000e+00	0
75%	3.350000e+02	2.087215e+05	1.073152e+05	1.442584e+05	9.430367e+05	1.111909e+06	0.000000e+00	0
max	7.430000e+02	9.244552e+07	5.958504e+07	4.958504e+07	3.560159e+08	3.561793e+08	1.000000e+00	1

6.1.4 Screenshot 4

```
df.isnull().sum()
```

```
step              0
type              0
amount            0
nameOrig          0
oldbalanceOrg     0
newbalanceOrig    0
nameDest          0
oldbalanceDest    0
newbalanceDest    0
isFraud           0
isFlaggedFraud    0
dtype: int64
```

Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

6.1.5 Screenshot 5

```
df.nameOrig.value_counts()
```

```
C1902386530    3
C363736674     3
C545315117     3
C724452879     3
C1784010646    3
              ..
C98968405      1
C720209255     1
C1567523029    1
C644777639     1
C1280323807    1
Name: nameOrig, Length: 6353307, dtype: int64
```

6.1.6 Screenshot 6

```
df.nameDest.value_counts()
```

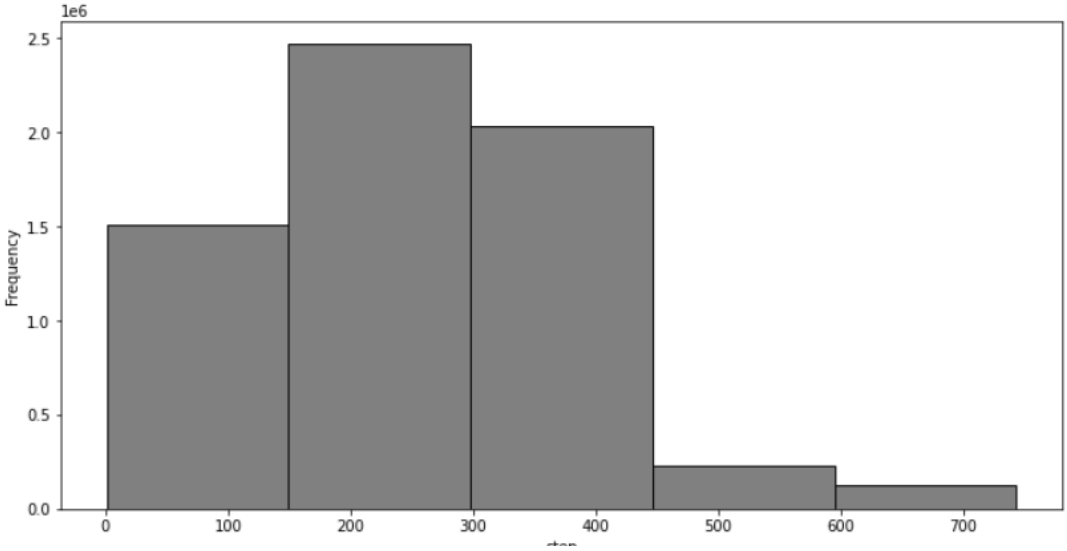
```
C1286084959    113
C985934102     109
C665576141     105
C2083562754    102
C248609774     101
              ...
