Using Machine Learning Techniques to Detect Credit Card Fraud

Brian Tiner

Abstract - Credit card fraud is a continually growing concern that is too complicated for humans to accurately detect without the aid of computers. Machine learning algorithms have been developed to help detect anomalies in credit card transactions so fraud can be identified. The focus of this paper is to develop a robust credit card fraud detection system using machine learning methods. In this paper, the dataset of credit card transactions that is used to test these methods is introduced and examined. The dataset's structure and issues is discussed to identify solutions. The issues of the dataset are resolved through preprocessing methods. Normalization, feature extraction, and SMOTE oversampling are conducted to provide clean data for classification. Various machine learning classification techniques are proposed and executed on the given data. A random forest classifier, naive Bayes classifier, logistic regression model, and support vector machine are implemented for the given dataset. The results of the classification are analysed for their accuracy and efficiency. The ideal system is identified from the proposed machine learning techniques by its ability to accurately identify credit card fraud.

Ⅰ - Introduction

A credit card refers to a personal card issued by a financial institution that is used to purchase goods and services within a credit limit of an individual. [1] It is a much easier form of payment than cash. Rather than paying for each item individually with cash, one can pay off every purchase made within a time period at a later date with a credit card. In recent years, credit cards have become one of the most popular forms of payment. However, growth in popularity has led to growth in malicious activity. The increase of e-banking and credit card use has led to an increase in fraudulent activity that can account for cumulative losses of billions of dollars every year. [6] There are several different types of fraudulent activity. These include: counterfeit credit cards, stolen credit cards, and manual credit card imprints. [7] The most common type is the result of a stolen credit card where the thief makes several transactions before the original owner cancels the card. While there are different types of credit card fraud, the main idea behind each is for an individual to make purchases that they have no intent of repaying. Given the problem of credit card fraud, there is no way a human can identify fraudulent activity with over a billion transactions world wide on a given day. To solve the issue, machine learning techniques have been developed to identify outliers in credit card transaction data which represent fraudulent activity.

Credit card fraud is a classification problem. [4] Every credit card transaction can be classified as either genuine or fraudulent. There are many factors that come into play when developing a system to detect credit card fraud. The first concern is the data itself. The data can contain several limitations when detecting fraudulent transactions. Problems in the data include: noise, irrelevant features, and highly skewed data. [4] Strong fraud detection systems need to account for any problems in the dataset that is being analysed. There are many preprocessing techniques that can be applied to credit card transaction data. Different preprocessing methods that are used for this type of problem are: data normalization, feature extraction, and adjusting class distribution. [2] Using a series of preprocessing methods

together can increase the classification accuracy of machine learning algorithms. After data preprocessing has been executed, machine learning algorithms need to be used to classify credit card transactions. There are numerous approaches used to classify fraud. Some popular machine learning techniques for credit card fraud detection are: random forest classifiers, naive Bayes classifiers, logistic regression models, and support vector machines. [9] There exist many other classification techniques for fraud detection, however only the four listed will be analysed in this paper. Each approach attempts to construct a classification model using the provided data which can then make predictions given new transactions. Each classification method has different accuracies and efficiencies when working with different types of datasets.

The remainder of this paper will be structured as follows. First, the dataset used for credit card fraud detection is discussed. The structure of the dataset as well as the different features and limitations are examined and preprocessing methods are proposed. Next, the preprocessing that has been implemented is considered. How the preprocessing has been implemented will be displayed as well as the impacts it has on the classification algorithms. Finally, the different machine learning classification techniques are proposed for credit card fraud detection. Each technique's results are examined and compared to see what the most effective classification method is for credit card fraud detection.

