Credit card fraud detection using data analysis

Darab Qasimi
Daqasimi18@earlham.edu
Earlham College
Richmond, Indiana, USA

ABSTRACT

This proposal details plans for a program to detect fraudulent activities of credit cards. The program will use a Random Forest Decision Tree (RFDT) to classify and distinguish between fraudulent activities and authentic ones. RFDT is a machine learning method that trains the program and the model to derive conclusions. A decision is derived when the majority vote of decision trees and branches points to one conclusion. RFDT often consists of multiple branches of decisions, and the algorithm ultimately produces the decision that is the strongest.

ACM Reference Format:
Darab Qasimi. 2022. Credit card fraud detection using data analysis. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

This project of detecting fraudulent activities in credit cards aims to classify transactions as authentic or non-authentic. The project is mainly focused on detecting fraudulent activities. Credit card fraud often happens due to theft, fake copy, identity theft, and among other forms. Credit card fraud predictor variables are time, cash amount, transaction class, location, etc. Often if any of the predictor variables are not normal such as location, then there is a red flag raised by the fraud detection algorithms. Part of the reason that some American credit card companies require customers to inform the bank about their travels is that they want to prevent the customers' transactions from getting red-flagged by fraud detection algorithms. Another example of a red flag to credit card fraud detection algorithms is the amount involved in the transactions. Most bank institutions have daily spending limits on credit cards, and if a transaction goes beyond the limit, that's most likely a sign of fraud because customers know about their spending limit, so if there is an excess, it has to be unauthentic. Although there are cases where a customer might not know their spending limit and go over, in that case, they will have to confirm the large transaction with their financial institution. One of the behaviors of detecting fraudulent activities in this project is looking at cases such as the above examples.

For this project, the aim of using RFDT is to maximize the results of detecting fraud in credit cards by outputting a stronger decision from so many decision trees. In the context of this project, after providing the dataset to the program, the RFDT will divide the data recursively among many decision trees. In the decision-making process, each tree will come up with a decision, and in the end, only the one with the most votes will produce a result. So, the RFDT is the steps to processing data and recursively making decisions until it arrives at the best decision and decides if a credit card activity is fraudulent. This project is about credit card fraud detection, and machine learning methods such as RFDT, and bagging methods will be used to analyze a dataset [9].

2 RANDOM FOREST DECISION TREE

In this project, the Random Forest Decision Tree (RFDT) will be used as a base algorithm for deciding whether a credit card activity is a fraud. RFDT is a collection of decision trees. In this project, the decision trees will represent multiple decisions. Some will claim a transaction is a fraud some won't, and in the end, the results aggregated into a final decision. RFDT is a machine-learning method. This project will use a bagging method to detect fraud activities of credit cards. Bagging is one of the best algorithms of RFDT in a predictive model, which is used in the banking system for fraud detection. Bagging is an RFDT approach, where a group of decision trees will build a "forest." The "forest" that this algorithm builds is also known as a decision tree ensemble, which is often trained with a bagging model. This approach of RFDT combines multiple models into one package. Based on the thesis, bagging is one of the best algorithms of RFDT in a predictive mode

3 BACKGROUND

To classify credit card activities, we must understand the factors that affect fraud. Credit cards that are stolen or misplaced, synthetic fraud, data breaches, mail interception, skimming, and merchant collusion are common examples of ways credit cards are hacked or used for fraudulent activities. One of the resources that will be used in the research paper is a thesis by Ayorinde [2] that looks at the common trends in credit card fraud in the banking, retail, financial services, and healthcare industries. The thesis has used machine learning models such as decision trees to classify fraudulent transactions on a dataset taken from Kaggle [9]. The dataset is a simulated credit card transaction containing legit fraudulent transactions and fraud transactions from the duration of 1st Jan 2019 - 31st Dec 2020. The thesis has used the bagging approach of RFDT, where a group of decision trees will build a forest. Bagging will help in assigning the number of trees I'll use in the project, and it will also help in organizing trees into categories.

