Credit Card Detection by Applying Interpretable Tree-Based Machine Learning Models

Shizhao Xiong(✉)

School of Mathematics, The University of Edinburgh, Edinburgh, UK
S2067924@ed.ac.uk

Abstract. Credit card security issues have emerged in recent decades as the usage of credit cards for payment has increased. As a result, more and more credit card fraud instances have occurred, drawing significant attention from financial and academic circles. This work intends to employ three interpretable tree-based models, namely decision tree classifier, random forest classifier, and extra tree classifier to detect credit card fraud instances and employ Area Under Curve, Accuracy, Positive Predicted Value, recall, and F1 score as indicators to evaluate their performance while dealing with the challenges of extensive sample data and severely imbalanced data in credit card fraud detection. In addition, the feature importance based on these three models is also presented to observe the degree of correlation between each input feature variable and the predicted label during the model training process. The experimental results indicate that the extra tree classifier, this ensemble model performs better in this detection, which can assist credit card users and institutions in completing credit card detection in organizing the occurrence of fraud events as much as feasible.

Keywords: Machine learning · credit card fraud detection · decision tree classifier · random forest classifier · extra tree classifier

1 Introduction

As electronic payments grow more commonplace and credit card purchases substitute for cash as the main payment terms, there are more cases of credit card fraud occurring globally. Credit card fraud is a domain in which perpetrators commit illegal behaviors that may have a negative impact on other individuals or businesses [1]. It happens when individuals access personal credit card information and attempt to conduct unauthorized purchases or other financial activities [2]. Credit card fraud incidents can be categorized as internal and external credit card fraud. Specifically, internal credit card fraud happens when the cardholder and issuer bank authorize it, and deceivable identities are used to carry out the crime. In contrast, external fraud happens when card information and other cardholder data are taken using dubious ways [3]. Since credit card fraud may occur in various unexpected ways and many credit card users are at risk of it, numerous credit card firms employ a range of measures that aim to deny transactions rather than

V. Gaikar et al. (Eds.): ECIT 2023, AHE 18, pp. 266–272, 2023.
https://doi.org/10.2991/978-94-6463-210-1_34

authorize them to prevent and identify such fraudulent behaviors [4]. Thus, credit card fraud detection is a corrective action that machine learning algorithms may control to manage data analytics, predictive modeling, decision-making, and fraud alerts.

Machine learning models often require large amounts of reliable data to perform accurate predictions [5]. When attempting to build a model to monitor credit card fraud, the distribution of the experimental data could be more evenly distributed. The amount of credit card fraud is frequently much lower than that of non-fraudulent specimens, and the model cannot proficiently gather information from deceivable samples during the data training process, which can easily result in misjudgment [6]. Therefore, choosing a class of machine learning models that can effectively manage this highly imbalanced data, weighing their benefits and drawbacks, and outlining how each model contributes to identifying credit card fraud occurrences has turned into a pressing issue that must be resolved right now.

In 1994, Ghosh and Reilly discussed the performance of neural networks based on datasets provided by credit card issuers and discovered that fraud detection systems by applying neural networks could considerably improve the accuracy and timeliness of fraud detection in the still-relatively-new field of recognizing credit card fraudulent activities [7]. Maes et al. found when comparing Bayesian and neural networks for the aim of identifying credit card fraud that both of them provided good results in fraud detection, whereas the former had a shorter training period, the latter had a faster fraud detection process [8]. The processing flow for credit card transactions was modeled using a hidden Markov model by Srivastava et al., who also demonstrated how it could be applied to fraud detection [9]. Bahnsen et al. described a cost-sensitive technique based on Bayes minimum risk with the proposed cost measure that can precisely represent the financial benefits and economic losses from fraud detection [10]. Sahin et al. provided the C5.0, C&RT, and CHAID decision tree algorithms together with four additional support vector machine classifiers, allowing institutions to employ models to compare valuable historical data, estimate the likelihood of a new fraudulent transaction, and provide a rationale for the transaction authorization method [11].

Much attention has been paid to the research on credit card fraud detection models. People set out to explore algorithms and models for monitoring credit card fraud cases as artificial intelligence took shape. Unfortunately, the techniques at the time were underdeveloped, and the detection of credit card fraud using neural networks had a poor understanding. Additionally, the results reached after comparing the performance of the model with that of the model are less persuasive since fewer models are included in those studies. Furthermore, there are several models utilized in the literature, but the coverage could be more substantial. In this regard, this paper will examine the detection of credit card fraud and evaluate the effectiveness of these methods using three interpretable tree-based machine learning models: decision tree, random forest, and extra tree from a broader perspective. While tree-based machine learning methods can analyze both categorical and numerical data and only require minor amounts of data preparation and computation during training. Overall, they are more computationally efficient than other machine learning models in terms of computing efficiency and are superior for pattern identification and trend detection.