M1470027725      1
M1330329251      1
M1784358659      1
M2081431099      1
C2080388513      1
Name: nameDest, Length: 2722362, dtype: int64
```

```
df.type.value_counts()
```

```
CASH_OUT    2237500
PAYMENT     2151495
CASH_IN     1399284
TRANSFER     532909
```

6.1.7 Screenshot 7

```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

feature=['step','amount','oldbalanceOrg','newbalanceOrig','oldbalanceDest','newbalanceDest']

for i in feature:
    plt.xlabel(i)
    df[i].plot(kind='hist', bins=5, figsize=(12,6), facecolor='grey',edgecolor='black')
    plt.show()
```



6.1.8 Screenshot 8

```
flagged_fraud_records = df[(df.isFraud==1) & (df.isFlaggedFraud==1)]
flagged_fraud_records
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFrau
2736446	212	TRANSFER	365423.309	C728984460	1822508.289	1970344.793	C639921569	0.0	0.0	1	1
3247297	250	TRANSFER	365423.309	C1100582606	1343002.080	1343002.080	C1147517658	0.0	0.0	1	1
3760288	279	TRANSFER	365423.309	C1035541766	536624.410	536624.410	C1100697970	0.0	0.0	1	1
5563713	387	TRANSFER	365423.309	C908544136	1822508.289	1970344.793	C891140444	0.0	0.0	1	1
5996407	425	TRANSFER	365423.309	C689608084	1822508.289	1970344.793	C1392803603	0.0	0.0	1	1
5996409	425	TRANSFER	365423.309	C452586515	1822508.289	1970344.793	C1109166882	0.0	0.0	1	1
6168499	554	TRANSFER	365423.309	C193696150	1822508.289	1970344.793	C484597480	0.0	0.0	1	1
6205439	586	TRANSFER	353874.220	C1684585475	353874.220	353874.220	C1770418982	0.0	0.0	1	1
6266413	617	TRANSFER	365423.309	C786455622	1822508.289	1970344.793	C661958277	0.0	0.0	1	1
6281482	646	TRANSFER	365423.309	C19004745	1822508.289	1970344.793	C1806199534	0.0	0.0	1	1
6281484	646	TRANSFER	365423.309	C724693370	1822508.289	1970344.793	C1909486199	0.0	0.0	1	1
6296014	671	TRANSFER	365423.309	C917414431	1822508.289	1970344.793	C1082139865	0.0	0.0	1	1
6351225	702	TRANSFER	365423.309	C1892216157	1822508.289	1970344.793	C1308068787	0.0	0.0	1	1
6362460	730	TRANSFER	365423.309	C2140038573	1822508.289	1970344.793	C1395467927	0.0	0.0	1	1
6362462	730	TRANSFER	365423.309	C1869569059	1822508.289	1970344.793	C1861208726	0.0	0.0	1	1
6362584	741	TRANSFER	365423.309	C992223106	1822508.289	1970344.793	C1366804249	0.0	0.0	1	1

Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

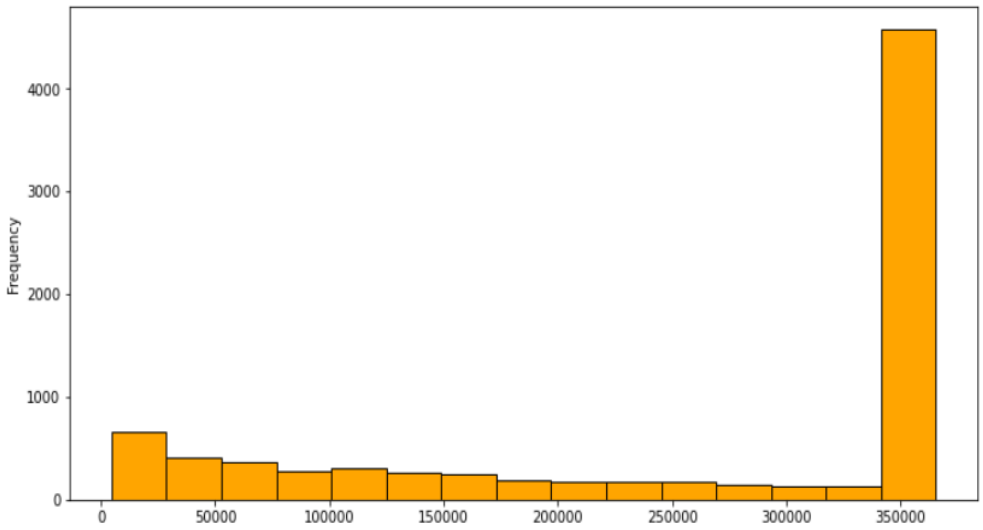
6.1.9 Screenshot 9

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFrau
5188057	367	CASH_OUT	365423.309	C1210833971	435867.160	0.000	C363013236	186826.40	622693.550	1	0
5990227	416	CASH_OUT	365423.309	C2110305720	1822508.289	0.000	C225008798	523626.59	3194869.671	1	0
5990225	416	CASH_OUT	365423.309	C246726057	1822508.289	0.000	C1786144514	2423749.18	3194869.671	1	0
5990224	416	TRANSFER	365423.309	C298387535	1822508.289	1970344.793	C662194461	0.00	0.000	1	0
5988262	415	CASH_OUT	365423.309	C2137951962	1675153.280	0.000	C309573869	12832.26	1687985.530	1	0
...	...	...	...	...	...	...	...	...	...	...	...
