Available online www.jsaer.com

Journal of Scientific and Engineering Research, 2023, 10(5):372-378

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Credit Card Fraud Detection Using Machine Learning

Khirod Chandra Panda

Senior Engineer, Asurion Insurance LLC
Email: khirodpanda4bank@gmail.com

Abstract Credit card fraud detection is presently the most frequently occurring problem in the present world. This is due to the rise in both online transactions and e-commerce platforms. Credit card fraud generally happens when the card was stolen for any of the unauthorized purposes or even when the fraudster uses the credit card information for his use. In the present world, we are facing a lot of credit card problems. To detect the fraudulent activities the credit card fraud detection system was introduced. This project 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. Every day the modern world is moving towards digitalization and cashless transactions are becoming more common, credit cards are rapidly becoming more popular. Online and offline purchases using credit cards have become increasingly popular, which results in more fraudulent transactions every day. Many credit card fraud incidents occur every year and lead to huge financial losses. accordingly, it could be important to choose the best fraud detection method is essential so that it can detect fraud before criminal consumers a stolen card. To detect fraud, one method is to evaluate historical transaction data, as well as both normal and fraudulent transactions, to obtain usual and fraudulent behavior features by using machine learning techniques. we can use machine learning algorithms to solve this problem if we have access to enough data. In this study, our goal is to compare three algorithms for detecting credit card fraud (Decision Tree, Regression Logistic and Random Forest). we want to use a model that is new and based on a hybrid approach for detecting credit card fraud. According to this study, the proposed model is more capable of identifying fraudulent transactions than previous studies.

Keywords Random Forest, Adaboost, Fraud detection, Fraudulent behavior detection, Support Vector Machine, Decision Tree, Unsupervised learning

Introduction

Fraud is a criminal activity of a human being, which might be illegal act of money transferring from one's account without notifying. In [1], the author explains Fraud as to misuse of someone's money or assets for one's own advancement. In the last few years, we have always seen increasing businesses, online services, and Internet users in the USA. Also, over the last few years, many people use internet banking systems to transfer money, debit and credit cards for their purchases, and online payment services for all kinds of bills or invoices [2]. This technology offers various benefits such as cashless shopping, avoiding long queues to pay bills, and making our lives easier. However, despite the positive aspects of online transactions, financial fraud and unauthorized payments pose significant risks. [3].

Credit card fraud can have many reasons, such as the use of the card by an unauthorized card holder using fake identities, or it may be due to the usage of stolen credit cards. Many algorithms have been developed to overcome this obstacle.

The best method to solve this problem is to use various detection approaches [4]. Transactions are accepted or rejected within a very short time frame, which may span from microseconds to milliseconds. Detection of this transaction should be immensely first and effective. Another obstacle that exists is large amount of similar transaction that occur Hence, fraud cannot be detected by monitoring individual transaction.

There are different ML techniques to tackle credit card fraud detection, but we can classify them into main groups, including supervised, unsupervised, and reinforcement learning. The supervised learning techniques are applicable for classification and prediction problems, and data should be labeled for these techniques. This group contains techniques such as Support Vector Machine (SVM), Logistic Regression, Decision Tree, Naïve Bayes, K-Nearest Neighbor, Random Forest, Artificial Immune System, and Artificial Neural Network. On the other hand, the unsupervised learning techniques work with the unlabeled data and cluster the inputs based on their similarities. Some unsupervised ML techniques are K-means, Hidden Markov Model, Genetic Algorithm, Gradient Descendent, and DBSCAN (Zareapoor, 2015).

Literature Review

According to data mining concept of classification fraud detection falls in the bucket of classification problem [6]. As fraud detection works on the algorithm of data mining to classify the credit card transaction as an original or fraudulent one. The author proposed in [6] that Credit Card Fraud Detection is a problem of Data Mining and there are two major reasons for which credit card fraud detection is becoming more complex & challenging. They also performed a performance test on the bases of comparison on European cardholders having 284,807 transactions by using three techniques Knearest, Naive Bayes & Logistic Regression. They conclude by showing the effect of hybrid sampling.

Researchers have categorized machine learning algorithms that are useful for analyzing results. In one study, LR, SVM, GB, and RD were combined on a European dataset, yielding an overall accuracy of 91% [7]. Another study combined LR, DT, and RF, with RF achieving the highest performance at 95.5%, followed by DT at 94.3%, and LR at 90% [4]. The focus of these studies is on classifying credit card transactions as genuine or fraudulent, based on the behavior of the card owner. Predicting variables play a crucial role in this classification, significantly impacting the performance of fraud detection systems [8].

