Credit Card fraud Detection using Machine Learning Approach (Random Forest Algorithm).

Asim Shaik, Mohammed K Nasirulla Duvvur

April 25, 2023

1 Introduction

Machine Learning (ML) is increasingly important in credit card fraud detection due to its ability to automatically learn from data and detect patterns that are difficult for human analysts to identify. Some of its abilities includes handling large and complex data-sets, detect fraud in real-time, adapt to new fraud patterns, increase accuracy, and reduce the cost of fraud detection. Due to increasing success of ML algorithms in fraud detection of credit cards transactions, we decided to use Random Forest Algorithm for detecting the fraud-lent transactions using the behaviour of spender.

1.1 Background

Preventing credit card fraud is a critical issue that impacts financial institutions, merchants, and consumers. Credit card fraud detection is a challenging and impactful initiative that uses advanced technologies, such as machine learning and big data analytics, to detect and prevent fraudulent transactions in real time. By working on this project, we will contribute to the overall safety and trust of financial systems, which is crucial for a stable and healthy economy.

Building a credit card fraud detection model is an essential step to protect customer's financial assets and ensure the integrity of financial transactions. Credit card fraud is a constantly evolving problem, and implementing sophisticated algorithms and data analysis techniques can help detect and prevent fraudulent activities in real time. By building a credit card fraud detection model, we will contribute to maintaining the safety and trust of financial systems, providing a sense of fulfillment and accomplishment.

Scikit-learn and TensorFlow are the trending frameworks used in credit card fraud detection models. Platforms like Apache Spark, and Hadoop are data processing platforms that are capable of handling large datasets, can be used and integrated with various machine learning frameworks. Algorithms such as Support Vector Machine(SVM), Logistic Regression, Random Forest, and Deep Neural Networks are being implemented for this model these days.

1.2 Problem Definition

This project is used to determine the fraud in the credit card transactions as fraud is increasing along with the development in technology in today's world, So as the increase in the use of credit card. The main objective of this approach is to detect frauds with very high accuracy as well as in number. Here we have chosen a classification approach using Random Forest. The Random Forest algorithm has the ability to separate the data and can handle huge number of transactions with decent result. This approach gives high accuracy to detect the frauds.

1.2.1 Importance of Random Forest in Credit Card fraud Detection

Random Forest Algortithm(RFA) is a popular machine learning algorithm for credit card fraud detection due to its ability to handle high-dimensional datasets with many features and the potential for feature selection. Here are some of the reasons why Random Forest is significant for credit card fraud detection:

- **Accurate Predictions** Random Forest is an ensemble learning method that combines multiple decision trees to produce a more accurate model. This will reduce the variance of the model and improves its ability to generalize to new data. This results in more accurate predictions of whether a given transaction is fraudulent or not.

- **Feature Selection** Random Forest can automatically perform feature selection, which is the process of selecting the most relevant features for the model. This can be especially useful in credit card fraud detection, where there may be many features that are not relevant to the detection of fraud.

- **Scalablity** Random Forest is scalable and can handle large datasets with many features. This is important in credit card fraud detection, where there may be millions of transactions to analyze.

- **Speed and Robustness** Random Forest can quickly train a model and make predictions, which is important in real-time fraud detection. Random Forest is a robust algorithm that can handle noisy and missing data, which is common in credit card fraud detection.

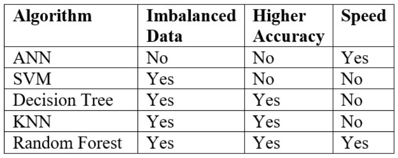
1.3 Challenges

The main and biggest challenge here is the process of detection of frauds as well as to determine which ones are non-frauds.Credit card fraud patterns can change over time, which can lead to concept drift, where the relationship between the features and the target variable changes. Ml algorithms can be computationally intensive, especially when dealing with large data-sets.