Ⅱ - Dataset

The dataset used for credit card fraud detection is essential. Depending on the structure of the dataset, different machine learning algorithms can have different classification accuracies. Some datasets have more categorical variables while others are based on numerical variables. The dataset that will be used for the classification in this paper was found on kaggle. [10] It is a very common dataset used for credit card fraud detection research. [1, 2, 8] The dataset consists of 284,807 credit card transactions by european cardholders over a two day period in September of 2013. The dataset has thirty numerical input features and one binary output feature. The two readable input features are time and amount. The time feature is represented as seconds starting from the first transaction in the dataset. The amount feature represents the dollar amount of the transaction. The output feature named class represents if the transaction was fraudulent or not. The other twenty eight input features are vague due to confidentiality reasons. Each of the twenty eight features are a result of a principal component analysis transformation.

From this dataset, there are several problems that need to be addressed. First, the distribution of the dataset is unbalanced. While there are 284,807 transactions, only 492 were recorded as fraud. This means only 0.172% of transactions from this dataset are considered fraud. This is a common problem addressed in most credit card fraud detection systems. [2, 4, 9] The dataset also consists of many input features. Given so many input features, there may be some that are not relevant to the final result. Furthermore, the two features of time and amount have a much larger range of values than the other input features. Several preprocessing methods need to be executed to account for the obscurities. First the data should be normalized to account for a wide range in values. Then the irrelevant features should be extracted to help classification accuracy. Lastly, an oversampling technique should be applied to help with the unbalanced data.

Ⅲ - Preprocessing

Preprocessing is one of the most important components to a strong fraud detection system. Datasets provided for credit card fraud detection always have several problems that must be resolved before the transactions can be classified. The most robust systems must account for issues such as: skewed data, noise, or irrelevant features. [4] There are several different ways to account for these obscurities. Given the dataset described in the previous section, the first thing that needs to be accounted for is the wide range in values. Using python library sklearn, the entire dataset is z-score normalized to put each input feature into the same range. This aids the accuracy by accounting for the time and amount features that have much larger values than the rest of the dataset.

After the dataset is normalized, more in depth preprocessing is conducted. The next obscurity that needs to be corrected is irrelevant features in the dataset. For this, a feature selector tool by Will Koehrsen [5] is used that is referenced in [2]. The feature selector is a python class with five different methods for the purpose of data preprocessing. The methods used from the feature selector class for this credit card transaction dataset is to identify low importance features. With this method, an importance percentage can be specified to find features below a specific importance threshold. However, the method has some varying results.

60 Iterations, 95% Importance,	60 Iterations, 99% Importance,	500 Iterations, 95% Importance	500 Iterations, 95% Importance
V5 : 10	V23 : 14	V10 : 6	V10 : 0.012
V20 : 9	V24 : 8	V11 : 94	V11 : 0.188
V13 : 7	V6 : 13	V2 : 240	V2 : 0.48
V9 : 27	V2 : 21	V27 : 158	V27 : 0.316
V11 : 11	V11 : 6	V9 : 182	V9 : 0.364
V24 : 12	V5 : 4	V6 : 262	V6 : 0.524
V2 : 27	V20 : 3	V3 : 55	V3 : 0.11
V23 : 38	V27 : 2	V18 : 127	V18 : 0.254
V6 : 31	V9 : 6	V17 : 35	V17 : 0.07
V16 : 5	V3 : 3	V23 : 273	V23 : 0.546
V27 : 11	V25 : 3	V5 : 138	V5 : 0.276
V15 : 1	V1 : 1	V24 : 110	V24 : 0.22
V18 : 26	V19 : 1	V20 : 67	V20 : 0.134
V3 : 6	V13 : 1	V19 : 24	V19 : 0.048
V19 : 2	V18 : 3	V28 : 28	V28 : 0.056
V17 : 10		V13 : 56	V13 : 0.112
V1 : 1		V21 : 26	V21 : 0.052
V25 : 1		V16 : 60	V16 : 0.12
V8 : 1		V25 : 27	V25 : 0.054
V28 : 1		V15 : 5	V15 : 0.01
V21 : 1		V22 : 10	V22 : 0.02
		V8 : 10	V8 : 0.02
		V12 : 1	V12 : 0.002
		Time : 3	Time : 0.006
		V1 : 1	V1 : 0.002
		V7 : 1	V7 : 0.002

Fig. 1

Many tests are conducted on the credit card transaction dataset provided. As illustrated in (Fig. 1), tests are run using importance levels of 95% and 99% with different amounts of iterations. The outputs of each iteration are tallied and the totals are output as well as a percentage of occurrences for the largest test. From these tests, the values that occur the most for low importance features are V2, V6, and V23. These features are removed from the dataset to increase efficiency and accuracy.

