IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 4, April 2020

# Credit Card Fraud Detection using Kmeans_SMOTE and Luhn Algorithm

**Dastagir Pasha M Z[1], Savita Sheelavant[2]**

P.G. Student, Department of MCA, Rashtreeya Vidyalaya College of Engineering, Bengaluru, Karnataka, India[1]

Assistant Professor, Department of MCA, Rashtreeya Vidyalaya College of Engineering, Bengaluru, Karnataka, India[2]

**Abstract:** This project mainly focuses on credit card fraud detection in real world scenarios. Nowadays credit card frauds are increasing in number as compared to previous years. Criminals are using various methods like identity theft, fraudulent phone calls or messages, site cloning etc to trap the users and get the money out of them and also few bank personnel cheats bank by entering wrong data like wrong card number and amount etc. 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. Many techniques based on Artificial Intelligence, Neural Networks, Decision Trees, Genetic Algorithm etc were developed to detect fraudulent transactions. This paper presents kmeans_smote oversampling technique, Random Forest Algorithm (RFA) along with Luhn Algorithm for credit card fraud detection

**Keywords:** Credit Card fraud, Random Forest Algorithm, kmeans_SMOTE, Luhn Algorithm

## I. INTRODUCTION

As many countries are going digital, the number of online transactions are increasing drastically and hence credit card users are also increasing. As credit card offers many services compared to other modes of transactions. According to global online banking statistics, cybercrime cost the average global financial institutions $18.4 million dollars in 2018. 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.

Fraud Detection is the process of monitoring the transaction behaviour of a cardholder like amount, time, location etc to detect whether the transaction is authentic and authorized if not it will be detected as illicit. Credit card data is highly imbalanced as the authentic transactions are more than fraudulent transactions. The dataset used consists 284,807 transactions from which 492 transactions are fraudulent. The dataset is highly imbalanced as there are only 0.172% fraudulent transactions.

To handle this highly imbalanced data, oversampling technique is used. Oversampling is the process of generating synthetic data that tries to randomly generate a sample of the attributes from observations in the minority class. There are different oversampling techniques like SMOTE, ADASYN, XGBoost etc. Kmeans_SMOTE is a combination of kmeans clustering and SMOTE, which is as oversampling technique in the proposed system, along with kmeans_SMOTE RFA(Random Forest Algorithm) classification algorithm is used for achieving higher precision and accuracy.

## II. LITERATURE SURVEY

In [1] the author discusses various methods available for balancing the data for efficient analysis. Many available techniques like sampling, cost sensitive learning, ensemble learning and feature selection are discussed in this paper for balancing the datasets. . Of all the papers reviewed for this research SMOTE technique is most commonly used one and feature selection is the second most used technique. In [2] the author primarily talks about using Random Forest Algorithm for credit card fraud detection, it also states the advantages of using RFA and its implementation. The proposed system has achieved an accuracy rate of 0.99480286 using only the RFA algorithm. In [3] the author proposed a system for credit card fraud detection using SMOTE technique. The author uses SMOTE for oversampling the data and experiments this with different Machine Learning methods like logical Regression, random Forest, Naïve Bayes and Multilayer Perceptron. The author concludes that using SMOTE oversampling technique there was an increase in the rate of accuracy and precision. In [4] the author talks about Luhn Algorithm, how Luhn algorithm is used for credit card number validation. The author also discusses the weakness the algorithm suffers and proposed an enhanced Luhn algorithm to overcome those weaknesses.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 9, Issue 4, April 2020

## III. EXPERIMENTAL SETUP

The experimental setup consists following:

A. Data set used

The dataset used is taken from kaggel[5], the datset contains transactions made by credit-cards in September 2013 by European cardholders, this dataset consists transactions occurred in two days and has a total of 284,807 transactions and few attributes data is modified to maintain confidentiality.

B. Tools used:

1. Programming language: Python
2. IDE: Anaconda
3. Libraries Used: Numpy, Pandas, Matplotlib, Scikit-learn, kmeans_smote

## IV. METHODOLOGY

I.

