International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

# Credit Card Fraud Detection using Ensemble Learning

**Mohini Devikar[1], Apurv Khadke[2], Amogh Lad[3], Rhishikesh Sapkal[4], Saurabh Nikalje[5]**

*[1]Assistant Professor, Department of Computer Engineering, M. E. S. College of Engineering*
*[2,3,4,5]Student, Department of Computer Engineering, M. E. S. College of Engineering, Pune, India*

---

**Abstract –** *This paper elaborates on the implementation of a Credit card fraud detection system using Ensemble Learning techniques. It provides information regarding the System design, architecture, and model. Credit card frauds are increasing considerably with an increase in the number of digital transactions. Credit card frauds cause huge financial loss to companies and consumers however, there is a lack of published literature on credit card fraud detection techniques. The major contribution to this is the confidentiality of data used to work with. We decided to construct the fraud detection system using Ensemble Learning. We studied various Machine Learning algorithms such as KNN, Random Forest, and GaussianNB(Naive Bayes). In this paper, we worked with publicly available European union credit card fraud dataset.*

***Key Words***: **K-Nearest Neighbors(KNN), Random Forest, GuassianNB (Naive Bayes), Support Vector Machine(SVM), Ensemble Learning, Principal Component Analysis(PCA), Accuracy, Recall, Precision.**

## 1. INTRODUCTION

Fraud detection concerns a large number of financial institutions and banks, as this crime costs them around $ 60 billion per year. Credit card fraud is concerned with illegal use of credit card information for purchases. These frauds are executed either physically or digitally. Credit card frauds are of various types: Bankruptcy fraud, Application fraud, Behavioral fraud, and Theft/Counterfeit fraud. Counterfeit frauds are also known as Card Holder not Present Fraud. These kinds of frauds are generally irrevocable and very challenging to detect [1].

Nowadays, digital transactions are considerably increasing, leading to inefficient detection of such frauds. Machine Learning works with a huge amount of sample data of the underlying domain to classify data encountered in the future. The main goal was to deal with the class imbalance problem. With the help of machine learning algorithms, we were able to overcome this obstacle and correctly classify most of the available data[1].

Supervised learning consists of labeled class data available which aids in training the model to classify unlabeled data. Standard models are used to create a hybrid model. Well known techniques used to achieve this are Bagging, Boosting, AdaBoost(Adaptive Boosting), and Majority voting [3].

### 1.1 Ensemble Learning

Ensemble Learning is used to solve computational intelligence problems. It is a method of combining multiple classifiers to form a strategic structure. The resultant prediction output is more accurate compared to the individual constituents. Ensemble Learning is used to enhance the performance of Classifiers for classification and prediction. It comprises of various techniques such as Bagging, Boosting, Stacking which in turn contributes to the existence of a more flexible structure.

### 1.2 Bagging

Bagging (Bootstrap aggregating) consists of multiple models voting with equal weight. Model variance is promoted when bagging trains each model in the Ensemble using a random sampling of the training set. Random forest algorithm uses Bagging to achieve high classification accuracy.

### 1.3 Boosting

Boosting is a technique in which incrementally an Ensemble is built by training each new model instance to emphasize the training instances that previous models had misclassified. AdaBoost(Adaptive Boosting) is the most common implementation of Boosting.

### 1.4 Stacking

Stacking is the technique in which various models are trained on the data and then a combiner algorithm is trained to make the predictions based on the predictions of all the models combined.

## 2. SYSTEM ARCHITECTURE

The system architecture consists of a Training module and Prediction module. The prediction module uses the resultant of the training module. The System operates in two phases, initially existing data needs to be fed to the Ensemble Model so that it absorbs the characteristics of

the data. It is then capable of classifying data belonging to 'class 0' being legit transactions and 'class 1' being fraudulent transactions with minimum loss. In the second phase of operation, the model is deployed to predict the incoming data and generate class labels as mentioned above.

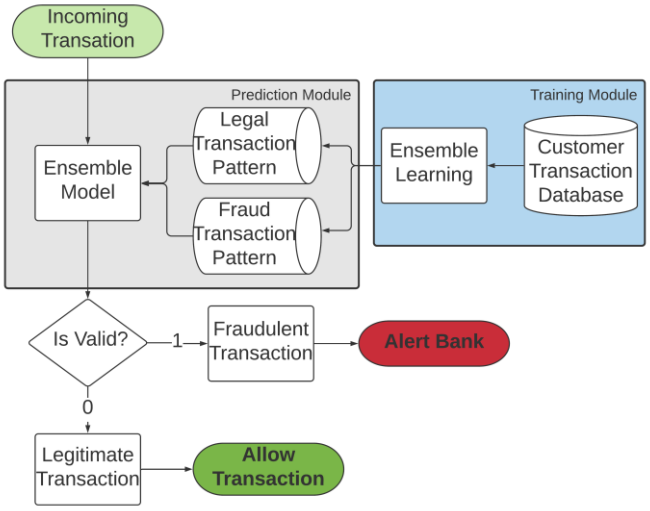


**Fig -1:** System Architecture

## 3. PROCESS WORKFLOW

### 3.1 Dataset

The Dataset comprises of varied credit card transactions made by credit cardholders over a period of 2 days in September 2013 by European cardholders. This Dataset presents transactions of a diverse nature, it contains 492 fraud transactions out of 284,807 transactions. Looking at the amount of fraud transactions in the Dataset it shows a highly imbalanced[1] nature. The positive class (frauds) accounts for 0.172% of all transactions. The data available has been transformed using PCA in order to reduce the dimensions and protect the interest of the customers who have provided their data. Unfortunately, due to confidentiality issues, there is no information about the original features.

Features available at hand V1, V2, . . . V28 are the principal components obtained after application of PCA, the only features available in their natural state are 'Time' and 'Amount'. Feature 'Time' represents the time in seconds elapsed between individual transactions and the very first transaction in the Dataset. The feature 'Amount' is the amount credited/debited to/from the account of the user. Feature 'Class' is the key to train a model because it is the response variable. It takes value 1 in case of fraud and value 0 for any other type of transaction giving us the labeled information needed.

### 3.2 Feature Selection

Feature Selection is one of the core concepts of machine learning. It helps by boosting the performance of your model. The raw data is available after cleaning and removing any and all anomalies. It still has a few features which do not contribute to the performance or negatively impact it. Such features if added lead to inaccurate and inconsistent results. Feature Selection is the process where you automatically or manually select features that have the highest relevance and contribution to the performance metrics such as precision and sensitivity. The challenge we faced while selecting features was to identify which one was relevant to the context because the data had been transformed and was not in its original state.

Feature Selection provides other benefits such as reduced risk of over-fitting, improved accuracy, reduced training time because of reduced data. Various techniques are available which can be used to select relevant features for training a model such as Univariate Selection, Feature Importance, Correlation Matrix with Heatmap. The method we adopted to tackle this problem was to use a method similar to a heatmap. We plotted the density distribution graphs as shown in Fig -2, of the individual attributes starting with V1, V2, . . . , V28, Time, and amount. The plotted data was with respect to the class label which helped us understand the trend that this dataset followed. The features were selected based on their relevance and observed distribution.

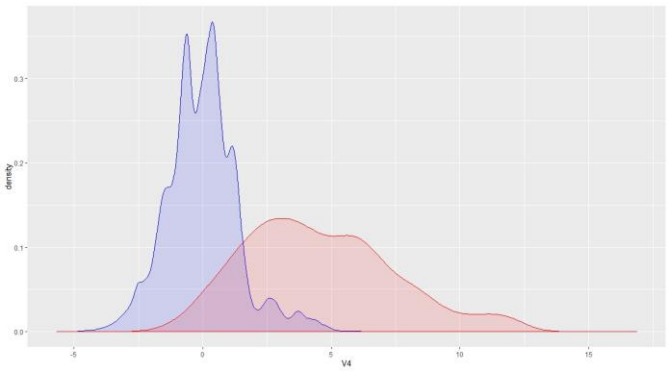


**Fig -2:** Density Distribution of V4

### 3.3 Model Training

Model Training involves training of the Classifier using the available dataset. In this phase, various algorithms were analyzed for the available dataset. Classifiers were selected based on their accuracy and recall score over a randomly selected test data, as shown in Table 1. KNN[5], Random Forest[6], and GuassianNB[7] are the algorithms that were selected for creating the final Ensemble Model.

| Method | Accuracy | Sensitivity |
|---|---|---|
| KNN | 99.961 | 79.268 |
| Random Forest | 99.991 | 92.918 |
| GuassianNB | 99.268 | 80.894 |
| SVM | 99.961 | 80.487 |
| Logistic Regression | 99.916 | 60.162 |
| Bernoulli Naive Bayes | 99.980 | 47.764 |

**Table -1:** Algorithms Performance in percentage

### 3.4 Ensemble Learning

The Ensemble Learning Model is created as shown in Fig - 3. Bagging technique is used in which three Trained Models KNN[5], Random Forest[6], and GuassianNB[7] are used for voting with equal weights for classification of transactions. Hard voting is carried out in this process.

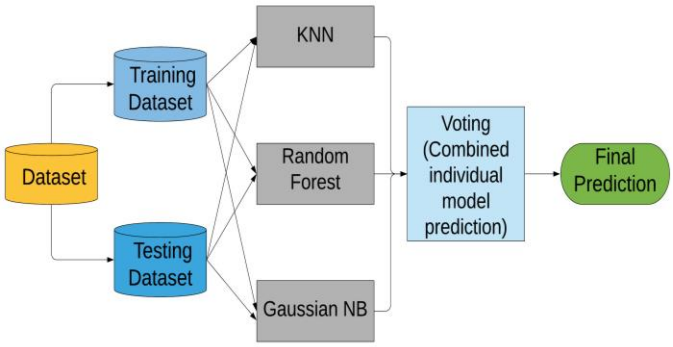


**Fig -3**: Block Diagram of Ensemble Learning Model

K-Nearest Neighbor is a supervised machine learning algorithm that uses Euclidean, Manhattan, or Minkowski distance functions. K-Nearest Neighbor is an algorithm which classifies transactions by similarity based on the distance in multidimensional space. The record is assigned to the class of the nearest neighbors.

Random forest is a Bagging Classifier that builds decision trees to classify the data objects. The model selects a variable that enables the best splitting of records and repeats the splitting process multiple times. To make predictions more precise it trains multiple decision trees on random subsets from a general dataset. To decide whether a transaction is fraud, Trees vote is taken and the model provides a consensus judgment. The Random Forest is an Ensemble Method Classifier that combines various Tree predictors. The advantage of using Random Forest is that it is robust to noise, outliers and works very well over an imbalanced dataset.

A Gaussian Naive Bayes algorithm is a special type of NB algorithm. It's specifically used when the features have continuous values. It's also assumed that all the features have a Gaussian Distribution i.e, normal distribution. Besides the Gaussian Naive Bayes there exists the Multinomial Naive Bayes and the Bernoulli Naive Bayes. A Gaussian distribution is also called as Normal distribution. We picked the Gaussian Naïve Bayes because it is the simplest and the most popular one.

### 3.5 Finalizing and experimenting with Ensemble Models

In the initial stages as we moved towards making Ensembles out of existing models that were laid out based on their performance metrics( accuracy, recall, precision) the initial permutations made consisted of models in pairs of two. The observed result did present a sizable improvement compared to their constituents. The improvement was significant however at the huge trade-off between accuracy and recall scores. If in case the accuracy of the overall hybrid model was high then the recall score would drop and vice versa. To reduce this problem permutations of 3 models were considered which not only reduced the trade-off in accuracy and recall but also resulted in an increased precision value of 100%.

## 4. PROJECT WORKFLOW:

**Step-1:** Available Dataset was cleaned to obtain a consistent and error-free data to avoid any incorrect classification.

**Step-2:** Once clean data was available it needed to be reduced in size. To achieve this we made use of the density distribution graphs of transformed attributes. Attribute selection was carried out in a way that the meaning of data did not change and the information was preserved.

**Step-3:** Model selection phase was an important one and would determine the success. Supervised machine learning algorithms were taken into consideration because labeled data was available. Individual models were built using the available data. These models were tested again using randomly selected test data. Algorithms providing the highest performance measures were chosen and narrowed down to KNN[5], SVM[4], Random Forest[6], and GaussianNB[7].

**Step-4:** Permutations of the selected models were considered in pairs of two and three. Their performance based on accuracy, precision, and recall was compared. Out of all available results the most promising result obtained was from the Majority Voting[3] based Ensemble of KNN using Minkowski distance, Random Forest using Gini index, and GaussianNB .

## 5. RESULTS

Fraud has been increasing at an alarming rate and preventive measures are in place however, these can still be exploited. We need to have more efficient systems to counter these losses. Out of all types of frauds we have observed that credit card frauds amount to a huge number and have raised concerns globally. The cost of maintenance of a system that covers all possible cases is not feasible to most vendors and banks. The major problem observed is the class imbalance problem. Several solutions have been proposed to counter these. Here we studied the available solutions to implement a better one.

We initially compared the existing solutions which gave a high yield in detecting fraudulent transactions. We discussed their performance metrics to further our work. Once a list of models was laid out, we experimented by combining compatible models to check if they show any improvement. On experimentation, we were able to settle with an Ensemble of 3 models which provided the highest measure out of all. The final model comprises of K-nearest neighbors, Random Forest and Gaussian-NB. The model implemented using majority voting[3] Ensemble showed a significant growth and provided accuracy of 99.625% and sensitivity of 94% with a precision of 100% and F1-score of 96.91% on publicly available data. This meant that the implemented model is capable of handling and classifying most transactions.

| Actual | Predicted | |
|---|---|---|
| | **False** | **True** |
| **False** | 1500 | 0 |
| **True** | 6 | 94 |

**Table -2:** Confusion Matrix for Ensemble Model

In Table 2, given above we can see the result obtained on the test dataset. The test dataset consisted of 1600 transactions in total with 1500 legitimate transactions and 100 fraud transactions. The model was able to successfully classify all the legitimate transactions i.e. 1500. It was also able to classify 94 out of 100 fraud transactions over all the test datasets of the same size.

As observed there is a significant increase in the performance however, this result was generated over data that is comparatively old and had a low imbalance. The amount of credit card fraud transactions does contribute to a major amount however in a real-world scenario the data accumulated over a longer period would increase the skewed nature of data in-comparison. As the trade-off observed in accuracy and sensitivity in previous models has now been resolved we would like to implement this model in production on real-time data. The challenge of not being able to explain which attributes contribute to the detection of a fraud will be our focus going forward.

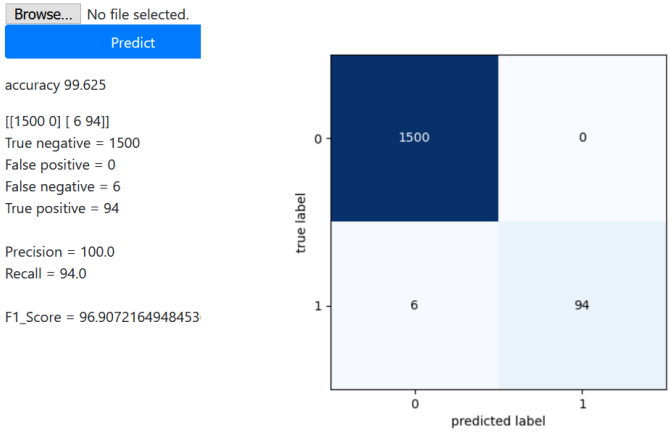


**Fig -4**: Performance Metrics of Ensemble Model

## ACKNOWLEDGEMENT

It gives us great pleasure and satisfaction to have worked on “Credit Card Fraud Detection”. We are thankful to and fortunate enough to get constant encouragement, support, and guidance from our guide Prof. M. G. Devikar. She encouraged us to keep moving forward under her guidance and vigilant support. Also, We would like to extend our sincere esteems to all friends and family for their motivation in times that we hit a wall. We would also like to thank our project team members who showed immense patience and understanding throughout the project. We would like to thank all those, who have directly or indirectly helped us for the completion of the work during this project.

## REFERENCES

[1] Sara Makki, Zainab Assaghir,Mohand-Said Hacid: ”An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection”M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 2019.

[2] Sangeeta Mittal, Shivani Tyagi: "Performance Evaluation of Machine Learning Algorithms for Credit Card Fraud Detection" may 2019.

[3] Credit card fraud detection using AdaBoost and majority voting Kuldeep Randhawa1, Chu Kiong Loo Manjeevan Seera.

[4] Y. Sahin and E. Duman, "Detecting Credit Card Fraud by Decision Trees and Support Vector Machines," International Multiconference of Engineers and computer scientists, vol. 1, pp. 442–447, 2011

[5] V. R. Ganji and S. N. P. Mannem, "Credit card fraud detection using anti-k nearest neighbor algorithm," International Journal on Computer Science and Engineering (IJCSE), vol. 4, no. 06, pp. 1035–1039, 2012.

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

[7] Olawale Adepoju, Julius Wosowei, Shiwani lawte, "Comparative Evaluation of Credit Card Fraud Detection Using Machine Learning Techniques" 2019.