

Credit Card Fraud Detection Using Supervised Machine Learning

A PROJECT REPORT

Submitted by:

Piyush Kumar (20BCS9107)
Shashank Singh (20BCS9109)
Akash Kumar (20BCS9110)
Rithik Sharma (20BCS9135)
Muskan Kumari (20BCS8285)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING



Chandigarh University

MAY 2022



BONAFIDE CERTIFICATE

Certified that this project report “**Credit Card Fraud Detection Using Supervised Machine Learning**” is the bonafide work of “**Piyush Kumar, Shashank Singh, Muskan Kumari, Akash Kumar and Rithik Sharma**” who carried out the project work under my/our supervision.

**SIGNATURE OF
SUPERVISOR**

**SIGNATURE OF HEAD
OF DEPARTMENT**

Submitted for the project viva-voce examination held on 19/05/2022

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We are grateful to our respectable teacher, Er. Nidhi, whose insightful leadership and knowledge benefited us to complete this project successfully. Thank you so much for your continuous support and presence whenever needed.

We would also like to thank Er. Nidhi for his advice and contribution to the project and the preparation of this report.

TABLE OF CONTENTS

Sr. no	Topics	Page no
1	ABSTRACT	7
2	Chapter 1: Introduction	8
3	Chapter 2: Literature survey	9-10
4	Chapter 3: Design flow/Process	11-15
5	Chapter 4 Results analysis and validation	16-24
6	Chapter 5: Conclusion and future work	25-30
7	References	31-33
8	Biography	34

LIST OF FIGURES

Fig. no	Fig. name	Page no
1	Proposed System Block Diagram	11
2	Frauds Using Card Not Present Transaction	13
3	flow chart	13
4	Architecture diagram	14
5	Basic concept of research or architecture diagram	14
6	Chart	15
7	Count of Fraudulent vs Non-Fraudulent Transactions	15
8	User interface for train and test data	20
9	Detection of fraud or normal transaction	20
10	Confusion matrix for Logistic regression	21
11	Confusion matrix for Naive Bayes	22
12	Confusion matrix for Decision Tree	22
13	Confusion matrix for ANN	23

LIST OF TABLES

Table no	Content	Page no
1	The following table shows the data of students having their names and roll numbers, age and gender.	18
2	The following table gives us the dimension and description about above mentioned data structures used in Pandas	18
3	Accuracy, precision, recall comparison table for different ML algorithms	24

ABSTRACT

This Project is focused on credit card fraud detection in real world scenarios. Nowadays credit card frauds are drastically increasing in number as compared to earlier times. Criminals are using fake identity and various technologies to trap the users and get the money out of them. Therefore, it is very essential to find a solution to these types of frauds. In this proposed project we designed a model to detect the fraud activity in credit card transactions. This system can provide most of the important features required to detect illegal and illicit transactions. As technology changes constantly, it is becoming difficult to track the behavior and pattern of criminal transactions. To come up with the solution one can make use of technologies with the increase of machine learning, artificial intelligence and other relevant fields of information technology; it becomes feasible to automate this process and to save some of the intensive amounts of labor that is put into detecting credit card fraud. Initially, we will collect the credit card usage data-set by users and classify it as trained and testing dataset using a random forest algorithm and decision trees. Using this feasible algorithm, we can analyze the larger data-set and user provided current data-set. Then augment the accuracy of the result data. Proceeded with the application of processing of some of the attributes provided which can find affected fraud detection in viewing the graphical model of data visualization. The performance of the techniques is gauged based on accuracy, sensitivity, and specificity, precision. The results is indicated concerning the best accuracy for Random Forest are unit 98.6% respectively.

CHAPTER 1- INTRODUCTION:

Nowadays Credit card usage has been drastically increased across the world, now people believe in going cashless and are completely dependent on online transactions. The credit card has made the digital transaction easier and more accessible. A huge number of dollars of loss are caused every year by the criminal credit card transactions. Fraud is as old as mankind itself and can take an unlimited variety of different forms. The PwC global economic crime survey of 2017 suggests that approximately 48% of organizations experienced economic crime. Therefore, there's positively a necessity to unravel the matter of credit card fraud detection. Moreover, the growth of new technologies provides supplementary ways in which criminals may commit a scam. The use of credit cards is predominant in modern day society and credit card fraud has been kept on increasing in recent years. Huge Financial losses have been fraudulent effects on not only merchants and banks but also the individual person who are using the credits. Fraud may also affect the reputation and image of a merchant causing non-financial losses that. For example, if a cardholder is a victim of fraud with a certain company, he may no longer trust their business and choose a competitor. Fraud Detection is the process of monitoring the transaction behavior of a cardholder to detect whether an incoming transaction is authentic and authorized or not otherwise it will be detected as illicit. In a planned system, we are applying the random forest algorithm for classifying the credit card dataset. Random Forest is an associate in the nursing algorithmic program for classification and regression. Hence, it is a collection of decision tree classifiers. The random forest has an advantage over the decision tree as it corrects the habit of over fitting to their training set. A subset of the training set is sampled randomly so that to train each individual tree and then a decision tree is built, each node then splits on a feature designated from a random subset of the complete feature set. Even for large data sets with many features and data instances, training is extremely fast in the random forest and because each tree is trained independently of the others. The Random Forest algorithm has been found to provide a good estimate of the generalization error and to be resistant to overfitting.

Chapter 2: Literature survey

Multiple Supervised and Semi-Supervised machine learning techniques are used for fraud detection, but we aim is to overcome three main challenges with card frauds related dataset i.e., strong class imbalance, the inclusion of labelled and unlabelled samples, and to increase the ability to process a large number of transactions. Different Supervised machine learning algorithms like Decision Trees, Naïve Bayes Classification, Least Squares Regression, Logistic Regression and SVM are used to detect fraudulent transactions in real-time datasets. Two methods under random forests are used to train the behavioural features of normal and abnormal transactions. They are Random-tree-based random forest and CART based. Even though random forest obtains good results on small set data, there are still some problems in case of imbalanced data. The future work will focus on solving the above-mentioned problem. The algorithm of the random forest itself should be improved. Performance of Logistic Regression, K-Nearest Neighbour, and Naïve Bayes are analysed on highly skewed credit card fraud data where Research is carried out on examining meta-classifiers and meta-learning approaches in handling highly imbalanced credit card fraud data. Through supervised learning methods can be used there may fail at certain cases of detecting the fraud cases. A model of deep Auto-encoder and restricted Boltzmann machine (RBM) that can construct normal transactions to find anomalies from normal patterns. Not only that a hybrid method is developed with a combination of Ada boost and Majority Voting methods.

Problem Definition

With the growth of e-commerce websites, people and financial companies rely on online services to carry out their transactions that have led to an exponential increase in the credit card frauds. Fraudulent credit card transactions lead to a loss of huge amount of money. The design of an effective fraud detection system is necessary in order to reduce the losses incurred by the customers and financial companies. Research has been done on many models and methods to prevent and detect credit card frauds. Some credit card fraud transaction datasets contain the problem of imbalance in datasets. A good fraud detection system should be able to identify the fraud transaction accurately and should make the detection possible in real-time transactions. Fraud detection can be divided into two groups: anomaly detection and misuse detection. Anomaly detection systems bring normal transaction to be trained and use techniques to determine novel frauds. Conversely, a misuse fraud detection system uses the labelled transaction as normal or fraud transaction to be trained in the database history. So, this misuse detection system entails a system of supervised learning and anomaly detection system a system of unsupervised learning. Fraudsters masquerade the normal behavior of customers and the fraud patterns are changing rapidly so the fraud detection system needs to constantly learn and update. Credit card frauds can be broadly classified into three categories, that is, traditional card related frauds (application, stolen, account takeover, fake and counterfeit), merchant related frauds (merchant collusion and triangulation) and Internet frauds (site cloning, credit card generators and false merchant sites).

Objectives

We propose a Machine learning model to detect fraudulent credit card activities in online financial transactions. Analyzing fake transactions manually is impracticable due to vast amounts of data and its complexity. However, adequately given informative features, could make it is possible using Machine Learning. This hypothesis will be explored in the project. To classify fraudulent and legitimate credit card transaction by supervised learning Algorithm such as Random forest. To help us to get awareness about the fraudulent and without loss of any financially.

Chapter 3: Design flow/Process

Increase in online transactions using payment methods like credit card has also increased the fraudulent activities. Every year, a large amount of financial losses are caused by these illegal credit card transactions. No system is 100% secure and there is always a loophole in them. Therefore there is need to solve the issues of detecting fraud in transactions done by credit cards. To overcome this problem the proposed system for fraud identification in credit card transactions is designed using Random Forest algorithm. This algorithm uses combination of Decision Tree to solve the problem. Each tree is trained using dataset and based on this training each tree gives probability of transaction been fraud or legal. After that model predicts the result.

