BMS COLLEGE OF ENGINEERING

(Autonomous College under VTU)
Bull Temple Road, Basavanagudi, Bangalore - 560019

A Project report on

"Credit Card Fraud Detection"

Submitted in partial fulfillment of the requirements for the award of degree

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

By

Rajath K - 1BM19IS125
Rohith R Kashyap - 1BM19IS131
Roopesh Reddy C - 1BM19IS132
Sacheth B M-1BM19IS135

Under the guidance of
Ms Shobana T S - Assistant Professor

Department of Information Science and Engineering

2021 - 22

BMS COLLEGE OF ENGINEERING

(Autonomous College under VTU)

Bull Temple Road, Basavanagudi, Bangalore - 560019

Department of Information Science and Engineering

CERTIFICATE

This is to certify that the project entitled "***Credit Card Fraud Detection***" is a bona-fide work carried out by **Rajath K (1BM19IS125), Rohith R Kashyap (1BM19IS131), Roopesh Reddy C (1BM19IS132) & Sacheth B M (1BM19IS135)** in partial fulfillment for the award of degree of Bachelor of Engineering in Information Science and Engineering from Visvesvaraya Technological University, Belgaum during the year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering Degree.

Signature of the Faculty
Name and Designation

Signature of the HOD
Name and Designation

Table of Contents

SL.NO	Contents	Page No.
1.	Abstract	4
2.	Introduction	5
3.	Problem Statement	6
4.	Literature Survey	6-9
5.	System Requirements Specification	10
6.	System Design	10
7.	Implementation	11-15
8.	Test Results	16
9.	Conclusion	16-17
10.	References	18

ABSTRACT

Financial fraud is a growing problem with long term consequences in the financial industry and while many techniques have been discovered to solve this problem faced by various companies, data mining has been successfully applied to finance databases to automate analysis of huge volumes of complex data. Data mining has also played a salient role in the detection of credit card fraud in online transactions.

Fraud detection in credit cards is a data mining problem. It becomes challenging due to two major reasons – first, the profiles of normal and fraudulent behaviour change frequently and second, the credit card fraud data sets are highly skewed. This paper investigates and checks the performance of Decision Tree, Random Forest,SVM,K-Means and Logistic Regression on highly skewed credit card fraud data. Dataset of credit card transactions is sourced from European cardholders containing 284,786 transactions. These techniques are applied on the raw and pre-processed data. The performance of the techniques is evaluated based on accuracy, sensitivity, specificity, precision. The results indicating the optimal accuracy for Logistic Regression, Decision Tree, Random Forest, K-Means and SVM classifiers are 94.4%, 91.9%, 92.9%, 93.9%, 93.4% and 94.95% respectively

INTRODUCTION

As we are moving towards the digital world — cybersecurity is becoming a crucial part of our life. When we talk about security in digital life then the main challenge is to find the abnormal activity.

When we make any transaction while purchasing any product online — a good amount of people prefer credit cards. The credit limit in credit cards sometimes helps us make purchases even if we don't have the amount at that time. but, on the other hand, these features are misused by cyber attackers.

To tackle this problem we need a system that can abort the transaction if it finds fishy.

Here, comes the need for a system that can track the pattern of all the transactions and if any pattern is abnormal then the transaction should be aborted.

Today, we have many machine learning algorithms that can help us classify abnormal transactions. The only requirement is the past data and the suitable algorithm that can fit our data in a better form.

