2022 IEEE International Conference on Data Science and Information System (ICDSIS)

Credit Card Fraud Detection Web Application using Streamlit and Machine Learning

1st Vipul Jain
Department of ISE
Siddaganga Institute of Technology
Tumakuru, India
vipuljain.hp@gmail.com

2nd Kavitha H
Department of ISE
Siddaganga Institute of Technology
Tumakuru, India
hkavitha@sit.ac.in

3rd Mohana Kumar S
Department of CSE
Ramaiah Institute of Technology
Bengaluru, India
mohanks@msrit.edu

2022 IEEE International Conference on Data Science and Information System (ICDSIS) | 978-1-6654-9801-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDSIS55133.2022.9915901

Abstract—**Credit Card Fraud is one of the major threads in the financial industry. Due to the covid-19 pandemic and the advance in technologies, the number of users is increasing, with the increased use of credit cards. Due to more use of credit cards, Fraud cases also increase day by day. The research community striving hard to explore myriad credit card fraud detection techniques, but changes in technology and the varying nature of credit card fraud make it difficult to develop an effective technique for the detection of credit card fraud. This research work used a real-world credit card dataset. To detect the fraud transaction within this dataset three machine learning algorithms are used (i.e. Random Forest, Logistic regression, and AdaBoost) and compared the machine learning algorithms based on their Accuracy and Mathews Correlation Coefficient (MCC) Score. In these three algorithms, the Random Forest Algorithm achieved the best Accuracy and MCC score. The Streamlit framework is used to create the machine learning web application.**

Keywords—Credit Card Fraud Detection, Machine Learning, Streamlit, Accuracy, MCC Score.

I. Introduction

Online money transaction increases day by day. All countries focus on digital money transactions and also focus on the security of the transaction. But day by day fraud occurs and increases the loss of money. According to Nilson Report 2021, out of $47.229 trillion transactions,$32.20 billion in fraud transactions were reported, and expected fraud cases will increase up to $49.32 in 2030 [1].In online mode payments, the expiry date of the card, card CVV, card number and additionally OTP is required for the transaction of money. Sometimes, the people or cardholders may lose their money through physical theft of mobile and card. In some other cases, people may lose their money through phishing links by entering their personal information on the phishing website.

From social engineering, the attacker uses the psychological trick to mind wash the cardholder and take the personal information, and uses for the money theft.

The financial institutes also responsible for the protection of the money and should focus on the customer transaction behaviours and based on their previous money transactions history predict the transaction fraud or not. To do that machine learning is the best technique to predict whether the transaction is fraud or not. The financial organization network thread also makes a money loss for both organization and the customers [2] throws cyber- attack.

The detection of deceitful transactions in the larger dataset is a challenging task. To train the model data plays a major role and to predict the fraudulent transaction we need to confirm which algorithm is best suited for larger detect the fraudulent transactions in the larger transaction and predict accurately with the limited amount ofmisclassification.

In this research work, the Kaggle credit card dataset is used. This dataset is also called the European credit card dataset. Based on the algorithm performance in the literature survey papers, we used three algorithms in our research work to implement the credit card fraud detection webapplication. In the credit card fraud detection-related work mentioned in this paper, the researcher did not implement machine learning-based web applications for fraud detection. In this research work, we used the Streamlit framework to implement the machine learning-based credit card fraud detection web application.

II. Related Works

This section provides a related work review of previous research that used Machine Learning technique for credit card fraud detection.

In [3], the researcher proposed a credit card fraud detection engine using an AdaBoost and Majority Voting. In this research work, the researchers used a publicly available European dataset. First, tested the individual algorithm's accuracy and MCC score. Then combining each algorithm with the AdaBoost improves the model performance. The majority voting introduced and tested the performance by adding the noise to the dataset. In Majority voting the Decision Tree + Gradient boosting technique achieved a 1.000 MCC score and Naive-Bias with AdaBoost also achieved a 1.000 MCC score. After adding the noise also the accuracy rate of the combined algorithms achieved a stable result.

