ISSN: 0974-5823 Vol. 7 No. 1 January, 2022

International Journal of Mechanical Engineering

DECEPTION FINDING IN CREDIT CARD WITH DETECTION TECHNIQUES

Dr. Jayasri Kotti[1], Srinivas Adapa[2], B. Padmaja[3], A. Maheswara Rao[4]

[1] Professor, CSE Dept., Vignan's Institute of Engineering for women (VIEW) ,K.J.peta, VSEZ (Post), Visakhapatnam-530049, AP, INDIA

[2]Assistant Professor, CSE Dept., Vignan's Institute of Engineering for women (VIEW) ,K.J.peta, VSEZ (Post), Visakhapatnam-530049, AP, INDIA

[3]Assistant Professor, CSE Dept., Vignan's Institute of Engineering for women (VIEW) ,K.J.peta, VSEZ (Post), Visakhapatnam-530049, AP, INDIA

[4]Assistant Professor, CSE Dept., Vignan's Institute of Engineering for women (VIEW) ,K.J.peta, VSEZ (Post), Visakhapatnam-530049, AP, INDIA

ABSTRACT

Financial scam is an ever-growing threat with far significances in the economic industry. Online transactions are increased radically noteworthy and number of online transactions are done with credit cards. Deception finding in credit card occurs regularly and leads to enormous economic losses. So, banks and other monetary establishments care the progress of credit card deception applications. Fake dealings can occur in dissimilar ways and they can be placed into numerous types. Due to the growth of fake dealings, there is a necessity to find the efficient deception finding model. The best method to notice this kind of deception is to examine the spending patterns on every card and to figure out any discrepancy with respect to the usual spending patterns. In order to identify deception finding in credit cards, here we adopted an optimized Support Vector Machine model along with Random Forest and k nearest neighbor algorithms. These algorithms combinedly will give accurate result. The fake dealings are mixed up with genuine dealings and the simple recognition techniques which include comparison of both the deception and the genuine data are never sufficient to detect the deception dealings precisely. This work aims to exemplify the modelling of data using Machine Learning with deception finding of credit cards dealings which has happened prior with the data of deception dealings. Proposed model will regulate whether a new transaction tends to be deception or genuine. We have an objective to detect 100% of the deception dealings while reducing illegal deception classifications.

Keywords: Credit Card Deception, Fake dealings, Support Vector Machine, Random Forest

1. INTRODUCTION

The expansion of economic industry makes the credit card business become one of the bank's maximum significant profits. But along with these developments, worldwide credit fake dealings increase at shocking percentage. Economic corporations cannot efficiently notice fake dealings; as a significance, the loss is fetching gradually serious. How to recognize the Deception in credit card dealings efficiently, speedily and correctly is becoming the mostly worried problem.

Now a days usage of credit cards is necessary and more convenient. It may be used in regular or in online shopping. In recent years credit cards are used for buying things and amenities with in online or offline transactions for more convenience. In offline mode users physically presents their card for making payment. In this kind of buying an attacker has to steal the credit card for fake communications. But in online mode payment attackers need only little information about card number, secure code and expiration date ect., (only Card details). Monetary scam is an ever-rising threat with far penalties in the economic trade. Identification of fraudulent dealings or transactions is important to credit card companies for their future growth.

In other words, Credit Card Deception can be defined as a situation where an individual impost somebody else's credit card for individual reasons while the card holder and the card issuing companies are unaware of the fact that the card is being used. Deception in credit card dealings is illegal and annoying practice of an account by someone other than the card holder of that account. Essential anticipation events can be taken to stop this misuse and the behavior of such fake performs can be deliberate to minimize it and protect against comparable incidences in the upcoming.