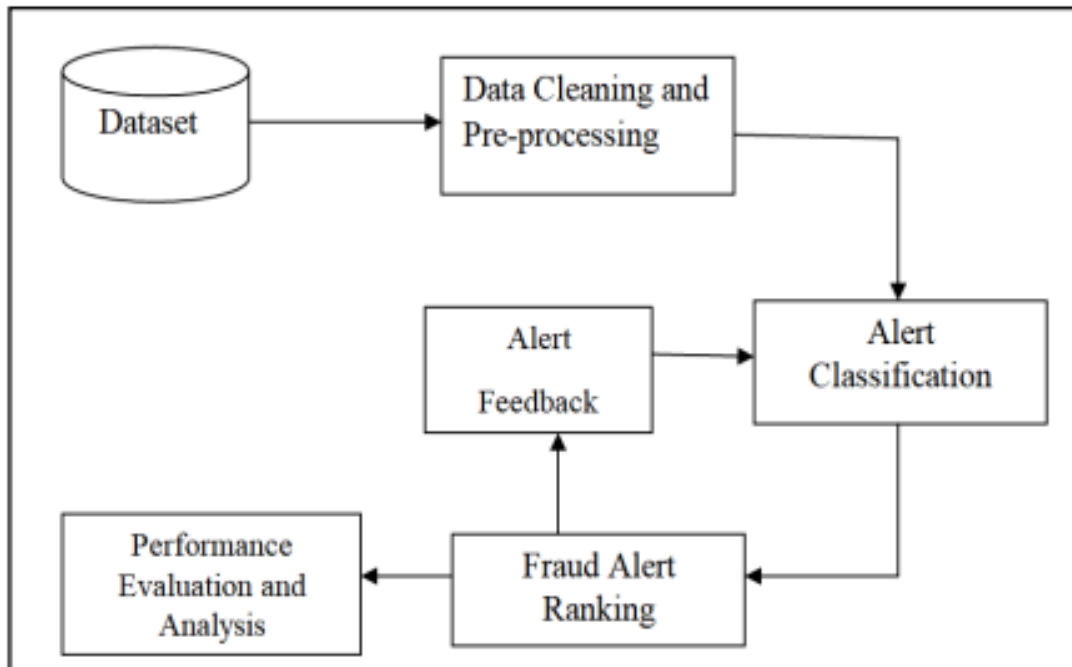


Figure 1 Proposed System Block Diagram

Modules

As shown in the block diagram, following are the modules of system:

- Data Cleaning and Preprocessing
- Alerts Classification
- Alert Ranking
- Performance Analysis

Data Cleaning and Preprocessing

The model accuracy depends on amount of data on which it is trained. The more amounts of data better will be the performance of model. In this first module the selected data is cleaned and preprocessed as follow:

- a. Cleaning: Fixing of missing data or removal of duplicate data from dataset is called as cleaning. The dataset may contain record which may be duplicate, incomplete or may have null values. Such records need to remove by cleaning.
- b. Sampling: As number of frauds in dataset is less than overall transaction, class distribution is unbalanced in credit card transaction. Hence sampling method is used to solve this issue.

Alert Classification

Here machine learning model is used that trains the model based on features associated with transactions like location from where transaction is made, zip code, IP address, time and identity of customer. All this dataset is fed as input to the classifier and classifier splits them into multiple decision trees. The sub-trees check this input for an authorized transaction and give probabilities of transaction to be fraud or legal. Combining the results of all sub-trees, the model will alert the fraudulent transaction.

Alert Ranking

This module ranks each alert using learning to rank algorithm. The algorithm ranks each alert identified by the model using likelihood. If it is found that alert has greater rank then a security question is generated. If the individual answers the security question correctly then the transaction is allowed otherwise it is blocked. The IP address and location of fraudster is then tracked by the system. This security questions will be created every time whenever the transaction is identified to be suspicious and rank of alert is highest. This makes the FDS user friendly and helps to launch complaint against fraudster. Also the number of alert generated by the system is reduced as compared to rule based approach system.

Algorithm of Proposed Strategy

User inputs v_1, v_2, \dots, v_n

Dataset D

Step1: Initiate User Input

Step2: for each transaction x do deploy pattern P compute probability PD generate alerts A

Step3: Implement RankNet Model to generate rank return rank

Step 4: if rank $\geq n$ then Display Security Question SQ verify SQ

Step 5: if SQ verified then allow transaction else block transaction and track fraudster location

[end if]

