iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 11 **Issue:** VI **Month of publication:** June 2023

**DOI:** https://doi.org/10.22214/ijraset.2023.53990

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 11 Issue VI Jun 2023- Available at www.ijraset.com*

# Credit Card Fraud Detection Using Random Forest Classification

Soumyajit Nandi[1], Soumyanil Dey[2], Soham Bhattacherjee[3], Shakir Ahmed[4], Aratrika Ghosh[5], Moloy Dhar[6], Nirupam Saha[7], Biswajit Chaki Choudhury[8]
*Guru Nanak Institute 0f Technology, Kolkata, India*
[1]soumyajitnandi123@gmail.com, [2]soumyanildey@gmail.com, [3]sohambhattaacherjer@gmail.com, [4]shakirfriendahmed@gmail.com, [5]aratrikaghosh5@gmail.com, [6]moloy.dhar@gnit.ac.in, [7]nirupam.saha@gnit.ac.in, [8]biswajit.chakichoudhury@gnit.ac.in

***Abstract:*** ***In recent years credit card became one of the essential parts of the people. Sudden increase in E-commerce, customer started using credit card for online purchasing therefore risk of fraud also increases. Instead of carrying a huge amount in hand it is easier to keep credit cards. But nowadays that too becomes unsafe. Nowadays we are facing a big problem on credit card fraud which is increasing in a good percentage. The main purpose is the survey on the various methods applied to detect credit card frauds. From the abnormalities, in the transaction, the fraudulent one is identified. We address this issue in order to implement some machine learning algorithm like random forest, logistic regression in order to detect this kind of fraud. In this paper we increase the efficiency in finding the fraud. However, we discussed and evaluated employee criteria. Currently, the issues of credit card fraud detection have become a big problem for new researchers. We implement an intelligent algorithm which will detect all kind of fraud in a credit card transaction. We handled the problem by finding a pattern of each customer in between fraud and legal transaction. Isolation Forest Algorithm and Local Outlier Factor are used to predict the pattern of transaction for each customer and a decision is made according to them. In order to prevent data from mismatching, all attribute are marked equally.***

***Keywords:*** ***Decision Tree, Scaling,*** *Random Forest*, ***biometrics, credit card***

## I. INTRODUCTION

Nowadays as we can see that there is a huge increase online payment and the payment is mostly done with the help of credit cards. It becomes a big problem for marketing company to overcome with the credit card fraudulent activities. Fraudulent can be done in many ways such as tax return in any other account, taking loans with wrong information etc. Therefore, we need an efficient fraudulent detection model to minimize fraudulent activity and to minimize their losses. There are a huge number of new techniques which provide different algorithms which help in detecting number of credit card fraudulent activity. Basic understanding of these algorithms will help us in making a significant credit card fraudulent detection model. This paper helps us in finding doubtful credit card transaction by proposing a machine learning algorithm.

This type of fraud is happening from past, and till now not much research has done here in this particular area. The types of credit fraud in transactions are bankruptcy fraud, behavioral fraud, counterfeit fraud, application fraud [3]. There are experiments done before on credit card fraudulent activity on basis of meta-learning. There is certain limit of meta-learning. There are two features which is introduced here in our report is True Positive and False alarm. Both these features play an important role in catching fraudulent because the rate of determining fraudulent behavior is quick. For the better performance of model, we need a better classifier. Different classifier can be combined together with help of meta-learning.

### *A. Problem Statement*

Credit card fraud has emerged as a significant challenge in today's digital age. With the increasing reliance on electronic transactions and online shopping, fraudulent activities targeting credit cards have become more sophisticated and prevalent. This problem statement focuses on the detrimental effects of credit card fraud and its impact on individuals, businesses, and the overall economy. Credit card fraud refers to unauthorized and fraudulent use of someone else's credit card information to make purchases or gain access to funds. Fraudsters employ various techniques such as skimming, phishing, identity theft, and card-not-present transactions to obtain sensitive card details and exploit them for financial gain.

The consequences of credit card fraud are far-reaching. For individuals, it can result in financial loss, damaged credit scores, and psychological distress due to the invasion of privacy. Businesses also suffer substantial losses due to charge backs, reputation

damage, and increased security measures. Moreover, the economy suffers as a whole, as the costs of fraud are ultimately borne by consumers in the form of higher fees and interest rates.