Deception finding in credit cards includes monitoring the actions of people of handlers in order to guess, observe or avoid offensive performance, which consist of deception and disturbance. The act of fraud detection in credit card transactions is greatly affected by the sampling approach on dataset, range of variables and recognition techniques used. Initial research of deception finding in

credit cards emphases on the arrangement and documentation of approaches and representations, including single pattern recognition methods such as decision trees and neural networks, the combination method, distributed data mining. But due to the difficulty and sparse of the transaction data, these tactics are often challenged with the issue of model selection, model parameter settings, improper selection when dealing with large scale transaction data, which frequently lead to owe study, over fitting and the local optimal problem.

The best technique to notice this kind of deception is to examine the spending patterns on every card and to figure out any discrepancy with respect to the "usual" spending patterns. In order to identify deception finding in credit cards, here we proposed an optimized SVM model, Naive Bayes, Random Forest and k-nearest algorithm. These algorithms will give accurate result. SVM is a comparatively new field in the data mining field. The method is first mapped data from the input-space to feature-space, and then construct a linear discriminate function in feature space.

2. LITERATURE REVIEW

Lots of research work is carried out for deception finding in credit cards. Deception is any malicious action that goals to root monetary loss to the other party. Deceptions caused by Credit Cards have charges clients, banks and billions of moneys worldwide. Attackers are unceasingly annoying to discover new ways and tricks to commit deception, even after there are several tools to stop deception. Thus, in order to disintegrate all these deceptions, we need an authoritative deception finding system which not only notices the deception but also senses it before it takes place and in an exact method.

To perceive the credit card deception there are numerous methods which are based on Deep-learning, Logistic Regression, Naive Bayesian, SVM, Neural-network, K-Nearest, Datamining, Decision tree, Genetic algorithm etc.,[1][9]. Credit Card Deception finding is a typical sample of classification. In this process, we have concentrated on analysing and pre-processing data sets as well as the deployment of multiple anomaly detection algorithms such as Local Outlier Factor and Isolation Forest algorithm on the PCA transformed Credit Card Transaction data.[2][7]

The deception finding appearance in credit card dealings is significantly affected by the sampling approach on dataset, selection of variables and detection technique(s) used. This work examines the performance of naive Bayes, k-nearest and logistic regression on highly skewed credit card deception data. Dataset of credit card communications is obtained from European cardholders covering nearly 264,856 dealings [3]. Numerous methods are at hand for a deception finding system such as support vector-machine, Bayesian Network, K-Nearest neighbor, Markov model and decision trees [4][6][11]

Finding of deception contains nursing and analyzing the behavior of dissimilar operators in order to estimate finding unwanted behavior. To efficiently notice credit card deception, we want to distinguish the various advancements, algorithms and types involved in detecting credit card deception. There are numerous algorithms to notice the credit card deception and each have their own advantages and correctness. Some of those algorithms are K-nearest-neighbor, Linear regression, Naive Bayes, Random Forest algorithm etc.,[5][8][12]. Deception finding in credit card has been divided into online fraud and offline fraud. Maximum frauds happened in Online mode only. Online deception may occur via internet, cell phone, shopping in absence or card holder [10].

3. PROPOSED METHODOLOGY

Now-a-days Payments using credit cards have increased. Due to the growth of fake dealings, there is a necessity to treasure the effective deception finding model. The foremost objective of this work is to find the fake dealings of credit card. Here we proposed a data model which adopts optimized Support Vector Machine model, Random Forest and k nearest neighbor algorithms and all together will give accurate result.

Proposed data model takes input dataset as credit card type, transaction id, transaction method, amount, bank, time. By applying random forest algorithm, the classifier that contains decision trees on various subsets of the given dataset and takes average to improve the predictive accuracy of the dataset. After that the SVM algorithm is used to create the boundary that can segregate n-dimensional space into classes so that we can put the new data in the correct category in the future. Finally, the algorithm K-Nearest Neighbor undertakes the comparison among the available data and new data and place the original data in to the group that is most similar to the existing groups. Through this technique we can get the fake transactions and stop them from happening.