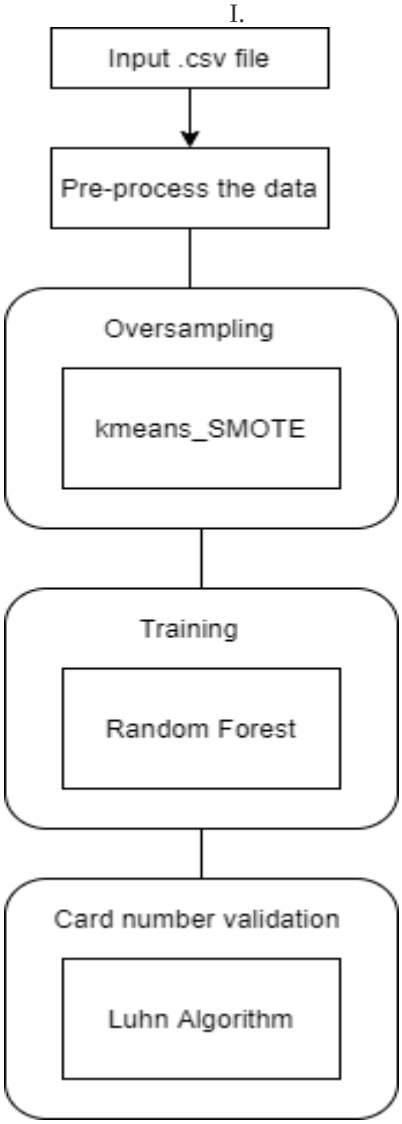


Fig. 1 Block Diagram

The steps shown in Fig. 1 are explained as followed:

A. Data Collection

The data used is obtained from kaggel[5]. It consists credit card transactions information. Using pandas library package the credit card data is imported, the same is depicted in Fig. 2.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 4, April 2020

```
#importing packages
%matplotlib inline
import scipy.stats as stats
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('ggplot')
```

```
df = pd.read_csv('C:/Users/Dastagir Pasha/Documents/creditcard.csv')
```

```
#shape
print('This data frame has {} rows and {} columns.'.format(df.shape[0], df.shape[1]))
```

```
This data frame has 284807 rows and 31 columns.
```

```
#peek at data
df.sample(5)
```

| | Time | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | ... | V21 | V22 | V23 | V24 | V25 | V26 | V27 | V28 | Amount |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **110519** | 71858.0 | -1.170 | -0.265 | 1.332 | 0.550 | -0.895 | 3.351 | 1.239 | 0.746 | -0.080 | ... | 0.047 | 0.317 | 0.499 | -1.362 | -1.234 | 0.644 | 0.187 | 0.092 | 399.15 |
| **160040** | 113094.0 | 2.074 | -0.098 | -1.481 | 0.178 | 0.169 | -0.842 | 0.093 | -0.219 | 0.880 | ... | 0.259 | 0.859 | 0.027 | 0.674 | 0.313 | -0.444 | 0.001 | -0.050 | 1.00 |
| **277962** | 167957.0 | -0.225 | -0.486 | 2.713 | -0.094 | -0.903 | 1.490 | -0.439 | 0.066 | 0.918 | ... | -0.252 | 0.524 | -0.158 | 0.635 | -0.620 | -0.643 | -0.109 | -0.238 | 50.00 |
| **252971** | 156039.0 | -0.228 | 1.318 | -1.859 | -0.318 | 0.178 | -1.048 | 0.295 | 0.689 | -0.102 | ... | -0.080 | -0.546 | 0.326 | 0.522 | -0.857 | 0.128 | -0.379 | -0.100 | 30.00 |
| **96879** | 65980.0 | -1.243 | 1.509 | 0.861 | -0.702 | 0.162 | -0.650 | 0.451 | 0.411 | -0.715 | ... | -0.125 | -0.568 | -0.025 | -0.043 | -0.212 | -0.113 | -0.092 | 0.117 | 0.69 |

5 rows × 31 columns

Fig. 2 Importing required python package for importing the data

B. Pre-processing

Data pre-processing consists following:

1. Formatting: Formatting the data means, putting the data in a legitimate way which would be suitable to work with project
2. Data cleaning: It is the important step in data science. In this step data which is inappropriate for the task and noisy data is removed

In Fig. 3 the attributes that are required are extracted from the data is shown, later it is processed to use in the algorithm.

```
pd.set_option('precision', 3)
df.loc[:, ['Time', 'Amount']].describe()
```

| | Time | Amount |
|---|---|---|
| **count** | 284807.000 | 284807.000 |
| **mean** | 94813.860 | 88.350 |
| **std** | 47488.146 | 250.120 |
| **min** | 0.000 | 0.000 |
| **25%** | 54201.500 | 5.600 |
| **50%** | 84692.000 | 22.000 |
| **75%** | 139320.500 | 77.165 |
| **max** | 172792.000 | 25691.160 |

