CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING

MINI PROJECT REPORT

Submitted by

Varusha S	**210701304**
Udhayakumar G	**210701293**

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

ANNA UNIVERSITY
PROGRESS THROUGH KNOWLEDGE

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

ANNA UNIVERSITY:: CHENNAI 600 025

APRIL 2024

RAJALAKSHMI ENGINEERING COLLEGE,

CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled "**Credit card fraud detection using machine learning**" is the bonafide work of **"Varusha S (210701304), Kanaga Udhayakumar G(210701293)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Karthick V

Assistant Professor,

Department of Computer Science and Engineering,

Rajalakshmi Engineering College,

Chennai – 602015

Submitted to Mini Project Viva-Voce Examination held on ___________________

Internal Examiner **External Examiner**

ABSTRACT

In current scenario when the term "fraud" comes into our mind, credit card fraud clicks to mind so far. With the great increase in credit card transactions, credit card fraud has increasing excessively in recent years. Our goal is to develop a predictive algorithm that can differentiate between authentic and fraudulent activity by utilizing past transaction data. The project starts with thorough data preprocessing, which includes resolving missing values, normalizing features, and using methods like oversampling and undersampling to handle class imbalance. Modern techniques based on Data mining, Machine learning, Sequence Alignment, Fuzzy Logic, Genetic Programming, Artificial Intelligence etc., has been introduced for detecting credit card fraudulent transactions. This paper shows how data mining techniques can be combined successfully to obtain a high fraud coverage combined with a low or high false alarm rate.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S.MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Rahul Chiranjeevi V** Professor, Department of Computer Science and Engineering. Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

Varusha S-210701304
Udhayakumar G-210701293

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

EDA	Exploratory Data Analysis
SVM	Support Vector Machines
KNN	K Nearest Neighbours
FPR	False Positive Rate

CHAPTER 1
INTRODUCTION

1.1 GENERAL

This project aims to develop a machine learning model to identify fraudulent transactions from a dataset of credit card usage. By analyzing patterns and anomalies in transaction data, the model seeks to accurately distinguish between legitimate and fraudulent activities. This project involves data preprocessing, feature selection, model training, and evaluation using metrics like accuracy, precision, recall, and the False Positive Rate (FPR). The ultimate goal is to enhance the security of financial transactions by minimizing false positives and effectively detecting fraudulent behavior, thereby protecting both consumers and financial institutions from potential losses.

1.2 OBJECTIVE

The objective of this project is to develop an efficient and accurate machine learning model that can identify and prevent fraudulent credit card transactions in real-time. By analyzing historical transaction data and identifying patterns indicative of fraud, the model aims to minimize financial losses for credit card companies and protect customers from unauthorized transactions. The project seeks to achieve a high detection rate while maintaining a low false positive rate, ensuring that legitimate transactions are not mistakenly flagged, thereby enhancing the overall security and trustworthiness of the credit card system.

1.3 EXISTING SYSTEM

Machine learning-based credit card fraud detection is a crucial and an active field of study. Using various machine learning approaches and techniques, a number of systems and approaches have been created to address this problem. Historically, rule-based fraud detection systems have been the mainstay of many financial organizations. These systems make use of preset criteria and regulations, like transactions coming from strange places, several transactions in a brief amount of time and transactions that go over a specific threshold. Logistic regression is a straightforward also a powerful linear model for binary classification .Models that perform well with non-linear linkages and interactions are decision trees and random forests. SVMs, or support vector machines, work well in high-dimensional spaces.Deep learning models that can recognise intricate patterns are called neural networks.

1.4 PROPOSED SYSTEM

The proposed techniques are used in this paper, for detecting the frauds in credit card system. The comparison are made for different machine learning algorithms such as Logistic Regression, Decision Trees, Random Forest, to determine which algorithm gives suits best and can be adapted by credit card merchants for identifying fraud transactions.