2 Literature-review

[1] In this research, they've used ANN wherein the recall rate is lower. Here, data preprocessing, normalization and under-sampling has been carried out to overcome the problems with the data set. [2] Here they have used SVM wherein the data preprocessing was good but they have not achieved best accuracy when we try to compare with other algorithms. [3] Using C4.5 decision tree algorithm. in this research, they've predicted fraud transactions with success of 92.74 percentage correctly predicted. However, their dataset is imbalanced as result of low rate of fraud transaction dataset for that reason better indicator for algorithm performance is PR curve than receiver operating characteristic (ROC) rate [4] Here they've used Naive Bayes and KNN classifier and achieved good accuracy but with taking less time for the execution as well as to read the data.

2.1 Performance-Comparison



Algorithm	Imbalanced Data	Higher Accuracy	Speed
ANN	No	No	Yes
SVM	Yes	No	No
Decision Tree	Yes	Yes	No
KNN	Yes	Yes	No
Random Forest	Yes	Yes	Yes

Figure - performance comparison

3 Milestone's(Plan)

In this project, we propose a credit card fraud detection model using RFA that accurately identifies fraudulent transactions, minimizes false positives, and provides a scalable and efficient solution for detecting credit card frauds.

3.0.1 Milestone 1

Data Loading and exploration: The very first step is to load the data-set which is in the csv format and we will try to explore the data like what features we have and try to understand it fundamentally. We also look what are the rows and columns we have in the data set.

3.0.2 Milestone 2

Data Analysis: Next, to get better understanding of data we have used python packages like pandas and NumPy to identify the hidden trends and data framing which will be used further in the analysis and model building.

3.0.3 Milestone 3

Data pre-processing, Training the Model: The next step is to split the data as train and test and build a random forest classifier model. we will train the Random Forest model using the training data. The model should be trained to predict whether a given transaction is fraudulent or valid.

3.0.4 Milestone 4

Evaluating Performance: Once the model has been trained, its performance should be evaluated using the testing data. Common metrics used to evaluate the performance of a credit card fraud detection model include accuracy, precision, recall, and F1 score. Based on this we look for any improvements required and will look into parameters to improve the accuracy.

3.0.5 Milestone 5

Finally, we will try to deploy the model in a production environment, where it can be used to detect fraudulent transactions in real-time.

4 Background

4.1 Credit Card fraud Detection

Credit card fraud detection is a common application of machine learning where our goal is to detect fraudulent transactions in credit card transactions. It is a critical problem for banks and financial institutions to prevent fraudulent activities and minimize financial losses. The model is trained on a labeled dataset that contains both legitimate and fraudulent transactions, and it learns to distinguish between the two. We're using Confusion Matrix to predict fraudulent transactions. The performance of our credit card fraud detection model will be evaluated using metrics such as accuracy, precision, recall, F1 score, and Matthews correlation coefficient.

4.2 Random Forest, Classification, Random Classifier

4.2.1 Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to create a more accurate and robust model. In a Random Forest, a collection of decision trees are constructed independently using bootstrap samples of the original data. During the construction of each tree, a random subset of the features is selected for splitting at each node. This helps to reduce the correlation between the trees and increase the diversity of the model. When making a prediction with a Random Forest model, the input data is passed through each decision tree and the output of each tree is aggregated to produce a final prediction. The aggregation can be done using simple majority voting (in classification problems) or averaging (in regression problems).

4.2.2 Classification,Random Classifier

Classification is a type of supervised learning task in machine learning where the goal is to predict the category or class to which an input data point belongs. It is a fundamental problem in machine learning and is used in a variety of applications, including image recognition, spam filtering, and sentiment analysis. In classification, a machine learning model is trained using a labeled data-set, where each data point is associated with a class label. The model learns to identify patterns and relationships between the input features and the corresponding output labels. The trained model can then be used to predict the class of new, unseen data points. Random Forest Classifier is an implementation in machine learning which is used for classification tasks.

