International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 5 ǁ 2022 ǁ PP. 09-14

Credit Card Fraud Detection System using Machine Learning

Shubham Shah
Department Information Technology
Shah & Anchor Kutchhi
Engineering College
Mumbai, India

Dhairya Shah
Department Information Technology
Shah & Anchor Kutchhi
Engineering College
Mumbai, India

Nirmit Shah
Department Information Technology
Shah & Anchor Kutchhi
Engineering College
Mumbai, India

PranaliWagh
Department Information Technology
Shah & Anchor Kutchhi
Engineering College
Mumbai, India

Abstract *–The recent advances of e-commerce and e-payment systems have sparked an increase in financial fraud cases such as credit card fraud. It is therefore crucial to implement mechanisms that can detect the credit card fraud. Features of credit card frauds play important role when machine learning is used for credit card fraud detection, and they must be chosen properly. This paper proposes a machine learning (ML) based credit card fraud detection engine using the genetic algorithm (GA) for feature selection. After the optimized features are chosen, the proposed detection engine uses the following ML classifiers: Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Artificial Neural Network (ANN), and Naive Bayes (NB). To validate the performance, the proposed credit card fraud detection engine is evaluated using a dataset generated from European cardholders. The result demonstrated that our proposed approach outperforms existing systems.*
Keywords – *Credit Card, Machine Learning, cardholders, genetic algorithm.*

Date of Submission: 01-05-2022 Date of acceptance: 12-05-2022

I. Introduction

Credit card fraud is receiving increasing attention in today's world, and fraud in government offices, corporate industries, financial industries, and many other organizations is increasing. In today's world, the high dependence on the Internet is the reason for the high rate of credit card fraudulent transactions, but fraud is not only increasing online, but also offline transactions. Despite the use of data mining techniques, the results are not very accurate in detecting these credit card frauds. The only way to minimize these losses is to use efficient algorithms for fraud detection, which is a promising way to reduce credit card fraud. With the increase in internet usage, financial companies issue credit cards. Having a credit card means that we can borrow funds. Funds can be used for any purpose. When a card is issued, the conditions involved are that the cardholder repays the original amount they borrowed and the additional fees they agreed to pay. When someone else uses your credit card instead of you without your authorization, the credit card is called fraud. Scammers steal your credit card PIN or account details to perform any unauthorized transactions without stealing the original physical card. By detecting credit card fraud, we can find out if a new transaction is fraudulent or a real transaction. Fraud can involve cards such as credit or debit cards. In this case, the card itself acts as a source of fraud in the transaction. The purpose of the crime may be to obtain property without payment or it may be to obtain funds illegally. Credit cards are a good target for fraud. The reason is that you can make a lot of money in a short period of time without taking too much risk. Even crimes can take weeks to discover. Commonly used algorithms for credit card fraud detection are: RF, KNN, and SVM. In modern times, technology must be used efficiently, correctly and correctly. This project will help develop the technological infrastructure of our country.

II. Literature Survey

Many Authors and Researchers have imposed machine learning algorithm to detect the type of credit card fraud. We have reported some of the method in the literature.

Maniraj [1] focuses on data set analysis and preprocessing, and the application of multiple anomaly detection algorithms such as Local Outlier Factor and Isolation Forest Algorithm to PCA-transformed credit card transaction data.

Asha RB [2] evaluated and compared two algorithms used namely naïve bayes and random forest algorithm, and both showed high accuracy percentage i.e., NB=97.37% and RF=90%.

AndhavarapuBhanusri [3] have explained various techniques available for a fraud detection system such as Support Vector Machine (SVM), Artificial Neural Networks (ANN), Bayesian Network, K- Nearest Neighbor (KNN), Hidden Markov Model, Fuzzy Logic Based System and Decision Trees. An extensive review is done on the existing and proposed models for credit card fraud detection and has done a comparative study on these techniques on the basis of quantitative measurements such as accuracy, detection rate and false alarm rate. The conclusion of our study explains the drawbacks of existing models and provides a better solution in order to overcome them.

Siddhant Bagga et.al [4] compares the performance of logistic regression, K-nearest neighbors, random forest, naive bayes, multilayer perceptron, ada boost, quadrant discriminative analysis, pipelining and ensemble learning on the credit card fraud data.