The final step of preprocessing is dealing with the highly unbalanced dataset. The amount of actual fraud in relation to the entire dataset is incredibly low. However this is a good real world scenario as there is only a very small amount of fraudulent activity among real transactions. To account for this, synthetic minority oversampling technique (SMOTE) is used. SMOTE is a very popular preprocessing method when it comes to credit card fraud datasets. [1, 2, 8] It is used to oversample the minority class of the dataset to correct the data imbalance. For this instance, it is implemented using the python imblearn package. The SMOTE oversampling has major effects on the results on the classification algorithms. This will be examined in the results section. After each preprocessing method has been executed, the dataset is partitioned into 75% training set and 25% test set. Using the preprocessed training and test sets, the machine learning techniques are applied and analysed.

Ⅳ - Methods

There are many different machine learning classification algorithms that can be used to predict credit card fraud. Each has varying efficiencies and accuracies. The methods that are examined in this paper are: random forest classifier, naive Bayes classifier, logistic regression models, and support vector machines. These are some of the most popular

methods used for credit card fraud detection. [1-4,6-9] The methods displayed are implemented using python's sklearn library. Each is examined based on their strengths and limitations with respect to credit card fraud detection.

Random forest classifiers are a strong method of credit card fraud detection. They are a supervised learning algorithm that consists of multiple decision trees. The output of the random forest classifier is the statistical mode of the internal decision tree's output. They have relatively fast training time [9] and have better accuracy with more decision trees in the forest. For this instance, the random forest contains 100 decision trees. Random forest classifiers are a better classification method than a single decision tree. They avoid overfitting by having many different samples with different output rather than a single tree with one output.

The naive Bayes classifier is a supervised learning algorithm that is based on Bayes theorem. [2] There are several different distributions of the naive Bayes classifier. For this dataset, a gaussian naive Bayes classifier is used. It has very fast training time and can work well with high dimensional data. However, some features are classified based on a bias because each feature is processed independently. This can introduce a classification problem and result with a lower accuracy.

Logistic regression is a supervised learning algorithm that uses a logistic function to construct a model for classification. It interprets the input features for classification and models a relationship between the input and output. [2] Using the created model it can predict binary outputs. It has a fast training time and fast prediction time. The drawback is that a large number of entries in the training set can cause overfitting. It is also difficult to represent more complex relationships using a logistic regression model.

Support vector machines are a supervised learning algorithm that constructs a model of the data by defining a hyperplane that separates the output classes. [7] The hyperplane is represented on a graph as a line separating the points representing different classes of output. Ideally the hyperplane is located equally between the output classes using support vectors. The training and prediction time for a support vector machine is very long. It can be a robust classification technique but as the data grows larger it can only classify with a medium degree of accuracy. [9]

V - Results

To identify the ideal credit card fraud detection system, each classification method must be analysed and compared. The methods described in the previous section are all executed on the credit card fraud detection dataset found on kaggle. [10] The evaluation metrics used to compare between methods are: accuracy, recall, and precision. Accuracy represents the percentage of correctly classified transactions. Recall represents the percentage of correctly detected fraudulent transactions out of all fraudulent transactions. Precision represents the percentage of correctly detected fraudulent transactions out of the total detected fraudulent transactions. [6] Each of these metrics is calculated from a confusion matrix that records the amount of true positive, false positive, true negative, and false negative predictions.