Addressing the problem of credit card fraud requires a multi-faceted approach involving collaboration between financial institutions, law enforcement agencies, and individuals. Enhanced security measures, robust fraud detection systems, and consumer education are vital to minimizing the risk of credit card fraud and protecting the interests of individuals and businesses alike.

## II. ALGORITHM

In this paper, we focused on one main algorithms for recommendations: RANDOM FOREST ALGORITHM

### *A. Random Forest Algorithm*

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:

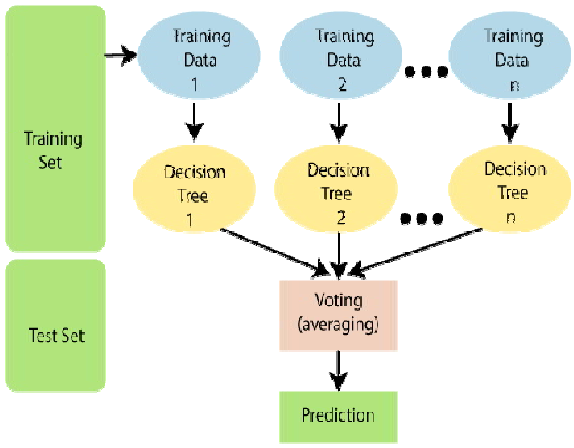



Fig. 1

### *B. Data Source*

The dataset was retrieved from an open-source website, Kaggle.com. it contains data of transactions that were made in 2013 by credit card users in Europe, in two days only. The dataset consists of 31 attributes, 284,808 rows. 28 attributes are numeric variables that due to confidentiality and privacy of the customers have been transformed using PCA transformation, the three remaining attributes are “Time” which contains the elapsed seconds between the first and other transactions of each attribute, “Amount” is the amount of each transaction, and the final attribute “Class” which contains binary variables where “1” is a case of fraudulent transaction, and “0” is not as case of fraudulent transaction.

Dataset Link: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

## III. VARIOUS LIBRARIES USED

1) *NNUMPY:* NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basicstatistical operations, random simulation and much more. It is used for numerical data processing

2) *PPANDAS:* It is a data analysis library which provides data frames. It can select data in rows and columns. It labels the data into rows & columns.
3) *Mmatplotlib:* It is a comprehensive library for creating static, animated, and interactive visualizationsin Python. It is a two-dimensional plotting library for creating graphs and plots (visualization of data). Matplotlib is basically a python package for 2D graph plotting.
4) *SScikit-Learn:* It is the most useful library for machine learning in Python. It provides a selectionof efficient tools for machine learning including classification, regression, clustering and dimensionalityreduction via a consistence interface in Python. Scikit-Learn is a higher-level library that includesimplementations of several machine learning algorithms.
5) *SSeaborn:* Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

### *A. Benefits of Random Forest*

Here are several key points highlighting the use of Random Forest Classification for this purpose:

1) *EEnsemble Learning:* Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It creates a "forest" of decision trees and aggregates their results, making it highly robust and accurate for credit card fraud detection.
2) *HHandling Imbalanced Data:* Credit card fraud datasets often suffer from imbalanced class distributions, with a majority of transactions being non-fraudulent. Random Forest handles imbalanced data well by automatically adjusting the weights during training, preventing biased predictions.
3) *RRobustness:* Random Forest is resistant to overfitting and performs well even with noisy and incomplete data. It can handle a large number of input variables and effectively deal with missing values, making it suitable for real-world credit card fraud detection scenarios.
4) *HHigh Accuracy:* Random Forest can achieve high accuracy in fraud detection due to its ability to capture complex relationships and patterns in the data. It can identify fraudulent transactions based on various features such as transaction amount, location, time, and customer behavior.
5) *SSpeed and Efficiency:* Random Forest is parallelizable, allowing it to handle large datasets and perform computations efficiently. It can process numerous transactions in real-time, making it suitable for online fraud detection systems that require quick responses.
6) *IInterpretability:* Random Forest provides insights into the decision-making process by measuring feature importance and constructing decision paths. This transparency aids in understanding the underlying factors contributing to fraud detection and can assist in fraud prevention strategies.
7) IIn summary, Random Forest Classification offers a robust and accurate approach to credit card fraud detection. Its ability to handle imbalanced data, feature importance analysis, and efficiency in handling large datasets make it a valuable tool in combating fraudulent activities and protecting individuals and businesses from financial losses.