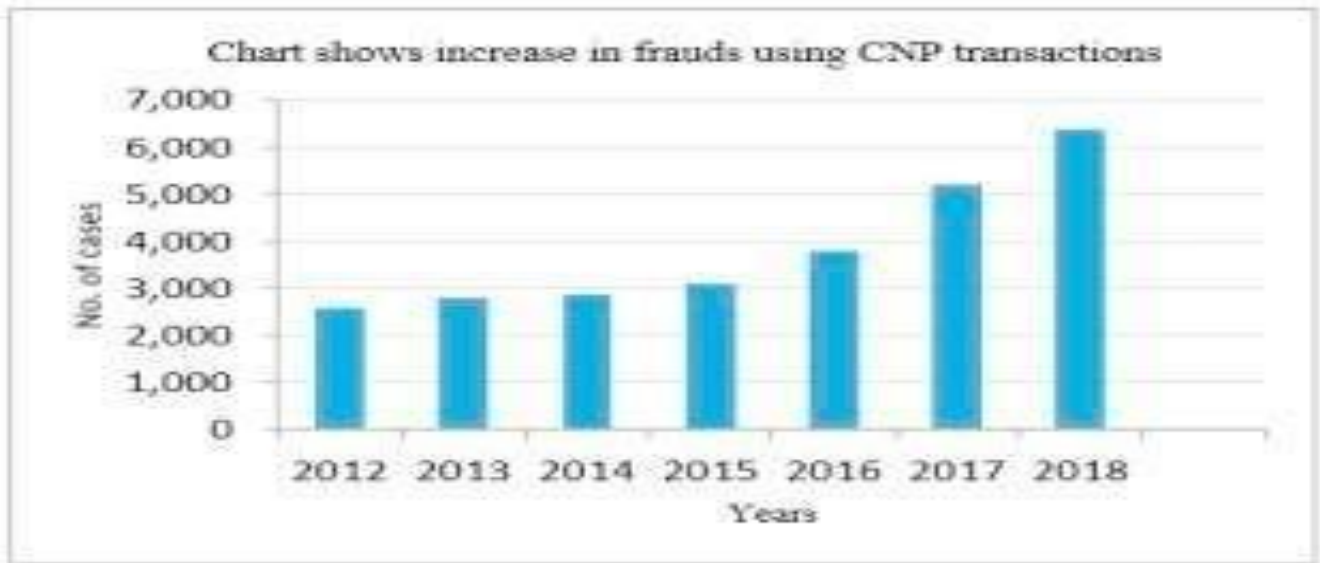


Fig. 2: Frauds Using Card Not Present Transaction

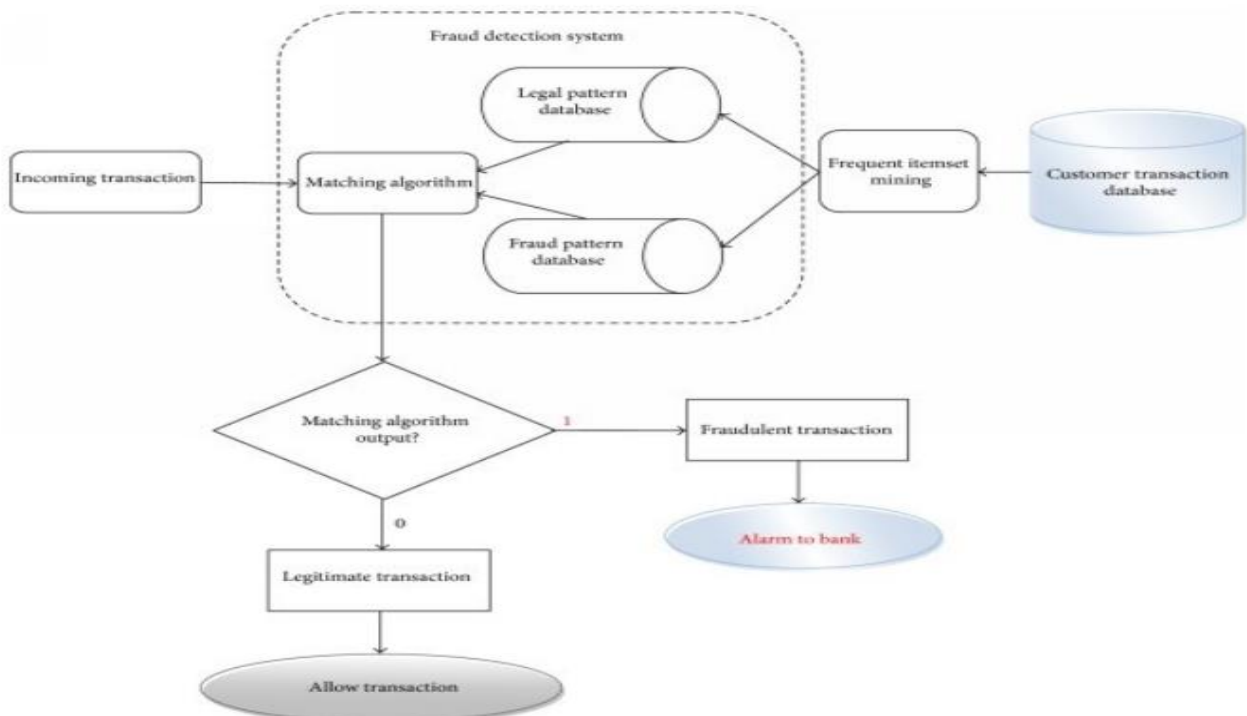


Fig. 3: flow chart

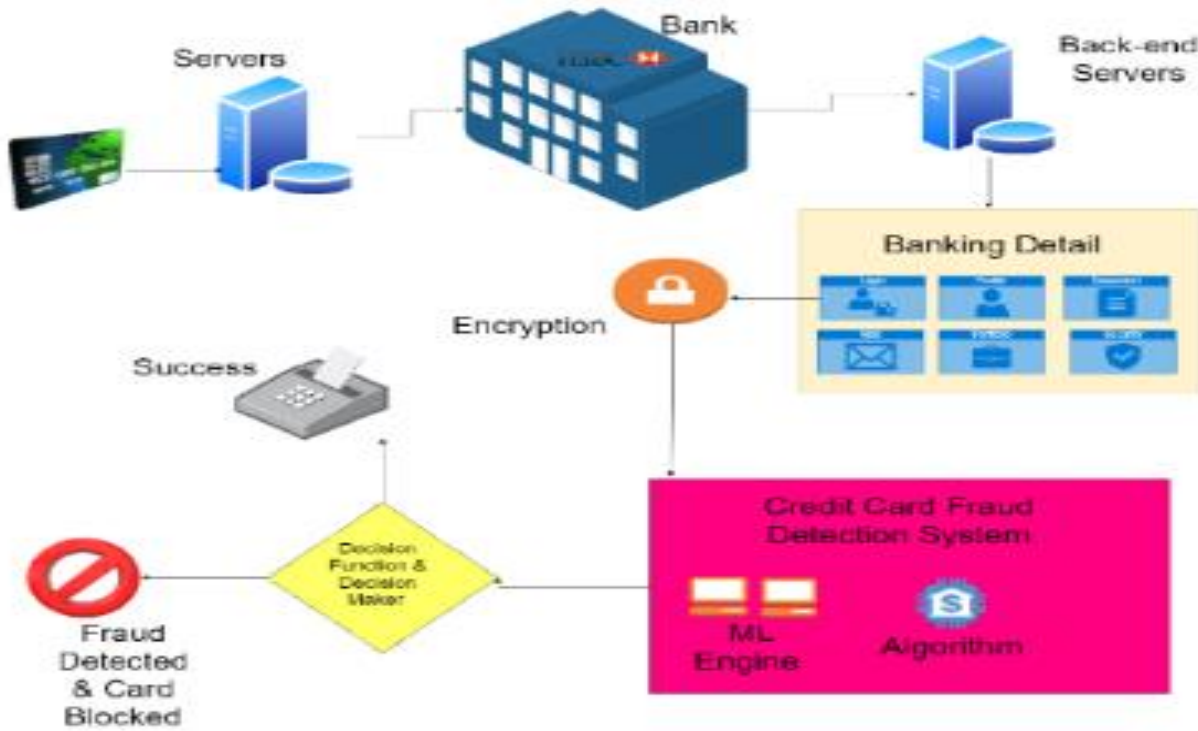


Fig. 4: Architecture diagram

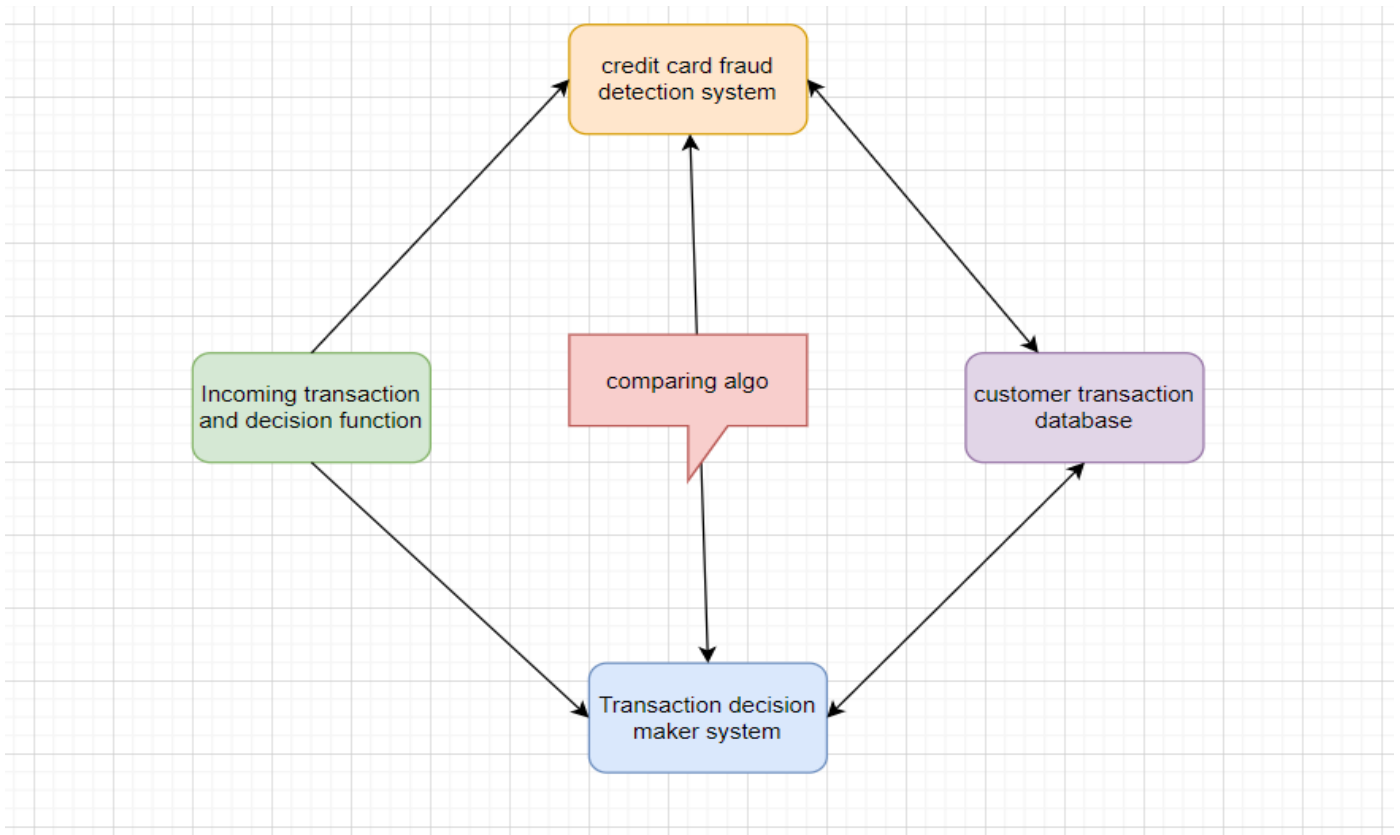


Fig. 5: Basic concept of research or architecture diagram

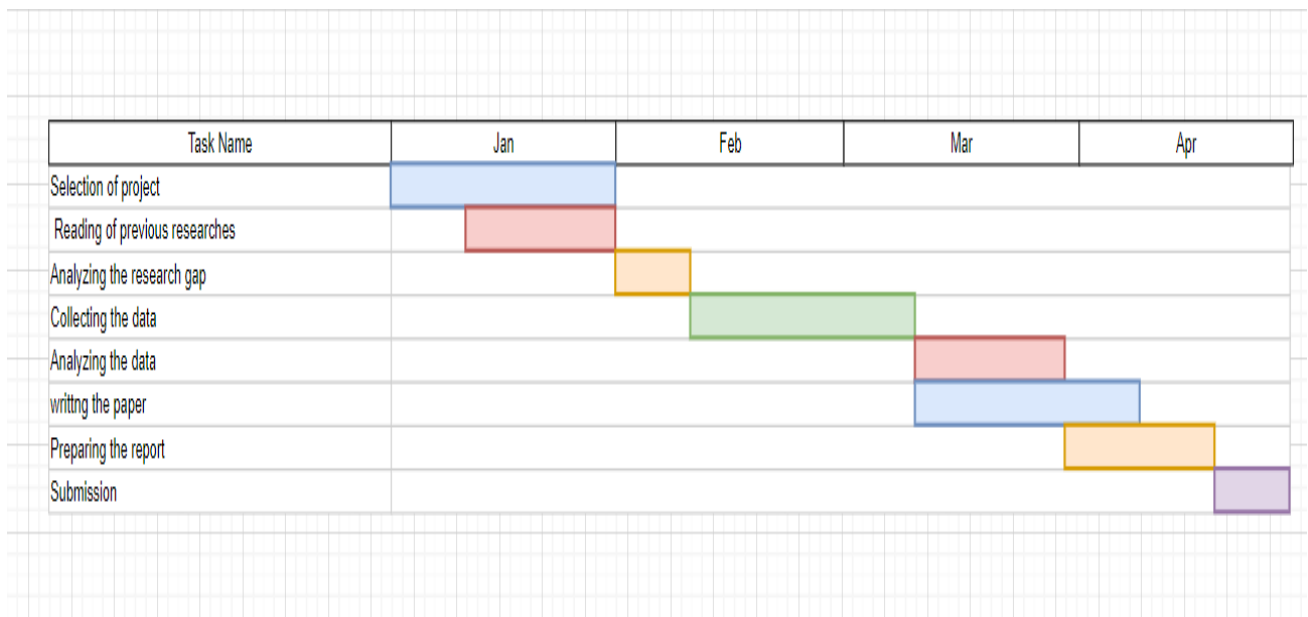


Fig. 6: Chart

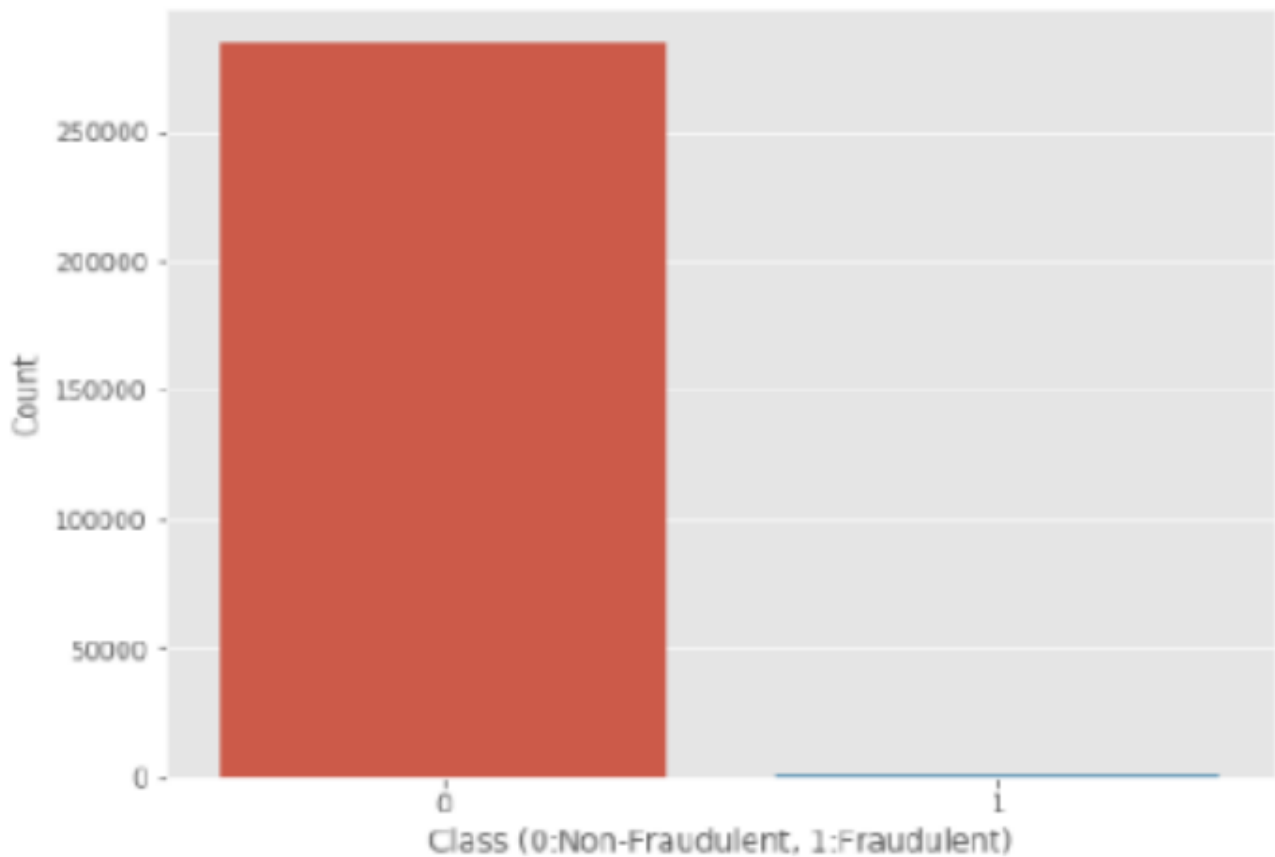


Fig. 7: Count of Fraudulent vs Non-Fraudulent Transactions

Chapter 4 Results analysis and validation

1. Jupyter Notebook

Computation notebooks have been used as electronic lab notebooks to document procedures, data, calculations, and findings. Jupyter notebooks provide an interactive computational environment for developing data science applications.

Jupyter notebooks combine software code, computational output, explanatory text, and rich content in a single document. Notebooks allow in-browser editing and execution of code and display computation results. A notebook is saved with an .ipynb extension. The Jupyter Notebook project supports dozens of programming languages, its name reflecting support for Julia (Ju), Python (Py), and R.

The following are some of the features of Jupyter notebooks that makes it one of the best components of Python ML ecosystem –

- Jupyter notebooks can illustrate the analysis process step by step by arranging the stuff like code, images, text, output etc. in a step by step manner.
- It helps a data scientist to document the thought process while developing the analysis process.
- One can also capture the result as the part of the notebook.
- With the help of jupyter notebooks, we can share our work with a peer also.

2. DataSets

- **NumPy**

It is another useful component that makes Python as one of the favorite languages for Data Science. It basically stands for Numerical Python and consists of multidimensional array objects. By using NumPy, we can perform the following important operations –

- Mathematical and logical operations on arrays.
- Fourier transformation
- Operations associated with linear algebra.

We can also see NumPy as the replacement of MatLab because NumPy is mostly used along with Scipy (Scientific Python) and Matplotlib (plotting library).

Execution :

```
import numpy as np
```

- **Pandas**