Actual working of random forest has two phases, first is to create the forest by combining N decision tree and second is to make predictions for each tree, that was created in first phase. Working process can be explained in following septs and figure 1.

- Step 1: select random K data points from the training data
- Step 2: Construct the decision tree associated with the selected subsets
- Step 3: Select the number N for decision trees that we want to build
- Step 4: Repeat step1 and step 2

- Step5: Find the predictions of each decision tree and assign the new data points to the category that wins the maximum polls for new data points.

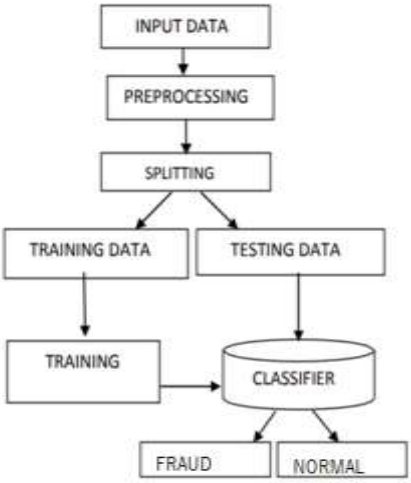


Figure 1. Data model

In training Data, group of examples used for learning acceptable parameters of the classifier in the SVM, will use the training set to find the optimal support vectors. Testing Data uses a group of examples only to assess the performance of a fully-trained classifier in SVM case, we would use the test to estimate the error rate, FP rate or TP rate after we have chosen the final model. For validation group of examples used to tune the parameters of a classifier For SVM case, we would use the validation set to find the "optimal" number of support vectors or determine a stopping point for the algorithm.

4. RESULTS

Source Code Snippets with Explanation*:*

Dataset features:

CreditCardFraudDetection

V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amou
3.049106	-1.763406	-1.559738	0.160842	1.233090	...	-0.503600	0.984460	2.458589	0.042119	-0.481631	-0.621272	0.392053	0.949594	46.(
-1.555434	-0.720961	-1.080664	-0.053127	-1.978682	...	-0.177650	-0.175074	0.040002	0.295814	0.332931	-0.220385	0.022298	0.007602	5.(
-1.191209	1.309109	-0.878586	0.445290	-0.446196	...	-0.295583	-0.571955	-0.050881	-0.304215	0.072001	-0.422234	0.086553	0.063499	231.1
1.129566	1.696038	0.107712	0.521502	-1.191311	...	0.143997	0.402492	-0.048508	-1.371866	0.390814	0.199964	0.016371	-0.014605	34.(
0.428804	0.089474	0.241147	0.138082	-0.989162	...	0.018702	-0.061972	-0.103855	-0.370415	0.603200	0.108556	-0.040521	-0.011418	2.:
-1.314394	-0.150116	-0.946365	-1.617935	1.544071	...	1.650180	0.200454	-0.185353	0.423073	0.820591	-0.227632	0.336634	0.250475	22.1
2.941968	2.955053	-0.063063	0.855546	0.049967	...	-0.579526	-0.799229	0.870300	0.983421	0.321201	0.149650	0.707519	0.014600	0.(
0.295198	-0.959537	0.543985	-0.104627	0.475664	...	-0.403639	-0.227404	0.742435	0.398535	0.249212	0.274404	0.359969	0.243232	26.(
-0.172577	-0.916054	0.369025	-0.327260	-0.246651	...	0.067003	0.227812	-0.150487	0.435045	0.724825	-0.337082	0.016368	0.030041	41.(
-0.836758	-0.831083	-0.264905	-0.220982	-1.071425	...	-0.284376	-0.323357	-0.037710	0.347151	0.559639	-0.280158	0.042335	0.028822	16.(
0.007443	-0.200331	0.740228	-0.029247	-0.593392	...	0.077237	0.457331	-0.038500	0.642522	-0.183891	-0.277464	0.182687	0.152665	33.(
-0.786002	0.578435	-0.767084	0.401046	0.699500	...	0.013676	0.213734	0.014462	0.002951	0.294638	-0.395070	0.081461	0.024220	12.(
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
-0.045791	-1.345452	0.227476	-0.378355	0.665911	...	0.235758	0.829758	-0.002063	0.001344	0.262183	-0.105327	-0.022363	-0.060283	1.(
0.901528	-0.760802	0.758545	0.414698	-0.730854	...	0.003530	-0.431876	0.141759	0.587119	-0.200998	0.267337	-0.152951	-0.065285	80.(
2.327804	3.664740	-0.533297	0.842937	1.128798	...	0.086043	0.543613	-0.032129	0.768379	0.477688	-0.031833	0.014151	-0.066542	25.(
-0.884199	0.793083	-0.527298	0.866429	0.853819	...	-0.094708	0.236818	-0.204280	1.158185	0.627801	-0.399981	0.510818	0.233265	30.(
1.451777	0.093598	0.191353	0.092211	-0.062621	...	-0.191027	-0.631658	-0.147249	0.212931	0.354257	-0.241068	-0.161717	-0.149188	13.(
0.639105	0.186479	-0.045911	0.936448	-2.419986	...	-0.263889	-0.857904	0.235172	-0.681794	-0.668894	0.044657	-0.066751	-0.072447	12.(
2.199572	3.123732	-0.270714	1.657495	0.465804	...	0.271170	1.145750	0.084783	0.721269	-0.529906	-0.240117	0.129126	-0.080620	11.(
2.833960	3.240843	0.181576	1.282746	-0.893890	...	0.183856	0.202670	-0.373023	0.651122	1.073823	0.844590	-0.286676	-0.187719	40.(
2.932315	3.401529	0.337434	0.925377	-0.165663	...	-0.266113	-0.716336	0.108519	0.688519	-0.460220	0.161939	0.265368	0.090245	1.1
1.982785	3.732950	-1.217430	-0.536644	0.272867	...	2.016666	-1.588269	0.588482	0.632444	-0.201064	0.199251	0.438657	0.172923	8.(
2.424360	-2.956733	0.283610	-0.332656	-0.247488	...	0.353722	0.488487	0.293632	0.107812	-0.935586	1.138216	0.025271	0.255347	9.(
-0.715798	-0.751373	-0.458972	-0.140140	0.959971	...	-0.208260	-0.430347	0.416765	0.064819	-0.608337	0.268436	-0.028069	-0.041367	3.(
0.578957	-0.605641	1.253430	-1.042610	-0.417116	...	0.851800	0.305268	-0.148093	-0.038712	0.010209	-0.362666	0.503092	0.229921	60.(

Loading Dataset and collecting features and labels in x, y variables.

```
import numpy as np
import pandas as pd
import sklearn


df = pd.read_csv('creditcard.csv', low_memory=False)
df.head()
x = df.iloc[:,:-1]
y = df['Class']
# x.head()
frauds = df.loc[df['Class'] == 1]
non_frauds = df.loc[df['Class'] == 0]
print("We have", len(frauds), "fraud data points and", len(non_frauds), "regular data points.")
```

Preprocessing Stage:

```
from sklearn.preprocessing import scale
x = scale(x)

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y,test_size=0.3, random_state=42)
print("Size of training set: ", X_train.shape)
```

Dividing data into train and test data as shown in code using train_test_split functions.