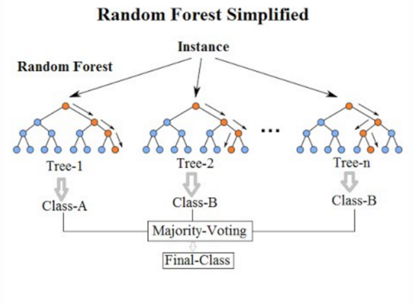
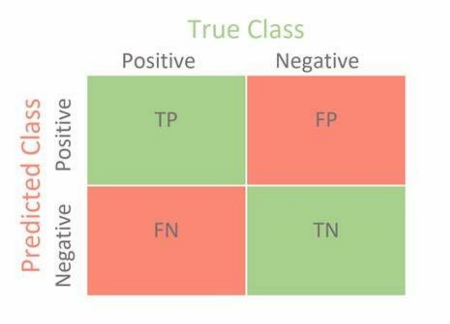


Figure explains how random forest algorithm works

4.2.3 Confusion matrix



		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure:Confusion Matrix

A confusion matrix is a table that summarizes the performance of a machine learning model on a test dataset. Using confusion matrix values, the classification metrics can be calculated. **True Positive (TP):** The number of transactions that are truly fraudulent and are correctly classified as fraudulent by the model. **False Positive (FP):** The number of transactions that are not fraudulent but are incorrectly classified as fraudulent by the model. **True Negative (TN):** The number of transactions that are not fraudulent and are correctly classified as not fraudulent by the model. **False Negative (FN):** The number of transactions that are truly fraudulent but are incorrectly classified as not fraudulent by the model.

5 Data-set

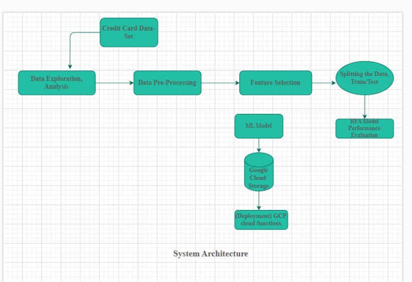
The data set consists of Time, User identities and sensitive features from V1 to V28, Amount, and Class as columns. Due to some confidentiality issues, the original features are replaced with V1, V2, … V28 columns which are the result of PCA transformation applied to the original ones. The only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise. Time is the Number of seconds elapsed between current transaction and the first transaction in the dataset. Amount is the transaction amount here and For the Class, 1 represents transactions where 0 otherwise. Data set contains 284,808 records (148MB).

6 Infrastructure

Model Building: We will use Machine Learning library in order to provide ml algorithms, data pre-processing, model selection evaluation etc.
OS: Windows 8 and above (will be using windows 11)
GPU: Random forest does not support GPU acceleration (need to use third party implementations services if required.)
Version Control: Git, Github.
Network-Requirements: Considering the data set size we have(148 mb), we may need upto 2 to 20 Mbps for uploading the data.
Deployment: Current Plan is to use GCP, if our model does not run on our local machine.

7 System-Architecture

Here is our system architecture or you can say it an outline design of how things work in the model we are building right from the beginning, it covers all of our milestones and required modules to deliver the project successfully.


Credit Card Data-Set
Data Exploration, Analysis
Data Pre-Processing
Feature Selection
Splitting the Data, Train/Test
ML Model
RFA Model Performance Evaluation
Google Cloud Storage
(Deployment) GCP cloud functions
System Architecture

System Architecture

To describe our system architecture, first we have our credit card data-set which will be having records of valid and fraud-lent transaction details withe multiple features. As the very first step we will import the data using python and will try to explore the data to identify and get a clear understanding of the data values/features present in it. We further did some analysis and data framing to make the data ready. We then prepossessed the data for the selection of features, splitting the data into train and test.
We will build the machine learning fraud detection model using the Random Forest classifier and we train it using the train dataset and further we used test data to evaluate the performance metrics like accuracy, f1 call etc.

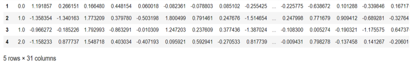
So, performance evaluation can be helpful in improving accuracy and other metrics. Then we can make some predictions based on the data provided and our model will provide the class saying which transaction is fraud, which is valid.

