International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 09 | Sept - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Credit Card Fraud Detection Using Machine Learning

Madhuri R. Gangwe[1], Dr Smita S Ponde[2]

1Department of Computer Engineering, DIEMS, BATU University, Lonere (M. S) India
1Department of Computer Engineering, DIEMS, BATU University, Lonere (M. S) India

---

Abstract -The aim of the project is to restrict fraudsters from using consumers' accounts for unauthorised purposes by using machine learning algorithms to detect fraudulent credit card transactions. Because credit card fraud is becoming more and more commonplace globally, steps need to be done to put an end to those who commit fraud. Setting a restriction on those activities will benefit the clients since the project's primary objective will be achieved—their money will be recovered and refunded into their accounts, and they won't be charged for goods or services that they did not acquire. Three machine learning techniques—KNN, SVM, and Logistic Regression—will be used to detect fraudulent transactions. The models are going to be applied to a dataset of credit card transactions.

Key Words: Credit Card Fraud Detection, Fraud Detection, Fraudulent Transactions, K-Nearest Neighbors, Support Vector Machine, Logistic Regression, NaïveBayes.

1.INTRODUCTION

With the rise in the number of people using credit cards on a daily basis, credit card firms have to pay more attention to their clients' protection and safety. A total of 2.8 billion credit cards were used globally in 2019, and 70% of those users had at least one card, according to Credit Card Statistics 2021.
In the US, there were 393,207 reports of credit card fraud in 2020 compared to 271,927 incidents in 2019, an increase of 44.7%. There are two types of credit card theft. The first involves identity thieves opening a credit card account in your name; between 2019 and 2020, complaints of this fraudulent activity rose by 48%.The second kind involve an identity thief using an account you already have, typically by using your credit card information. Reports of this kind of theft rose by 9% between 2019 and 2020 (Daly, 2021).
These statistics drew my attention because they show a sharp increase in the numbers over time. This motivated me to attempt an analytical solution by employing various machine learning techniques to identify fraudulent credit card transactions among a large number of transactions.

2. Literature Review

In order to identify the model that performed the best in identifying fraudulent transactions, Zareapoor and his research team employed a variety of methodologies. The model's accuracy, detection speed, and cost were taken into consideration. Neural networks, Bayesian networks, SVM, KNN, and other models were employed. The study paper's comparison table demonstrated how quickly and accurately the Bayesian Network identified fraudulent transactions. The NN functioned well and detected objects quickly while maintaining a medium level of accuracy. KNN achieved an excellent speed and medium accuracy, whereas SVM, with a poor speed and medium accuracy, had one of the lowest ratings. Regarding the price every model constructed was large [1].

Alenzi and Aljehane employed a Logistic Regression model with a 97.2% accuracy, 97% sensitivity, and 2.8% error rate to identify credit card fraud. Their model and the Voting Classifier and KNN, two further classifiers, were compared. In contrast, with KNN, when k = 1:10, the model's accuracy was 93%, its sensitivity was 94%, and its error rate was 7%. VC scored 90% in accuracy, 88% in sensitivity, and 10% in error rate [2].

In addition to aiming to reduce the number of fraudulent transactions that are mistakenly labelled, Maniraj's team developed a model that can determine if a new transaction is fraudulent or not. They were able to identify 99.7% of the fraudulent transactions, demonstrating the effectiveness of their methodology [3].

Using Support Vector Machine, Dheepa and Dhanapal employed a behavior-based classification approach in which they examined customer behavioral patterns such as the quantity, date, time, location, and frequency of card usage to differentiate between instances of credit card fraud. Their method produced an accuracy of greater than 80% [4].

In order to detect credit card fraud, Mailini and Pushpa suggested using KNN and outlier detection. After testing their model on sampled data, the authors discovered that KNN is the most effective method for identifying target instance anomalies and for detecting fraud with memory limitations. Regarding outlier detection, it requires a lot less memory and compute and operates faster and more effectively in online huge datasets for credit card fraud detection. However, their

research and findings demonstrated that KNN was more precise and effective [5].