## IV. WORK FLOW

The workflow of credit card fraud detection using Random Forest can be summarized in the following steps:

1) *DData Preparation:* Collect and preprocess the credit card transaction data, including cleaning, transforming, and normalizing the features.
2) *FFeature Selection:* Identify relevant features that contribute to fraud detection, considering factors such as transaction amount, location, time, and customer behavior.
3) *TTrain-Test Split:* Split the dataset into a training set and a testing set to evaluate the model's performance.
4) *RRandom Forest Training:* Train the Random Forest model using the training data, building an ensemble of decision trees based on the selected features.
5) *MModel Evaluation:* Evaluate the model's performance using the testing set, considering metrics such as accuracy, precision, recall, and F1-score.
6) *FFeature Importance Analysis:* Analyze the feature importance provided by the Random Forest model to identify the most influential factors in fraud detection.
7) *PPrediction:* Apply the trained model to new, unseen credit card transactions to predict whether they are fraudulent or not.

*8) FFraud Detection and Prevention:* Take appropriate actions based on the model predictions, such as blocking suspicious transactions, investigating potential fraud cases, and implementing fraud prevention strategies.

By following this workflow, credit card issuers and financial institutions can leverage the power of Random Forest to detect and mitigate credit card fraud effectively.

### *A. Importing Required Libraries and DataFrame Loading:*

Required libraries such Pandas,Numpy,Seaborn etc are imported and the dataset is imported form Kaggle for data preparing.

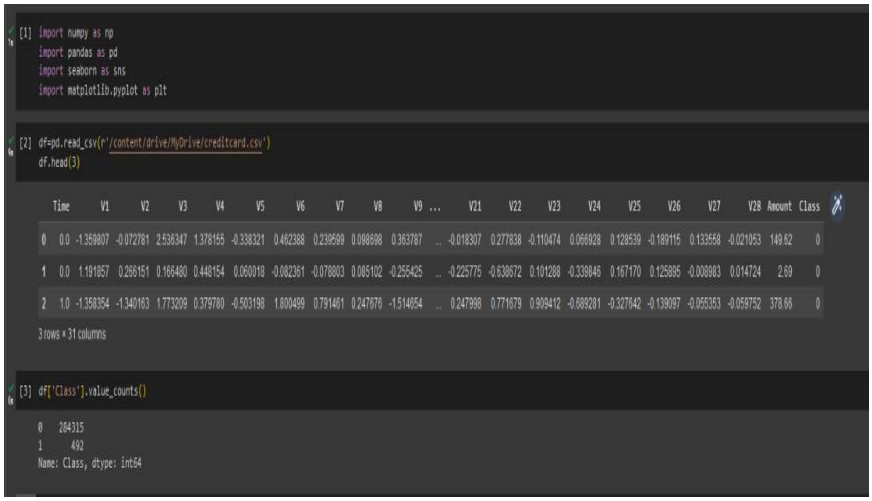



Fig. 2

### *B. Data Visualization and Correlation Mapping*

Heatmap is used to view data correlation to get a good knowledge of data dependencies and perfectly structured dataset with less dependencies and if correlation value of two or more features is greater than +0.7 or -0.7 then they are correlated highly.

In this case, the dependencies are quite less that is good enough.