CHAPTER 2

LITERATURE SURVEY

Fraud act as the unlawful or criminal deception intended to result in financial or personal benefit. It is a deliberate act that is against the law, rule or policy with an aim to attain unauthorized financial benefit. Numerous literatures pertaining to anomaly or fraud detection in this domain have been published already and are available for public usage. A comprehensive survey conducted by Clifton Phua and his associates have revealed that techniques employed in this domain include data mining applications, automated fraud detection, adversarial detection. In another paper, Suman, Research Scholar, GJUS&T at Hisar HCE presented techniques like Supervised and Unsupervised Learning for credit card fraud detection. Even though these methods and algorithms fetched an unexpected success in some areas, they failed to provide a permanent and consistent solution to fraud detection. A similar research domain was presented by Wen-Fang YU and Na Wang where they used Outlier mining, Outlier detection mining and Distance sum algorithms to accurately predict fraudulent transaction in an emulation experiment of credit card transaction data set of one certain commercial bank. Outlier mining is a field of data mining which is basically used in monetary and internet fields. It deals with detecting objects that are detached from the main system i.e. the transactions that aren't genuine. They have taken attributes of customer's behaviour and based on the value of those attributes they've calculated that distance between the observed value of that attribute and its predetermined value. Unconventional techniques such as hybrid data mining/complex network classification algorithm is able to perceive illegal instances in an actual card transaction data set, based on network reconstruction algorithm that allows creating representations of the deviation of one instance from a reference group have proved efficient typically on medium sized online transaction. There have also been efforts to progress from a completely new aspect. Attempts have been made to improve the alertfeedback interaction in case of fraudulent transaction. In case of fraudulent transaction, the authorised system would be alerted and a feedback would be sent to deny the ongoing transaction. Artificial Genetic Algorithm, one of the approaches that shed new light in this domain, countered fraud from a different direction. It proved accurate in finding out the fraudulent transactions and minimizing the number of false alerts. Even though, it was accompanied by classification problem with variable misclassification costs.

CHAPTER 3
SYSTEM DESIGN

3.1 DEVELOPMENT ENVIRONMENT

3.1.1 HARDWARE SPECIFICATIONS

This project uses minimal hardware but in order to run the project efficiently without any lack of user experience, the following specifications are recommended

Table 3.1.1 Hardware Specifications

PROCESSOR	Intel Core i5
RAM	4GB or above
HARD DISK	6GB
PROCESSOR FREQUENCY	1.5 GHz or above

3.1.2 SOFTWARE SPECIFICATIONS

The software specifications in order to execute the project has been listed down in the below table. The requirements in terms of the software that needs to be preinstalled and the languages needed to develop the project has been listed out below.

Table 3.1.2 Software Specifications

BACK END	Python
SOFTWARES USED	Jupyter Notebook

3.2 SYSTEM DESIGN

3.2.1 ARCHITECTURE DIAGRAM

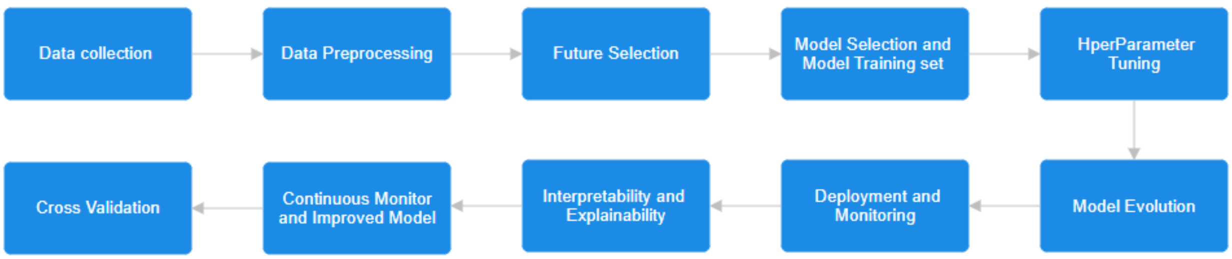

Fig 3.2.1 Architecture Diagram

DATA COLLECTION :

The first phase of the project involves data from trusted sources such as kaggle.The data set collected should have desired data columns and be able provide better results and the size should be sufficient enough.