Maes and his group suggested employing neural networks and Bayesian algorithms to detect credit card fraud. According to their findings, Bayesian performance has 8% higher fraud detection efficiency than ANN, which implies that sometimes BBN identifies 8% more fraudulent transactions. Aside from the learning periods, ANN can take many hours, while BBN simply needs twenty minutes [6].
The Awoyemi team examined the application of three machine learning techniques KNN, Naïve Bayes, and logistic regression in the identification of credit card fraud. To see the varied results, they sampled from several distributions. Naïve Bayes has the highest accuracy of the 10:90 distributions at 97.5%, followed by KNN at 97.1%.With an accuracy of only 36.4%, logistic regression did not perform well. They observed that a different distribution, 34:66, was also examined. KNN appeared first on the chart, showing a marginal improvement in accuracy to 97.9%, followed by Naïve Bayes with 97.6%. Logistic Regression fared better in this distribution, showing an increase in accuracy to 54.8% [7].

To identify credit card fraud, Jain's team employed a number of machine learning algorithms, including three: SVM, ANN, and KNN. They then computed the true positive (TP), false negative (FN), false positive (FP), and true negative (TN) generated in order to compare the results of each model. 99.71% accuracy, 99.68% precision, and 0.12% false alarm rate were achieved with ANN.
SVM accuracy is 94.65%, precision is 85.45%, and false alert rate is 5.2%. Lastly, the KNN has an accuracy of 97.15%, precision of 96.84%, and a false alarm rate of 2.88% [8].

Gupta and his colleagues focused on putting into practice an automated model that employs several machine learning approaches to identify fraudulent occurrences that are economically tied to consumers, with a focus on credit card transactions. Among all the methods they employed, Naïve Bayes demonstrated exceptional performance in identifying fraudulent transactions, with an accuracy of 80.4% and an area under the curve of 96.3% [9].

Adepoju and his colleagues employed a variety of machine learning techniques, including logistic regression; support vector machines (SVMs), naive bayes, and KNNs (K-Nearest Neighbours), on skewed credit card fraud data. All of the models had accuracy scores of 99.07% for Logistic Regression, 95.98% for Naïve Bayes, 96.91% for K-nearest neighbor, and 97.53% for the final model, Support Vector Machine (SVM) [10].

3. Project Description

3.1 Introduction

Several actions must be completed in order to fulfill the project's purpose, which is to identify the best model for detecting credit card fraud. The first two processes are selecting the best available data and preparing/preprocessing it. Once the data is ready, the modeling phase begins with the creation of four models: K-Nearest Neighbour (KNN), Naïve Bayes, SVM, and Logistic Regression. Two Ks, K = 3 and K = 7, were selected for the KNN model. With the exception of SVM, which was generated only in Weka, all models were constructed using both R and Weka software. All visualizations were also created using both programmers.

3.2 Data Source

The dataset was obtained from Kaggle.com, an open-source website. It includes information on transactions performed in only two days in 2013 by European credit card users. There are 284,808 rows and 31 characteristics in the dataset. The remaining three attributes are "Time," which contains the amount of each transaction, "Amount," which is the amount of each transaction, and the final attribute "Class," which contains binary variables where "0" indicates that a transaction is not fraudulent and "1" indicates that a fraudulent transaction occurs. The 28 attributes are numeric variables that have been transformed using PCA transformation due to the confidentiality and privacy of the customers.

4. Data Analysis

4.1 Data Preparation

The dataset's structure is depicted in the first figure below, which also includes a glimpse of each attribute's variables. As can be seen at the end of the figure, the Class type is integer, so I had to change it to identify the 0 as Not Fraud and the 1 as Fraud in order to make the process of building the model and obtaining visualizations easier