In [4], the researchers used machine learning algorithms to detect the fraudulent activities in the real- world dataset. First, under-sampled the data and then applied algorithms to that data. The random forest achieved 96.77 % accuracy.

In [5], the researcher’s implemented a machine learning framework to detect credit card fraud in the real-world larger dataset. To manage the imbalanced dataset first re- sampled the data using the Synthetic Minority over- sampling Technique (SMOTE), then divided data into training and testing data. For training Random Forest (RF), Extreme Gradient Boosting (XGBoost) and Extra Tree (ET) algorithms are used in the Phase-1 experiment. In Phase-2, AdaBoost Algorithm is used to boost the performance of the phase-1 used algorithms. Finally, the ET-AdaBoost and XGB-AdaBoost achieved a 0.99 MCC score, and all the algorithms with AdaBoost achieved above 99% accuracy.

In [6], the researcher implemented Credit Card Fraud detection method using: k-Nearest Neighbor (kNN) and other machine learning algorithms. The European cardholder’s dataset is used to evaluate the performance of the dataset. The kNN algorithm achieved 91.11% precision.

978-1-6654-9801-2/22/$31.00 ©2022 IEEE

In [7], the researcher used European cardholder's dataset. To detect the fraudulent transaction in the dataset, hybrid technique of over-sampling and the under-sampling is carried out then the logistic regression, k-nearest neighbours and the naïve bayes techniques are applied on the raw and pre-processed data. The k-nearest neighbors algorithm achieved better result i.e accuracy 97 %.

In [8], the researcher used European credit card holder's dataset. Then applied Synthetic Minority Over-sampling Technique (SMOTE) to Over-sample the dataset. Then split the data into training and testing data. To train the model, the Logistic Regression, Random Forest, Naive Bayes and Multilayer Perceptron (MLP) algorithms are used. The MLP and Random Forest achieved 99 % accuracy.

In [9], the researcher used real- dataset that is collected from the e-commerce company of China; the original dataset contains over 30,000,000 individual transactions. This transaction happens between November 2016 to January 2017. The dataset are divided into training and testing dataset. Fore training 70% and for testing 30%. To train the model Random Forest algorithm are used and achieved 97.77% accuracy.

In [10], the researchers used Support Vector Machine, Decision tree and the Random Forest algorithms to detect the fraudulent transaction in the dataset. The main objective of this research is to detect credit card fraud in the real world. Using above mentioned algorithms the researcher achieved the best accuracy rate in Random Forest algorithm.

III. Methods

A. Dataset

The dataset used in this research work is the European cardholder dataset, which is publicly available through Kaggle [11]. This dataset has the transactions made by credit cards in September 2013 by credit cardholders. These are the two-day transaction information in it. This dataset has 492 frauds out of 2,84,807 transactions. The fraud class in this dataset is 0.172% and normal transactions are 99.828%. It has the 30 features V1, V2…V28, time, and amount.

Table I. Credit Card Dataset

Time	V1	V2	.	Amount	Class
0	-1.35981	-0.07278	.	149.62	0
0	1.191857	0.266151	.	2.69	0
1	-1.35835	-1.34016	.	378.66	0
1	-0.96627	-0.18523	.	123.5	0
2	-1.15823	0.877737	.	69.99	0
2	-0.42597	0.960523	.	3.67	0
4	1.229658	0.141004	.	4.99	0

The features from V1 to V28 are in numerical form. The last column has the class column that has the values 0 and 1. Class 0 is the legitimate transaction and Class 1 represents the fraudulent transaction. The Table I shows the dataset information.

- The dataset cleaning already made by the Kaggle dataset provider, for this dataset no need to do data cleaning. In this dataset no duplicates and no missing values are there.
- The dataset is in .csv format.
- The type conversion is also not required for this dataset. The dataset is already in numerical value 0 or 1. The class 0 represents the legitimate transaction and the class 1 represents the fraudulent transaction.