DATA PREPROCESSING :

The Data collected won't be in a state that can be used for training purposes hence, the data should undergo the step of preprocessing in which common problems are eradicated such as missing values , improper spelling in data or incorrectness in data etc. Various python libraries specialized for data analysis can be utilized for this purpose such as Numpy, Pandas.This step is crucial for the project as these may cause inefficiency if they are fed directly to the model.

CHAPTER 4
PROJECT DESCRIPTION

4.1 MODULE DESCRIPTION

4.1.1 DATA COLLECTION :

The first phase of the project involves data from trusted sources such as kaggle.The data set collected should have desired data columns and be able provide better results and the size should be sufficient enough.

4.1.2 DATA PREPROCESSING :

The Data collected won't be in a state that can be used for training purposes hence, the data should undergo the step of preprocessing in which common problems are eradicated such as missing values , improper spelling in data or incorrectness in data etc. Various python libraries specialized for data analysis can be utilized for this purpose such as Numpy, Pandas.This step is crucial for the project as these may cause inefficiency if they are fed directly to the model.

4.1.3 EDA :

EDA stands for Exploratory Data Analysis in which the entire acquired data is analyzed for its relation within the data. Any outliers or deviation of data can be inferred at this point and also this helps to gain the significance of each data column. The common libraries utilized for this step include Matplotlib and Seaborn. Both of these are visualization tools commonly used in the project. Through EDA, we concluded that several attributes of users such as phone number, user id etc. are redundant and thus they are dropped. Heatmaps are extensively used to know the correlation between various attributes.

4.1.4 MODEL TRAINING :

The vectorized text data is used to train a convolutional neural network model . During training, the model adjusts its internal parameters iteratively to minimize a defined loss function. Dropout layers are included to prevent overfitting, ensuring the model generalizes well to unseen data. The model is trained using a portion of the data, while performance is monitored using a separate validation set.

4.1.5 MODEL EVALUATION :

Once training is complete, the model's performance is evaluated using a separate test dataset. Performance metrics such as accuracy, precision, and recall are calculated to assess the model's effectiveness in classifying legal descriptions.

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 IMPLEMENTATION

First of all, we obtained our dataset from Kaggle, a data analysis website which provides datasets. Inside this dataset, there are 31 columns out of which 28 are named as v1-v28 to protect sensitive data. The other columns represent Time, Amount and Class. Time shows the time gap between the first transaction and the following one. The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

First of all, importing all the necessary libraries and loading the data,

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import gridspec
```

```
C:\Users\admin\AppData\Local\Temp\ipykernel_10380\118333752.py:2: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
```

```
data = pd.read_csv("creditcard.csv")
```

Then understanding and describing of data includes implementation of simple methods such as head() [which is used to return the first 5 rows of the dataset] , describe() [returns the overview of our imported dataset], etc.,

```
data.head()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	
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	0.128539	-0.18
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	0.167170	0.12
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	-0.327642	-0.13
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	0.647376	-0.22
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	-0.206010	0.50

```
print(data.shape)
print(data.describe())
```

```
(284807, 31)
                Time            V1            V2            V3            V4  \
count  284807.000000  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05
mean    94813.859575  1.168375e-15  3.416908e-16 -1.379537e-15  2.074095e-15
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

                 V5            V6            V7            V8            V9  \
count  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05
mean   9.604066e-16  1.487313e-15 -5.556467e-16  1.213481e-16 -2.406331e-15
std    1.380247e+00  1.332271e+00  1.237094e+00  1.194353e+00  1.098632e+00
min   -1.137433e+02 -2.616051e+01 -4.355724e+01 -7.321672e+01 -1.343407e+01
25%   -6.915971e-01 -7.682956e-01 -5.540759e-01 -2.086297e-01 -6.430976e-01
50%   -5.433583e-02 -2.741871e-01  4.010308e-02  2.235804e-02 -5.142873e-02
75%    6.119264e-01  3.985649e-01  5.704361e-01  3.273459e-01  5.971390e-01
max    3.480167e+01  7.330163e+01  1.205895e+02  2.000721e+01  1.559499e+01

       ...           V21           V22           V23           V24  \
count  ...  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05
mean   ...  1.654067e-16 -3.568593e-16  2.578648e-16  4.473266e-15
std    ...  7.345240e-01  7.257016e-01  6.244603e-01  6.056471e-01
min    ... -3.483038e+01 -1.093314e+01 -4.480774e+01 -2.836627e+00
25%    ... -2.283949e-01 -5.423504e-01 -1.618463e-01 -3.545861e-01
50%    ... -2.945017e-02  6.781943e-03 -1.119293e-02  4.097606e-02
75%    ...  1.863772e-01  5.285536e-01  1.476421e-01  4.395266e-01
max    ...  2.720284e+01  1.050309e+01  2.252841e+01  4.584549e+00
```

```
                V25           V26           V27           V28         Amount  \
count  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05  284807.000000
mean   5.340915e-16  1.683437e-15 -3.660091e-16 -1.227390e-16      88.349619
std    5.212781e-01  4.822270e-01  4.036325e-01  3.300833e-01     250.120109
min   -1.029540e+01 -2.604551e+00 -2.256568e+01 -1.543008e+01       0.000000
25%   -3.171451e-01 -3.269839e-01 -7.083953e-02 -5.295979e-02       5.600000
50%    1.659350e-02 -5.213911e-02  1.342146e-03  1.124383e-02      22.000000
75%    3.507156e-01  2.409522e-01  9.104512e-02  7.827995e-02      77.165000
max    7.519589e+00  3.517346e+00  3.161220e+01  3.384781e+01   25691.160000

               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]
```