Fig. 3 Data pre-processing

C. Balancing the data

The credit card transaction data consists 99.8% consists non-fraudulent data whereas only 0.2% of the data was considered as fraudulent. This is highly imbalanced data. To create a training dataset with balanced class distribution kmeans_smote over-sampling method is used. In Fig. 4 the count of fraudulent vs non-fraudulent is shown and it is clear that the data is highly imbalanced whereas in Fig. 5 count of fraudulent vs non-fraudulent is shown after using kmeans_SMOTE oversampling technique and the data is balanced.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 9, Issue 4, April 2020

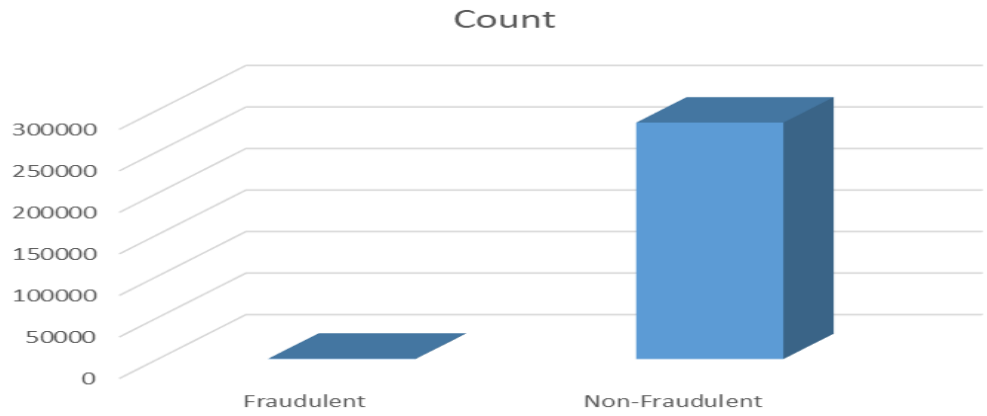


Fig. 4 Data before balancing

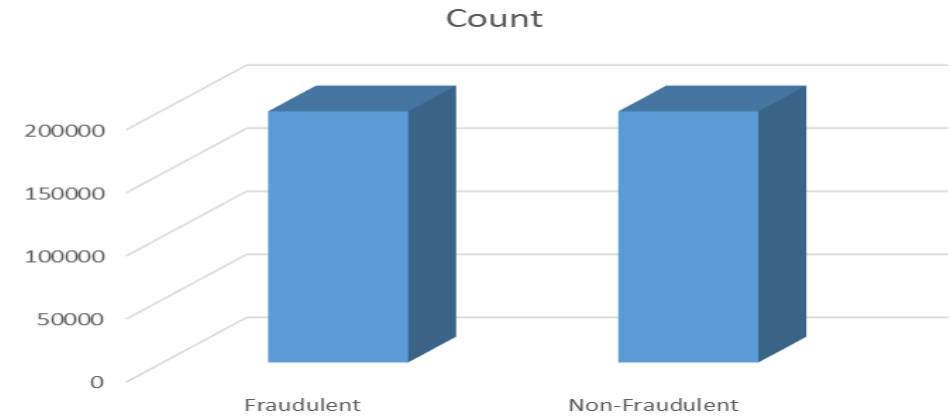


Fig. 5 Data after balancing

D. Implementing Random Forest Algorithm

Random Forest Algorithm is used for training the model. It is a classifier algorithm and is very popular as it provides relatively better performance. In Fig. 6 the output of Random Forest along with kmeans_SMOTE is shown. The output consists accuracy, confusion matrix and precision.

```
RFA using kmeans_Smote
Accuracy:  0.9996839998595555
RF:
 [[85290     5]
 [   22   126]]
RF-AUC:  0.9648456152969899
RF:
 Precision:  [0.99974212 0.96183206] Recall:  [0.99994138 0.85135135]
```

Fig. 6 Output of RFA using kmeans_SMOTE

E. Validating Credit card numbers

The credit card number is validated using Luhn Algorithm, as the model does not recognize valid credit card number. In some cases some bank personnel shows transactions that have not occurred that is fraud transactions by entering records like invalid card number, amount etc. Hence Luhn Algorithm is used to check whether the credit card number is valid or not.

Working of kmeans_SMOTE

The following are the steps which are used to implement kmeans_SMOTE[9]:

A. Cluster the entire input space using k-means
B. Distribute the number of samples to generate across clusters
C. Select clusters which have a high number of minority class samples
D. Assign more synthetic samples to clusters where minority class samples are sparsely distributed
E. Oversample each filtered cluster using SMOTE

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 4, April 2020

## V. RESULT ANALYSIS

This section is all about comparison of the proposed work with the existing work. Existing work uses regular SMOTE.

Table I Comparing Results of Smote With Kmeans_Smote

| | Accuracy | Precision | Recall |
|---|---|---|---|
| RFA with SMOTE | 0.999438 | 0.999613 | 0.999824 |
| RFA with kmeans_SMOTE | 0.999602 | 0.999683 | 0.999917 |

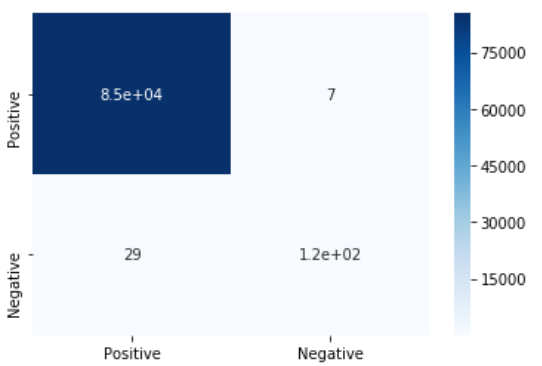


Fig. 7 Confusion Matrix Using kmeans_SMOTE

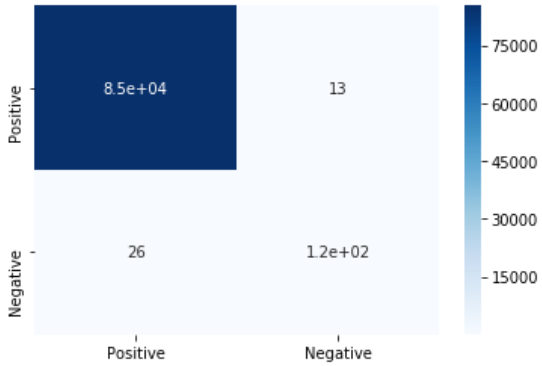


Fig. 8 Confusion Matrix using SMOTE

In Fig. 7 confusion matrix is shown using kmeans_SMOTE, the FP(False Positive) is 7 and in Fig. 8 confusion matrix using SMOTE is illustrated, the FN is 13. From the two figures it is clear that kmeans_SMOTE provides better precision than regular SMOTE.

## VI. CONCLUSION

Using Random Forest Algorithm with kmeans_smote, accurate value of credit card fraud detection i.e. 0.999602 (99.97%) is obtained. In comparison to existing modules, this proposed module is applicable for the larger dataset and provides more accurate results. The Random Forest algorithm will provide better performance, but the speed during testing and application will still be an issue. In future work we will try to implement this as a web application. And also try to provide the solution for credit card fraud using modern technologies like Artificial Intelligence and Deep Learning.

## REFERENCES

[1]. P.Pavithra and S.Babu.' Data Mining Techniques for Handling Imbalanced Datasets: A Review', International Journal of Scientific Research and Engineering Development, Vol. 2, 2019
[2]. Vaishnave Jonnalagadda, Priya Gupta and Eesita Sen, 'Credit card fraud detection using Random Forest Algorithm', International Journal of Advance Research, Ideas and Innovations in Technology. Vol 5, Issue. 2, 2019
[3]. D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic and A. Anderla, "Credit Card Fraud Detection - Machine Learning methods," 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, 2019, pp. 1-5, doi: 10.1109/INFOTEH.2019.8717766
[4]. Khalid Waleed Hussein, Dr. Nor Fazlida Mohd. Sani, Professor Dr. Ramlan Mahmod and Dr. Mohd. Taufik Abdullah, 'Enhance Luhn Algorithm for Validation of Credit Cards Numbers', International Journal of Computer Science and Mobile Computing, Vol. 2, Issue. 7, July 2013
[5]. https://www.kaggle.com/mlg-ulb/creditcardfraud/data
[6]. M. S. Kumar, V. Soundarya, S. Kavitha, E. S. Keerthika and E. Aswini, "Credit Card Fraud Detection Using Random Forest Algorithm," 2019 3rd International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, 2019, pp. 149-153, doi: 10.1109/ICCCT2.2019.8824930
[7]. D. Viji and S. Kothbul Zeenath Banu, 'An Improved Credit Card Fraud Detection Using K-Means Clustering Algorithm', International Journal of Engineering Science Invention, pp. 59-64, 2018
[8]. Yashvi Jain, NamrataTiwari, ShripriyaDubey and Sarika Jain. 'A Comparative Analysis of Various Credit Card Fraud Detection Techniques', International Journal of Recent Technology and Engineering, Vol. 7, Issue. 5S2, January 2019
[9]. https://kmeans-smote.readthedocs.io/en/latest/