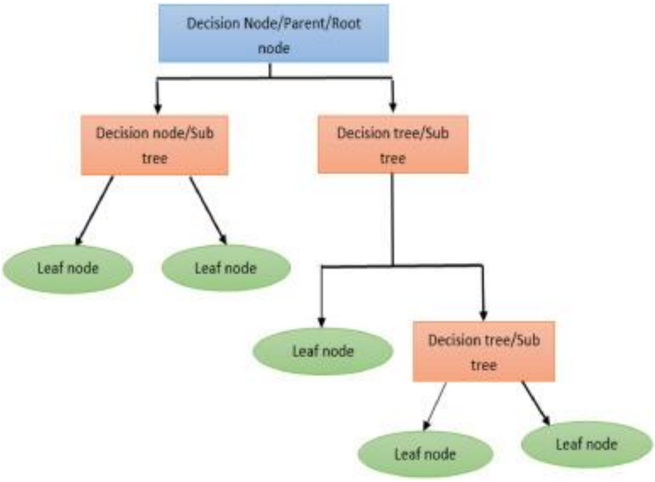
Additionally, the analysis of machine learning algorithms and Bayesian networks has shown better economic efficiency. In another study, the author highlights two major challenges in fraud detection: changes in the profile of a fraudster or normal behavior, and highly skewed data. Various methods, such as quadrant discriminative analysis, pipelining, and ensemble learning on CCFD, were compared to identify the most effective algorithm [8].

Decision Tree

A decision tree is a type of supervised learning algorithm [9] that is used for classification and regression tasks. It’s a non-parametric supervised learning algorithm for classification and regression tasks. It is represented in a tree-like structure, consisting of nodes that represent decisions or choices, and branches that represent the outcomes of those decisions. The tree starts with a root node and then splits into child nodes based on specific conditions related to the input variables. This splitting process continues until no further splits are required or until a certain stopping criterion is met [10].

One of the key advantages of decision trees is their ability to handle different types of data attributes, such as categorical and numerical, making them versatile and easy to use [11]. However, there is a risk of overfitting, where the model learns the training data too well and performs poorly on new, unseen data. To address this issue, pruning techniques are often used to remove certain nodes from the tree, improving its generalization performance.

Decision trees are commonly used in credit card fraud detection due to their ability to effectively classify data based on various attributes. By analyzing transaction data, decision trees can help identify patterns indicative of fraudulent activity, aiding in the detection and prevention of fraud.

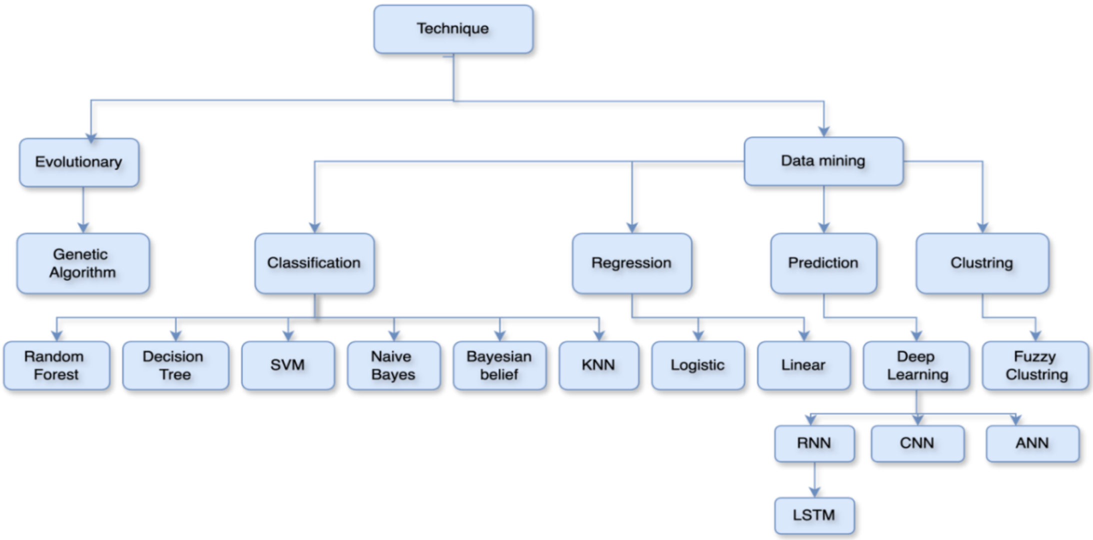


Decision trees, like many other machine learning algorithms, are subject to potentially overfitting the training data. Trees that are too deep can lead to models that are too detailed and don't generalize on new data. On the other hand, trees that are too shallow might lead to overly simple models that can't fit the data

Random Forest of Decision Trees

The instability in single trees and sensitivity to some training data led to development of another model that is random forests. Putting it in other way, It's another supervised technique that uses the bagging idea to improve the results by combining multiple single trees. With each tree being built independent of each other computational efficiency of random forest is comparatively better [12].

This approach uses a random subset of each tree's features and a training dataset to overcome the disadvantages of a single decision tree. It is basically an ensemble of regression and/or classification trees with it obtaining variance amongst its trees and hence are easy to use because of use of only two randomness sources or parameters that is building trees using trained data separate bootstrapped along samples with considering only a random data attribute subset to build each tree as specified [13].

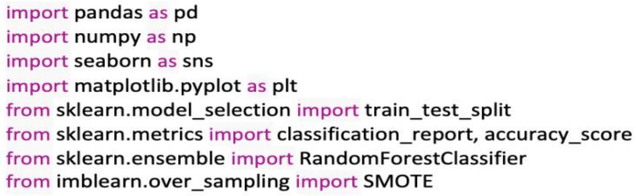


Stages of My Analysis

Library Used

I used Python for my Use case. Python libraries are collections of pre-written code and functions that extend the capabilities of the Python programming language. They provide a wide range of tools and modules for various tasks, making it easier for developers to work on specific tasks without reinventing the wheel

I used pandas, NumPy, seaborn, and matplotlib. pyplot libraries, which are used for data manipulation, analysis, and visualization in Python.



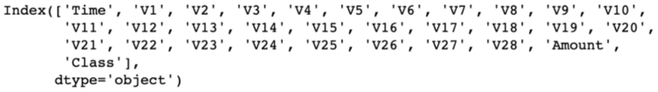
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score
from sklearn.ensemble import RandomForestClassifier
from imblearn.over_sampling import SMOTE
```

import specific modules from the scikit-learn and imbalanced-learn libraries. The train_test_split function is used to split data into training and testing sets. The classification_report and accuracy_score functions are used for evaluating classification models. The RandomForestClassifier class represents a random forest classifier. The SMOTE class is used for oversampling techniques.

Data Set

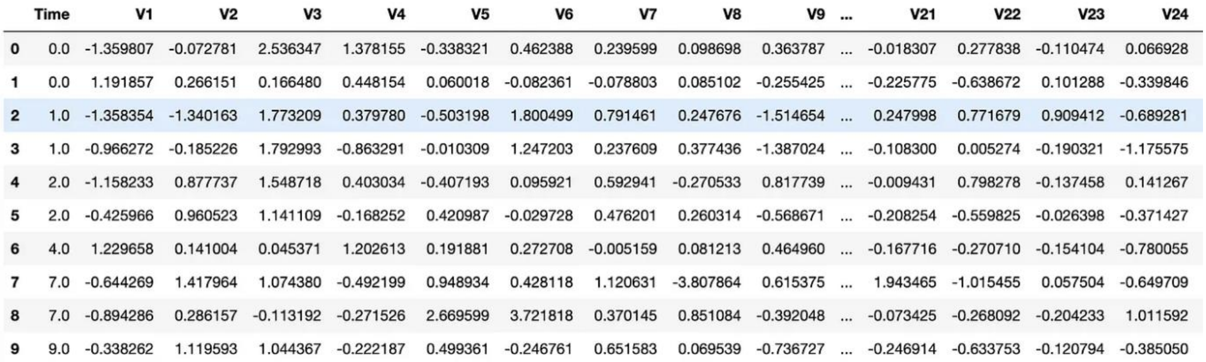
We will explore a dataset focused on detecting credit card fraud, thoroughly examining its essential attributes. This dataset comprises details about credit card transactions, encompassing diverse numerical characteristics and a target variable indicating the fraudulent status of each transaction. Our analysis will encompass an overview of the dataset's organization, a detailed discussion on the interpretation of its columns, and an emphasis on significant aspects warranting.

Columns of data Involved



```
Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
       'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',
       'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount',
       'Class'],
      dtype='object')
```

The dataset consists of 30 numeric attributes labeled from V1 to V28, which are likely transformed features designed to protect sensitive information. Although the exact meaning of these attributes is not specified, they are presumed to represent various aspects of the transactions. Examining these attributes may reveal patterns or irregularities associated with fraudulent transactions.



	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267
5	2.0	-0.425966	0.960523	1.141109	-0.168252	0.420987	-0.029728	0.476201	0.260314	-0.568671	...	-0.208254	-0.559825	-0.026398	-0.371427
6	4.0	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.005159	0.081213	0.464960	...	-0.167716	-0.270710	-0.154104	-0.780055
7	7.0	-0.644269	1.417964	1.074380	-0.492199	0.948934	0.428118	1.120631	-3.807864	0.615375	...	1.943465	-1.015455	0.057504	-0.649709
8	7.0	-0.894286	0.286157	-0.113192	-0.271526	2.669599	3.721818	0.370145	0.851084	-0.392048	...	-0.073425	-0.268092	-0.204233	1.011592
9	9.0	-0.338262	1.119593	1.044367	-0.222187	0.499361	-0.246761	0.651583	0.069539	-0.736727	...	-0.246914	-0.633753	-0.120794	-0.385050

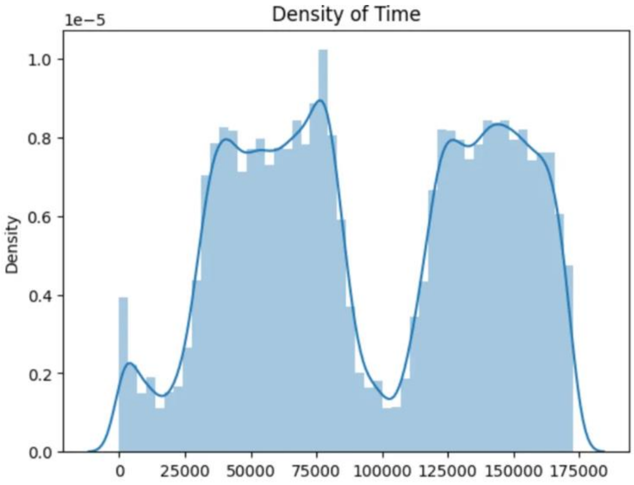
It contains only numerical input variables, which are the result of a PCA transformation. Unfortunately, I am not able to provide the original features and more background information about the data due to confidentiality issues. Features V1, V2, … V28 are the principal components obtained with PCA. The only features that have not been transformed with PCA are 'Time' and 'Amount.' Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount; for example-dependent cost-sensitive learning. Feature ‘Class’ is the response variable, and it takes value 1 in case of fraud and 0 otherwise.

Upon reviewing the dataset, it becomes evident that the majority of transactions are categorized as non-fraudulent (Class 0), while only a small fraction are identified as fraudulent (Class 1). This observation highlights a class imbalance, a common occurrence in datasets used for fraud detection. Addressing this imbalance is crucial to ensure the development of accurate predictive models.

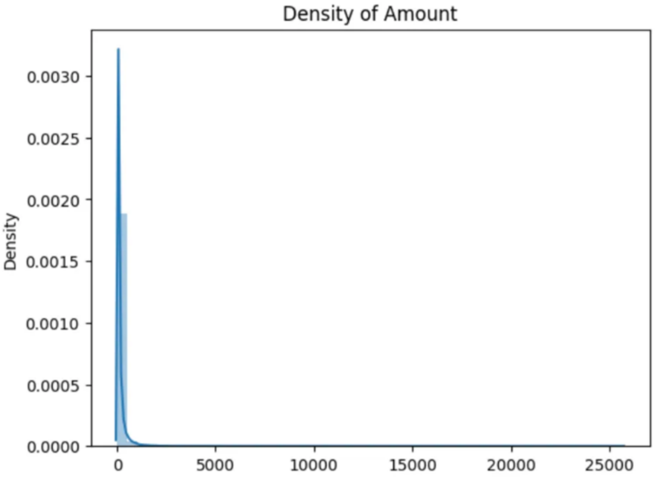
Before applying any machine learning algorithms or exploring data patterns, it is imperative to comprehend the characteristics of the credit card fraud detection dataset thoroughly. This blog post has offered an overview of the dataset’s structure, examined key columns, and underscored the presence of class imbalance. With this understanding, we are now equipped to proceed with feature engineering, model selection, and evaluation techniques to construct effective fraud detection models.

The "Time" feature in a credit card fraud detection dataset provides crucial information about the timing of transactions. By creating a density plot of this feature, we can visualize the distribution of transaction times, revealing patterns and trends in the data. For instance, we may observe peaks in transaction activity during certain hours or days, indicating periods of high transaction volume.

Analyzing the density plot can help us identify unusual patterns that may indicate fraudulent activities.



For example, if we notice a spike in transactions during typically low-activity periods, it could be a sign of fraudulent behavior. Similarly, if there are frequent transactions at unusual times, such as late at night, it may warrant further investigation.



The density plot of the “Amount” feature is a valuable tool for understanding the distribution of transaction values in a credit card fraud detection dataset. This plot provides insights into the density and concentration of transactions at different amounts, allowing us to identify typical transaction ranges and detect any unusual spikes or outliers that may indicate fraudulent activity.

By examining the density plot, we can visualize the distribution of transaction amounts and gain an understanding of the typical range of values. This information is crucial for identifying potential anomalies associated with fraudulent transactions, such as unusually large or small transaction amounts. Additionally, the density plot helps us develop robust fraud detection models by providing insights into the patterns and trends in transaction values.

Overall, analyzing the distribution of the "Amount" feature through a density plot is essential for gaining insights into transaction value distribution, identifying potential anomalies, and developing effective fraud detection strategies.

Conclusion

In conclusion, while current fraud detection techniques have limitations in detecting fraud as it occurs, our exploration of a credit card fraud detection dataset has provided valuable insights. We have highlighted the need for a technology that can detect fraud with equal precision and accuracy across all circumstances and datasets. By analyzing key dataset characteristics, exploring feature distributions, and implementing machine learning algorithms such as the Random Forest Classifier, we have demonstrated an approach to developing effective fraud detection models. Combining data analysis, visualization, and machine learning techniques can enhance fraud detection efforts and mitigate risks associated with fraudulent transactions.

References: -

[1]. Aisha Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," J. Netw. Comput. Appl., vol. 68, pp. 90–113, Jun. 2016, doi: 10.1016/j.jnca.2016.04.007.

[2]. Zhenchuan Li, Guanjun Liu, Senior Member and Changjun Jiang, Deep Representation Learning With Full Center Loss for Credit Card Fraud Detection, vol. 7, no. 2, pp. 569-579, 2020.

[3]. Zahra Karimi Zandian and Mohammad Reza Keyvanpour, "SSLBM: A New Fraud Detection Method Based on Semi-Supervised Learning", Journal of Computer and Knowledge Engineering, vol. 2, no. 2, 2019.

[4]. Gayan K. Kulatilleke, "Challenges and Complexities in Machine Learning based Credit Card Fraud Detection", Cryptography and Security (cs.CR); Machine Learning, 2022.

[5]. Naoufal Rtayli and Nourddine Enneya, "Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyperparameters optimization", Journal of Information Security and Applications, vol. 55, 2020.

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

[7]. "Amusan et al. - 2021 - Credit Card Fraud Detection on Skewed Data using M.pdf."

[8]. A. Saleh Hussein, R. Salah Khairy, S. M. Mohamed Najeeb, and H. Th. S. Alrikabi, "Credit Card Fraud Detection Using Fuzzy Rough Nearest Neighbor and Sequential

[9]. P. Suraj, N. Varsha and S. P. Kumar, "Predictive modelling for credit card fraud detection using data analytics," Procedia Computer Science, 132, 385-395, 2018.

[10]. S. Yusuf, E. Duman, "Detecting credit card fraud by decision trees and support vector machines," IMECS 2011- International multiconference of Engineers and Computer Scientists 2011, 1, 442-447, 2011.

[11]. P. H. Swain and H. Hauska, "The decision tree classifier: Design and potential," IEEE Transactions on Geoscience Electronics, 15(3), 142-147, 1977

[12]. T. K. Ho, "Random decision forests," In Proceedings of 3rd international conference on document analysis and recognition, Vol. 1, pp. 278-282, IEEE, August-1995.

[13]. S. Bhattacharyya, S. Jha, K. Tharakunnel, J. Westland, “Data mining for credit card fraud: A comparative study,” Decision Support Systems, 50, 602-613, 2011.