Determining the imbalance in the data,

```
fraud = data[data['Class'] == 1]
valid = data[data['Class'] == 0]
outlierFraction = len(fraud)/float(len(valid))
print(outlierFraction)
print('Fraud Cases: {}'.format(len(data[data['Class'] == 1])))
print('Valid Transactions: {}'.format(len(data[data['Class'] == 0])))
```

```
0.0017304750013189597
Fraud Cases: 492
Valid Transactions: 284315
```

Only *0.17%* fraudulent transaction out all the transactions. The data is highly Unbalanced. Lets first apply our models without balancing it and if we don't get a good accuracy then we can find a way to balance this dataset. But first, let's implement the model without it and will balance the data only if needed.

Printing amount details for fraudulent transactions using the describe() method,

```
print("Amount details of the fraudulent transaction")
fraud.Amount.describe()
```

```
Amount details of the fraudulent transaction

count     492.000000
mean      122.211321
std       256.683288
min         0.000000
25%         1.000000
50%         9.250000
75%       105.890000
max      2125.870000
Name: Amount, dtype: float64
```

Printing amount details for normal/valid transactions using the describe() method,

```
print("details of valid transaction")
valid.Amount.describe()
```

```
details of valid transaction

count    284315.000000
mean         88.291022
std         250.105092
min           0.000000
25%           5.650000
50%          22.000000
75%          77.050000
max       25691.160000
Name: Amount, dtype: float64
```

As we can clearly notice from this, the average Money transaction for the fraudulent ones is more. This makes this problem crucial to deal with.

Building a Random Forest model using Scikit learn,

```python
from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier()
rfc.fit(xTrain, yTrain)

yPred = rfc.predict(xTest)
```

5.2 OUTPUT SCREENSHOTS

Plotting the correlation matrix, The correlation matrix graphically gives us an idea of how features correlate with each other and can help us predict what are the features that are most relevant for the prediction.

```python
corrmat = data.corr()
fig = plt.figure(figsize = (12, 9))
sns.heatmap(corrmat, vmax = .8, square = True)
plt.show()
```