	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	2.0	-0.425966	0.960523	1.141109	-0.168252	0.420987	-0.029728	0.476201	0.260314	-0.568671	...	-0.208254	-0.559825	-0.026398	-0.371427	-0.23279
6	4.0	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.005159	0.081213	0.464960	...	-0.167716	-0.270710	-0.154104	-0.780055	0.75013
7	7.0	-0.644269	1.417964	1.074380	-0.492199	0.948934	0.428118	1.120631	-3.807864	0.615375	...	1.943465	-1.015455	0.057504	-0.649709	-0.41526
8	7.0	-0.894286	0.286157	-0.113192	-0.271526	2.669599	3.721818	0.370145	0.851084	-0.392048	...	-0.073425	-0.268092	-0.204233	1.011592	0.37320
9	9.0	-0.338262	1.119593	1.044367	-0.222187	0.499361	-0.246761	0.651583	0.069539	-0.736727	...	-0.246914	-0.633753	-0.120794	-0.385050	-0.06973

10 rows × 31 columns

DATASET (real bank transactions made by European cardholders in the year 2013)

PROBLEM STATEMENT

Are the existing techniques used for detecting credit card frauds correctly and accurately providing efficient results or can it be improved using Machine Learning?Thus, our project tries to predict credit-card frauds using Machine Learning as it is believed to provide better results as compared to the existing techniques used to detect these frauds.

LITERATURE SURVEY

Random Forest Classifiers :A Survey and Future Research Directions

The intention of this paper was to present a review of current work related to Random Forest classifier and identify future research directions in the field of Random Forest classifier. Random Forest classifier is an ensemble technique and hence is more accurate, but it is time consuming compared to other individual classification techniques. We mainly tried to review the work done for accuracy improvement and performance improvement of Random Forest. As a result of our survey, we have presented Taxonomy of Random Forest algorithm and

performed analysis of various algorithms / techniques based on Random Forest algorithm. This analysis which is presented as Comparison chart will serve as a guideline for pursuing future research related to Random forest classifier.