Vaishnavi Nath Dornadula et.al [5] aim to design and develop a novel fraud detection method for Streaming Transaction Data, with an objective, to analyse the past transaction details of the customers and extract the behavioural patterns. Then using sliding window strategy, to aggregate the transaction made by the cardholders from different groups so that the behavioural pattern of the groups can be extracted respectively. Later different classifiers are trained over the groups separately. And then the classifier with better rating score can be chosen to be one of the best methods to predict frauds.

Abdulsattar et.al [6] reviews the binary classification problem considered where the transaction can be a fraud or legitimate transaction. The goal is to classify transactions using five different machine learning algorithms, namely SGD, DT, RF, J48 and IBk. After applying classifiers the results are compared to see which algorithms works the best.

Ljiljana et.al [7] compares the performance of 3 machine learning algorithms which are RF, SVM, Logistic Regression. To mitigate imbalance class sizes, we use SMOTE sampling method. The performance of the algorithm is evaluated based on precision and recall values.

J. O. Awoyemi et.al [8] investigates the performance of naïve bayes, k-nearest neighbour and logistic regression on highly skewed credit card fraud data. Dataset of credit card transactions is sourced from European cardholders containing 284,807 transactions. A hybrid technique of under-sampling and oversampling is carried out on the skewed data

R. Sailusha et.al [9]aims to focus mainly on machine learning algorithms. The algorithms used are random forest algorithm and the Adaboost algorithm. The results of the two algorithms are based on accuracy, precision, recall, and F1-score. The ROC curve is plotted based on the confusion matrix. The Random Forest and the Adaboost algorithms are compared and the algorithm that has the greatest accuracy, precision, recall, and F1-score is considered as the best algorithm that is used to detect the fraud.

Selvani Deepthi Kavila et.al [10] evaluates and compares machine learning technique like Logistic regression, Decision Tree and Random forest which were used to detect the fraud in credit card system. Sensitivity, Specificity, accuracy and error rate are used to evaluate the performance for the proposed system. The accuracy for logistic regression, Decision tree and random forest classifier are 90.0, 94.3, and 95.5 respectively.

In ref. [11], the authors implemented a credit card fraud detection system using several ML algorithms including logistic regression (LR), decision tree (DT), support vector machine (SVM) and random forest (RF). These classifiers were evaluated using a credit card fraud detection dataset generated from European cardholders in 2013. In this dataset, the ratio between non-fraudulent and fraudulent transactions is highly skewed; therefore, this is a highly imbalanced dataset. The researcher used the classification accuracy to assess the performance of each ML approach. The experimental outcomes showed that the LR, DT, SVM and RF obtained the following accuracy scores: 97.70%, 95.50%, 97.50% and 98.60%, respectively. Although these outcomes are good, the authors suggested that the implementation of advanced pre-processing techniques could have a positive impact on the performance of the classifiers.

Varmedja et al. [12] proposed a credit card fraud detection method using ML The authors used a credit card fraud dataset sourced from Kaggle [19]. This dataset contains transactions made within 2 days by European credit card holders. To deal with the class imbalance problem present in the dataset, the researcher implemented the Synthetic Minority Oversampling Technique (SMOTE) oversampling technique. The following ML methods were implemented to assess the efficacy of the proposed method: RF, NB, and multilayer perceptron (MLP). The experimental results demonstrated that the RF algorithm performed optimally with a fraud detection

accuracy of 99.96%. The NB and the MLP methods obtained accuracy scores of 99.23% and 99.93%, respectively. The authors concede that more research should be conducted to implement a feature selection method that could improve on the accuracy of other ML methods.

Khatri et al. [13] conducted a performance analysis of ML techniques for credit card fraud detection. In this research, the authors considered the following ML approaches: DT, k-Nearest Neighbor (KNN), LR, RF and NB. To assess the performance of each ML method, the authors used a highly imbalanced dataset that was generated from European cardholders. One of the main performance metric that was used in the experiments is the precision which was obtained by each classifier. The experimental outcomes showed that the DT, KNN, LR, and RF obtained precisions of 85.11%, 91.11%, 87.5%, 89.77%, 6.52%, respectively.