3.1 Implementation skills

In an online blog on DataFlair called Detect Credit Card Fraud with Machine Learning in R [1], many algorithms are used in R

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

to detect fraudulent credit card activity. The blog has complete instructions and implementations of credit card fraud detection algorithms. The program has used algorithms and methods such as Decision Trees, Logistic Regression, Artificial Neural Networks, and Gradient Boosting Classifier for data analysis. The blog has used the Card Transactions dataset [9], which contains a mix of fraudulent and non-fraudulent transactions. DataFlair has used the same dataset this project will use to train the RFDT algorithm and machine learning model. The material on the DataFlair blog is important for this capstone because it has shown how algorithms such as RFDT are done on the implementation, and programming side, and how the dataset is used in the program.

3.2 Unbalanced Data

Unbalanced data is having unequal instances of two different classes. In detecting credit card fraud, unbalanced data refers to having unequal instances of acceptable and fraudulent transactions. In Sharma's [8] thesis, the paper has considered that the dataset is unbalanced. This dataset, credit card fraudulent transactions, is unbalanced, likely because there are often a high percentage of credit cards without any fraudulent transactions and only a few with fraud. Before data analysis, the data in the dataset is scaled so that the overall dataset is ready for standardized modeling and analysis. In the dataset that Sharma has used in their thesis, there is only 0.172 percent fraud activity in credit cards of 284,807 transactions. Knowing how small the fraud percentage can be in a dataset, there is a high chance that if my program is tested with some dataset It might be unbalanced. Working with an unbalanced dataset can cause severely skewed class distribution which is risky given that most traditional machine learning models and methods assume balanced datasets. To avoid the risk of getting a false result by working with an unbalanced dataset modern machine learning methods have to be used, and the program should be trained not to assume a balanced dataset. I'll use Sharma's implementation of how data is balanced and how machine learning is used to avoid false results and conclusions in case this project is tested with random datasets.

3.3 Artificial Neural Networks vs. RFDT

Sulaiman et al.'s research [3], suggests that machine learning is an effective way of determining which transactions might be fraudulent. This work has used the Random Forest algorithm for constructing decision trees for training and machine learning purposes. The research paper describes the Random Forest decision tree as a slow algorithm in real-time fraud analysis. In the research paper, some of the Artificial Neural Network (ANN) Method is seen as convenient algorithm for being an unsupervised method for predicting fraud. ANN and RFDT find patterns in detecting fraudulent activities in credit cards, but ANN seems to be an effective solution. I believe RFDT is an effective way to produce a result aggregated from many other decision trees in supervised machine learning. RFDT is more effective because it's computationally less expensive [6]. In my project, I'll be using the bagging method, meaning there is less variance and complexity to the program. For my research paper, RFDT will be an appropriate way to analyze my intended dataset.

3.4 Supervised and unsupervised

There are two approaches to detecting credit card fraud; supervised and unsupervised. Supervised machine learning is defined as using labeled datasets, and unsupervised machine learning doesn't use labeled training data. I'm interested in supervised credit card fraud detection for this capstone project. Mekterović [5] has looked at detecting fraudulent activities of credit cards when cards are not present during transactions. The research has taken two approaches to predict fraudulent activities of credit cards; the first approach is supervised learning, and the second approach is unsupervised learning. The novel dataset has used 197,471 transactions that came from industrial partners' real-world credit card transactions over three months. The transactions are divided chronologically to achieve a realistic scenario when an actual fraudulent credit card activity happens. The dataset has also been modeled visually, where it's easier to see the data flow.

4 DESIGN AND IMPLEMENTATION

The credit card fraud detection project will be implemented using Python. There will be three steps in analyzing the fraudulent activities of a credit card and then deriving a conclusion as to whether a credit card's activity is legit or fraudulent. The generation implementation of the program will follow the structure shown in Figure 1.

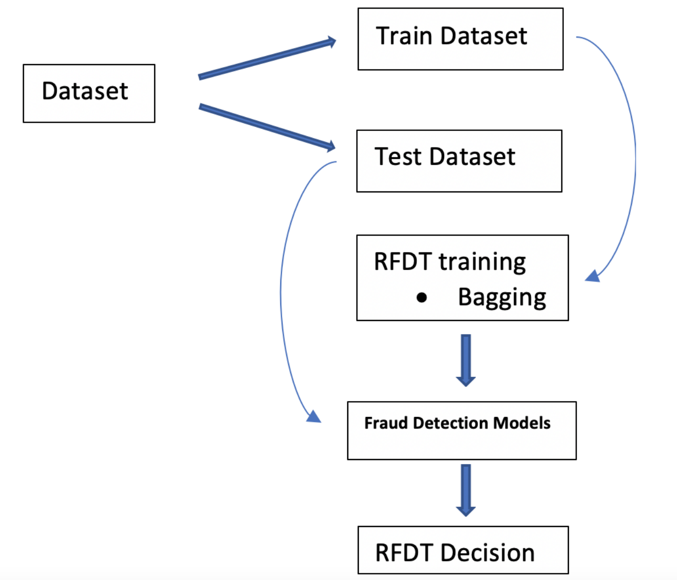

Figure 1: Software Architecture

Some Python libraries required for RFDT are pandas, numpy, and seaborn [7]. Through Pandas library, I'll be analyzing data, numpy library will be used to perform mathematical operations, and seaborn will be used to visualize random distributions. For data analysis, first, the program will take a dataset as its input which contains information on credit cards that have been used in fraudulent or normal activities. Second, based on figure 2 above,

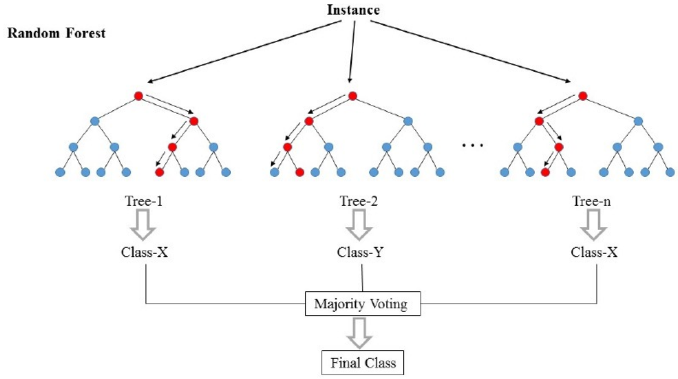

Figure 2: RFDT performance [4]

the program will analyze the dataset in the second step using the RFDT algorithm. During step two, small steps will be taken in the program getting rid of outliers and set up the dataset for analysis. Third, after the dataset or the instance is given to the algorithm, the RFDT will recursively divide the data among many decision trees and each tree will make a decision. During step three there, small steps will be taken, such as assigning the number of decision trees. Step four will be analyzing the dataset, each of the RFDT trees will deduce conclusions or results, and in the end, the best decision will go in the final class step. In the output step, out of many conclusions and decisions from the decision branches, it will show the best conclusion as a result.

5 MAJOR RISKS

There are some risks associated with detecting credit card fraud. Some of the risks are only bound to this project, but there are some other risks that the program might face if it's used in a professional work environment. There will be steps taken to minimize potential risks such as a wrong red flags that the program might output during data analysis.

5.1 Wrong red flags

One of the obvious risks is flagging a legit activity as a fraudulent activity. There are bad implications if this program flags a legit activity as fraudulent, which can lead to user dissatisfaction and distrust of credit cards. I'll have to be precise in the dataset analyzing steps to solve the issue of false positives and false negatives.

5.2 Phishing and hacking

Currently, phishing and hacking are not a risk within the scope of my project capstone, but if it is used in a professional work environment for business purposes, phishing might be a risk factor. The ultimate purpose of this project is to work with banks and credit card companies where the program will need access to the personal information of credit card holders. The program will follow steps to be protective of personal information to avoid phishing and hackers. Although hacking is a major risk to my program, it won't be a major risk in the project since I won't be working with the personal information of credit card holders directly. Part of my plan is to find a way to not work with the personal information of credit card users, which will make it easier to not worry about the security implications of my project but still perform the purpose of detecting credit card fraud.

6 TIMELINE

The overall process for writing the project should take about fourteen to fifteen weeks. The following are the breakdown of what will be accomplished in each week;

- Week 1: Write the basic steps of the program, e.g., steps to reading the dataset.
- Week 2: Write the initial steps to analyze the dataset. At this time, I'll make sure the program can read the dataset properly and that each column of the dataset is properly labeled and ready for analysis.
- Week 3: Develop the program and start implementing RFDT algorithm. This will be the initial stages of RFDT and there might still be no output through the algorithm.
- Week 4: Develop the RFDT algorithm and implement the bagging method. In the bagging stage, the RFDT algorithm implementation will almost be finished. Meanwhile, I'll be working on writing the first draft of the paper, which will include an introduction of the program, a literature review, data architecture diagram.
- Week 5: At this time, the dataset will get classified and it will be easier for the RFDT algorithm to know what activity is fraud and what's not. The first draft of the software will be ready to present this week.
- Week 6: During this week, the RFDT algorithm will be trained with the dataset that has already been classified.
- Week 7: Review the program, and polish it. At this time, the program will also be checked for time performance and make sure it doesn't take too long to detect a fraudulent credit card activity.
- Week 8: Test the algorithm with a different dataset that is close to how the current dataset is set up, and it's yet to be identified.
- Week 9: Keep testing the algorithm with the new dataset and check for accuracy. During this week the result of the program will be checked against some of the authentic RFDT algorithm implementations online. During this week the second draft of the paper will be ready which includes initial results and initial visualizations.
- Week 10: At this stage the program will be at its latest stages of development. This week the program will be tested again for accuracy and make sure that it's doing what it's intended to do. The first draft of demonstration video will also be prepared this week.
- Week 11: Now that the program is completely done and ready to submit, I'll take this week as a chance to create a diagram or a poster about what's happening in the program. In the poster, I'll draw how the RFDT algorithm analyzes information and the flow of data. This diagram will be similar to Figure 2 in the design and implementation section. The draft of poster will be ready this week.
- Week 12: Speak with a computer science faculty member to overlook the program. Showing the program to a faculty

member will help me in finding issues in the program that I missed during the past eleven weeks. I'll also implement some of the feedback that I'll receive this week that will improve the performance of the program. The third draft of paper will be ready by the end of week 12.

- Week 13 and 14: During these two weeks I'll be working on the second draft of the demonstration video, and the second draft of the poster.
- Week 15: Everything including the project, final paper, demonstration video, and the poster will be ready to submit this week.

7 CONCLUSION

Detecting credit card activities is an area where a good understanding of the intended algorithm is required. To familiarize myself with the methods of analyzing fraud activities and analyzing datasets, the above-mentioned online resources will be used as references and tutorials. The aim is to study the supervised approach of detecting or analyzing credit card fraud using the RFDT algorithm and implement it in Python. The process of writing a supervised machine learning will not only help me learn about machine learning, but it will also help me better understand how credit card transactions can be classified to detect fraudulent activities.

8 ACKNOWLEDGEMENT

I would like to thank David Barbella for all the feedback he has given me throughout the process of producing this proposal.

REFERENCES

[1] 2022. Data Science Project – Detect Credit Card Fraud with Machine Learning in R.

[2] Kayode Ayorinde. 2021. *A Methodology for Detecting Credit Card Fraud.* Ph. D. Dissertation. Minnesota State University, Mankato.

[3] Rejwan Bin Sulaiman, Vitaly Schetinin, and Paul Sant. 2022. Review of Machine Learning Approach on Credit Card Fraud Detection. *Human-Centric Intelligent Systems* (2022), 1–14.

[4] Stavros I Dimitriadis, Dimitris Liparas, Alzheimer's Disease Neuroimaging Initiative, et al. 2018. How random is the random forest? Random forest algorithm on the service of structural imaging biomarkers for Alzheimer's disease: from Alzheimer's disease neuroimaging initiative (ADNI) database. *Neural regeneration research* 13, 6 (2018), 962.

[5] Igor Mekterović, Mladen Karan, Damir Pintar, and Ljiljana Brkić. 2021. Credit card fraud detection in card-not-present transactions: Where to invest? *Applied Sciences* 11, 15 (2021), 6766.

[6] Aakash Parmar, Rakesh Katariya, and Vatsal Patel. 2018. A review on random forest: An ensemble classifier. In *International Conference on Intelligent Data Communication Technologies and Internet of Things.* Springer, 758–763.

[7] Abhishek Sharma. 2022. Age detection using CNN with Keras-with source code-easiest way-easy implementation. https://medium.com/mlearning-ai/age-detection-using-cnn-with-keras-with-source-code-easiest-way-easy-implementation-57c107b23bc4

[8] Nishant Sharma. 2020. Credit Card Fraud Detection Predictive Modeling. (2020).

[9] Kartik Shenoy. 2020. Credit Card Transactions Fraud Detection Dataset. (2020).