SVM Algorithm Loading and accuracy calculation:

```
from sklearn import svm
from sklearn import svm
clf = svm.SVC()
clf.fit(X_train, y_train)

predictions = clf.predict(X_test)
print("Size of training set: ", X_test.shape)
print(predictions.shape)

from sklearn.metrics import classification_report,confusion_matrix
print(confusion_matrix(y_test,predictions))


print(classification_report(y_test,predictions))
from sklearn.metrics import accuracy_score
accuracy_score(y_test, predictions)
```

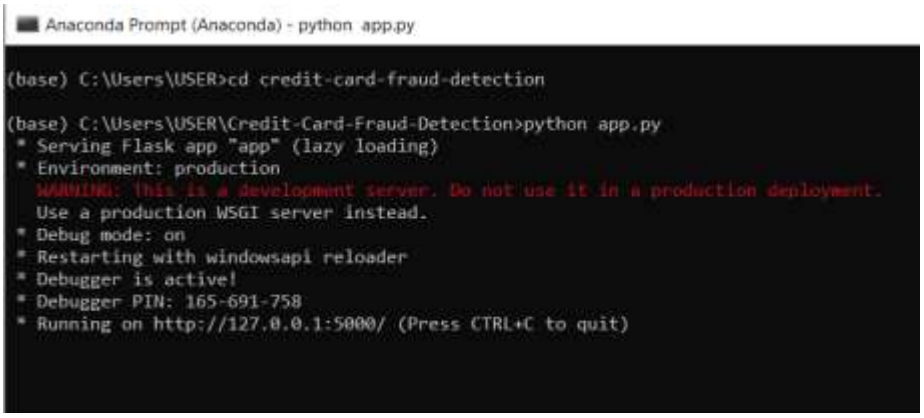
By initializing SVM algorithm train data is given as input to fit function for SVM algorithm and then predict function is called with test data which has half of train records features based on output prediction values will be labels of test set and then accuracy is calculated using accuracy score function by comparing with existing test labels in dataset.

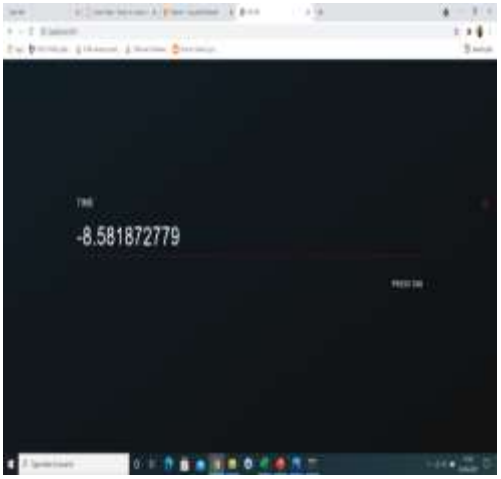
INPUTS:

*Untitled - Notepad

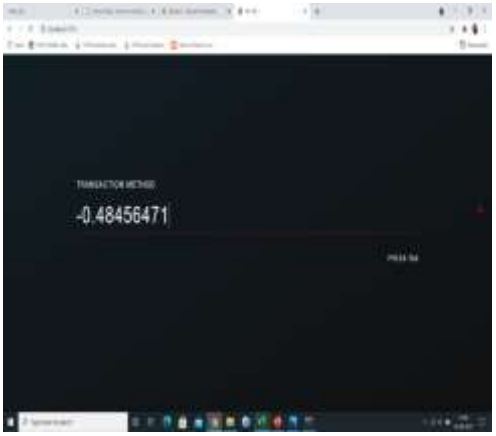
File Edit Format View Help

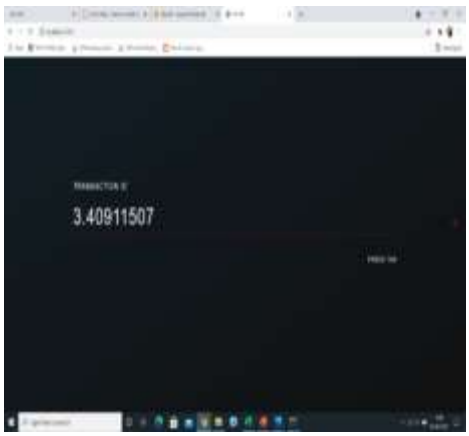
Time	Amount	Transaction Method	Transaction Id	Type of Card	Location	Bank	Class
-9.343495424	0.479454917	-0.892450823	-0.223973018	-0.780906931	0.51599532	0.127688873	0
-8.581872779	0.140151109	-0.48456471	3.40911507	3.229685886	0.206835106	0.856151016	1