B. Machine Learning Algorithm

1) Logistic Regression (LR): Logistic Regression is the machine learning algorithm used for classification problems [12].It is called classification algorithm, because it gives output like true or false, Yes or No and 0 or 1. Instead of giving 0 or 1 value which gives the probabilistic value which lies between 1 and 0.

The eqn (1) shows the principle and how the logistic regression algorithm works. The b_0, b_1 …. b_n are coefficients and x_1, x_2… x_n are the independent variables and p is the outcome.

$$p = \frac{1}{1+e^{-(b_0+b_1x_1+b_2x_2+...+b_nx_n)}} \qquad (1)$$

2) Random Forest (RF): Random Forest is the ensemble Machine Learning algorithm. The Random Forest is called ensemble algorithm, because it constructs the multiple decision tree and takes the majority voting method for the classification problem and it takes the average for the regression problem. It is used for both regression and classification problems [13].

3) *AdaBoost:* AdaBoost is an ensemble method, which is help to increase the efficiency of binary classifiers [14]. AdaBoost uses a repetitive approach to learn from the mistakes of weak classifiers, and turn them into strong ones. The AdaBoost algorithm represented in (2).

$$\mathrm{G}_N(x) = \sum_{i=1}^{N} g_t(x) \qquad (2)$$

Where g_t is the weak learner, that outputs a prediction given an input vector x.

C. Streamlit

Streamlit is an application framework, which helps to create a machine learning web application in an easy way. It is well-matched with main python libraries that are used in the machine learning for data visualization and analysis [15].

D. Experimental Setup

The classification experiment is conducted on the visual studio code. The Streamlit framework is used to create a machine learning web application.

E. Performance Metrics

The experiment performance is evaluated using the accuracy (AC), recall (RC), precision (PR), Matthews correlation coefficient (MCC), and confusion matrix.

The mathematical formula of this indicator is as follows:

- False positive (FP): The normal transactions that are incorrectly labelled as fraudulent.
- False Negative (FN): The fraudulent transaction that is incorrectly labelled as normal transaction.
- True positive (TP): The fraudulent transaction is accurately classified as a fraudulent transaction.
- True Negative (TN): The legitimate transactions are positively classified as the legitimate transaction.

1) Accuracy: The accuracy is a measure used to determine which model is best for identifying relationships and patterns between database variations based on input, or training, data. eqn (3) shows the precision formula.

$$Accuracy = \frac{TN+TP}{TP+TN+FN+FP} \qquad (3)$$

2) Precision: precision indicates the quality of a positive prediction made by the model. The Precision eqn (4) shows the Precision formula.

$$Preceision = \frac{TP}{TP+FP} \qquad (4)$$

3) Recall: The recall measures the model's ability to detect positive samples. If the recall is higher it means the more positive classes are detected. The eqn (5) shows the recall formula.

$$Recall = \frac{TP}{TP+FN} \qquad (5)$$

4) Matthews correlation coefficient (MCC): The Matthews correlation coefficient (MCC), produces a high score only if the prediction obtained good results in all of the four confusion matrix categories (i.e, TP, FN, TN, and FP). The eqn (6) shows the MCC formula.

$$\mathrm{MCC} = \frac{(\mathrm{TP}\times\mathrm{TN})-(\mathrm{FP}\times\mathrm{FN})}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}} \qquad (6)$$

F. Credit Card Fraud Detection Workflow

The Fig. 1 shows the workflow of the Credit Card fraud detection. The Credit Card Fraud Detection Workflow as discussed below:

- Firstly, the credit card dataset is taken form the Kaggle dataset. The Kaggle dataset is in the .csv format and all the values in the dataset are not having any duplicate and null value in it. So this dataset with the numerical value type conversion is not required. The Fig. 2 shows the credit card dataset.
- Split the data for training and testing. In this project 80% of the data taken for training and 20% of data is taken for testing.
- To train the model Logistic Regression, Random Forest and AdaBoost algorithm are used.
- After model is trained with 80% data testing is done with 20% data to check the fraudulent transactions.
- After testing and detecting the fraudulent transaction in the dataset. The module is deployed in the form of web application using the Streamlit framework.