Awoyemi et al. [14] presented a comparison analysis of different ML methods on the European cardholders credit card fraud dataset. In this research, the authors used an hybrid sampling technique to deal with the imbalanced nature of the dataset. The following ML were considered: NB, KNN, and LR. The experiments were carried out using a Python based ML framework. The accuracy was the main performance metric that was utilized to assess the effectiveness of each ML approach. The experimental results demonstrated that the NB, LR,and KNN achieved the following accuracies, respectively: 97.92%, 54.86%, and 97.69%. Although the NB and KNN performed relatively well, the authors did not explore the possibility to implement a feature selection method.

In ref. [16] the authors utilized several ML learning based methods to solve the issue of credit card fraud. In this work, the researchers used the European credit cardholder fraud dataset. To deal with the highly imbalanced nature of this dataset, the authors employed the SMOTE sampling technique. The following ML methods were considered: DT, LR, and Isolation Forest (IF). The accuracy was one of the main performance metrics that was considered. The results showed that the DT, LR, and IF obtained the accuracy scores of 97.08%, 97.18%, and 58.83%, respectively.

Manjeevan et al. [15] implemented an intelligent payment card fraud detection system using the GA for feature selection and aggregation. The authors implemented several machine learning algorithms to validate the effectiveness of their proposed method. The results demonstrated that the GA-RF obtained an accuracy of 77.95%, the GA-ANN achieved an accuracy of 81.82%, and the GA-DT attained an accuracy of 81.97%.

III. Proposed System

The Proposed system uses the Artificial Neural Network to find the fraud in the credit card transactions. Performance is measured and accuracy is calculated based on prediction. And also classification algorithms such as Support vector machine and k-Nearest Neighbor are used to build a credit card fraud detection model. We compare all the three algorithms used in the experiment and made a decision that artificial neural networks predicts well than system developed using support vector machine and k-nearest neighbour algorithms. The dataset used in the experiment consist of 31 attributes out of which 30 attributes consist of information related to name, age, account information and so on and last attribute give the outcome of the transaction in either 0 or 1.

ANN is biologically inspired by human brain. The neurons are interconnected in the human brain like the same nodes are interconnected in artificial neural network. depicts the structure of ANN with input, output and hidden layers. Inputs are x1, x2…Xn and output is y. w1…wn are the weights associated with inputs x1…xn respectively. There are 15 hidden layers used in this neural network. The activation function used in our credit card fraud detection model is RELU.

IV. Research Methodology

In this research, we use a dataset that includes credit card transactions that were made by European cardholders for 2 days in September 2013. This dataset contains 284807 transactions in total in which 0.172% of the transactions are fraudulent. The dataset has the following 30 features (*V1*,.., *V28*), *Time* and *Amount*. All the attributes within the dataset are numerical. The last column represents the class (type of transaction) whereby the value of 1 denotes a fraudulent transaction and the value of 0 otherwise. The features *V1* to *V28* are not named for data security and integrity reasons. This dataset has and one of the key issues that we discovered is the low detection accuracy score that was obtained by those models because of the highly imbalanced nature of the dataset. In order to solve the issue of class imbalance, we applied the Synthetic Minority Oversampling Technique (SMOTE) method in the Data-Preprocessing phase of the proposed framework. The SMOTE method works by picking samples that are close to each other within the feature space, drawing a line between the data points in the feature space and creating a new instance of the minority class at a point along the line.

The Genetic Algorithm (GA) is a type of Evolutionary inspired Algorithm (EA) that is often used to solve a number of optimization tasks with a reduced computational overhead. EAs generally possess the following attributes:

- **Population** EAs approaches maintain a sample of possible solutions called *population.*

- **Fitness** A solution within the population is called an *individual*. Each individual is characterized by a gene representation and a fitness measure.
- **Variation** The individual evolves through *mutations* that are inspired from the biological gene evolution.