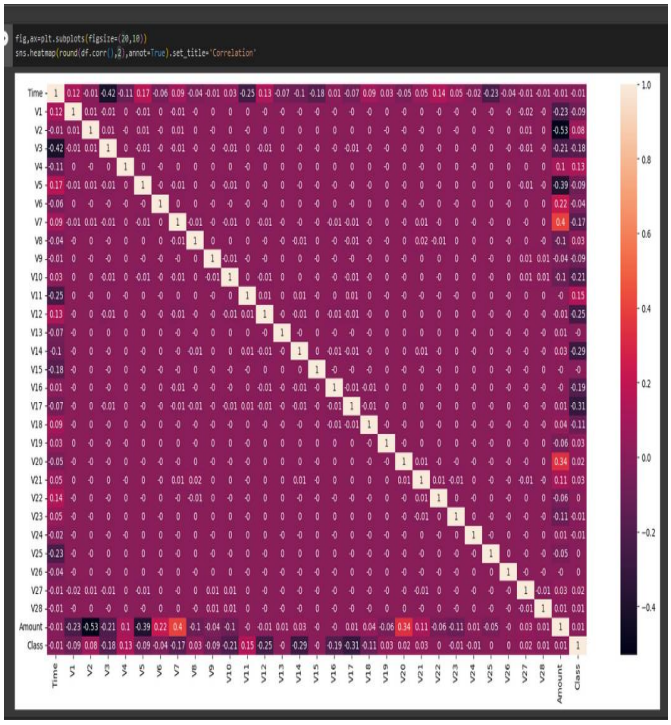



Fig. 3

### *C. Data Information*

df.info()provides with a summary of the dataset, including the number of rows, column names, data types, and the presence of missing values.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 283726 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   Time    283726 non-null  float64
 1   V1      283726 non-null  float64
 2   V2      283726 non-null  float64
 3   V3      283726 non-null  float64
 4   V4      283726 non-null  float64
 5   V5      283726 non-null  float64
 6   V6      283726 non-null  float64
 7   V7      283726 non-null  float64
 8   V8      283726 non-null  float64
 9   V9      283726 non-null  float64
 10  V10     283726 non-null  float64
 11  V11     283726 non-null  float64
 12  V12     283726 non-null  float64
 13  V13     283726 non-null  float64
 14  V14     283726 non-null  float64
 15  V15     283726 non-null  float64
 16  V16     283726 non-null  float64
 17  V17     283726 non-null  float64
 18  V18     283726 non-null  float64
 19  V19     283726 non-null  float64
 20  V20     283726 non-null  float64
 21  V21     283726 non-null  float64
 22  V22     283726 non-null  float64
 23  V23     283726 non-null  float64
 24  V24     283726 non-null  float64
 25  V25     283726 non-null  float64
 26  V26     283726 non-null  float64
 27  V27     283726 non-null  float64
 28  V28     283726 non-null  float64
 29  Amount  283726 non-null  float64
 30  Class   283726 non-null  int64  
dtypes: float64(30), int64(1)
memory usage: 69.3 MB
```

Fig. 4

### *D. Feature Scaling*

Feature scaling is a crucial preprocessing step in machine learning that aims to normalize the numeric features of a dataset to a consistent scale. It helps to ensure that features with different scales or units contribute equally to the learning process.

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
df[['Time','Amount']]=scaler.fit_transform(df[['Time','Amount']])
```

Fig. 5

```
df[['Time','Amount']].head(10)
```

| | Time | Amount |
|---|---|---|
| 0 | -1.996823 | 0.244200 |
| 1 | -1.996823 | -0.342584 |
| 2 | -1.996802 | 1.158900 |
| 3 | -1.996802 | 0.139886 |
| 4 | -1.996781 | -0.073813 |
| 5 | -1.996781 | -0.338670 |
| 6 | -1.996739 | -0.333399 |
| 7 | -1.996675 | -0.190387 |
| 8 | -1.996675 | 0.018879 |
| 9 | -1.996633 | -0.338630 |

Fig. 6

*E. Data splitting , Selecting Best Features and Model Train*

Feature selection is an important step in machine learning that involves selecting a subset of relevant features from the original set of features. It helps to improve model performance by reducing overfitting, increasing interpretability, and decreasing training time.

SelectKBest is a feature selection technique in scikit-learn that selects the top K features based on univariate statistical tests. It operates on the principle of evaluating the relationship between each feature and the target variable independently and selecting the K features with the highest scores.

Train-Test Split is a common technique used in machine learning to evaluate the performance of a model on unseen data. It involves splitting the available dataset into two separate sets: a training set and a testing set.