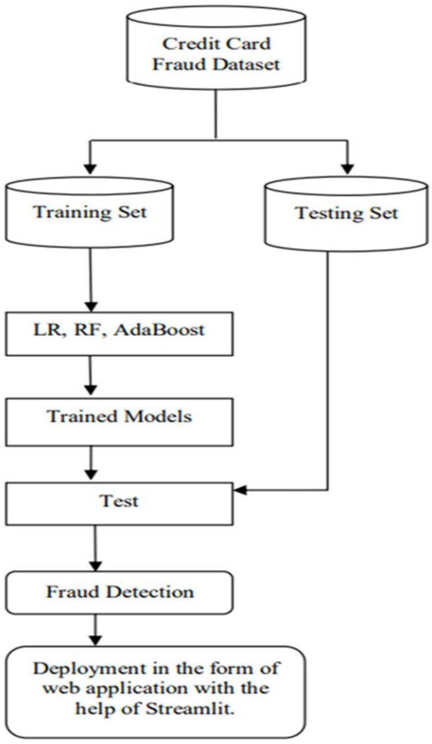
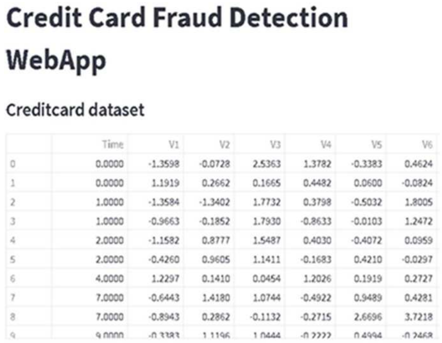

Credit Card Fraud Dataset
Training Set
Testing Set
LR, RF, AdaBoost
Trained Models
Test
Fraud Detection
Deployment in the form of web application with the help of Streamlit.

Fig. 1. Credit Card Fraud Detection Workflow



Credit Card Fraud Detection WebApp

Creditcard dataset

	Time	V1	V2	V3	V4	V5	V6
0	0.0000	-1.3598	-0.0728	2.5363	1.3782	-0.3383	0.4624
1	0.0000	1.1919	0.2662	0.1665	0.4482	0.0600	-0.0824
2	1.0000	-1.3584	-1.3402	1.7732	0.3798	-0.5032	1.8005
3	1.0000	-0.9663	-0.1852	1.7930	-0.8633	-0.0103	1.2472
4	2.0000	-1.1582	0.8777	1.5487	0.4030	-0.4072	0.0959
5	2.0000	-0.4260	0.9605	1.1411	-0.1683	0.4210	-0.0297
6	4.0000	1.2297	0.1410	0.0454	1.2026	0.1919	0.2727
7	7.0000	-0.6443	1.4180	1.0744	-0.4922	0.9489	0.4281
8	7.0000	-0.8943	0.2862	-0.1132	-0.2715	2.6696	3.7218
9	9.0000	-0.3383	1.1196	1.0444	-0.2222	0.4994	-0.2468

Fig. 2. Credit Card Dataset Loaded in Web Application.

IV. Results And Discussion

As shown in Fig. 2, In the first step we collected a real-world dataset from the Kaggle dataset. Then separate the data into training and testing parts. In the next step, we trained the machine learning model with the help of the training dataset. In this research work used random forest, logistic regression, and AdaBoost algorithm to train the model. After training a model we evaluated each algorithm or model performance with the help of the accuracy, recall, precision, confusion matrix, and the MCC score.