An important note to make before the classification methods are compared is the impact of the SMOTE preprocessing. Classification tests are conducted both with and without the SMOTE preprocessing in place. First the results without SMOTE preprocessing are examined. As seen in (Fig. 2) are the resulting confusion matrices from each classification algorithm. The initial observation that can be made is that the naive Bayes

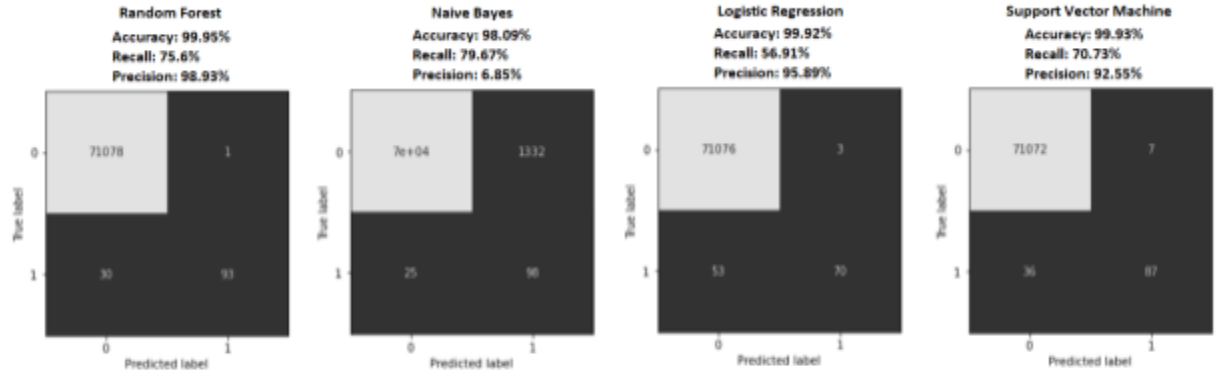

Fig. 2

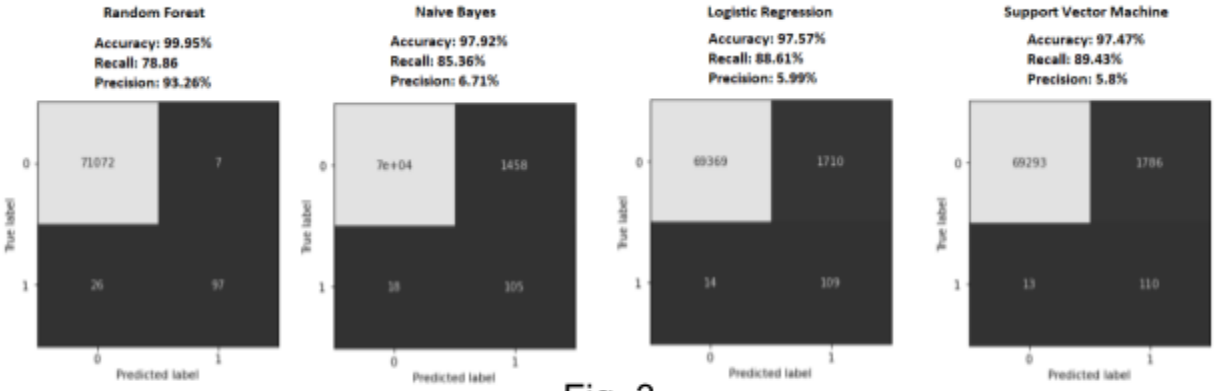

Fig. 3

classifier has a very high false positive rate. This causes the naive Bayes to have a very low precision score of 6.71%. While it has very low precision, it has the highest recall with 79.67%. However, the high recall score does not make up for the incredibly low precision score. Of the three remaining methods, the random forest has the most favourable results. The random forest has a precision score of 98.93% and recall score of 75.6%. This means that it classifies about 75% of fraudulent cases correctly and has almost no false positives. The support vector machine and logistic regression model each have lower recall and precision values than the random forest. Of the provided methods, the random forest is the most applicable.