It is another useful Python library that makes Python one of the favorite languages for Data Science. Pandas is basically used for data manipulation, wrangling and analysis. It was developed by Wes McKinney in 2008. With the help of Pandas, in data processing we can accomplish the following five steps –

- Load
- Prepare
- Manipulate
- Model
- Analyze

Data representation in Pandas

The entire representation of data in Pandas is done with the help of following three data structures –

Series – It is basically a one-dimensional ndarray with an axis label which means it is like a simple array with homogeneous data. For example, the following series is a collection of integers 1,5,10,15,24,25...

1	5	10	15	24	25	28	36	40	89
---	---	----	----	----	----	----	----	----	----

Data frame – It is the most useful data structure and used for almost all kind of data representation and manipulation in pandas. It is basically a two-dimensional data structure which can contain heterogeneous data. Generally, tabular data is represented by using data frames. For example, the following table shows the data of students having their names and roll numbers, age and gender.

Name	Roll number	Age	Gender
Aarav	1	15	Male
Harshit	2	14	Male
Kanika	3	16	Female
Mayank	4	15	Male

Table: 1

Panel – It is a 3-dimensional data structure containing heterogeneous data. It is very difficult to represent the panel in graphical representation, but it can be illustrated as a container of DataFrame.

The following table gives us the dimension and description about above mentioned data structures used in Pandas –

Data Structure	Dimension	Description
Series	1-D	Size immutable, 1-D homogeneous data
DataFrames	2-D	Size Mutable, Heterogeneous data in tabular form
Panel	3-D	Size-mutable array, container of DataFrame.

Table: 2

Execution:

```
import pandas as pd
```

- **Scikit-learn**

Supervised learning algorithms: Think of any supervised machine learning algorithm you might have heard about and there is a very high chance that it is part of scikit-learn. Starting from Generalized linear models (e.g Linear Regression), Support Vector Machines (SVM), Decision Trees to Bayesian methods – all of them are part of scikit-learn toolbox. The spread of machine learning algorithms is one of the big reasons for the high usage of scikit-learn. I started using scikit to solve supervised learning problems and would recommend that to people new to scikit / machine learning as well.