The Streamlit framework is used to build a Web Application for this machine learning experiment. Fig. 3 shows the Credit Card Fraud Detection Web Application.

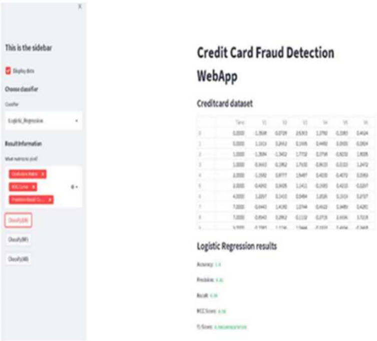
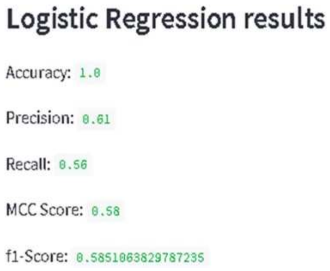


Fig. 3. Credit Card Fraud Detection Web Application

The after performing the experiment using machine learning algorithms, the performance of the algorithms are evaluated. The comparing to Logistic Regression and the AdaBoost algorithms the Random Forest algorithm achieved better accuracy and MCC score.

Each algorithm performance is evaluated using accuracy, MCC score, Precision, recall. The confusion matrix is used to know the TP, FN, TP and TN.

The Logistic Regression accomplished an accuracy of 100%, precision 61%, recall 56%, MCC score of 58%, f1-score 58%. The accuracy and other performance results are shown in Fig. 4.



Logistic Regression results

Accuracy: 1.0

Precision: 0.61

Recall: 0.56

MCC Score: 0.58

f1-Score: 0.5851063829787235

Fig. 4. Logistic Regression Result.

Fig. 5 Shows the Logistic Regression confusion matrix.

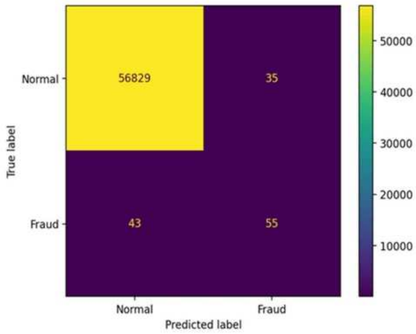
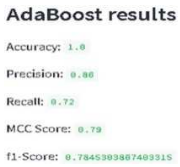


Fig. 5. Logistic Regression Confusion Matrix.

Fig. 6, shows, The AdaBoost algorithm results, the accuracy is 100 %, precision 86 %, recall 72 % and MCC score 79 %, f1- score is 78 %.



AdaBoost results

Accuracy: 1.0

Precision: 0.86

Recall: 0.72

MCC Score: 0.79

f1-Score: 0.7845303867403315

Fig. 6. AdaBoost algorithm result.

Fig. 7, Shows the AdaBoost algorithm confusion matrix.

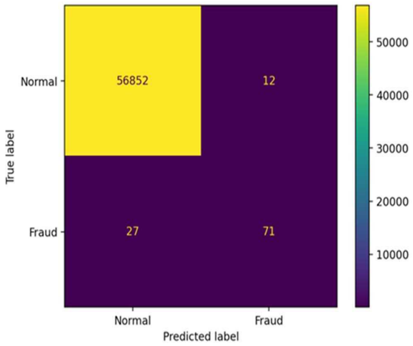
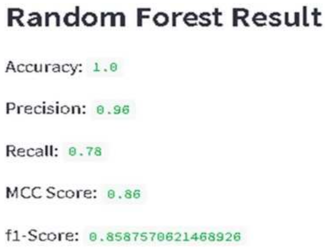


Fig. 7. AdaBoost Algorithm Confusion Matrix

The Random Forest algorithm achieved accuracy 100%, precision 96%, recall 78%, MCC score 86%, F1-score is 85.87 %, and the result of the random forest algorithm is shown in Fig. 8. Fig. 9 Shows the Random Forest algorithm confusion matrix.