Basically our model(RFA classifier) will not create any trouble in running on the local server as it does not require any GPU acceleration. But anyhow, for the future use and scalability we are planning to run our model on Google's cloud platform(GCP), we will try import our model into the google cloud storage and deploying the model using google cloud functions.

8 Implementation

8.1 Data Loading and exploration:

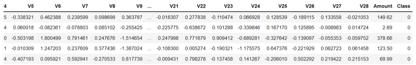
As our first step, we loaded the data-set we got from kaggle.com, we uploaded the csv format of the data and perform some basic operations like exploring the data, looking for number of records we have, what are the values data set containing and also performed basic mathematical operations like mean, count, standard deviation, min, max etc. Below are the results of the data exploration;



	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V2
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.12853
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.16717
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.32764
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.64737
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.20601

5 rows × 31 columns

Rows, Columns of the data

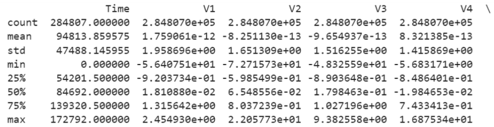


4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
5	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
4	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
0	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
1	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

The above figure describes the Rows,Columns of the data-set. Time - Time here is the number of seconds elapsed between two transactions. Amount - Transaction amount. Class - In class attribute, 0 represents the Non-Fraudulent transaction and 1 represents the Fraudulent Transaction.

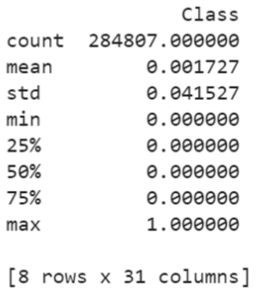
8.2 Data Describing

In data describing we calculate values such as Count, Mean, Standard Deviation, Min, 25percent, 50 percent, 75 percent Max for the attributes. From this we can do further analysis on how our data actually is, how we can use it and what can we learn from it. The below picture shows the mathematical operations of our data set; In the dataset, V1 to V28 columns contains the user information such as user name, address, ssn, transaction id etc, this information is hidden by PCA Dimensionality reduction by the dataset owners for privacy concerns.



```
                Time            V1            V2            V3            V4  \
count  284807.000000  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05
mean    94813.859575  1.759061e-12 -8.251130e-13 -9.654937e-13  8.321385e-13
std     47488.145955  1.958696e+00  1.651309e+00  1.516255e+00  1.415869e+00
min         0.000000 -5.640751e+01 -7.271573e+01 -4.832559e+01 -5.683171e+00
25%     54201.500000 -9.203734e-01 -5.985499e-01 -8.903648e-01 -8.486401e-01
50%     84692.000000  1.810880e-02  6.548556e-02  1.798463e-01 -1.984653e-02
75%    139320.500000  1.315642e+00  8.037239e-01  1.027196e+00  7.433413e-01
max    172792.000000  2.454930e+00  2.205773e+01  9.382558e+00  1.687534e+01
```

figure shows Mean,Count,STD of the data



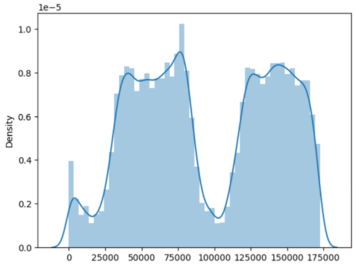
```
                Class
count   284807.000000
mean         0.001727
std          0.041527
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max          1.000000

[8 rows x 31 columns]
```

figure shows mean,count,max,min,st class(feature)

8.2.1 Data Preparation: Data Framing, Data Analysis and visualization:

In this stage of the project we focused on Data Analysis, feature selection, and visualization, more insights from the data set by identifying the distribution of each feature we have in the data. Below is the distribution of value ‘Amount’ feature:

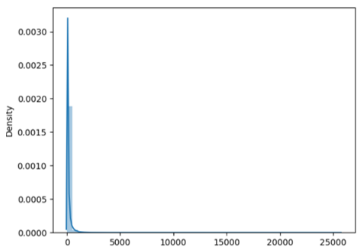


Here x-axis = Amount

y-axis = Density

above figure shows Distribution of ”Amount”

Below is the distribution of value ‘Time’ feature:



Here x-axis = Amount

y-axis = Density

However, we have done the visualization for all the features by plotting histograms where the x-axis is a particular feature (Time, V1 to V28, Amount, Class) and the y-axis is the density which we have represented in the implementation of this project.