Another useful and most important python library for Data Science and machine learning in Python is Scikit-learn. The following are some features of *Scikit-learn* that makes it so useful –

- It is built on NumPy, SciPy, and Matplotlib.
- It is an open source and can be reused under BSD license.
- It is accessible to everybody and can be reused in various contexts.
- Wide range of machine learning algorithms covering major areas of ML like classification, clustering, regression, dimensionality reduction, model selection etc. can be implemented with the help of it.

Execution:

```
from sklearn.model_selection import train_test_split
```

RESULTS

The figure 8 shows the user interface for test and train the data. Train and Test buttons are given to the user where using train the algorithms are trained and then o predict the fraud by clicking predict button it will take to another window where the input is given and output is seen as fraud or non fraud.

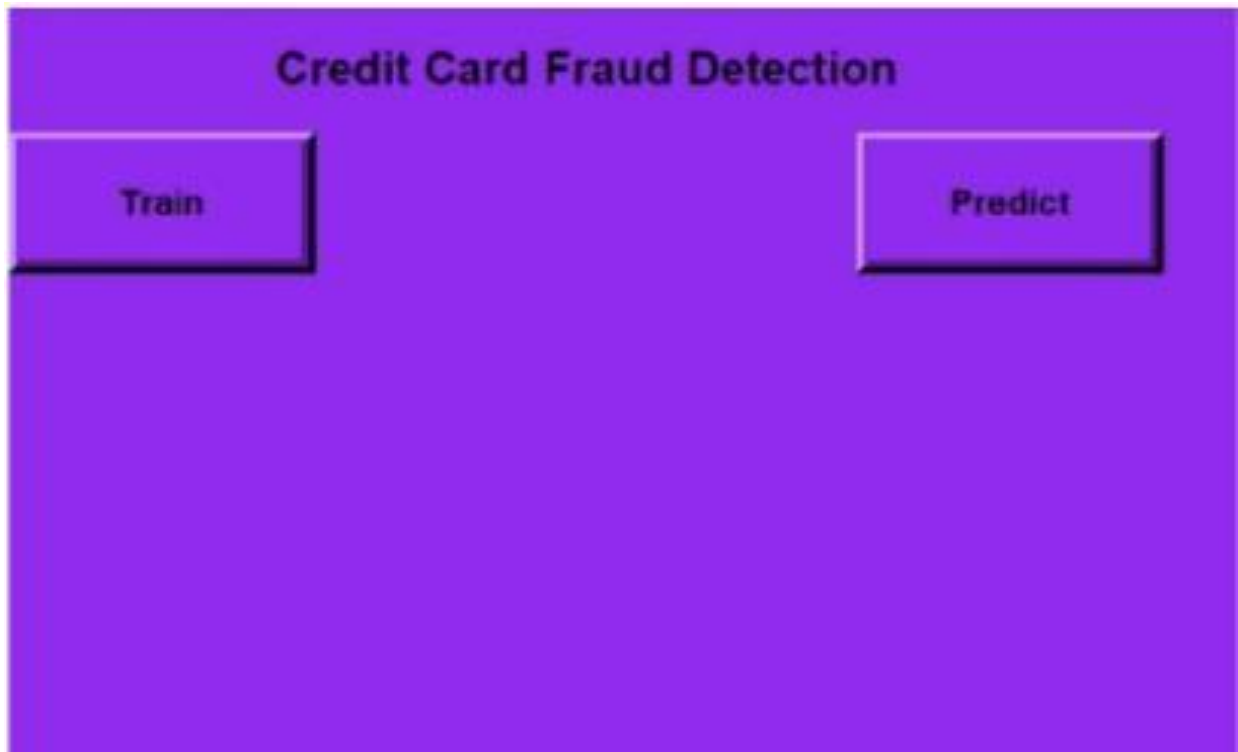


Fig. 8: User interface for train and test data

The figure 9 shows detection of fraud or nonfraud transaction. when predict button is clicked it will take to another window where it asks for data which is input to the machine learning algorithms and in the predict it will give output as fraud or nonfraud. comma separated 30 values are given including amount and time. Predicted result is displayed as fraud after providing the data. These results along with the as follows, where class 0 means the transaction was determined to be valid and 1 means it was determined as a fraud transaction.

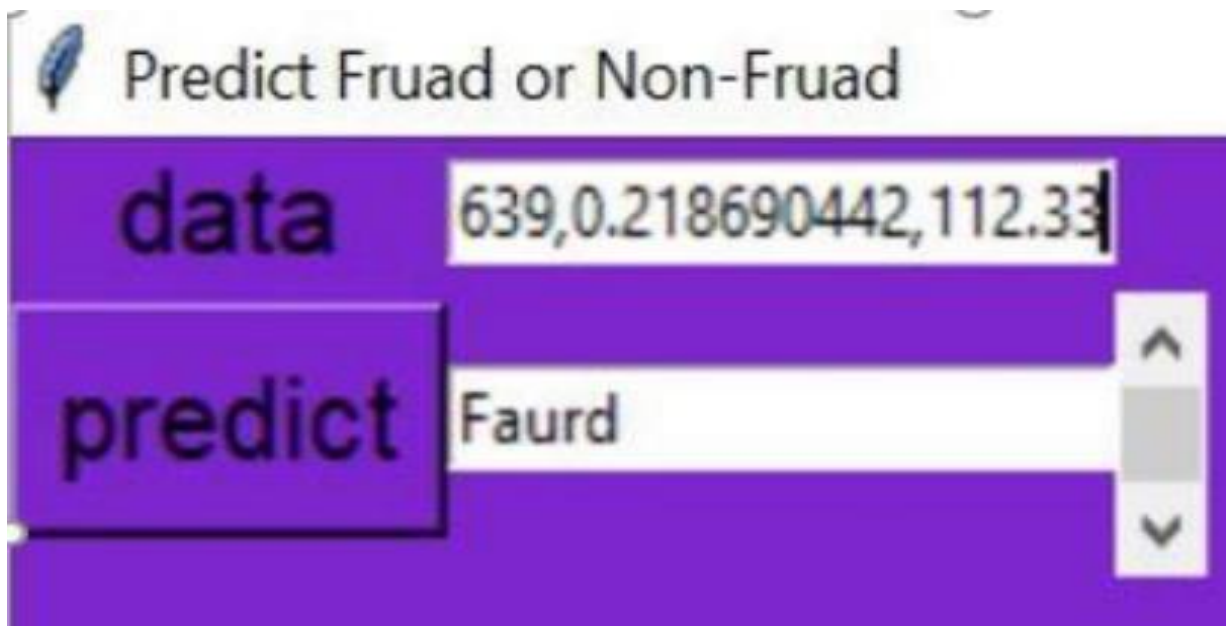


Fig. 9: Detection of fraud or normal transaction

1) Confusion matrix for Logistic regression Algorithm:

Fig 10 represents confusion matrix for Logistic regression algorithm. Which contains True Positive, True Negative, False Positive, False Negative. False positive value is lesser which shows fraud not detected cases are low. For logistic regression algorithm accuracy, recall, precision achieved are 94.84, 92.00, 97.58 respectively.