In this study, the RF approach is used as the fitness method inside the GA. Further, the RF method is employed because it resolves the problem of over-fitting that is generally encountered when using regular Decision Trees (DTs). Moreover, RF performs well with both continuous and categorical attributes and RF are known to perform optimally on datasets that have a class imbalance problem. Additionally, the RF is a rule-based approach; therefore, the normalising of data is not required [17]. The alternative to the RF include tree-based ML algorithms such as Extra-Trees and Extreme Gradient Boosting [18, 19]. The fitness method is defined a function that receives a candidate solution (a feature vector) and determines whether it is fit or not. The measure of fitness is determined by the accuracy that is yielded by a particular attribute vector in the testing process of the RF method within the GA. Algorithm 1 provides more details about the implementation of RF in the GA.

V. Comparative Analysis

Decision Tree

DT = DecisionTreeClassifier(max_depth = 4, criterion = 'entropy') DT.fit(X_train, y_train) dt_yhat = DT.predict(X_test)

Let's check the accuracy of our decision tree model.

print('Accuracy score of the Decision Tree model is {}'.format(accuracy_score(y_test, tree_yhat)))**Accuracy score of the Decision Tree model is 0.999288989494457**

Checking F1-Score for the decision tree model.

print('F1 score of the Decision Tree model is {}'.format(f1_score(y_test, tree_yhat)))**F1 score of the Decision Tree model is 0.776255707762557**

Checking the confusion matrix:

confusion_matrix(y_test, tree_yhat, labels = [0, 1])

```
array([[68782,    18],
       [   31,    85]], dtype=int64)
```

Here, the first row represents positive and the second row represents negative. So, we have 68782 as true positive and 18 are false positive. That says, out of 68782+18=68800, we have 68782 that are successfully classified as a normal transaction and 18 were falsely classified as normal — but they were fraudulent.

Let's now try different models and check their performance.

*K-Nearest Neighbors*n = 7

KNN = KNeighborsClassifier(n_neighbors = n)KNN.fit(X_train,y_train) knn_yhat = KNN.predict(X_test)

Let's check the accuracy of our K-Nearest Neighbors model.

print('Accuracy score of the K-Nearest Neighbors{}'.format(accuracy_score(y_test,knn_yhat)))

Accuracy score of the K-Nearest Neighbors model is 0.999506645771664

Checking F1-Score for the K-Nearest Neighbors model.

print('F1 score of the K-Nearest Neighbors model is {}'.format(f1_score(y_test, knn_yhat)))

F1 score of the K-Nearest Neighbors model is 0.8365384615384616

Logistic Regression

lr=LogisticRegression()

lr.fit(X_train,y_train)

lr_yhat = lr.predict(X_test)

Let's check the accuracy of our Logistic Regression model.

print('Accuracy score of the Logistic Regression model is {}'.format(accuracy_score(y_test, lr_yhat)))

Accuracy score of the Logistic Regression model is 0.9991148644726914

Checking F1-Score for the Logistic Regression model.

print('F1 score of the Logistic Regression model is {}'.format(f1_score(y_test, lr_yhat)))**F1 score of the Logistic Regression model is 0.6934673366834171**

Support Vector Machines

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

Let's check the accuracy of our Support Vector Machines model.

```
print('Accuracy score of the Support Vector Machinesmodel is {}'.format(accuracy_score(y_test,svm_yhat)))
```

Accuracy score of the Support Vector Machines model is 0.9993615415868594

Checking F1-Score for the Support Vector Machines model.

```
print('F1 score of the Support Vector Machines model is {}'.format(f1_score(y_test, svm_yhat)))
```

F1 score of the Support Vector Machines model is 0.7777777777777779

Random Forest

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

Let's check the accuracy of our Random Forest model.

```
print('Accuracy score of the Random Forest model is {}'.format(accuracy_score(y_test, rf_yhat)))
```

Accuracy score of the Random Forest model is 0.9993615415868594

Checking F1-Score for the Random Forest model.

```
print('F1 score of the Random Forest model is {}'.format(f1_score(y_test, rf_yhat)))
```

F1 score of the Random Forest model is 0.7843137254901961

XGBoost

```
xgb = XGBClassifier(max_depth = 4)
xgb.fit(X_train, y_train)
xgb_yhat = xgb.predict(X_test)
```

Let's check the accuracy of our XGBoost model.

```
print('Accuracy score of the XGBoost model is {}'.format(accuracy_score(y_test, xgb_yhat)))
```

Accuracy score of the XGBoost model is 0.9995211561901445

Checking F1-Score for the XGBoost model.

```
print('F1 score of the XGBoost model is {}'.format(f1_score(y_test, xgb_yhat)))
```

F1 score of the XGBoost model is 0.8421052631578947.