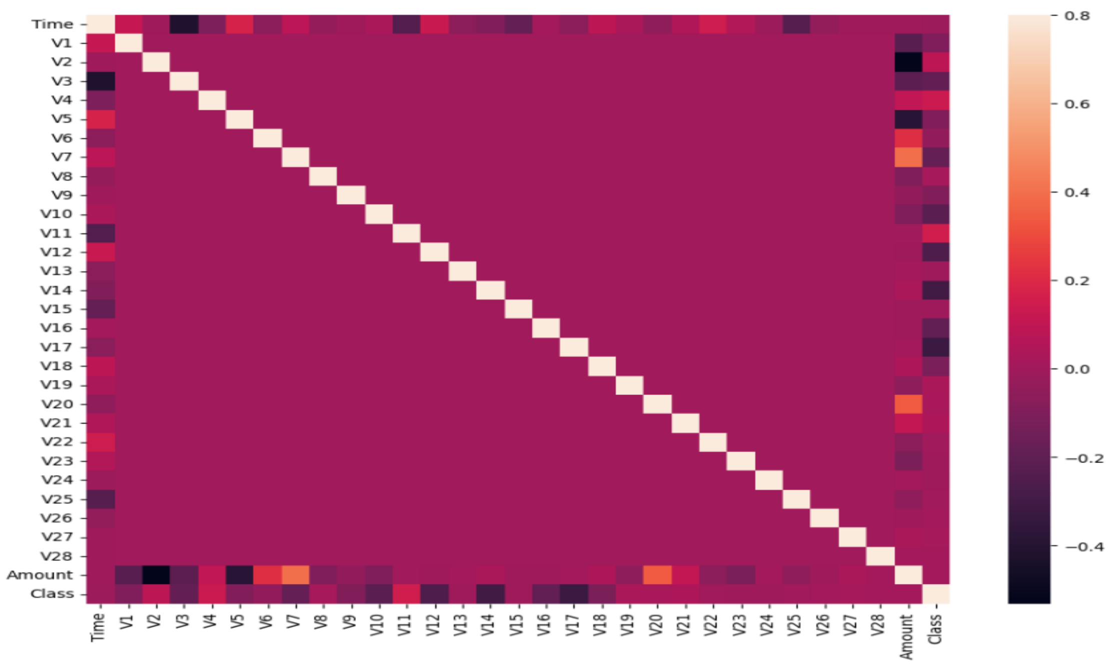

Fig 5.2.1 Correlation Matrix

In the HeatMap we can clearly see that most of the features do not correlate to other features but there are some features that either has a positive or a negative correlation with each other. For example, *V2* and *V5* are highly negatively correlated with the feature called *Amount*. We also see some correlation with *V20* and *Amount*. This gives us a deeper understanding of the Data available to us.

Building all kinds of evaluating parameters,

```python
from sklearn.metrics import classification_report, accuracy_score
from sklearn.metrics import precision_score, recall_score
from sklearn.metrics import f1_score, matthews_corrcoef
from sklearn.metrics import confusion_matrix
```

```python
n_outliers = len(fraud)
n_errors = (yPred != yTest).sum()
print("The model used is Random Forest classifier")

acc = accuracy_score(yTest, yPred)
print("The accuracy is {}".format(acc))

prec = precision_score(yTest, yPred)
print("The precision is {}".format(prec))

rec = recall_score(yTest, yPred)
print("The recall is {}".format(rec))

f1 = f1_score(yTest, yPred)
print("The F1-Score is {}".format(f1))

MCC = matthews_corrcoef(yTest, yPred)
print("The Matthews correlation coefficient is {}".format(MCC))
```

```
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 0.8694303688259544
```

Fig 5.2.2 Evaluating Parameters

Finally by visualizing the confusion matrix, we get the comparison with other algorithms without dealing with the imbalancing of the data,

```
LABELS = ['Normal', 'Fraud']
conf_matrix = confusion_matrix(yTest, yPred)
plt.figure(figsize =(12, 12))
sns.heatmap(conf_matrix, xticklabels = LABELS, yticklabels = LABELS, annot = True, fmt ="d");
plt.title("Confusion matrix")
plt.ylabel('True class')
plt.xlabel('Predicted class')
plt.show()
```