The training set is used to train the model, while the testing set is used to evaluate the model's performance on unseen data. By evaluating the model on data that it has not seen during training, we can get an estimate of how well the model generalizes to new, unseen examples.

Random Forest Classifier is an ensemble learning method that combines multiple decision trees to make predictions. Each tree is trained on a random subset of the data and uses a subset of features to make decisions. The final prediction is made by aggregating the predictions of all the individual trees.

In the code below:

1) n_estimators is set to 200, indicating the number of decision trees to be included in the random forest. Increasing the number of estimators can potentially improve the model's performance, but it also increases computational complexity.
2) criterion is set to 'entropy'. This criterion is used to measure the quality of a split during the tree building process. The 'entropy' criterion uses information gain to evaluate the purity of the target variable at each split. Alternative criteria include 'gini' (the default) and 'misclassification'. The 'entropy' criterion tends to favor more balanced splits and can work well when dealing with imbalanced datasets.

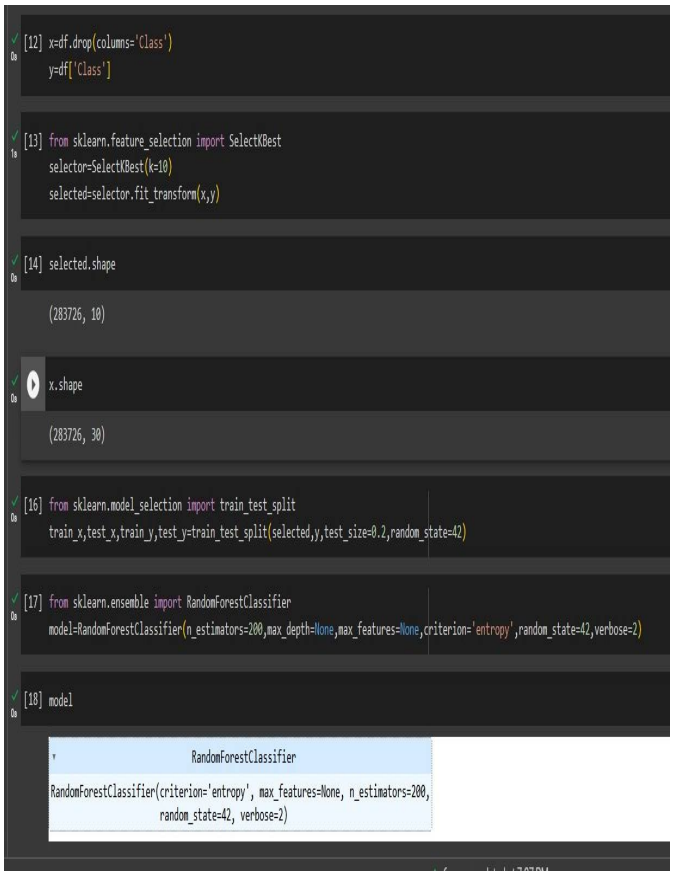



Fig. 7

```
model.fit(train_x,train_y)

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
building tree 1 of 200
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    4.1s remaining:    0.0s
building tree 2 of 200
building tree 3 of 200
building tree 4 of 200
building tree 5 of 200
building tree 6 of 200
building tree 7 of 200
building tree 8 of 200
building tree 9 of 200
building tree 10 of 200
building tree 11 of 200
building tree 12 of 200
building tree 13 of 200
building tree 14 of 200
building tree 15 of 200
building tree 16 of 200
building tree 17 of 200
building tree 18 of 200
building tree 19 of 200
building tree 20 of 200
building tree 21 of 200
building tree 22 of 200
building tree 23 of 200
building tree 24 of 200
building tree 25 of 200
building tree 26 of 200
building tree 27 of 200
building tree 28 of 200
building tree 29 of 200
building tree 30 of 200
```

Fig. 8

```
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed: 12.4min finished
RandomForestClassifier
RandomForestClassifier(criterion='entropy', max_features=None, n_estimators=200,
                       random_state=42, verbose=2)
```

Fig. 9

*F. Model Evaluation with Classification Report and Confusion Matrix*

The classification_report() function generates a comprehensive classification report that includes metrics such as precision, recall, F1-score, and support for each class. It provides valuable insights into the model's performance for each class and overall.