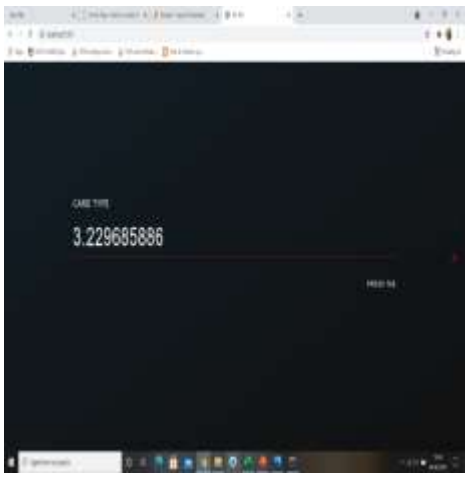

Anaconda Prompt (Anaconda) - python app.py
(base) C:\Users\USER>cd credit-card-fraud-detection
(base) C:\Users\USER\Credit-Card-Fraud-Detection>python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Restarting with windowsapi reloader
* Debugger is active!
* Debugger PIN: 165-691-758
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

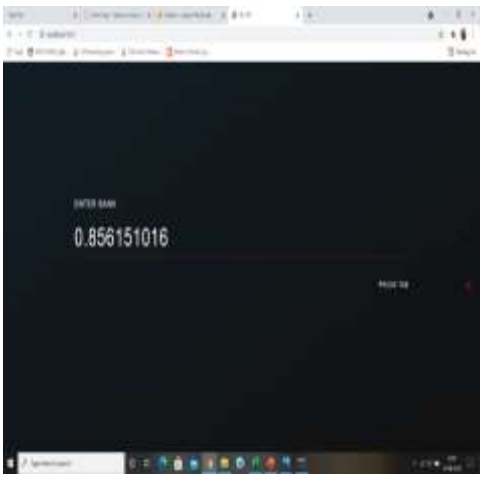

-8.581872779


0.140151109

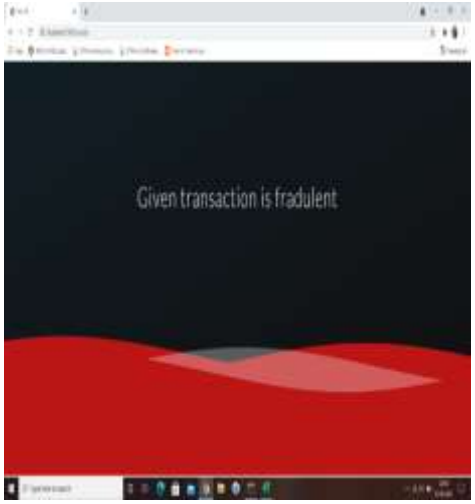

-0.48456471

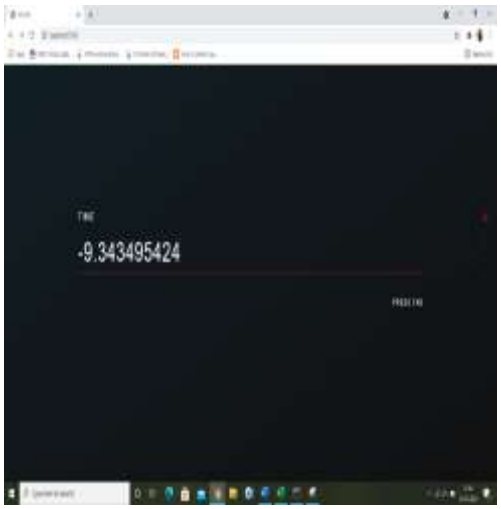

3.40911507


3.229685886


0.856151016


0.206835106

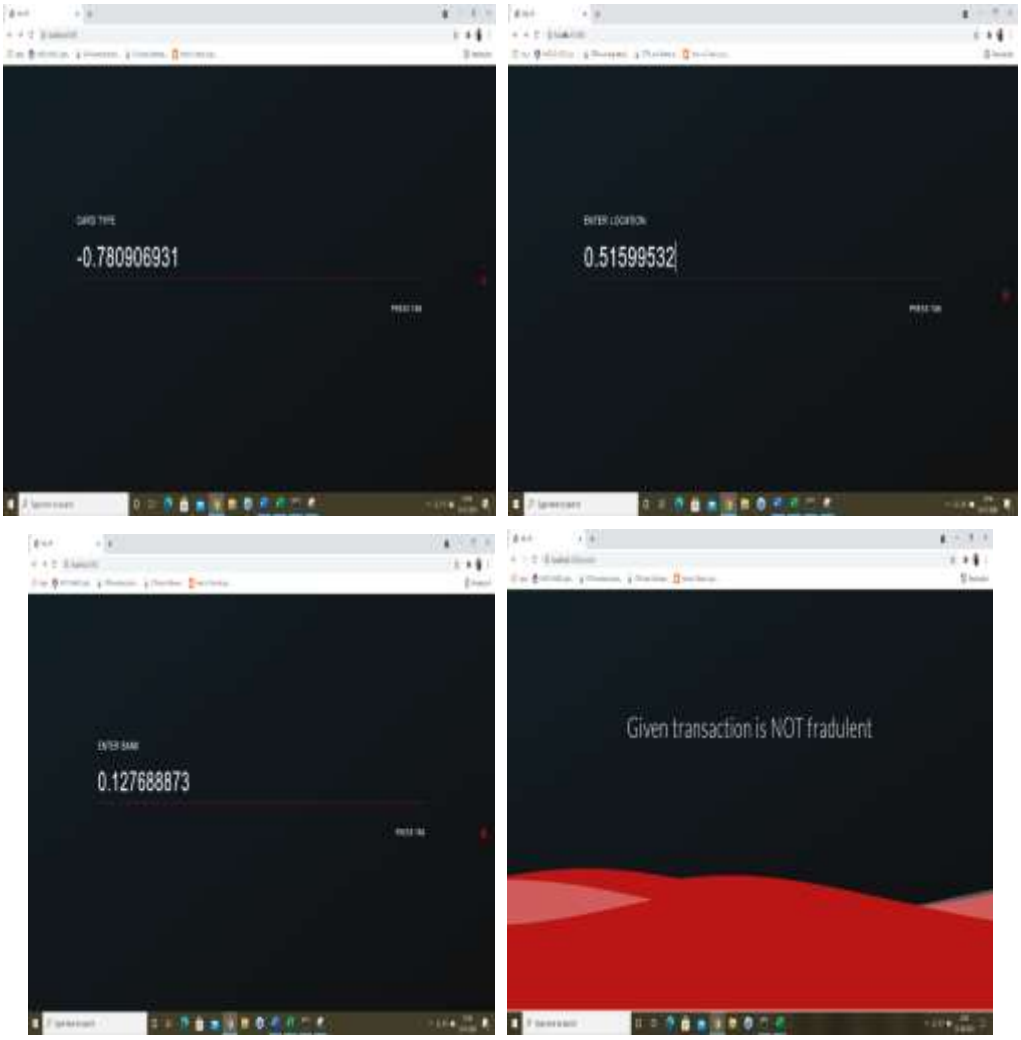

Given transaction is fradulent


-9.343495424


0.479454917


-0.892450823


-0.223973018



5. CONCLUSION

Proposed work investigates the performance of SVM for deception finding in credit cards. For performance assessment, uses a test data set with much lower fault rate 0.5% than the training datasets with different levels of under sampling. It indicates a sign of performance that may be predictable when representations are functional for fault finding where the percentage of fault dealings are classically low. Support vector machine predicts 95% users efficiently only 5% fake users are predicted as good users, and 14% good users are predicted as bad. To compare single tree datamining method with ensemble methods, considering the two incorrectly forecast situation the same bad, Support vector machine and Random Forest algorithms applied on dataset. All approaches describe user credentials and duration period to predict their credit risk. Without exception ensemble approaches are lower misclassification rates than the single tree method support vector