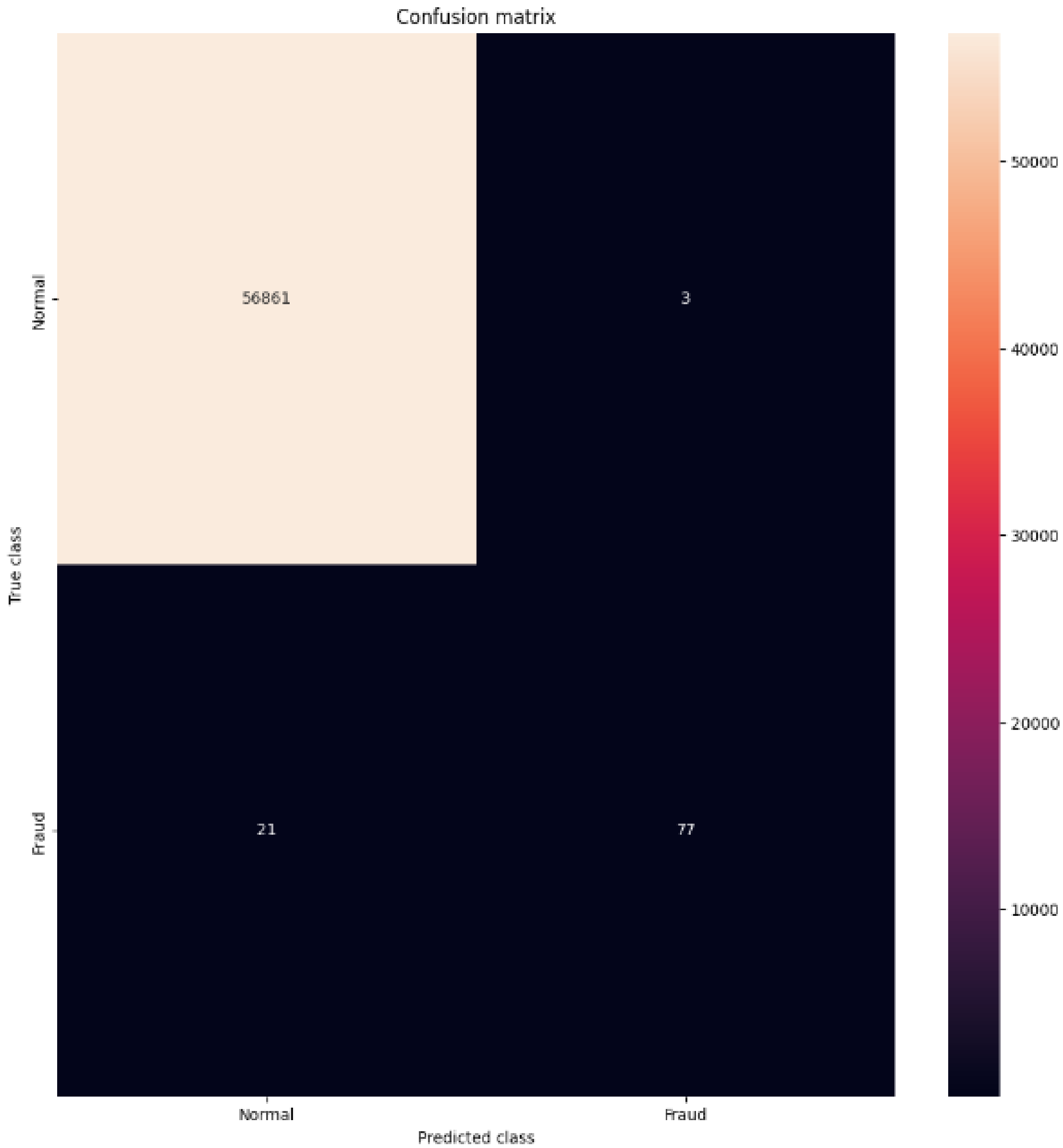

Fig 5.2.3 Confusion Matrix

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENTS

6.1 CONCLUSION

Credit card fraud has without hesitation an expression of criminal deception. Fraud identification seems to be a complicated problem that requires a significant amount of skill until throwing algorithms regarding machine learning into it. However, it is an implementation for both the better of machine learning as well as artificial intelligence, ensuring that perhaps the funds of both the customer seems to be secure and therefore not manipulated. The whole research article addressed an effective system of identifying fraud depending on machine learning methodologies, with such a feedback system. Its feedback process relates to enhancing the classifier's detection rate as well as effectiveness. An observational analysis has been conducted on respective machine learning strategies except for random forest, tree classifiers, artificial neural networks, vector supporting machine, Naïve Baiyes, logistic regression as well as gradient boosting classifier techniques, but also multiple performances evaluating parameters have been calculated such as precision, recall, F1-score, accuracy, and FPR percentage, for any method which has better results for evaluation parameters can be treated as best performing method. Here Random forest is showing better results as compared to other machine learning classifiers.