The confusion matrix function generates a confusion matrix that represents the counts of true positive, true negative, false positive, and false negative predictions. It helps to assess the model's performance in terms of correctly and incorrectly classified instances for each class. By examining the classification report and confusion matrix, we can gain insights into the model's accuracy, precision, recall, and F1-score for different classes. It can help us understand how well the model is performing and identify areas for improvement.

```
from sklearn.metrics import classification_report,ConfusionMatrixDisplay,confusion_matrix
print(classification_report(test_y,model.predict(test_x)))
print("\n\n")
ConfusionMatrixDisplay(confusion_matrix(test_y,model.predict(test_x)),display_labels=None).plot()
plt.show()
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    1.2s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.0s remaining:    0.0s
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56656
           1       0.98      0.72      0.83        90

    accuracy                           1.00     56746
   macro avg       0.99      0.86      0.92     56746
weighted avg       1.00      1.00      1.00     56746


[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    1.2s finished
```

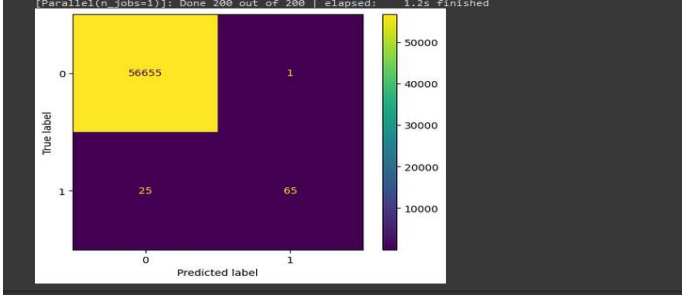



Fig. 10

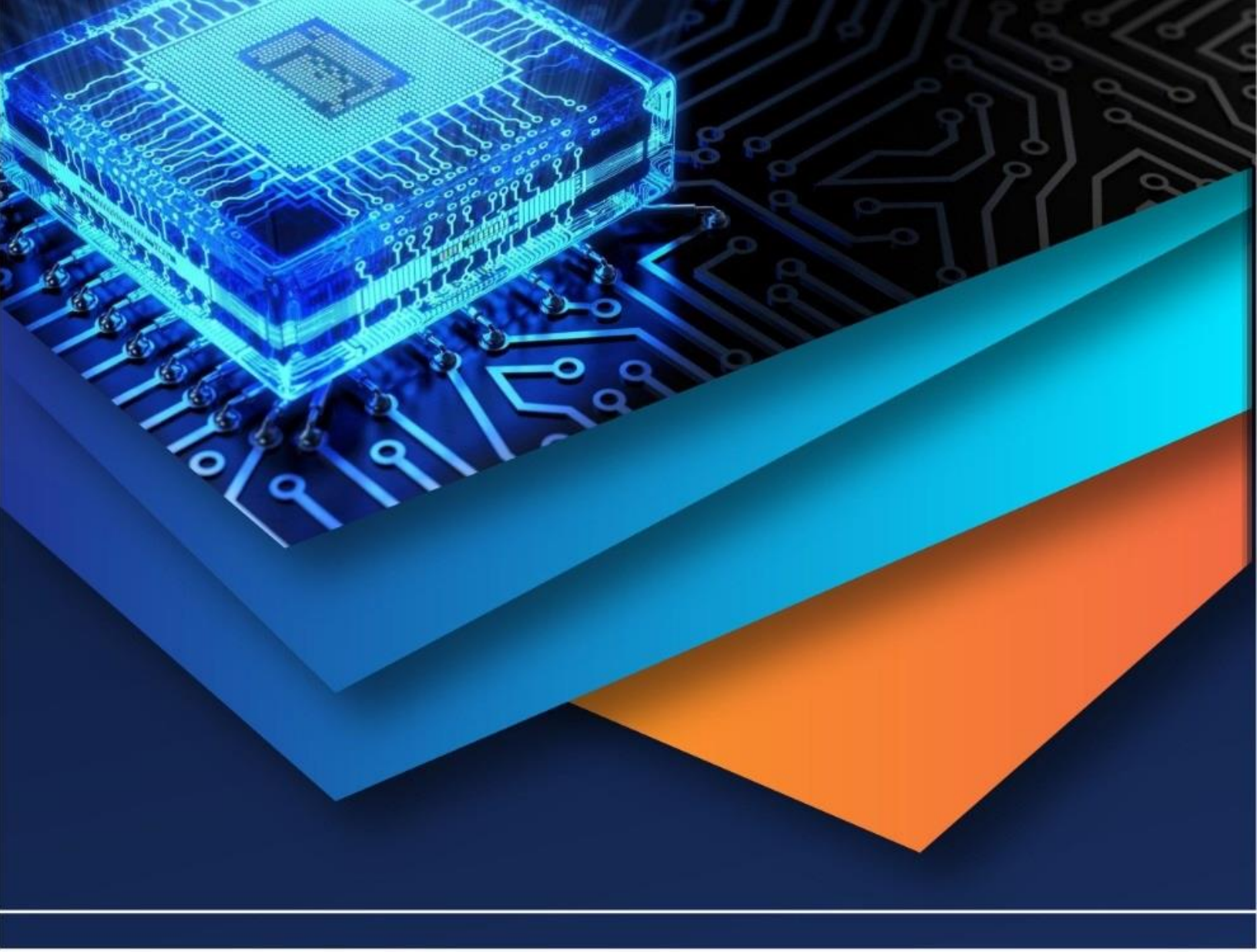
## V. CONCLUSION

In conclusion, credit card fraud detection is a critical area that requires continuous improvement and innovation to combat evolving fraud techniques. Machine learning algorithms, such as Random Forest and deep learning, have shown promise in detecting fraudulent activities by analyzing transaction data and identifying patterns and anomalies. The future of credit card fraud detection holds great potential with advancements in real-time monitoring, unsupervised learning, behavioral biometrics, and collaborative networks. By leveraging these technologies and techniques, financial institutions can enhance their fraud prevention strategies, protect customers from financial losses, and maintain the integrity of the credit card