#	Column	Non-Null Count	Dtype
0	Time	284807 non-null	float64
1	V1	284807 non-null	float64
2	V2	284807 non-null	float64
3	V3	284807 non-null	float64
4	V4	284807 non-null	float64
5	V5	284807 non-null	float64
6	V6	284807 non-null	float64
7	V7	284807 non-null	float64
8	V8	284807 non-null	float64
9	V9	284807 non-null	float64
10	V10	284807 non-null	float64
11	V11	284807 non-null	float64
12	V12	284807 non-null	float64
13	V13	284807 non-null	float64
14	V14	284807 non-null	float64
15	V15	284807 non-null	float64
16	V16	284807 non-null	float64
17	V17	284807 non-null	float64
18	V18	284807 non-null	float64
19	V19	284807 non-null	float64
20	V20	284807 non-null	float64
21	V21	284807 non-null	float64
22	V22	284807 non-null	float64
23	V23	284807 non-null	float64
24	V24	284807 non-null	float64
25	V25	284807 non-null	float64
26	V26	284807 non-null	float64

Figure 1 - Dataset Structure

4.2 Data Preprocessing

The dataset was prepared simply because there are no NAs or duplicate variables. The first change made to enable the dataset to be opened on the Weka program was to change the class attribute's type from Numeric to Class and use the

Sublime Text program to identify the class as {1,0}. To be able to generate the model and the visualization, another change was made to the type in the R program.

4.3 Data Modeling

Weka and R were used to generate the four models after ensuring that the data was prepared for modeling. Weka alone was used to generate the SVM model; R plus Weka were used to create KNN, Naïve Bayes, and Logistic Regression.

5. Results

5.1 KNN

I made the decision to construct two models with K=3 and K=7 while building the KNN model. The R-created model is depicted in Figure 5, where it achieved an accuracy of 99.83% and successfully identified 91,719 transactions while missing 155. Regarding the Weka program, the model misclassified 52 transactions and had an accuracy score of 99.94%.

The average accuracy is 99.89% since there are variations in accuracy.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	56656
1	0.90	0.73	0.81	90
accuracy			1.00	56746
macro avg	0.95	0.87	0.90	56746
weighted avg	1.00	1.00	1.00	56746

Figure 2 - Weka K=3

5.2 Naïve Bayes

Figure 9 displays the results of the Naïve Bayes model, which is the second model developed by R. It misclassified 2,051 transactions overall, misclassifying 33 fraudulent as nonfraudulent and 2018 nonfraudulent as fraudulent. The model's accuracy was 97.77%. The Naïve Bayes model developed in Weka has a slightly different accuracy of 97.73% and 1,938 misclassification incidents.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	56656
1	0.90	0.63	0.75	90
accuracy			1.00	56746
macro avg	0.95	0.82	0.87	56746
weighted avg	1.00	1.00	1.00	56746

Figure 3 - Weka Naïve Bayes

5.3 Logistic Regression

The final model developed with Weka and R is called Logistic Regression. It achieved 99.92% accuracy in R (figure 11) with 70 misclassified examples, and 99.91% accuracy in Weka (figure 10), with 77 misclassified instances.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	56656
1	0.95	0.70	0.81	90
accuracy			1.00	56746
macro avg	0.98	0.85	0.90	56746
weighted avg	1.00	1.00	1.00	56746

Figure 4 - Weka Logistic Regression

5.4 Support Vector Machine

Eventually, the Support Vector Machine model, as seen in Figure 12, achieved an accuracy score of 99.94% while misclassifying 51 cases.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	56656
1	0.98	0.64	0.78	90
accuracy			1.00	56746
macro avg	0.99	0.82	0.89	56746
weighted avg	1.00	1.00	1.00	56746

Figure 5 - Support Vector Machine

The total number of properly predicted occurrences is known as accuracy. Accuracy is displayed as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) in a confusion matrix. The term "True Positive" refers to transactions that the model properly identified as fraudulent.

The term "True Negative" denotes transactions that the model properly anticipated to be not fraudulent. False positive, the third rating, denotes transactions that are fraudulent but were mistakenly identified as non-fraudulent. Lastly, the confusion matrix is displayed in Table 1 below for False Negative, which are the non-fraudulent transactions that were mistakenly classified as such.