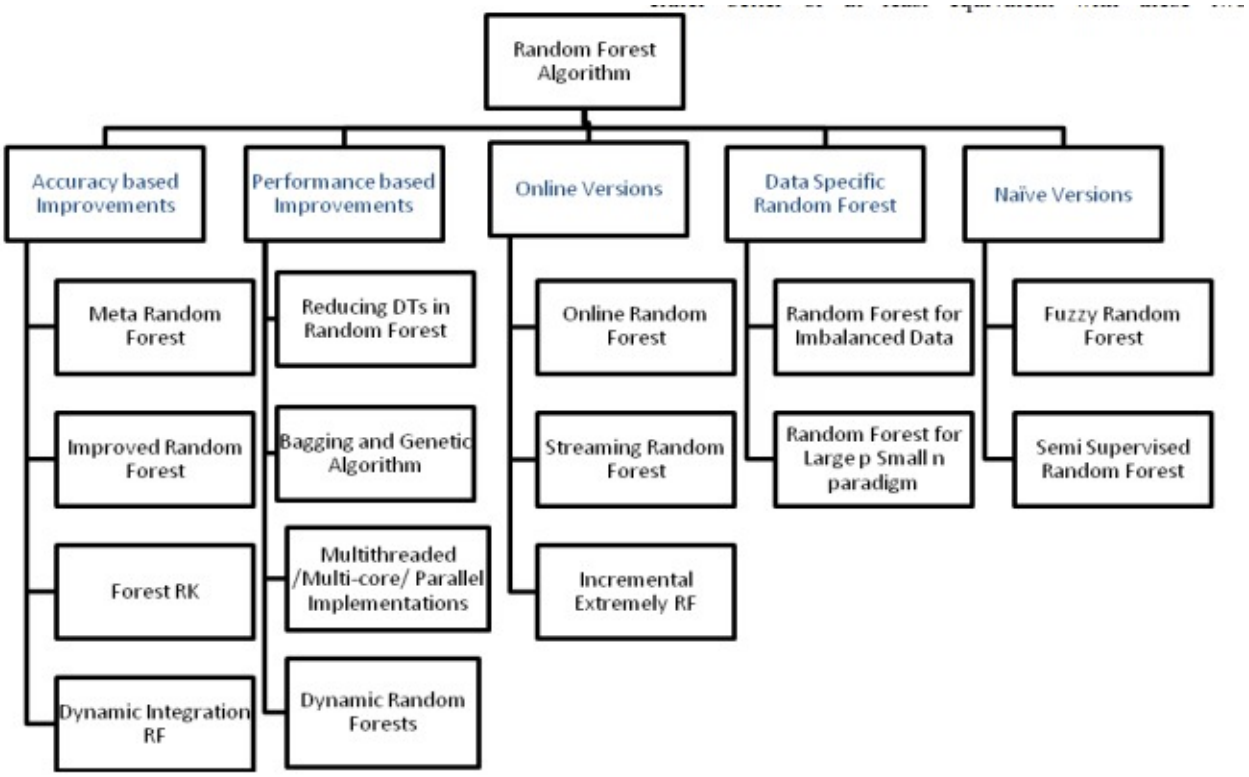


Fig.1 Taxonomy of Random Forest Classifier

A Survey on Decision Tree Algorithm For Classification

According to our observations, the performance of the algorithms strongly depends on the entropy, information gain and the features of the data sets. Various work has been done using the Decision tree Algorithm. But they all are like static in nature. Some recently improved algorithms reduce problems like replication, handle continuous data, and are biased to multi value attributes.

This paper provides students and researchers some basic fundamental information about decision tree, tools and recent issues.

Table 1 Comparisons between different Decision Tree Algorithm

	ID3	C4.5	C5.0	CART
Type of data	Categorical	Continuous and Categorical	Continuous and Categorical, dates, times, timestamps	continuous and nominal attributes data
Speed	Low	Faster than ID3	Highest	Average
Pruning	No	Pre-pruning	Pre-pruning	Post pruning
Boosting	Not supported	Not supported	Supported	Supported
Missing Values	Can't deal with	Can't deal with	Can deal with	Can deal with
Formula	Use information entropy and information Gain	Use split info and gain ratio	Same as C4.5	Use Gini diversity index

Analysis And Study Of K-Means Clustering Algorithm

In this paper we presented an algorithm for performing K-means clustering. Our experimental result demonstrated that our scheme can improve the direct K-means algorithm. This paper also explains the time complexity of K-means and our proposed algorithm. There are several improvements possible to the basic strategy presented in this paper. One approach will be to use the concept of Nearest Neighbor Clustering Algorithm to improve the compactness of clusters.

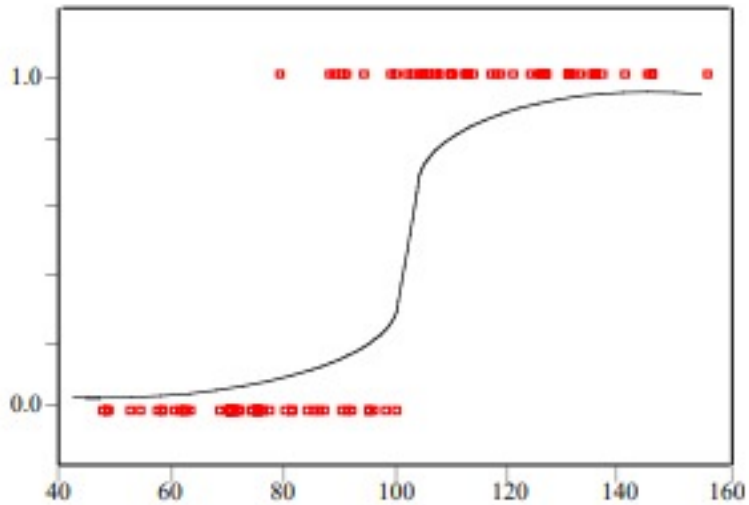
Table1 Comparison of algorithm's running time

Name of algorithm	Worst case	Average case	Best case
k-means	$O(n^i)$ where $2<=i<3$	$O(n^2)$	$O(n)$
Proposed Algorithm	$O(n^2)$	$O(n^i)$ where $1<=i<=2$	$O(n)$