6002113	428	CASH_OUT	4501.300	C1838531308	3037.670	0.000	C505532836	800854.71	803892.380	1	0
6002112	428	TRANSFER	4501.300	C1408814433	3037.670	0.000	C944070846	0.00	0.000	1	0
1796322	162	TRANSFER	4501.300	C1172437299	151.000	0.000	C315826176	0.00	0.000	1	0
1796323	162	CASH_OUT	4501.300	C790340353	151.000	0.000	C517676411	386163.34	386314.340	1	0
2	1	TRANSFER	4501.300	C1305486145	181.000	0.000	C553264065	0.00	0.000	1	0

6.1.10 Screenshot

```
fraud_amount.amount.plot(kind='hist', bins=15, figsize=(12,6), facecolor='orange',edgecolor='black')
```

```
<AxesSubplot:ylabel='Frequency'>
```



6.2 Result and Comparative Analysis

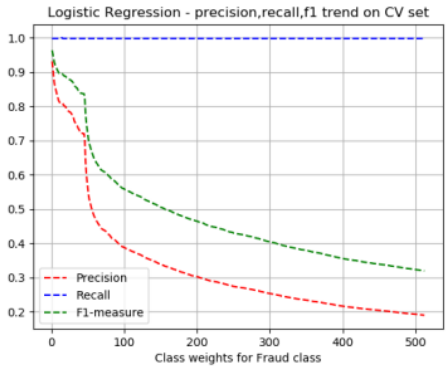
In this section, we discuss results obtained in training, validation and testing phases. We evaluated performance of our models by computing metrics like recall, precision, f1 score, area under precision-recall curve (AUPRC). A. Class weights selection In our experiments, we used increasing weights for fraud samples. We initially considered making class weights equal to imbalance ratio in our dataset. This approach seemed to give good recall but also resulted in very high number of false positives - >> 1 percent - especially for CASH OUT. Hence, we did not use this approach and instead tuned our models by trying out multiple combinations of weights on our CV split. Overall, we observed that higher class weights gave us higher recall at the cost of lower precision on our CV split. In figure 2, we show this observed behavior for CASH OUT transactions.

Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

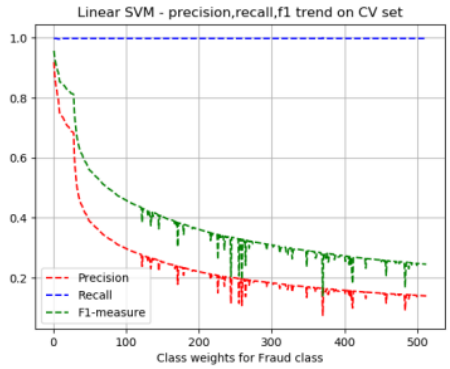
CASH OUT			
Split	Fraud	Non fraud	Total
Train	2881	1563369	1566250
CV	618	335007	335625
Test	617	335008	335625
Total	4116	2233384	2237500

TABLE III: Dataset split details

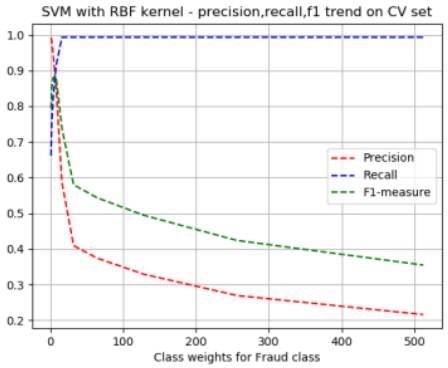
Figure 2: CASH OUT - Precision,Recall, F1 trend for increasing fraud class weights



(a) Logistic Regression



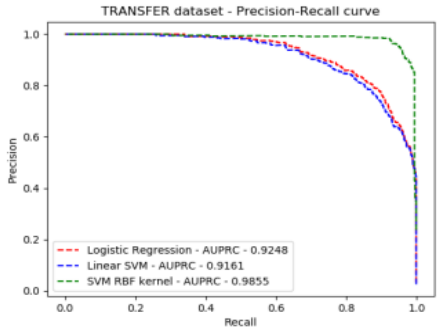
(b) Linear SVM



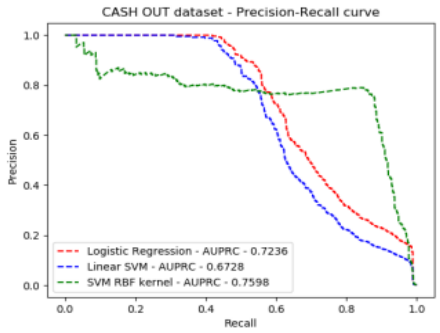
(c) SVM with RBF kernel

Figure 3

Figure 3: TRANSFER - Precision,Recall, F1 trend for increasing fraud class weights For TRANSFER dataset, the effect of increasing weights is less prominent, in particular for Logistic Regression and Linear SVM algorithms. That is, equal class weights for fraud and non-fraud samples give us high recall and precision scores. Based on these results, we still chose higher weights for fraud samples to avoid over-fitting on CV set. Figure 4 shows precision-recall curves obtained on CV set using chosen class weights for all three algorithms. Table IV summarizes these results via precision,recall,f1-measure and AUPRC scores. We chose to plot precision/recall curves (PRC) over ROC as PRCs are more sensitive to misclassifications when dealing with highly imbalanced datasets like ours. The final values of selected class weights are mentioned in table V.