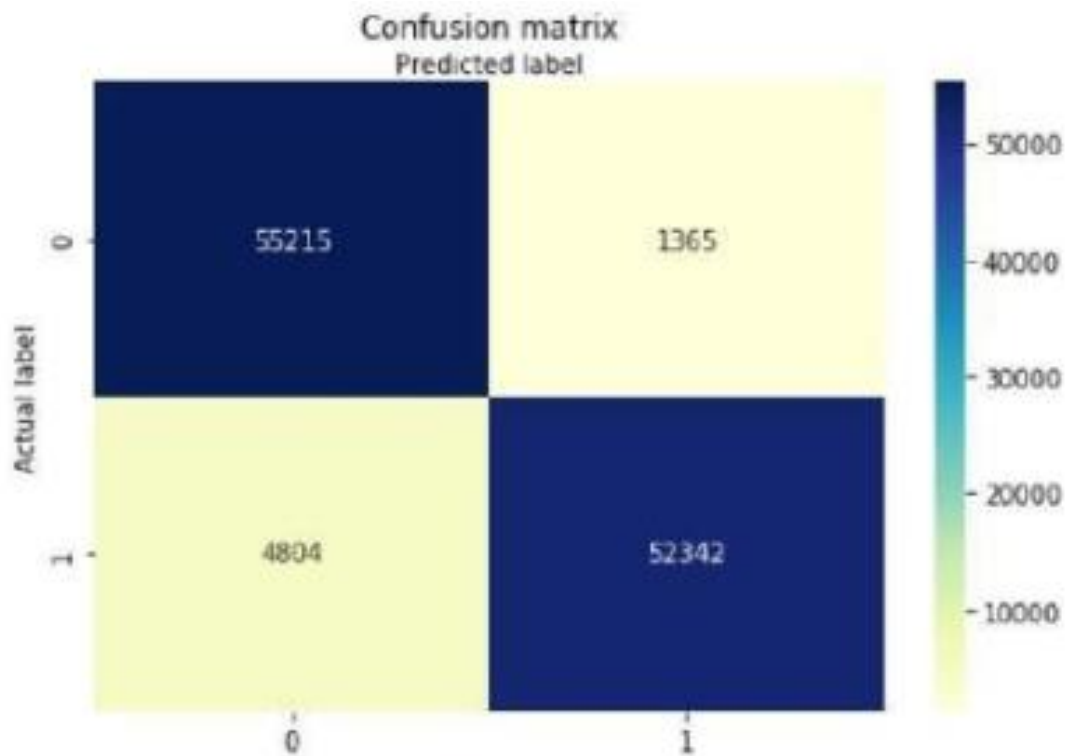


Fig. 10: Confusion matrix for Logistic regression

2) Confusion matrix for Naive Bayes Algorithm:

Fig 11 represents confusion matrix for Naive Bayes algorithm. Which contains True Positive, True Negative, False Positive, False Negative. False positive value is lesser which shows fraud not detected cases are low. For Naive Bayes algorithm accuracy, recall, precision achieved are 91.62, 84.82, 97.09 respectively.

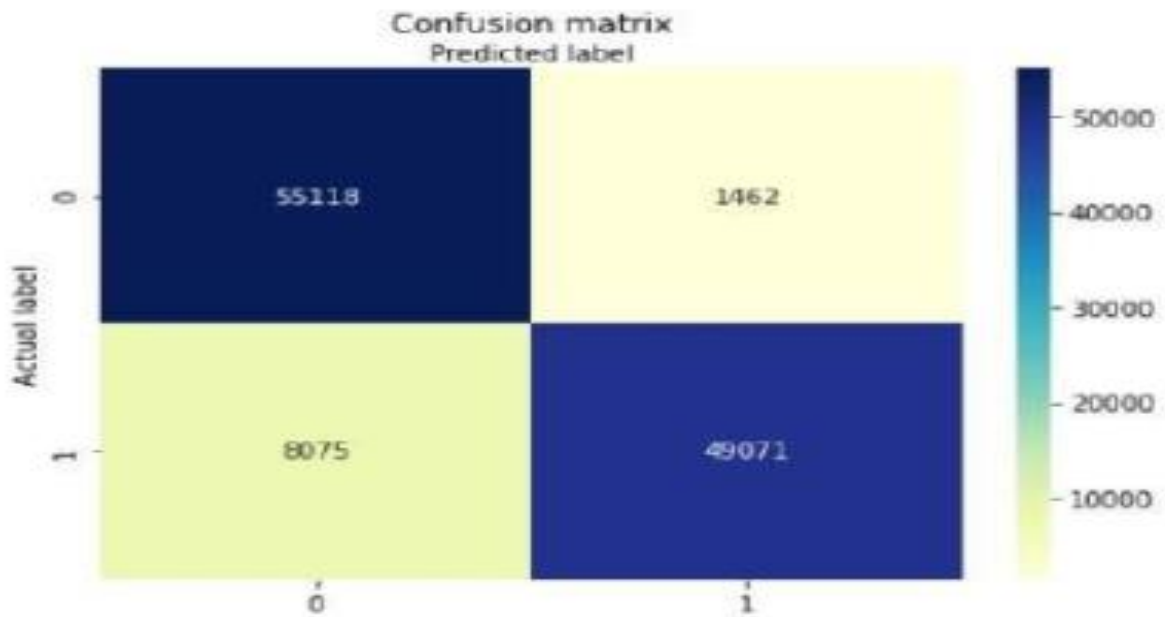


Fig. 11: Confusion matrix for Naive Bayes

3)Confusion matrix for Decision Tree Algorithm:

Fig 12 represents confusion matrix for Decision Tree algorithm. Which contains True Positive, True Negative, False Positive, False Negative. False positive value is lesser which shows fraud not detected cases are low. For Decision Tree algorithm accuracy, recall, precision achieved are 92.88, 98.98, 99.48 respectively.

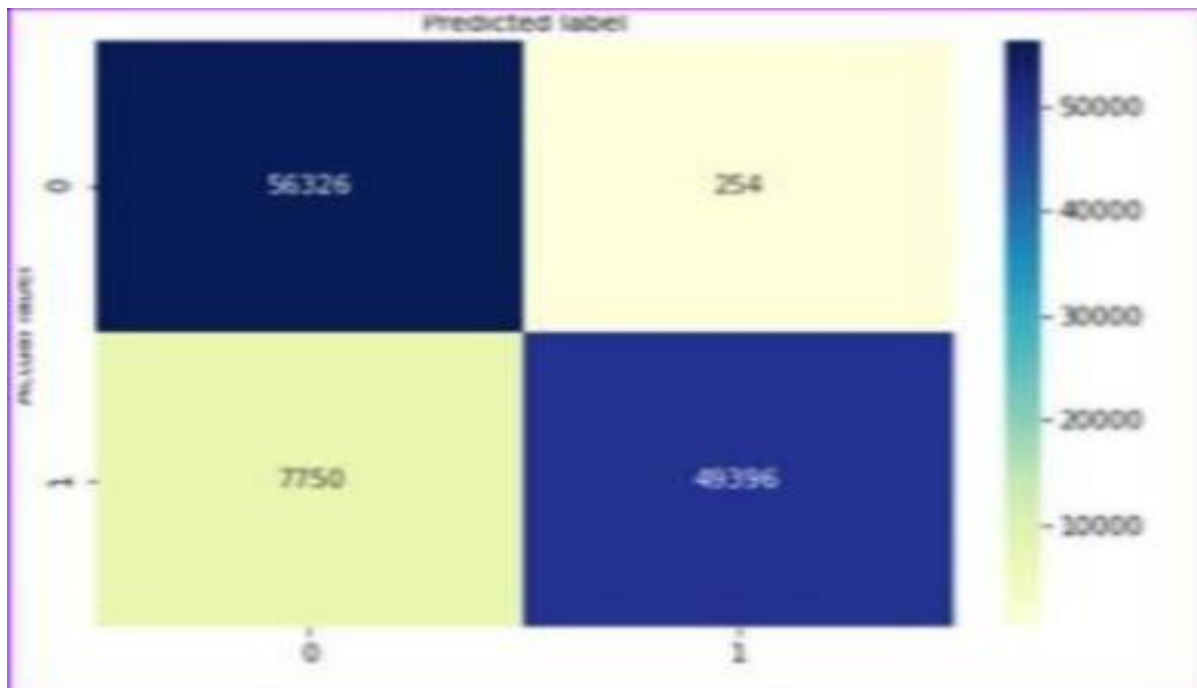


Fig. 12: Confusion matrix for Decision Tree

4) Confusion matrix for ANN model:

Fig 13 represents confusion matrix for ANN model. Which contains True Positive, True Negative, False Positive, False Negative. False positive value is lesser which shows fraud not detected cases are low. For ANN model algorithm accuracy, recall, precision achieved are 98.69, 98.98, 98.41 respectively.