2 Methods

This paper applied three different tree-based models for the research. In this instance, scikit-learn, a machine learning package for the Python language, will be employed in all three models to work properly. When every single tree model has been created, this study utilizes the training dataset to train the model and identify the optimal parameter settings. Then, using the validation set to monitor its accuracy before applying the remaining data to satisfy needs. In the end, a confusion matrix is applied to detect whether the model confuses two different classes.

2.1 Data Source and Preprocessing

The entire experimental data collection used in this study was downloaded from the Kaggle website, and this dataset was recorded and examined on research that Worldline and the Machine Learning Group at Université Libre de Bruxelles (ULB) collaborated on [12]. It comprised purchases with credit cards made by European cardholders in September 2013, which contained 284,807 transactions (referred to 284,807 columns) over two days. The 31 fields included 28 features of V1-V28 that were processed by Principal Components Analysis (PCA) due to confidentiality reasons. Moreover, the additional three features are "Time", "Amount", and "Class". More specifically, the amount of time that has passed since the first transaction in the dataset is explicitly stated in the feature "Time" for each transaction. The feature "Amount" refers to the transaction amount. The feature "Class" is to record whether the transaction is fraudulent. When there is a fraud, it is recorded as "1", when it does not exist, it is recorded as "0".

On the basis of the preprocessing, all data will be separated into three categories: a training set, a test set, and a validation set. Specifically, the training dataset accounting for 60% of all data will be used to train the model to find the most suitable values for the parameters. Afterwards, this study used the 20% validation set to monitor its accuracy. Then, using the remaining data as a test set to obtain the accurate performance of the model.

2.2 Tree-Based Machine Learning Algorithms

1) Decision Tree Classifier.

A classification decision tree is employed as a prediction model to draw a conclusion from a group of observations. In the tree structures, an internal node represents the feature, branches represent the output of the test set, and every leaf node represents class labels [13]. By applying the scikit-learn package, the function "criterion" quantifies the quality of a split, which is a parameter specific to the tree model; "splitter" is a strategy used to determine the split at each node, set "splitter" equals to "best", thus, supported strategies are "best" to find the best split; the randomization of an estimator is controlled by "random_state", and even when the splitter is set to "best", the features are always randomly permuted at each split [14].

2) Random Forest Classifier.

Multiple decision trees make up a random forest, a classifier whose output is the classification chosen by the most trees [15]. It uses averaging to increase predicted

Table 1. The Performance of Different Models

Numble	AUC	ACC	PPV	recall	F1_score
Decision Tree Classifier	0.8282004	0.9987589	0.9992228	0.9995335	0.9993782
Random Forest Classifier	0.8528641	0.9991798	0.9993338	0.9998445	0.9995891
Extra Tree Classifier	0.8528751	0.9992019	0.9993338	0.9998667	0.9996002

accuracy and manage overfitting [16]. To do this, scikit-learn will be used once more for this. Among this, a new parameter called "n_estimators" will be generated, it represents the number of trees in the forest; same with the application in the previous classifier mentioned here, "criterion" utilized to measure the quality of a split; and "n_jobs" indicates how many jobs will be run concurrently; when determining the best split at each node, the function "random_state" controls the sampling of the characteristics to consider as well as the randomness of the samples; "verbose" regulates the verbosity when fitting and forecasting, and set it to "FALSE" when building this model [14].

3) Extra Tree Classifier.

An extra tree classifier uses an estimator that fits a number of randomized decision trees on multiple subsamples of the dataset to optimize prediction accuracy and reduce overfitting. The only parameter that needs to be supplied in the model is "n_estimator" and set its value to 100 here [14].

3 Result and Discussion

3.1 Classification Performance of Different Models

In this paper, Area Under Curve (AUC), Accuracy (ACC), Positive Predictive Value (PPV), recall and F1 score are employed as indicators to measure the performance of the model. To clarify, ACC = (True Positive + True Negative)/(Positive + Negative), PPV = True Positive/(True Positive + False Positive), recall = True Positive/(True Positive + False Negative), F1_score = 2*True Positive/(2*True Positive + False Positive + False Negative) [16].