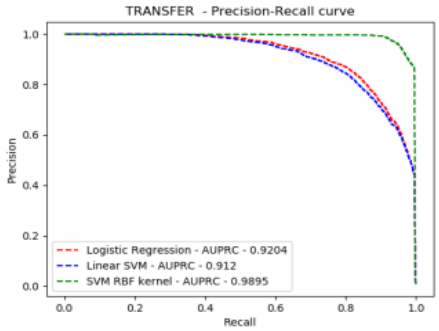


(a) TRANSFER

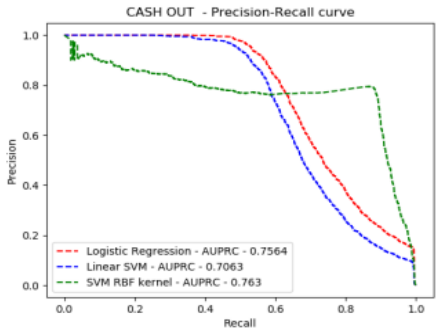


(b) CASH OUT

Fig. 4: CV set - Precision-Recall curve



(a) TRANSFER



(b) CASH OUT

Fig. 5: Train set - Precision-Recall curve

TRANSFER				
Algorithm	Recall	Precision	f1-measure	AUPRC
Logistic Regression	0.9983	0.4416	0.6123	0.9248
Linear SVM	0.9983	0.4432	0.6139	0.9161
SVM with RBF kernel	0.9934	0.5871	0.7381	0.9855
CASH OUT				
Algorithm	Recall	Precision	f1-measure	AUPRC
Logistic Regression	0.9822	0.1561	0.2692	0.7235
Linear SVM	0.9352	0.1263	0.2226	0.6727
SVM with RBF kernel	0.9773	0.1315	0.2318	0.7598

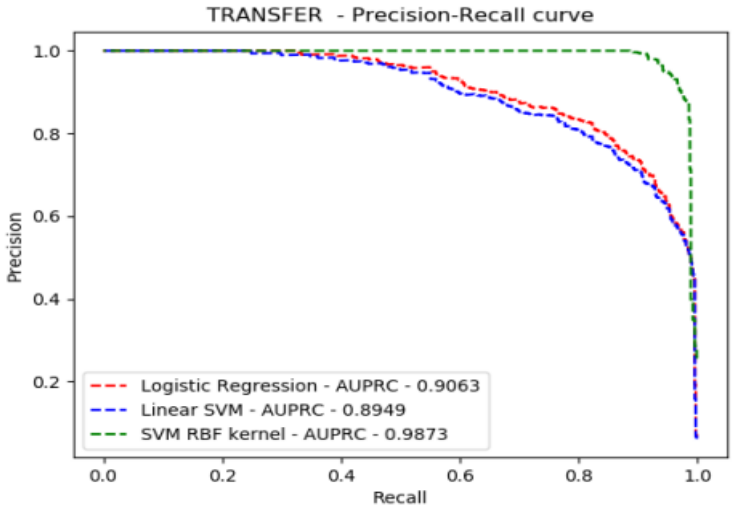
TABLE IV: Results on CV set

TRANSFER				
Algorithm	Recall	Precision	f1-measure	AUPRC
Logistic Regression	0.9958	0.4452	0.6153	0.9204
Linear SVM	0.9958	0.4431	0.6133	0.9121
SVM with RBF kernel	0.9958	0.6035	0.7515	0.9895
CASH OUT				
Algorithm	Recall	Precision	f1-measure	AUPRC
Logistic Regression	0.9847	0.1541	0.2664	0.7564
Linear SVM	0.9361	0.1225	0.2119	0.7063
SVM with RBF kernel	0.9875	0.1355	0.2383	0.7631

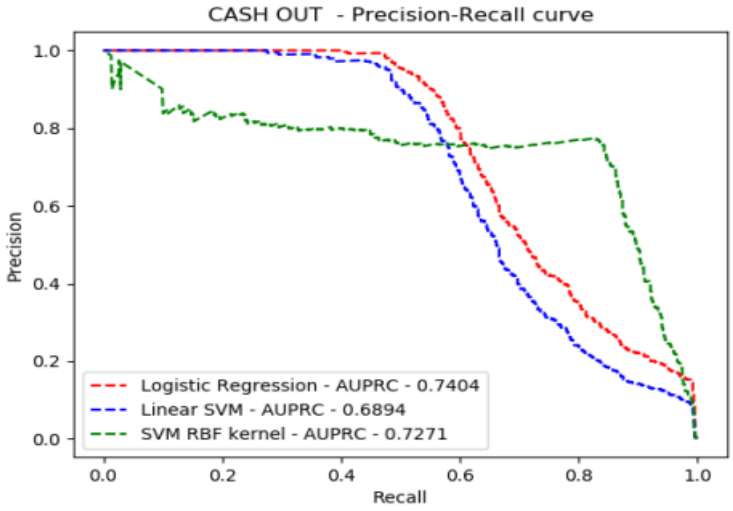
TABLE VI: Results on Train set

We get very high recall and AUPRC scores for TRANSFER transactions with ∼ 0.99 recall score for all three algorithms. In particular, SVM with RBF kernel gives us the best AUPRC value because it has much higher precision compared to the other two algorithms.

Table VIII displays corresponding confusion matrices obtained on test set of TRANSFER. We are able to detect more than 600 fraud transactions for all three algorithms with less than 1 percent false positives. TRANSFER transactions had shown a high variability across their two principal components when we performed PCA on it. This set of transactions seemed to be linearly separable - with all three of our proposed algorithms expected to perform well on it. We can see this is indeed the case. For CASH OUT transactions, we obtain less promising results compared to TRANSFER for both train and test sets Logistic regression and linear SVM have similar performance (and hence similar linear decision boundaries and PR curves). SVM with RBF gives a higher recall but with lower precision on average