machine where catching represents the best predicting result that 95% users in the test sample are predicted rightly. Proposed work implemented in python. The performance of the methods evaluated based on precision and balanced classification rate. This paper mainly focused on deception finding in credit cards. Credit cards data set collected initially as qualified data set. Then provide questions on customers credit card to test the dataset. Then Random Forest applied on already evaluated data set and providing current dataset. The benefit and advantage of deception finding in credit card is to clear for both companies and customers. We will get accurate results and this system will identify classification errors. This will take less time to identify the fraud transactions with credit cards.

Funding: No funding was received for conducting this study.

REFERENCES

[1]. R. R. Popat and J. Chaudhary, "A Survey on Credit Card Fraud Detection Using Machine Learning," *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2018, pp. 1120-1125, doi: 10.1109/ICOEI.2018.8553963.

[2]. Maniraj.S.P, Aditya Saini, Ahmed Shadab & Swarna Deep Sarkar, "Credit Card Fraud Detection using Machine Learning and Data Science," International Journal of Engineering Research & Technology (IJERT), vol. 8, pp. 110-115, Sept. 2019. DOI:10.17577/IJERTV8IS090031

[3]. John O. Awoyemi, A. O. Adetunmbi, S. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," International Conference on Computing Networking and Informatics (ICCNI), pp. 1-9, 2017. DOI:10.1109/ICCNI.2017.8123782

[4]. Yashvi Jain, NamrataTiwari, ShripriyaDubey,Sarika Jain , "A comparative analysis of various credit card fraud detection techniques International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7 Issue-5S2, January 2019