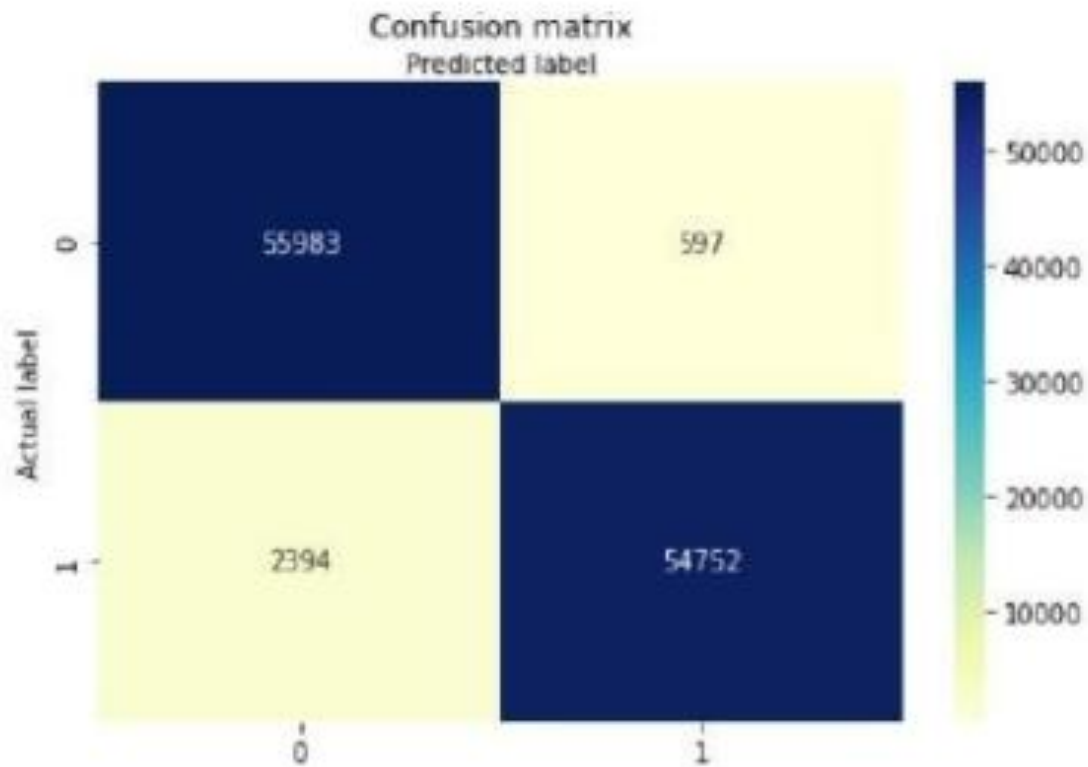


Fig. 13: Confusion matrix for ANN

Comparison of algorithms:

Table 3 represents the comparison table made using results obtained using simulation. Factors compared are accuracy, precision, recall. From table we can conclude that ANN model as best accuracy, precision and recall.

Achievement of accuracy is done using different algorithms and Ann model gives the best accuracy. confusion matrix gives visualization of results in the form table and minimum false positive rate is seen in all algorithms which is required results to achieve the objective. finally by providing the numerical data fraud or nonfraud detected using basic user interface design.

	Accuracy	Precision	Recall
Logistic Regression	94.84	97.58	92.00
Naive Bayes	91.62	97.09	84.82
Decision Tree	92.88	99.48	86.34
ANN model	98.69	98.41	98.98

Table 3: Accuracy, precision, recall comparison table for different ML algorithms

Chapter 5: Conclusion and future work

Credit card fraud is most common problem resulting in loss of lot money for peoples and loss for some banks and credit card company. This project want to help the peoples from their wealth loss and also for the banked company and fraud and fraud less transaction by using the time and amount using some machine learning algorithms such as logistic regression, decision tree, support vector machine, this all are supervised machine learning algorithm in machine learning.

In feature solving this problem statement using another part of project we used both and time and amount feature mainly for predicting the weather the transaction is fraud or Nonfraud transaction, in time series analysis we can reduce the number of parameters that is feature required for the model and we can achieve this model by using average method ,moving average or window method, naive method and sessional naive methods but all this method have some advantages and disadvantages.

FUTURE WORK

While we couldn't reach out goal of 100% accuracy in fraud detection, we did end up creating a system that can, with enough time and data, get very close to that goal. As with any such project, there is some room for improvement here. The very nature of this project allows for multiple algorithms to be integrated together as modules and their results can be combined to increase the accuracy of the final result. This model can further be improved with the addition of more algorithms into it. However, the output of these algorithms needs to be in the same format as the others. Once that condition is satisfied, the modules are easy to add as done in the code. This provides a great degree of modularity and versatility to the project.

More room for improvement can be found in the dataset. As demonstrated before, the precision of the algorithms increases when the size of dataset is increased. Hence, more data will surely make the model more accurate in detecting frauds and reduce the number of false positives. However, this requires official support from the banks themselves.

Credit Card Fraud Detection Using Supervised Machine Learning

SCREENSHOTS

```
import numpy as np
import pandas as pd
import os
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split

from sklearn.metrics import classification_report, accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
```

+ Code + Markdown Python

```
df = pd.read_csv(r"C:\Users\91705\Fraud_Detection\creditcard.csv")
df
```

Python

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	
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	-0.189115	0.
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	0.125895	-0.008983
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	-0.139097	-0.055353
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	-0.221929	0.062723
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	0.502292	0.219422
...
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.213454	0.111864	1.014480	-0.509348	1.436807	0.250034	0.943651
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.214205	0.924384	0.012463	-1.016226	-0.606624	-0.395255	0.068472

```
df.shape
```

(284807, 31)

```
df.columns
```

Python

```
Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10', 'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20', 'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'V29', 'V30', 'V31', 'Amount'], dtype='object')
```

Credit Card Fraud Detection Using Supervised Machine Learning

```
... Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
        'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',
        'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount',
        'Class'],
        dtype='object')

df.info()

... Output exceeds the size limit. Open the full output data in a text editor
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
#   column  Non-Null count  Dtype
---  ---
0    Time      284807 non-null  float64
1    V1        284807 non-null  float64
2    V2        284807 non-null  float64
3    V3        284807 non-null  float64
4    V4        284807 non-null  float64
5    V5        284807 non-null  float64
6    V6        284807 non-null  float64
7    V7        284807 non-null  float64
8    V8        284807 non-null  float64
9    V9        284807 non-null  float64
10   V10       284807 non-null  float64
11   V11       284807 non-null  float64
12   V12       284807 non-null  float64
13   V13       284807 non-null  float64
14   V14       284807 non-null  float64
15   V15       284807 non-null  float64
16   V16       284807 non-null  float64
17   V17       284807 non-null  float64
18   V18       284807 non-null  float64
19   V19       284807 non-null  float64
20   V20       284807 non-null  float64
21   V21       284807 non-null  float64
22   V22       284807 non-null  float64
23   V23       284807 non-null  float64
24   V24       284807 non-null  float64
25   V25       284807 non-null  float64
26   V26       284807 non-null  float64
27   V27       284807 non-null  float64
28   V28       284807 non-null  float64
29   Amount    284807 non-null  float64
30   Class     284807 non-null  int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

```
df.isna().sum()

Time      0
V1        0
V2        0
V3        0
V4        0
V5        0
V6        0
V7        0
V8        0
V9        0
V10       0
V11       0
V12       0
V13       0
V14       0
V15       0
V16       0
V17       0
V18       0
V19       0
V20       0
V21       0
V22       0
V23       0
V24       0
...
V26       0
```

Credit Card Fraud Detection Using Supervised Machine Learning

```
...
V20      0
V21      0
V22      0
V23      0
V24      0
...
V26      0
V27      0
V28      0
Amount    0
Class     0
dtype: int64

fraud = df.Class
type(fraud)

... pandas.core.series.Series

>
fraud_count = 0
for i in range(1,len(fraud)):
    if fraud[i] == 1:
        fraud_count += 1

fraud_count

... 492
```

```
fraud_count

... 492

genuine_trans = 0
for i in range(1,len(fraud)):
    if fraud[i] == 0:
        genuine_trans += 1

genuine_trans

... 284314

# Correlation matrix
corr_matrix = df.corr()
sns.heatmap(corr_matrix)

... <AxesSubplot:~>

</>
```

```
>
# correlation matrix
corr_matrix = df.corr()
sns.heatmap(corr_matrix)

... <AxesSubplot:~>

</>

...

X = df.drop(["Class"], axis = 1)
```

Credit Card Fraud Detection Using Supervised Machine Learning

```
X = df.drop(["Class"], axis = 1)

y = df["Class"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state = 42)

print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)

... (213605, 30)
(71202, 30)
(213605,)
(71202,)
```

```
logisticRegr = LogisticRegression()

logisticRegr.fit(X_train, y_train)
```

```
logisticRegr.fit(X_train, y_train)

... c:\Python39\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
  https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
  https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
n_iter_i = _check_optimize_result(
LogisticRegression())

y_pred = logisticRegr.predict(X_test)
accuracy = logisticRegr.score(X_test, y_test)
print(y_pred)
print(accuracy)

... [1 0 0 ... 0 0 0]
0.9986798123648212
```

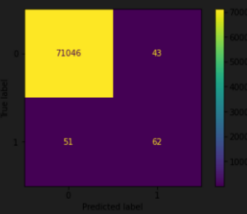
```
from sklearn.metrics import plot_confusion_matrix

conf_mat = metrics.confusion_matrix(y_test, y_pred)
print(conf_mat)
plot_confusion_matrix(logisticRegr, X_test, y_test)
```

```
conf_mat = metrics.confusion_matrix(y_test, y_pred)
print(conf_mat)
plot_confusion_matrix(logisticRegr, X_test, y_test)
plt.show()

... [[71046  43]
 [ 51  62]]