An Introduction to Logistic Regression Analysis and Reporting

In this paper, we demonstrate that logistic regression can be a powerful analytical technique for use when the outcome variable is dichotomous. The effectiveness of the logistic model was shown to be supported by (a) significance tests of the model against the null model, (b) the significance test of each predictor, (c) descriptive and inferential goodness-of-fit indices, (d) and predicted probabilities. During the last decade, logistic regression has been gaining popularity. The trend is evident in the JER and higher education journals. Such popularity can be attributed to researchers' easy access to sophisticated statistical software that performs comprehensive analyses of this technique. It is anticipated that the application of the logistic regression technique is likely to increase. This potential expanded usage demands that researchers, editors, and readers be coached in what to expect from an article that uses the logistic regression technique. What tables, charts, or figures should be included? What assumptions should be verified? And how comprehensive should the presentation of logistic regression results be? It is hoped that this article has answered these questions with an illustration of logistic regression applied to a data set and with guidelines and recommendations offered on a preferred pattern of application of logistic methods.

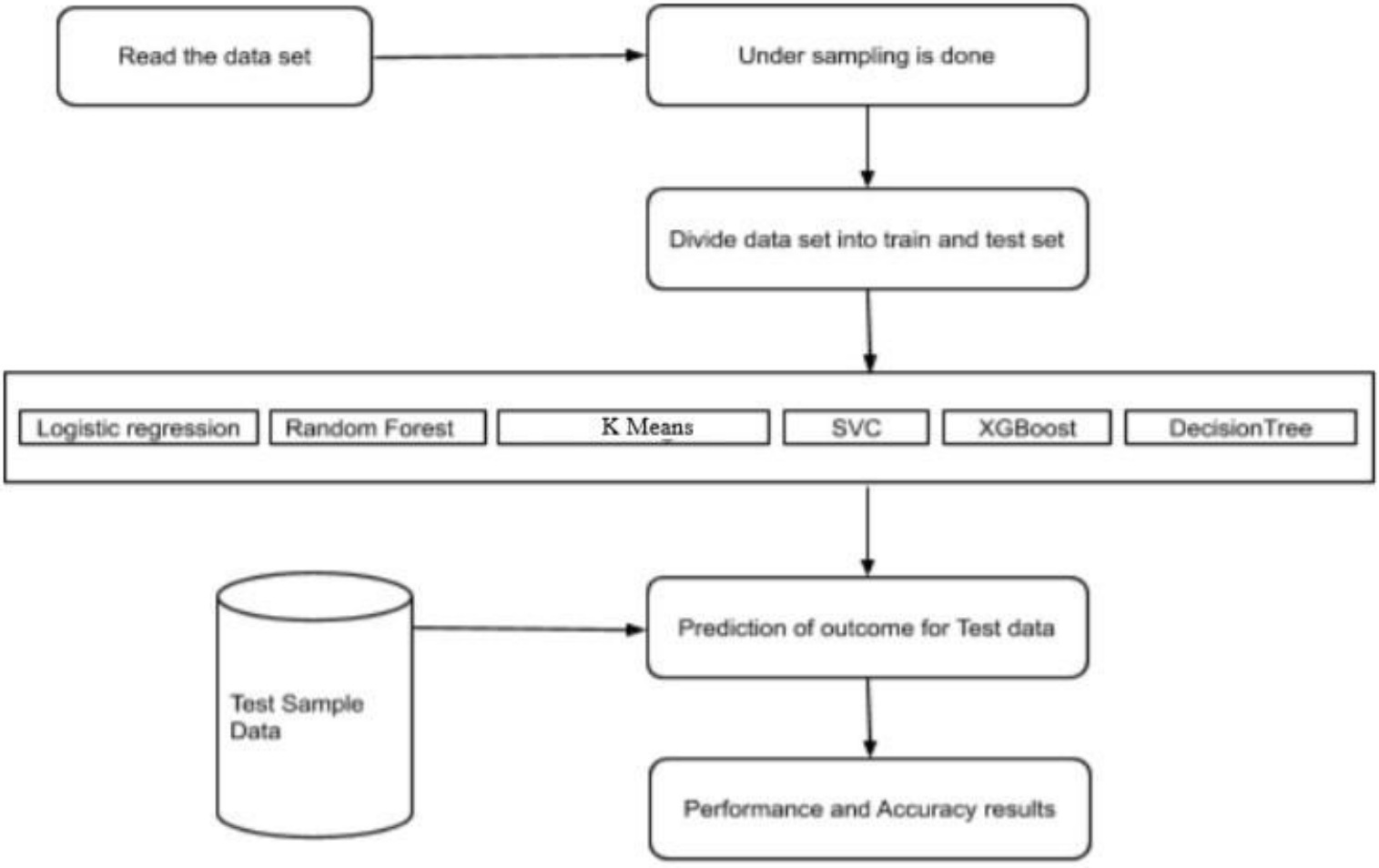
Figure 1. Relationship of a Dichotomous Outcome Variable, *Y* (1 = Remedial Reading Recommended, 0 = Remedial Reading Not Recommended) With a Continuous Predictor, Reading Scores