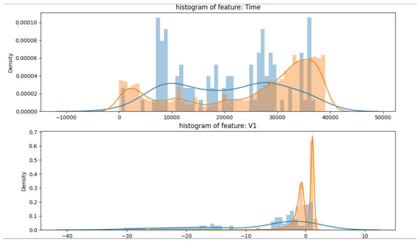
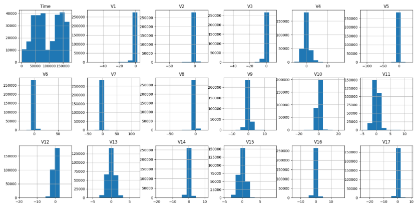


figure shows Histograms of ”time” and ”amount” features x-axis in both plots are features we are plotting

Some more histogram plots of the features present in the dataset;



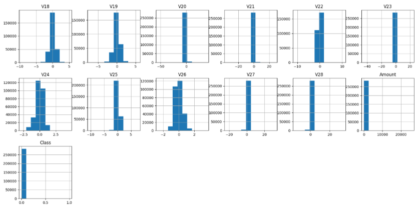


figure shows multiple histograms of individual features

By performing the analysis we got to know we have 0.17 percent fraudulent cases in the data-set. In numbers, 492 fraud transactions and 284315 valid transactions.

We also calculated the correlation matrix using a heat map which graphically helps us to know how the features correlate with each other so that it can help us predict which features are most relevant for the prediction.

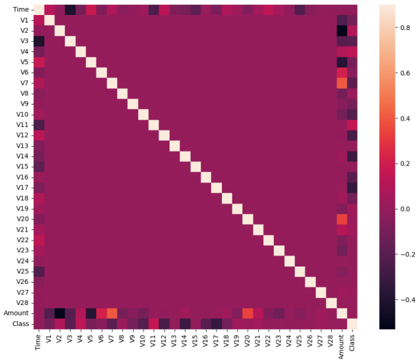


Figure: Co-relation Matrix

We observed that V2 and V5 features are more negatively correlated with Amount, also there is a correlation between V20 and Amount.

8.3 Random Forest Model:

We developed a random forest classifier using the scikit learn, we followed the below steps in order to create a fraud detection classifier with some performance metrics.

step1: Import the required modules and libraries whichever are necessary for model.

step2: Load the data into X and Y variables and separate the data.

Step 3: Split the data into training and testing sets using train,test, split function to split the dataset.

Step 4: Instantiate the Random forest classifier with the desired hyperparameters. (we have given random state=42)

Step 5: Train the model on the training data using the fit method, it will help in training the data.

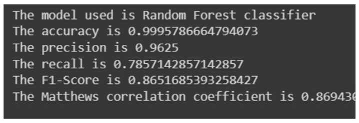
Step 6: Use the trained classifier to predict the labels for the test data using the predict method.

Step 7: Evaluate the performance of the model on the test data using metrics like accuracy, precision, recall, and F1-score etc.

Step 8: Adjust the hyperparameters and train again the model to improve its performance.

9 Results

After building the detection model we can evaluate the model based on the performance metrics like accuracy, f1 score, recall, mcc etc. Here are some results generated by our fraud detection model;



```
The model used is Random Forest classifier
The accuracy is 0.9995786664794073
The precision is 0.9625
The recall is 0.7857142857142857
The F1-Score is 0.8651685393258427
The Matthews correlation coefficient is 0.86943
```

Figure: Shows performance metrics of our model

Accuracy: Accuracy is the ratio of correctly predicted observations to the total number of observations. In other words, it is the percentage of correct predictions made by a model.