Now once the SMOTE preprocessing is included, the results change drastically. (Fig. 3) shows the resulting confusion matrices of the classification methods. The most obvious change is the support vector machine and the logistic regression. While they gained a remarkable amount in recall, the precision dropped significantly. The SMOTE preprocessing caused the two classification methods to predict a large amount of false positives. Similar to this, the naive Bayes classifier gained recall at the cost of precision. The random forest classifier had slight change from the SMOTE preprocessing. It also gained a small amount of recall for a small cost in precision. However, this is an improvement because with the small amount of fraudulent cases, an increase in correct fraud detection is important. Comparing the results to other research using the same dataset [2, 8], most classifiers have similar results. For both naive Bayes and logistic regression, there is a very high amount of false positives in both studies. However, the support vector machine has much worse results. In [8] the support vector machine yields a precision of 67.8%. This may be

because of differences in preprocessing. Regardless of preprocessing methods, the random forest classifier is the most ideal system. Both with and without SMOTE preprocessing, it has a high degree of recall and precision. Based on these comparisons, the random forest classifier would work the best given real credit card transaction data.

VI - Conclusion

Different credit card fraud detection systems were examined to find an ideal real world application. First the structure of the dataset was examined and flaws were pointed out. To account for the flaws in the dataset, preprocessing methods were applied. The preprocessing began with z-score normalization to account for the wide range in values. Low importance features were then extracted to aid the classification accuracy. Finally, SMOTE oversampling was done to help the unbalanced dataset. SMOTE introduces some discussion. The recall of the classification algorithms increased with SMOTE in place while the precision was lowered. The most robust method that had a positive impact from SMOTE was the random forest classifier. It has very high recall while still keeping a high precision score. The random forest classifier is the ideal method for a real world credit card fraud detection system.

References

[1] V. N. Dornadula and S. Geetha, "Credit Card Fraud Detection using Machine Learning Algorithms," *Procedia Comput. Sci.*, vol. 165, pp. 631–641, Jan. 2019, doi: 10.1016/j.procs.2020.01.057.

[2] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "Credit Card Fraud Detection - Machine Learning methods," in *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, East Sarajevo, Bosnia and Herzegovina, Mar. 2019, pp. 1–5, doi: 10.1109/INFOTEH.2019.8717766.

[3] P. Hajek and R. Henriques, "Mining corporate annual reports for intelligent detection of financial statement fraud – A comparative study of machine learning methods," *Knowledge-Based Systems*, vol. 128, pp. 139–152, Jul. 2017, doi: 10.1016/j.knosys.2017.05.001.

[4] A. O. Adewumi and A. A. Akinyelu, "A survey of machine-learning and nature-inspired based credit card fraud detection techniques" *Int J Syst Assur Eng Manag* 8, 937–953 (2017). https://doi.org/10.1007/s13198-016-0551-y

[5] Github (2019). Feature selector. [online] Available at: https://github.com/WillKoehrsen/feature-selector [Accessed 23 Nov. 2020].

[6] J. Forough and S. Momtazi, "Ensemble of deep sequential models for credit card fraud detection," *Appl. Soft Comput.*, p. 106883, Nov. 2020, doi: 10.1016/j.asoc.2020.106883.

[7] A. K. Shemar and B. K. Sidhu, "Credit Card Fraud Detection using Anomaly Detection" *Journal of Innovation in Computer Science and Engineering*, vol. 10(1), Nov. 2020 http://innovation-journals.org/10vi1-2.pdf

[8] V. Arora, R. S. Leekha, K. Lee, and A. Kataria, "Facilitating User Authorization from Imbalanced Data Logs of Credit Cards Using Artificial Intelligence," *Mobile Information Systems*, vol. 2020, Oct. 2020, doi: 10.1155/2020/8885269.

[9] P. Divya, D. Palanivel Rajan, and N. Selva Kumar, "Analysis of Machine and Deep Learning Approaches for Credit Card Fraud Detection," in *ICCCE 2020*, Springer, Singapore, 2021, pp. 243–254. https://doi.org/10.1007/978-981-15-7961-5_24

[10] Kaggle.com. (2019). Credit Card Fraud Detection. [online] Available at: https://www.kaggle.com/mlg-ulb/creditcardfraud [Accessed 20 Nov. 2020]