for this set of transactions. A possible reason for this outcome could be non-linear decision boundary computed using RBF kernel function. However, for all three algorithms, we can obtain high recall scores if we are more tolerant to false positives. In the real world, this is purely a design/business decision and depends on how many false positives is a payments company willing to tolerate.



(a) TRANSFER



TRANSFER				
Algorithm	Recall	Precision	f1-measure	AUPRC
Logistic Regression	0.9951	0.4444	0.6144	0.9063
Linear SVM	0.9951	0.4516	0.6213	0.8949
SVM with RBF kernel	0.9886	0.5823	0.7329	0.9873
CASH OUT				
Algorithm	Recall	Precision	f1-measure	AUPRC
Logistic Regression	0.9886	0.1521	0.2636	0.7403
Linear SVM	0.9411	0.1246	0.2201	0.6893
SVM with RBF kernel	0.9789	0.1321	0.2327	0.7271

TABLE VII: Results on Test set

Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

Overall, we observe that all our proposed approaches seem to detect fraud transactions with high accuracy and low false positives - especially for TRANSFER transactions. With more tolerance to false positives, we can see that it can perform well on CASH OUT transactions as well.

TABLE VIII: Confusion matrices

(a) Logistic Regression

True \ Pred	-	+
-	78557	765
+	3	612

(b) Linear SVM

True \ Pred	-	+
-	78579	743
+	3	612

(c) SVM with RBF kernel

True \ Pred	-	+
-	78886	436
+	7	608

VII. CONCLUSION

7.1 Conclusion

In fraud detection, we often deal with highly imbalanced datasets. For the chosen dataset (Paysim), we show that our proposed approaches are able to detect fraud transactions with very high accuracy and low false positives - especially for TRANSFER transactions. Fraud detection often involves a trade off between correctly detecting fraudulent samples and not misclassifying many non-fraud samples. This is often a design choice/business decision which every digital payments company needs to make. We've dealt with this problem by proposing our class weight based approach

7.2 Advantages

1. Risk Mitigation:

- Reduced Financial Losses: Fraud detection systems can identify and prevent fraudulent transactions in real-time, minimizing financial losses for both businesses and customers.