Precision: is the ratio of true positives (correctly predicted positive instances) to the total number of instances predicted as positive.

Recall: is the ratio of true positives to the total number of actual positive instances. It is a measure of the model's ability to correctly identify all positive instances.

F1 Score: It provides a balance between precision and recall and is often used as a single metric to evaluate a model's performance.

MCC: MCC ranges from -1 to +1, where a value of +1 represents a perfect prediction, 0 represents a random prediction, and -1 represents a completely wrong. In general it is a quality of measure of binary classification.

Confusion Matrix: Here is the result for the confusion matrix: In the below figure, X-axis has the Predicted class and Y-axis has the True Class. Also, the label DNormal represents a Positive Class and the label Fraud represents a Negative Class. TrueNegative(TN) = 56863 are the non-fraudulent transactions that are correctly classified as non-fraudulent, FalsePositive(FP) = 1 are the non-fraudulent transactions that were incorrectly classified as fraudulent, FalseNegative(FN) = 22 are fraudulent transactions that are incorrectly classified as non-fraudulent, TruePositive(TP) = 76 are the fraudulent transactions that are correctly classified as fraudulent. From this observation, we can say that the classifier performs well in detecting both fraud and non-fraud transactions however we've to consider other metrics as well to get a more comprehensive understanding of the model's performance.

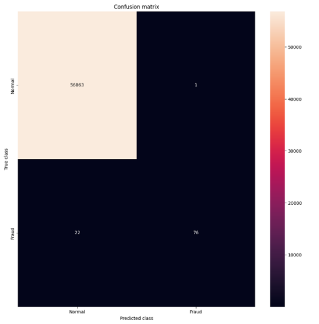
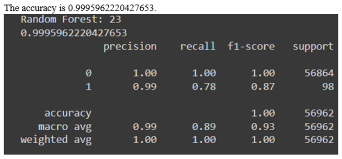


Figure shows Random Forest: 23 which is the count for the number of misclassifications made by the model.

Accuracy:

Below is the accuracy of our model and other metrics.

The accuracy is 0.9995962220427653.



```
Random Forest: 23
0.9995962220427653
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56864
           1       0.99      0.78      0.87        98

    accuracy                           1.00     56962
   macro avg       0.99      0.89      0.93     56962
weighted avg       1.00      1.00      1.00     56962
```

The accuracy of the model is

0.9995962220427653 which approaximately 99 percentage.

10 Conclusion and Future Work

Conclusion:
We've built a credit card fraud detection model using Python and Scikitlearn. We began by understanding the importance of detecting fraudulent transactions to protect customers and financial institutions. We've experimented with Random Forest Classifier. For beginning learning the model, we have set the hyperparameters and to evaluate the performance of this model, we used confusion matrix, accuracy, precision, recall, F1-score, and Matthew's correlation coefficient.Ultimately, we found that our Random Forest Classifier provided the best results in terms of detecting fraud transactions.

Future Work:
Further research, optimization, and larger as well as different datasets can help us to explore may be in a different way with the best results. It's crucial for financial institutions to continually enhance their fraud detection systems to stay ahead of the credit card financial fraud.

References

[1] RB Asha and Suresh Kumar KR. Credit card fraud detection using artificial neural network. *Global Transitions Proceedings*, 2(1):35–41, 2021.

[2] Vaishnavi Nath Dornadula and Sa Geetha. Credit card fraud detection using machine learning algorithms. *Procedia computer science*, 165:631–641, 2019.

[3] Admel Husejinovic. Credit card fraud detection using naive bayesian and c4. 5 decision tree classifiers. *Husejinovic, A.(2020). Credit card fraud detection using naive Bayesian and C*, 4:1–5, 2020.

[4] Sai Kiran, Jyoti Guru, Rishabh Kumar, Naveen Kumar, Deepak Katariya, and Maheshwar Sharma. Credit card fraud detection using naïve bayes model based and knn classifier. *International Journal of Advance Research, Ideas and Innovations in Technoloy*, 4(3):44, 2018.

[1][2][3][4].