SYSTEM REQUIREMENT SPECIFICATION

- Export Graphviz
- Back End Software : Anaconda, Jupyter notebook.
- Matplotlib

SYSTEM DESIGN



Algorithm steps:	
Step 1:	Read the Dataset.
Step 2:	Random Sampling is done on the data set to make it balanced.
Step 3:	Divide the dataset into two parts i.e., Train dataset and Test dataset.
Step 4:	Accuracy and performance metrics has been calculated to know the efficiency for different algorithms.
Step 5:	Then retrieve the best algorithm based on efficiency for the given dataset.

IMPLEMENTATION

Importing Necessary Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.metrics import f1_score, accuracy_score, confusion_matrix
```

Importing Dataset

```
from google.colab import drive
drive.mount('/content/drive')
```

```
data=pd.read_csv("/content/drive/MyDrive/creditcard.csv")
```

Data Processing & Understanding

```
Total_transactions = len(data)
normal = len(data[data.Class == 0])
fraudulent = len(data[data.Class == 1])
fraud_percentage = round(fraudulent/normal*100, 2)
print('Number of Transactions are ', Total_transactions)
print('Number of Normal Transactions are ', normal)
print('Number of fraudulent Transactions are ',fraudulent)
print('Percentage of fraud Transactions is ',fraud_percentage)
```

```
data.info()
```

```
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
amount = data['Amount'].values
data['Amount'] = sc.fit_transform(amount.reshape(-1, 1))
```

```
data.drop(['Time'], axis=1, inplace=True)
```

```
data.drop_duplicates(inplace=True)
```

```
X = data.iloc[:, :-1].values
y = data['Class'].values
```

Train & Test Split

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 1)
```

Model Building

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor
from sklearn.tree import plot_tree

DT = DecisionTreeClassifier(max_depth = 4)
DT.fit(X_train, y_train)
dt_pred = DT.predict(X_test)
```

```
print('Accuracy score of the Decision Tree model is ',accuracy_score(y_test, dt_pred))
print('F1 score of the Decision Tree model is ',f1_score(y_test, dt_pred))
plot_tree(DT)
confusion_matrix(y_test, dt_pred)
```

Logistic Regression

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()
lr.fit(X_train, y_train)
lr_pred = lr.predict(X_test)
```

```
print('Accuracy score of the Logistic Regression model is ',
accuracy_score(y_test, lr_pred))
print('F1 score of the Logistic Regression model is ',f1_score(y_test,
lr_pred))
```

Random Forest

```
from sklearn.ensemble import RandomForestClassifier,RandomForestRegressor

rf = RandomForestClassifier(max_depth = 4)
rf.fit(X_train, y_train)
rf_pred = rf.predict(X_test)
```

```
print('Accuracy score of the Random Forest model is
',accuracy_score(y_test, rf_pred))
print('F1 score of the Random Forest model is ',f1_score(y_test, rf_pred))
```