Random Forest Result

Accuracy: 1.0

Precision: 0.96

Recall: 0.78

MCC Score: 0.86

f1-Score: 0.8587570621468926

Fig. 8. Random Forest Result.

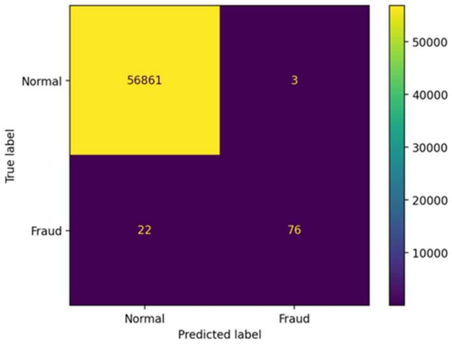


Fig. 9. Random Forest Confusion Matrix

V. Conclusion

In this research work implemented a credit card fraud detection Web Application using Machine Learning algorithms such as Logistic Regression, Random Forest, and AdaBoost. The Streamlit frameworks are used to build a Web Application. The performance of the algorithms is compared. In these algorithms, the Random Forest algorithm achieved 100 % accuracy, 96 % precision, 78 % recall, 85 % f1-score and 86 % MCC score. All the algorithm accuracy scores are similar, but algorithms have a difference in the MCC score due to some misclassifications in the results. In future works, we intend to use the majority voting method for the proposed framework and focusing on controlling misclassifications.

References

[1] Robertson, "Credit Card Fraud Nilson Report," Nilson Report, 2021.

[2] D. P, M. K. S, C. Raghavendra, K. P. SJ, K. H and D. SS, "Cyber Security Threats Detection Analysis and Remediation," in IEEE, 2021.

[3] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim and A. K. Nandi, “Credit Card Fraud Detection Using AdaBoost and Majority Voting,” IEEE, vol. 6, pp. 14277 - 14284, 2018.

[4] D. Tanouz, R. R. Subramanian, D. Eswar, “Credit Card Fraud Detection Using Machine Learning,” in IEEE, 2021.

[5] E. Ileberi, Y. Sun and Z. Wang, “Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost,” IEEE, vol. 9, no. 5, pp. 165286 - 165294, 2021.

[6] S. Khatri, A. Arora and A. P. Agrawal, “Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison,” in IEEE, 2020.

[7] J. O. Awoyemi, A. O. , “Credit card fraud detection using machine learning techniques: A comparative analysis,” in IEEE, 2017.

[8] M. K. Dejan Varmedja, “Credit Card Fraud Detection - Machine Learning methods,” in IEEE Xplore, 2019.

[9] [9] G. L. S. Xuan, “Random forest for credit card fraud detection,,” in IEEE, 15th International Conference on Networking, Sensing and Control (ICNSC), 2018.

[10] E. S. C. R. S K Saddam Hussain, “Fraud Detection in Credit Card Transactions Using SVM and Random Forest Algorithms,” in IEEE Xplore, 2021.

[11] Credit Card Fraud Detection Accessed: Jan. 20, 2022. [Online].Available:https://www.kaggle.com/mlgulb/creditcardfrad.

[12] Itoo, F. and Singh, S., 2021. Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. International Journal of Information Technology, 13(4), pp.1503-1511.

[13] M. S. Kumar, V. Soundarya, S. Kavitha, E. Keerthika and E. Aswini, “Credit Card Fraud Detection Using Random Forest Algorithm,” in IEEE, 2019.

[14] Ying, C., Qi-Guang, M., Jia-Chen, L. and Lin, G., 2013. Advance and prospects of AdaBoost algorithm. Acta Automatica Sinica, 39(6), pp.745-758.

[15] Singh, P., 2021. Machine learning deployment as a web service. In Deploy Machine Learning Models to Production (pp. 67-90). Apress, Berkeley, CA.