- Early Detection: Timely identification of suspicious activities allows for immediate action, preventing potential large-scale fraud.

2. Enhanced Security:

- Customer Trust: Implementing fraud detection measures enhances customer confidence in the security of online transactions, leading to increased trust in the business or financial institution.
- Data Protection: Fraud detection systems contribute to safeguarding sensitive customer information, protecting against data breaches and identity theft.

3. Operational Efficiency:

- Automation: Automated fraud detection systems can efficiently process large volumes of transactions in real-time, reducing the need for manual intervention and improving operational efficiency.
- Faster Response Time: Real-time detection allows for quick response and mitigation measures, preventing the escalation of fraudulent activities.

4. Regulatory Compliance:

- Meeting Regulatory Standards: Implementation of robust fraud detection systems ensures compliance with industry regulations and standards, avoiding potential legal and financial consequences.

5. Customer Experience:

- Seamless Transactions: Effective fraud detection systems can distinguish between legitimate and fraudulent transactions, allowing genuine transactions to proceed smoothly without unnecessary disruptions for customers.
- Reduced False Positives: Advanced algorithms and machine learning techniques help minimize false positives, ensuring that legitimate transactions are not mistakenly flagged as fraudulent.

6. Adaptability to Emerging Threats:

- Machine Learning and AI: Utilizing machine learning and artificial intelligence enables the system to adapt and evolve in response to changing fraud patterns and emerging threats.
- Continuous Improvement: Regular updates and improvements to the fraud detection algorithms keep the

Peer Reviewed & Refereed Journal
Volume-13, Issue-6 (December 2023)
https://doi.org/10.31033/ijemr.13.6.15

system effective against evolving fraud tactics.

7. Cost Savings:

- Fraud Prevention over Remediation: Preventing fraud in real-time is often more cost-effective than dealing with the aftermath of a successful fraudulent transaction, including chargebacks, legal expenses, and customer compensation.

8. Brand Reputation:

- Trust and Credibility: A commitment to robust fraud detection measures enhances the reputation of a business or financial institution, signaling a dedication to customer protection and security.
- Brand Loyalty: Customers are more likely to remain loyal to businesses that prioritize their security and well-being.

9. Data Insights:

- Analytical Capabilities: The data generated by fraud detection systems can be analyzed to gain insights into trends, patterns, and potential vulnerabilities, aiding in continuous improvement and proactive risk management.

10. Global Expansion:

- Facilitating International Transactions: A reliable fraud detection system enables businesses to expand their operations globally, facilitating secure online transactions across different regions.

7.2 Applications and Future Scope

We can further improve our techniques by using algorithms like Decision trees to leverage categorical features associated with accounts/users in Paysim dataset. Paysim dataset can also be interpreted as time series. We can leverage this property to build time series based models using algorithms like CNN. Our current approach deals with entire set of transactions as a whole to train our models. We can create user specific models - which are based on user's previous transactional behavior - and use them to further improve our decision making process. All of these, we believe, can be very effective in improving our classification quality on this dataset.

REFERENCES

[1] Samaneh Sorournejad, Zojah, Atani et.al. (2016). *A survey of credit card fraud detection techniques: Data and technique oriented perspective*.

[2] T.Singh, F.Di Troia, C.Vissagio & Mark Stamp. (2015). Support Vector machines and malware detection. *San Jose State University*.

[3] Wedge, Canter, Rubio et.al. (2017). *Solving the false positives problem in fraud prediction using automated feature engineering*.

[4] *PayPal Inc. Quarterly results*. Available at: https://www.paypal.com/stories/us/paypalreports-third-quarter-2018-results.

[5] Rajani, Padmavathamma. (2012). A model for rule based fraud detection in telecommunications. *IJERT*.

[6] A. Oza, R.Low & M.Stamp. (2014). HTTP attack detection using n−gram analysis. *Computers and Security Journal*.

[7] http://scikit-learn.org.

[8] https://www.kaggle.com/ntnu-testimon/paysim.