[5]. S. Abinayaa, H. Sangeetha, R. A. Karthikeyan, K. Saran Sriram, D. Piyush, "Credit Card Fraud Detection and Prevention using Machine Learning", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-4, April, 2020

[6]. K. R. Seeja, Masoumeh Zareapoor, "FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining", The Scientific World Journal, vol. 2014, Article ID 252797, 10 pages, 2014. https://doi.org/10.1155/2014/252797

[7]. Megha Banerjee , Reetodeep Hazra, " Fraudulent Credit Card Transactions Prediction Using 2D-Convolutional Neural Network", International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering(IJIREEICE), Vol. 8, Issue 9, September 2020 DOI 10.17148/IJIREEICE.2020.8910

[8] Adi Saputra, Suharjito, " Fraud Detection using Machine Learning in e-Commerce", International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 10, No. 9, 2019. (DOI) : 10.14569/IJACSA.2019.0100943

[9] Heta Naik and Prashasti Kanikar, " Credit card Fraud Detection based on Machine Learning Algorithms", International Journal of Computer Applications 182(44):8-12, March 2019

[10] C.Navamani, S.KRISHNAN, "Credit Card Nearest Neighbor Based Outlier Detection Techniques", International Journal of Computer Techniques -– Volume 5 Issue 2, ISSN- 2394-2231, 2018

[11] Y. Sahin and E. Duman, "Detecting Credit Card Fraud by Decision Trees and Support Vector Machines", International Multiconference of Engineers and computer scientists' volume 1,2011.

[12] Heta Naik and Prashasti Kanikar. Credit card Fraud Detection based on Machine Learning Algorithms. International Journal of Computer Applications 182(44):8-12, March 2019