Actual/Predicted	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Table 1 - Confusion Matrix

The table above shows all the components to calculate an accuracy of a model which is displayed in the below equation.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Model		Accuracy
KNN	k=3	99.89%
	K=3	
	K=7	99.88%
	K=7	
Naïve Bayes	Naïve Bayes	97.76%
	Naïve Bayes	
Logistic Regression	Logistic Regression	99.92%
	Logistic Regression	
Support Vector Machine	SVM	99.94%

Table 2 - Table of Accuracies

Table 2 displays the accuracy scores of every model developed for the study. Every model achieved high accuracy scores and demonstrated strong performance in identifying fraudulent transactions. With an accuracy of 99.94%, Support Vector Machine is the model that performed the best out of all of the models; Logistic Regression is in second place; KNN is in third place because both Ks had similar accuracy scores;

and Naïve Bayes is the model with the lowest accuracy, scoring 97.76%.

6. CONCLUSIONS

In conclusion, the primary goal of this project was to determine which of the four machine learning techniques selected for the project would be best suited for credit card fraud detection. This goal was achieved by building the models and comparing their respective accuracies; Support Vector Machine performed best in terms of accuracy, scoring 99.94% with just 51 misclassified cases. I think that by giving customers a better experience and a sense of security, implementing the model can assist reduce the amount of credit card theft and boost customer happiness.

Enhancing the model may be done in a number of ways, including applying it to diverse datasets of varying sizes and types, altering the data splitting ratio, and looking at it from other algorithmic angles. Combining telecom data can be used, for instance, to determine people's locations and determine where the credit card owner is when using the card. This will make fraud detection easier because, for example, if the card owner is at one place and a transaction is made with his card at another location, it will be readily identified as fraudulent.

REFERENCES

1. Zareapoor, M., Seeja.K.R, S. K. R., & Afshar Alam, M. (2012). Analysis on credit card fraud detection techniques: Based on certain design criteria. International Journal of Computer Applications, 52(3), 35–42. https://doi.org/10.5120/8184-1538.
2. Alenzi, H. Z., & Aljehane, N. O. (2020). Fraud detection in credit cards using logistic regression. International Journal of Advanced Computer Science and Applications, 11(12). https://doi.org/10.14569/ijacsa.2020.0111265.
3. Maniraj, S. P., Saini, A., Ahmed, S., & Sarkar, S. D. (2019). Credit card fraud detection using machine learning and Data Science. Credit Card Fraud Detection Using Machine Learning and Data Science, 08(09). https://doi.org/10.17577/ijertv8is090031.
4. Dheepa, V., & Dhanapal, R. (2012). Behavior based credit card fraud detection using support vector machines. ICTACT Journal on Soft Computing, 02(04), 391–397. https://doi.org/10.21917/ijsc.2012.0061.
5. Malini, N., & Pushpa, M. (2017). Analysis on credit card fraud identification techniques based on KNN and outlier detection. 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB). https://doi.org/10.1109/aeeicb.2017.7972424.
6. Maes, S., Tuyls, K., Vanschoenwinkel, B., & Manderick, B. (2002, January). Credit card fraud detection using Bayesian and neural networks. In Proceedings of the 1st international naiso congress on neuro fuzzy technologies (pp. 261-270).
7. Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017). Credit card fraud detection using Machine Learning Techniques: A Comparative Analysis. 2017 International Conference on Computing Networking and Informatics (ICCNI). https://doi.org/10.1109/iccni.2017.8123782.
8. Jain, Y., NamrataTiwari, S., & Jain, S. (2019). A comparative analysis of various credit card fraud detection techniques. International Journal of Recent Technology and Engineering, 7(5S2), 402-407.
9. Gupta, A., Lohani, M. C., & Manchanda, M. (2021). Financial fraud detection using naive Bayes algorithm in highly imbalance data set. Journal of Discrete Mathematical Sciences and Cryptography, 24(5), 1559–1572. https://doi.org/10.1080/09720529.2021.1969733.
10. Adepoju, O., Wosowei, J., lawte, S., & Jaiman, H. (2019). Comparative evaluation of credit card fraud detection using machine learning techniques. 2019 Global Conference for Advancement in Technology (GCAT). https://doi.org/10.1109/gcat47503.2019.8978372.