ecosystem. However, it is important to remain vigilant, adapt to new fraud trends, and collaborate across industries to stay one step ahead of fraudsters. The future scope and enhancement for credit card fraud detection include leveraging advanced techniques such as deep learning, unsupervised learning, and behavioral biometrics. Real-time monitoring, enhanced feature engineering, collaborative fraud detection networks, and explainable AI are also areas of focus. These advancements can lead to improved accuracy, faster detection, better understanding of fraud patterns, and increased collaboration among financial institutions, ultimately strengthening credit card fraud prevention and protection.

## REFERENCES

[1] "Credit Card Fraud Detection Using Enhanced Random Forest Classifier for Imbalanced Data" by AlsharifHasan Mohamad Aburbeian, Huthaifa I Ashqar (2023)
[2] "GAN-based Data Augmentation for Credit Card Fraud Detection" by EmilijaStrelcenia, SimantPrakoonwit (2022)
[3] "Credit card fraud detection using supervised learning approach" by Rashmi More, Chetan Awati, Suresh Shirgave, Rashmi Deshmukh, Sonam Patil (2021)
[4] "An efficient study of fraud detection system using Ml techniques" by S Josephine Isabella, Sujatha Srinivasan, G Suseendran (2020)
[5] "A cost-sensitive weighted random forest technique for credit card fraud detection" by Debashree Devi, Saroj K Biswas, Biswajit Purkayastha (2019)
[6] "Research on Credit Card Fraud Detection Model Based on Distance Sum" by Wen-Fang Yu, Na Wang
[7] 2009 International Joint Conference on Artificial Intelligence
[8] **"**Credit Card Fraud Detection Web Application using Streamlit and Machine Learning**" by** Vipul Jain;H Kavitha, S Mohana Kumar, 2022 IEEE International Conference on Data Science and Information System (ICDSIS)
**[9]** "Design and Implementation of Different Machine Learning Algorithms for Credit Card Fraud Detection" by Aditi Singh, Anoushka Singh, Anshul Aggarwal, Anamika Chauhan
[10] 2022 International Conference on Electrical,Computer, Communications and Mechatronics Engineering (ICECCME)



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY