Credit Card Fraud Detection Using Machine Learning

**Project Report**

**Course Code:** CS-3002

**Instructor:** Fahad Samad

**Group Members**

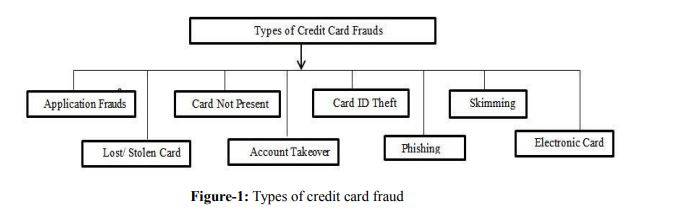
* Saman Khan 19K-0354
* Anusha Saad 19K-0281
* Hermain Qadir 19K-1517

# **Abstract**

Banks today lose customers and money if they are not able to detect fraudulent transactions made from credit cards issued by them. To protect customer rights, retain customers, and save money, credit card detection is a need of the hour in banking sector across the world. Data Science, more specifically machine learning, can be prove to be very relevant in this field of research [1]. When using machine learning to detect credit card fraud, features of these frauds must be carefully chosen because they play a significant part in the process. Our method relies on utilzing machine learning (ML) tecniques to perform credit card fraud detection based on attributes provided by the dataset made public by *www.kaggle.com*. The suggested methodology uses the K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), AdaBoost Classifier, LightGBM, and XGBoost ML classifiers after selecting the optimum features. A test dataset, which is a subset of the original data, is used to evaluate the performance the above-mentioned proposed credit card fraud detection model. The strategies' effectiveness is assessed using accuracy, recall, ROC, and precision.

# **Introduction**

Among other illegal activities, credit card fraud is rather frequent and can occur every day. Credit card fraud can occur in a number of ways. A burglar may find or steal a credit card and use it later [2]. Credit card theft also happens when one’s identity becomes public on some platforms. Criminals use the same credentials to login to one’s accounts. Phishing attempts are used by scammers to trick people into giving out confidential information like credit card number and card pin number [3]. Given the nature of this fraud, all credit card holders are equally vulnerable to the frauds. One way a person can keep him/herself safe is to practice utmost caution and not fall for phishing attempts. Numerous credit card firms have implemented various technologies to detect anomalous transactions using machine-learning. This helps to prevent and detect these types of fraudulent actions [4]. Globally, credit card fraud is the most widespread problem for individuals and financial institutions because of increasing trend of online shopping and making transactions digitally. The world is moving towards promoting cashless transactions, resulting in the loss of billions of dollars for banks and people equally. According to financial reports produced in the past few years, internet fraud reached 24.26 billion dollars in 2018, representing an annual increase of approximately 11%-18% [5]. Fraudulent transactions are made by newer and unique means as people as scammers are more dedicated than ever. They perform theft of a physical card and its credentials or even create a clone of one’s mobile phone device to steal login information stored on notepads or browsers/caches [6]. Internal credit card fraud happens when the cardholder and the bank permit the use of a fictitious identity to conduct a fraudulent transaction, whereas external credit card fraud occurs when card data and other personal credentials are taken by means like phishing [7]. In addition to this classification, there are additional types of fraudulent transactions, such as the classic approach in which application/account details are stolen and internet scams involving site cloning, card cloning, and questionable merchant sites, as seen in Figure 1 [8].



# **Existing System**

The authors of reference [9] constructed a credit card fraud detection system employing many ML techniques, including logistic regression (LR), decision tree (DT), support vector machine (SVM), and random forest (RF). In 2013, these classifiers were assessed against a dataset of European cardholders with a significant imbalance. The researchers hypothesize that sophisticated preprocessing techniques could improve the effectiveness of classifiers. Varmedja et al. [10] proposed an ML-based technique for detecting credit card fraud. The experimental findings revealed that the RF algorithm performed ideally with a 99.96% accuracy in detecting fraud. The authors admit that additional study is required to increase the precision of other ML approaches.

Khatri et al. [11] did a performance analysis of machine learning techniques for the identification of credit card fraud. In this study, the authors evaluated the following machine learning (ML) techniques: DT, k-Nearest Neighbor (KNN), LR, RF, and NB. To evaluate the effectiveness of each ML technique, the authors employed a dataset comprised of European cardholders that was very unbalanced. The precision attained by each classifier was utilized as one of the primary performance metrics during the tests. The experimental results revealed that the precisions of DT, KNN, LR, and RF were 85.11%, 91.11%, 87.5%, and 89.77%, respectively. Khatri et al. [11] did a performance analysis of machine learning techniques for the identification of credit card fraud. The authors investigated DT, k-Nearest Neighbor (KNN), LR, RF, and NB as ML techniques. The authors evaluated the effectiveness of each ML approach using a dataset of European cardholders with a significant imbalance. The accuracy was the primary performance indicator used to evaluate the efficacy of each ML technique. NB, LR, and KNN achieved accuracies of 97.92%, 54.86%, and 97.69%, respectively. Although the NB and KNN performed reasonably well, the authors did not consider implementing a feature selection approach.

# **System Design**

The system is not designed for end-to-end usage. It is only a model that can be later incorporated to build an end-to-end system that can be used in banks to detect credit card frauds in the least possible time.

For designing the system model, Google Collaboratory is used as it provides GPU support which was essential to train the model in less amount of time.

## **Data Dictionary**

The dataset was downloaded from [here](https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud).

This dataset contains credit card transactions made over a period of two days by European cardholders. It contains a total of 284, 807 transactions which are labelled as either fraudulent or non-fraudulent. The attribute class takes the value of ‘0’ if the transaction is non-fraudulent and value of ‘1’ if the transaction is fraudulent.

The dataset contains actual transactions that were in September 2013, so confidentiality is to be maintained. To do so, the features from V1 to V28 are the principal components obtained by performing Principal Component Analysis (PCA). The two features ‘time’ and ‘amount’ are not a part of the PCA modified data. ‘Amount’ is the amount of money deposited/withdrawn/cashed in the particular transaction. Feature ‘time’ is the time elapsed between the first transaction and the next on the same credit card.

# **System Implementation**

The system is implemented in a seven-step pipeline that is summarized as follows:

1. **Understanding Data**

The dataset is loaded by mounting Google Drive folder on Google Colab. Then, features are observed and analysed to determine the ones that will be used in the model building phase.

1. **Exploratory Data Analysis**

Univariate analysis is performed where each component vector from V1 to V28 is compared to make clear their occurrences and variation. Next, density plots of fraudulent vs non-fraudulent transactions are plotted to understand their concentration against time.

1. **Data Visualization**

Data is visualized by heatmaps, box plots, density plots, histograms, scatter plot to understand the correlation of each feature with the rest.

1. **Train and Test Split**

The dataset of more than 284,000 transactions is split for training and testing phases. 80% of the data is used for training and the rest 20% is used to test the accuracy of models. Hold-out cross-validation is used to correctly separate and two splits and test accordingly.

1. **Model-Building**

At this stage, several models are implemented. During implementation, their hyperparameters are fine-tuned to ensure the perform to the maximum capability. The models implemented are: Logistic Regression, K-Nearest Neighbours, Decision Tree, Random Forest, Ada Boost, LGBM, XGB.

1. **Model Evaluation**

In this last stage, models are evaluated using appropriate evaluation metrics. Since the data is highly imbalanced, it is very important to classify fraud transactions accurately rather than identifying non-fraudulent ones.

Model evaluation is performed on the basis of evaluation metrics as well as time taken.

## **Feature Selection** To determine which features are important in all classification models, feature selection is performed. This can open a new direction of research to understand what features are important to for models to classify transactions. To save time and effort, models should then be given the same features during training and features with less significance can be omitted.

# **System Evaluation**

## **Evaluation Methodology** The dataset contains 284,000 + transactions, out of which only 492 belong to class ‘1’ which means only 492 are the fraudulent transactions. This makes it evident how imbalanced the dataset is. At the same time, this is the reality. Fraudulent transactions occur much less than non-fraudulent ones. However, the need of the hour is to detect fraudulent transactions even if they constitute very less percentage of the total transactions made. Given the imbalance, it is very crucial to use the right evaluation metrics. Here, true-positives account for more importance because detecting fraudulent transactions is a necessity. Recall (also termed as sensitivity), therefore, is a better evaluation metric than precision or accuracy in this case. Specificity, also termed as precision, is not as important because focus is not identifying non-fraudulent transactions, but it cannot be ignored completely either. To satisfy both sensitivity and specificity requirements of evaluation of model, ROC score is a good measure. It is essentially used to depict the trade-off between sensitivity and specificity. The ROC score gives a value between 0 to 1. The higher the score, the better the model performs. A value between 0.8 to 0.9 is deemed excellent and a score of greater than 0.9 is termed outstanding.

## **Results of the Evaluation**

The results of evaluation are compiled in Table 1.0 below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr #** | **Model** | **Precision** | **Recall** | **ROC** |
| 1 | K-Neighbors Classifier | 1.00` | 0.09 | 0.546 |
| 2 | Decision Tree Classifier | 0.74 | 0.82 | 0.908 |
| 3 | Logistic Regression | 0.75 | 0.57 | 0.786 |
| 4 | Random Forest | 0.96 | 0.79 | 0.893 |
| 5 | Ada Boost | 0.86 | 0.72 | 0.862 |
| 6 | LGBM Classifier | 0.28 | 0.59 | 0.795 |
| 7 | XGB Classifier | 0.93 | 0.82 | 0.908 |

## **Table 1.0**

The time taken to train and test the models are compiled in Table 2.0 as follows:

|  |  |  |
| --- | --- | --- |
| **Sr #** | **Model** | **Time Taken/seconds** |
| 1 | K-Neighbors Classifier | 203.76 |
| 2 | Decision Tree Classifier | 17.89 |
| 3 | Logistic Regression | 2.89 |
| 4 | Random Forest | 182.24 |
| 5 | Ada Boost | 56.30 |
| 6 | LGBM Classifier | 3.66 |
| 7 | XGB Classifier | 48.82 |

## **Table 2.0**

## **Discussion of the Evaluation Results**

A total of seven models are implemented in this system.

Since area under the ROC curve (also termed as AUC score) was our primary measure to determine the best model, Decision Tree Classifier and Extreme Gradient Boosting (XGB) Classifier seems to offer the best results. With AUC score of 0.908, these outperform the rest. On the other hand, K-Neighbors Classifier can be termed as a model that fails to perform because its AUC score lies between 0.5 to 0.6. Logistic Regression and LGBM Classifier can be categorized as ‘fair’ models as their score lies between 0.7 to 0.8 whereas, Random Forest and Ada Boost Classifier are ‘good’ models because of score that falls in the range of 0.8 to 0.9.

Recall (also termed as sensitivity) is also an equally good measure to determine the performance of models. As per Recall obtained, again Decision Tree and XGB Classifier give the highest true-positive (where positive means the transaction is fraudulent) rate which means it predicts 82% of all fraudulent transactions.

On the basis of time elapsed during training and testing of models, Logistic Regression performs the best. However, time alone is not a good measure to obtain performance of the models. Often, complex models with good results require time to train without overfitting or underfitting. Thus, by continuing our results from recall, we will consider times taken for Decision Tree and XGB Classifier. While both have the same recall and AUC\_ROC score (correct to 2 decimal places), Decision Tree takes less time than the latter. It takes 17.89 seconds, whereas XGB Classifier takes 48.82 seconds, i.e. approximately three times as much time.

Additionally, both Decision Tree and Extreme Gradient Boost Classifier produce the same feature selection results. Both categorize V17 and V14 as the most important features in the same positions accordingly.

# **System Discussion**

Our implementation cannot be categorized as an end-to-end implementation, since this not a system. Instead, it is only a component of the system. Therefore, the model works-alone, so that is a limitation. Furthermore, the dataset is not in its actual form. It is modified by Principal Component Analysis, so there is no knowledge of what features V1 to V28 represent.

However, the project has strengths that can be used in future implementations to develop an end-to-end system that serves the purpose of credit card fraud detection in banks. The models are trained on large data or a large number of transactions, and models like Extreme Gradient Boost and Decision Tree Classifier give an excellent ACU score and Recall, so they are reliable enough to be incorporated in future work. Moreover, unlike some previous researches, feature selection is also performed to better understand what features are important. This would be greatly beneficial for banks because they would also know what features V1 to V28 represent, so banks can know what information is necessary to classify a transaction as fraudulent vs non-fraudulent.

Previous work done in the same research area have more or less used the same dataset as the one used in this project. The imbalance of dataset needs to be handled by further research and work. Previous work used the same models but compared model performance on the basis of accuracy which is not as good of a measure for this implementation as ROC score or Recall because as mentioned earlier the model performance can only be judged by how many fraudulent transactions are detected as fraud to prevent criminal activities as soon as possible.

# **Conclusion**

Banks aim to retain customers. However, credit card frauds pose a significant threat to banks as their business goal of getting new customers and retaining maximum number of customers is far from being met. As the trend of online payments and establishing a cash-less environment is promoted, credit card fraud is increasing. Criminals have found newer ways to commit frauds. To stop them, it has become a necessity for banks to have a credit card detecting system in place that is proactive and has appropriate fraud prevention mechanisms.

To assist banks in achieving this aim, we have developed a project that can be incorporated to build a system that fulfills the need of the hour. By implementing several different machine-learning algorithms and calculating ROC score of each, we have found Decision Tree and Extreme Gradient Boost to be the best-performing classifier models. They also produce a good recall which confirms the results of ROC score. Feature selection and time required to train models are add-ons that will prove to be a great assistance to build an end-to-end system in near sfuture or to improve on our work with another dataset.

# **References**

[1] S P, Maniraj & Saini, Aditya & Ahmed, Shadab & Sarkar, Swarna. (2019). Credit Card Fraud Detection using Machine Learning and Data Science. International Journal of Engineering Research and. 08. 10.17577/IJERTV8IS090031.

[2] Zheng, Lutao, et al. "Transaction fraud detection based on total order relation and behavior diversity." IEEE Transactions on Computational Social Systems 5.3 (2018): 796-806.

[3] Zojaji, Zahra, Reza Ebrahimi Atani, and Amir Hassan Monadjemi. "A survey of credit card fraud detection techniques: data and technique oriented perspective." arXiv preprint arXiv: 1611.06439 (2016).

[4] Bhattacharyya, Siddhartha, et al. "Data mining for credit card fraud: A comparative study." Decision Support Systems 50.3 (2011): 602-613.

[5] Gupta, Gaurav. "Analysis of Various Credit Card Fraud Detection Techniques." (2019).

[6] Dutta, Sharmistha, Ankit Kumar Gupta, and Neetu Narayan. "Identity Crime Detection Using Data Mining." 2017 3rd International Conference on Computational Intelligence and Networks (CINE). IEEE, 2017.

[7] Taha, Altyeb Altaher, and Sharaf Jameel Malebary. "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine." IEEE Access 8 (2020): 25579-25587

[8] Akanksha Bansal and Hitendra Garg 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1116 012181

[9] 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.

[10] 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.

[11] 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.