VI. Conclusion

In this research, a GA based feature selection method in conjunction with the RF, DT, ANN, NB, and LR was proposed. The GA was implemented with the RF in its fitness function. The GA was further applied to the European cardholders credit card transactions dataset and 5 optimal feature vectors were generated. The experimental results that were achieved using the GA selected attributes demonstrated that the GA-RF (using v5v5) achieved an overall optimal accuracy of 99.98%. Furthermore, other classifiers such as the GA-DT achieved a remarkable accuracy of 99.92% using v1v1. The results obtained in this research were superior to those achieved by existing methods. Moreover, we implemented our proposed framework on a synthetic credit card fraud dataset to validate the results that were obtained on the European credit card fraud dataset. The experimental outcomes showed that the GA-DT obtained an AUC of 1 and an accuracy of 100%. Seconded by the GA-ANN with an AUC of 0.94 and an accuracy of 100%. In future works, we intend to use more datasets to validate our framework.

References

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

[2]. Asha RB, Suresh Kumar KR, "*Credit card fraud detection using artificial neural network* ", Global Transitions Proceedings,Volume 2, Issue1,2021,Pages3541,ISSN2666285X,https://doi.org/10.1016/j.gltp.2021.01.006.

[3]. AndhavarapuBhanusri, "*Credit card fraud detection using Machine learning algorithms*", Quest Journals Journal of Research in Humanities and Social Science, vol. 08(02), 2020, pp. 04-11.

[4]. Siddhant Bagga, Anish Goyal, Namita Gupta, Arvind Goyal, "*Credit Card Fraud Detection using Pipeling and Ensemble Learning*", Procedia Computer Science,Volume 173,2020,Pages 104-112,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2020.06.014.

[5]. Vaishnavi Nath Dornadula, S Geetha,"*Credit Card Fraud Detection using Machine Learning Algorithms*", Procedia Computer Science, Volume165,2019,Pages631641,ISSN18770509,https://doi.org/10.1016/j.procs.2020.01.057(https://www.sciencedirect.com/science/article/pii/S187705092030065X).

[6]. Abdulsattar, Khadija & Hammad, Mustafa,"*Fraudulent Transaction Detection in FinTech using Machine Learning Algorithms*", 10.1109/3ICT51146.2020.9312025.

[7]. Puh, Maja &Brkic, Ljiljana, " *Detecting Credit Card Fraud Using SelectedMachineLearningAlgorithms*",12501255.10.23919/MIPRO.2019.8757212.

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

[9]. R. Sailusha, V. Gnaneswar, R. Ramesh and G. R. Rao, "*Credit Card Fraud Detection Using Machine Learning*", 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020, pp. 1264-1270, doi: 10.1109/ICICCS48265.2020.9121114.
[10]. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 24 (2018) pp. 16819-16824 © Research India Publications.http://www.ripublication.com
[11]. 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.
[12]. 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.
[13]. 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.
[14]. Awoyemi JO, Adetunmbi AO, Oluwadare SA. Credit card fraud detection using machine learning techniques: a comparative analysis. In: International conference on computer networks and Information (ICCNI); 2017. p. 1-9.
[15]. Seera M, Lim CP, Kumar A, Dhamotharan L, Tan KH. An intelligent payment card fraud detection system. Ann Oper Res 2021;1–23.
[16]. Dornadula VN, Geetha S. Credit card fraud detection using machine learning algorithms. Proc Comput Sci. 2019;165:631–41.
[17]. Khalilia M, Chakraborty S, Popescu M. Predicting disease risks from highly imbalanced data using random forest. BMC Med Inf DecisMak. 2011;11(1):1–13.
[18]. Abhishek L. Optical character recognition using ensemble of SVM, MLP and extra trees classifier. In: International conference for emerging technology (INCET) IEEE; 2020. p. 1–4.
[19]. Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H. Xgboost: extreme gradient boosting. R package version 04-2. 2015;1(4):1–4.