6.2 FUTURE ENHANCEMENTS

This project could include the integration of advanced machine learning techniques like deep learning and ensemble methods to improve detection accuracy. Additionally, implementing real-time processing and anomaly detection systems can help identify fraudulent transactions instantaneously. Enhancing the system with robust feature engineering, leveraging transaction metadata, and incorporating external data sources such as geolocation and merchant information can further refine the model. Regularly updating the model with the latest data and employing techniques like transfer learning can help maintain its effectiveness against evolving fraud patterns. Finally, adding explainability features will ensure that the decision-making process is transparent and understandable to stakeholders.

REFERENCES

[1] A. Shen, R. Tong, Y. Deng, "Application of classification models on credit card fraud detection", Service Systems and Service Management 2007 International Conference, pp. 1-4, 2007.

[2] A. C. Bahnsen, A. Stojanovic, D. Aouada, B. Ottersten, "Cost sensitive credit card fraud detection using Bayes minimum risk", Machine Learning and Applications (ICMLA). 2013 12th International Conference, vol. 1, pp. 333-338, 2013.

[3] B.Meena, I.S.L.Sarwani, S.V.S.S.Lakshmi," Web Service mining and its techniques in Web Mining" IJAEGT,Volume 2,Issue 1 , Page No.385-389.

[4] F. N. Ogwueleka, "Data Mining Application in Credit Card Fraud Detection System", Journal of Engineering Science and Technology, vol. 6, no. 3, pp. 311-322, 2011.

[5] G. Singh, R. Gupta, A. Rastogi, M. D. S. Chandel, A. Riyaz, "A Machine Learning Approach for Detection of Fraud based on SVM", International Journal of Scientific Engineering and Technology, vol. 1, no. 3, pp. 194-198, 2012, ISSN ISSN: 2277-1581.

[6] K. Chaudhary, B. Mallick, "Credit Card Fraud: The study of its impact and detection techniques", International Journal of Computer Science and Network (IJCSN), vol. 1, no. 4, pp. 31-35, 2012, ISSN ISSN: 2277-5420.

[7] M. J. Islam, Q. M. J. Wu, M. Ahmadi, M. A. SidAhmed, "Investigating the Performance of Naive-Bayes Classifiers and KNearestNeighbor Classifiers", IEEE International Conference on Convergence Information Technology, pp. 1541-1546, 2007.

[8] R. Wheeler, S. Aitken, "Multiple algorithms for fraud detection" in Knowledge-Based Systems, Elsevier, vol. 13, no. 2, pp. 93-99, 2000.

[9] S. Patil, H. Somavanshi, J. Gaikwad, A. Deshmane, R. Badgujar, "Credit Card Fraud Detection Using Decision Tree Induction Algorithm", International Journal of Computer Science and Mobile Computing (IJCSMC), vol. 4, no. 4, pp. 92-95, 2015, ISSN ISSN: 2320-088X.

[10] S. Maes, K. Tuyls, B. Vanschoenwinkel, B. Manderick, "Credit card fraud detection using Bayesian and neural networks", Proceedings of the 1st international naiso congress on neuro fuzzy technologies, pp. 261-270, 2002.

[11] S. Bhattacharyya, S. Jha, K. Tharakunnel, J. C. Westland, "Data mining for credit card fraud: A comparative study", Decision Support Systems, vol. 50, no. 3, pp. 602-613, 2011.

[12] Y. Sahin, E. Duman, "Detecting credit card fraud by ANN and logistic regression", Innovations in Intelligent Systems and Applications (INISTA) 2011 International Symposium, pp. 315-319, 2011.

[13] "Credit Card Fraud Detection Based on Transaction Behaviour -by John Richard D. Kho, Larry A. Vea" published by Proc. of the 2017 IEEE Region 10 Conference (TENCON), Malaysia, November 5-8, 2017