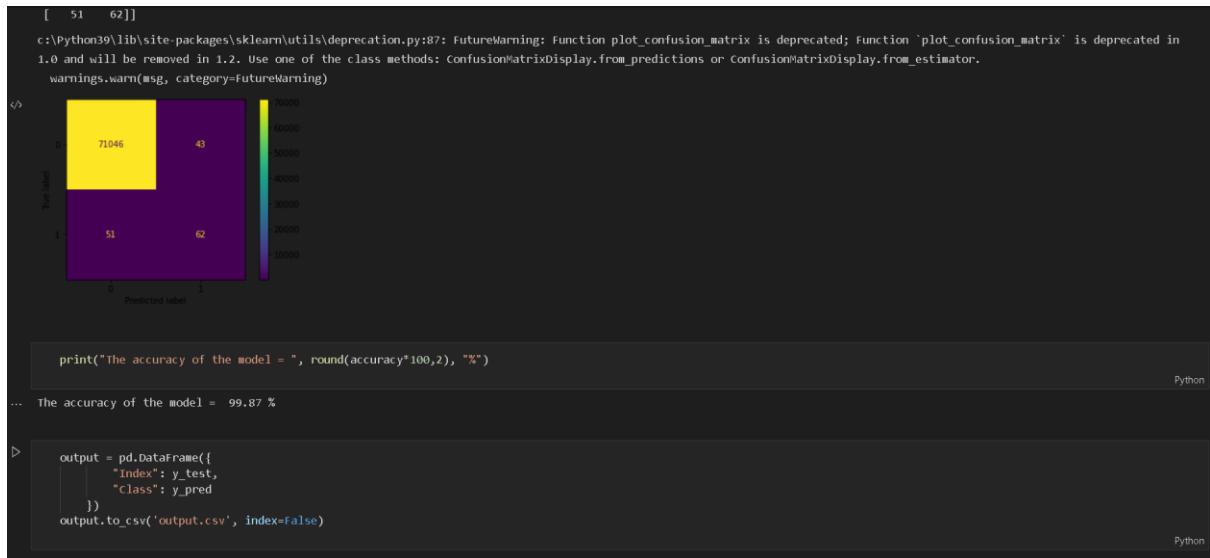
c:\Python39\lib\site-packages\sklearn\utils\deprecation.py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function 'plot_confusion_matrix' is deprecated in
1.0 and will be removed in 1.2. Use one of the class methods: confusion_matrix_display.from_predictions or confusion_matrix_display.from_estimator.
warnings.warn(msg, category=FutureWarning)
```



```
print("The accuracy of the model = ", round(accuracy*100,2), "%")

... The accuracy of the model = 99.87 %
```

Credit Card Fraud Detection Using Supervised Machine Learning



REFERENCES

1. Maniraj SP, Saini A, Ahmed S, Sarkar D. Credit card fraud detection using machine learning and data science. *Int J Eng Res* 2019; 8(09).
2. Dornadula VN, Geetha S. Credit card fraud detection using machine learning algorithms. *Proc Comput Sci*. 2019;165:631–41.
3. Thennakoon, Anuruddha, et al. Real-time credit card fraud detection using machine learning. In: 2019 9th international conference on cloud computing, data science & engineering (Confluence). IEEE; 2019.
4. Robles-Velasco A, Cortés P, Muñuzuri J, Onieva L. Prediction of pipe failures in water supply networks using logistic regression and support vector classification. *Reliab Eng Syst Saf*. 2020;196:106754.
5. Liang J, Qin Z, Xiao S, Ou L, Lin X. Efficient and secure decision tree classification for cloud-assisted online diagnosis services. *IEEE Trans Dependable Secure Comput*. 2019;18(4):1632–44.
6. Ghiasi MM, Zendehboudi S. Application of decision tree-based ensemble learning in the classification of breast cancer. *Comput in Biology and Medicine*. 2021;128:104089.
7. Lingjun H, Levine RA, Fan J, Beemer J, Stronach J. Random forest as a predictive analytics alternative to regression in institutional research. *Pract Assess Res Eval*. 2020;23(1):1.
8. Breiman L. Random forests. *Mach Learn*. 2001;45(1):5–32.
9. Ning B, Junwei W, Feng H. Spam message classification based on the Naive Bayes classification algorithm. *IAENG Int J Comput Sci*. 2019;46(1):46–53.
10. Katare D, El-Sharkawy M. Embedded system enabled vehicle collision detection: an ANN classifier. In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC); 2019. p. 0284–0289.
11. Campus K. Credit card fraud detection using machine learning models and collating machine learning models. *Int J Pure Appl Math*. 2018;118(20):825–38.
12. Varmedja D, Karanovic M, Sladojevic S, Arsenovic M, Anderla A. Credit card fraud detection-machine learning methods. In: 18th international symposium INFOTEH-JAHORINA (INFOTEH); 2019. p. 1-5.
13. Khatri S, Arora A, Agrawal AP. Supervised machine learning algorithms for credit card fraud detection: a comparison. In: 10th international conference on cloud computing, data science & engineering (Confluence); 2020. p. 680-683.
14. Awoyemi JO, Adetunmbi AO, Oluwadare SA. Credit card fraud detection using machine learning techniques: a comparative analysis. In: International conference on computer networks and Information (ICCNI); 2017. p. 1-9.

15. Seera M, Lim CP, Kumar A, Dhamotharan L, Tan KH. An intelligent payment card fraud detection system. *Ann Oper Res* 2021;1–23.
16. Guo S, Liu Y, Chen R, Sun X, Wang X. X, Improved SMOTE algorithm to deal with imbalanced activity classes in smart homes. *Neural Process Lett*. 2019;50(2):1503–26.
17. The Credit card fraud [Online]. <https://www.kaggle.com/mlg-ulb/creditcardfraud>
18. Kasongo SM. An advanced intrusion detection system for IIoT based on GA and tree based algorithms. *IEEE Access*. 2021;9:113199–212.
19. Mienye ID, Sun Y. Improved heart disease prediction using particle swarm optimization based stacked sparse autoencoder. *Electronics*. 2021;10(19):2347.
20. Hemavathi D, Srimathi H. Effective feature selection technique in an integrated environment using enhanced principal component analysis. *J Ambient Intell Hum Comput*. 2021;12(3):3679–88.
21. Pouramirarsalani A, Khalilian M, Nikravanshalmani A. Fraud detection in E-banking by using the hybrid feature selection and evolutionary algorithms. *Int J Comput Sci Netw Secur*. 2017;17(8):271–9.
22. Saheed YK, Hambali MA, Arowolo MO, Olasupo YA. Application of GA feature selection on Naive Bayes, random forest and SVM for credit card fraud detection. In: 2020 international conference on decision aid sciences and application (DASA); 2020. p. 1091–1097.
23. Davis L. *Handbook of genetic algorithms*; 1991.
24. Li Y, Jia M, Han X, Bai XS. Towards a comprehensive optimization of engine efficiency and emissions by coupling artificial neural network (ANN) with genetic algorithm (GA). *Energy*. 2021;225:120331.
25. Khalilia M, Chakraborty S, Popescu M. Predicting disease risks from highly imbalanced data using random forest. *BMC Med Inf Decis Mak*. 2011;11(1):1–13.
26. Abhishek L. Optical character recognition using ensemble of SVM, MLP and extra trees classifier. In: International conference for emerging technology (INCET) IEEE; 2020. p. 1–4.
27. Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H. Xgboost: extreme gradient boosting. *R package version 04-2*. 2015;1(4):1–4.
28. Harik GR, Lobo FG, Goldberg DE. The compact genetic algorithm. *IEEE Trans Evol Comput*. 1999;3(4):287–97.
29. Jain A, Nandakumar K, Ross A. Score normalization in multimodal biometric systems. *Pattern Recognit*. 2005;38(12):2270–85.
30. Kasongo SM, Sun Y. A deep long short-term memory based classifier for wireless intrusion detection system. *ICT Express*. 2020;6(2):98–103.
31. Norton M, Uryasev S. Maximization of auc and buffered auc in binary classification. *Math Program*. 2019;174(1):575–612.
32. Google Colab [Online]. Available: <https://colab.research.google.com/>

33. Scikit-learn : machine learning in Python [Online]. <https://scikit-learn.org/stable/>
 34. Altman ER. Synthesizing credit card transactions. 2019. arXiv preprint [arXiv:1910.03033](https://arxiv.org/abs/1910.03033)
-

BIOGRAPHY

Piyush Kumar

BE 2nd year,

Dept. of Computer Science
& Engineering

Shashank Singh

BE 2nd year,

Dept. of Computer Science
& Engineering

Muskan Kumari

BE 2nd year,

Dept. of Computer Science
& Engineering

Rithik Sharma

BE 2nd year,

Dept. of Computer Science
& Engineering

Akash Kumar

BE 2nd year,

Dept. of Computer Science
& Engineering