SVC

```
from sklearn.svm import SVC

svm = SVC()
svm.fit(X_train, y_train)
svm_pred = svm.predict(X_test)
```

```
print('Accuracy score of the Support Vector Machines model is ',
accuracy_score(y_test, svm_pred))
print('F1 score of the Support Vector Machines model is ',
f1_score(y_test, svm_pred))
```

K Means

```
from sklearn.neighbors import KNeighborsClassifier

k_range = range(1,11)
scores={}
scores_list=[]
for k in k_range:
  knn = KNeighborsClassifier(n_neighbors = k)
  knn.fit(X_train, y_train)
  knn_pred = knn.predict(X_test)
  scores[k]=accuracy_score(y_test, knn_pred)
  scores_list.append(accuracy_score(y_test, knn_pred))

plt.plot(k_range,scores_list)
plt.xlabel("K value for knn")
plt.ylabel("Testing accuracy")
plt.show()
print(scores)
```

```
from sklearn.neighbors import KNeighborsClassifier
n = 8
KNN = KNeighborsClassifier(n_neighbors = n)
KNN.fit(X_train, y_train)
knn_pred = KNN.predict(X_test)
```

```
print('Accuracy score of the K-Nearest Neighbors model is
',accuracy_score(y_test, knn_pred))
print('F1 score of the K-Nearest Neighbors model is ', f1_score(y_test,
knn_pred))
```

TEST RESULTS

SL.NO	TRAINING MODEL	ACCURACY SCORE	F1 SCORE
1.	Decision Tree	99.91	75.12
2.	Logistic Regression	99.89	66.67
3.	Random Forest Classifier	99.91	72.22
4.	Support Vector Classifier	99.93	78.14
5.	K Means	99.93	78.48

CONCLUSION

Although there are several fraud detection techniques available today, none is able to detect all frauds completely when they are actually happening, they usually detect it after the fraud has been committed. This happens because a very minuscule number of transactions from the total transactions are actually fraudulent in nature. So we need a technology that can detect the fraudulent transaction when it is taking place so that it can be stopped then and there and that too in a minimum cost. So the major task of today is to build an accurate, precise and fast detecting fraud detection system for credit card frauds that can detect not only frauds happening over the internet like phishing and site cloning but also

tampering with the credit card itself i.e. it signals an alarm when the tampered credit card is being used. The major drawback of all the techniques is that they are not guaranteed to give the same results in all environments. They give better results with a particular type of dataset and poor or unsatisfactory results with other type. Thus, the results are purely dependent on the dataset type used. In our undersample data, our model is unable to detect for a large number of cases the non fraud transactions correctly and instead, mis-classifies those non fraud transactions as fraud cases. Imagine that people that were making regular purchases got their card blocked due to the reason that our model classified that transaction as a fraud transaction, this will be a huge disadvantage for the financial institution. The number of customer complaints and customer dissatisfaction will increase. The next step of this analysis will be to do an outlier removal on our undersample dataset and see if our accuracy in the test set improves. In this paper, Machine learning techniques like Logistic regression, Decision Tree, Random forest, K-Means and SVC were used to detect fraud in the credit card system. Sensitivity, Specificity, accuracy and error rate are used to evaluate the performance for the proposed system. From the experiments, the result that has been concluded is that Logistic regression has an accuracy of 99.89% while SVC shows accuracy of 99.93% and Decision tree shows accuracy of 99.91% and Random forest shows accuracy of 99.91% but the best results are obtained by KNN with a precise accuracy of 99.93%. However when the learning curves of all the classifiers are evaluated we see that Random forest and decision tree overfit and only KNN is able to learn whereas others underfit. Hence we conclude that KNN is the best model for our system.

REFERENCES

1. Random Forest Classifiers :A Survey and Future Research Directions
2. Analysis And Study Of K-Means Clustering Algorithm
3. A Survey on Decision Tree Algorithm For Classification
4. An Introduction to Logistic Regression Analysis and Reporting