As can be seen from Table 1, the values of random forest and extra tree are relatively similar, regardless of the AUC, ACC or PPV. The value of AUC, ACC, PPV, recall and F1_score of the decision tree classifier is 82.82%, 99.88%, 99.92%, 99.95%, and 99.94%, respectively, slightly inferior to the first two. The possible reason is that noisy data points easily affect the decision tree. Hence, the performance of the decision tree is inferior to that of random forest and extra tree as ensemble models in this study.

3.2 Feature Importance

The term "feature importance" describes how each feature in the dataset is evaluated for a particular model, with the scores indicating the relative importance of each feature. When visualizing the feature importance of the decision tree classifier shown in Fig. 1, it can be discovered that other than the four features V17, V4, V10 and V7, the rest of

the features, especially the features “Time”, “V23”, “V2”, and “Amount”, which have values that are close to zero in this bar chart, are not very required for this model. In addition, it is clear from a detailed examination of the feature importance images for the random forest classifier and the extra tree classifier that V17, V12, V14, V16, and V11 are the five most indicative features of each in Fig. 2 and Fig. 3. Moreover, these five properties of the two models have highly similar values. Although many features in the decision tree classifier have extremely low scores, most features in the remaining two models still have high scores. As a result, eliminating these features to minimize the dimensionality of the model takes much work to achieve.

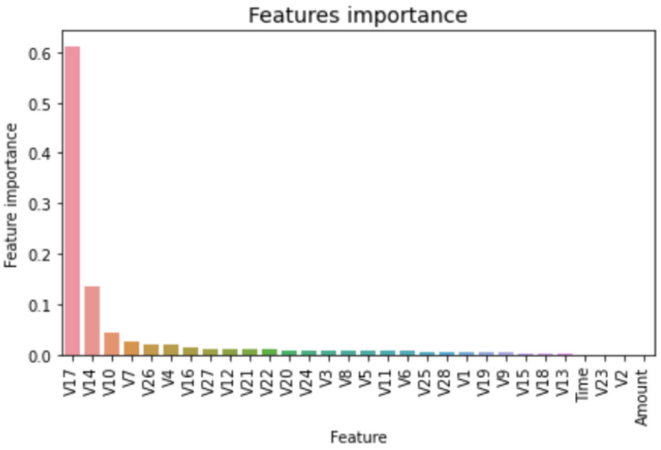

Fig. 1. Feature Importance Obtained from Decision Tree Classifier

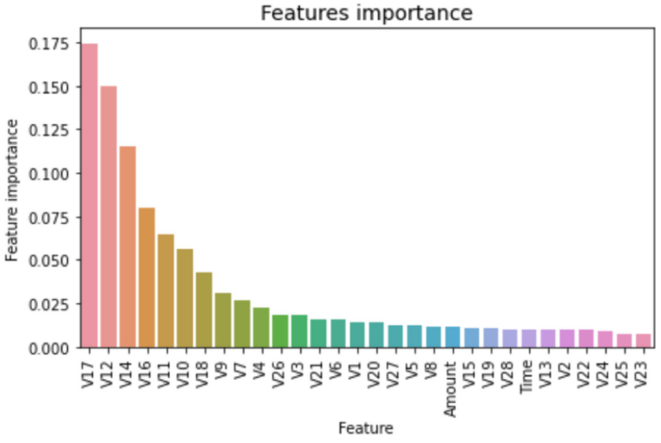

Fig. 2. Feature Importance Obtained from Random Forest Classifier

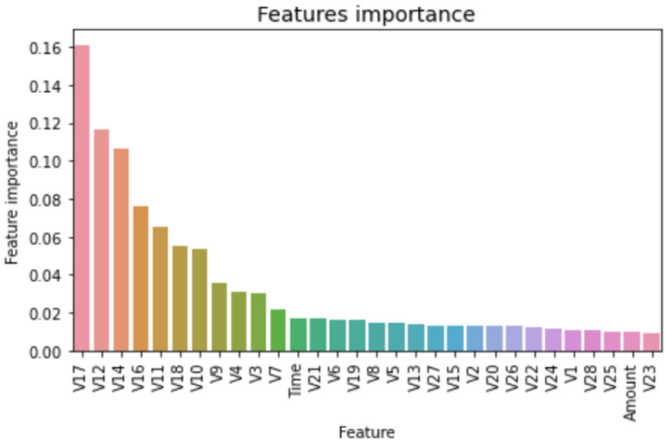

Fig. 3. Feature Importance Obtained from Extra Tree Classifier

4 Conclusion and Future Work

In this work, three tree-based models: a decision tree classifier, random forest classifier, and extra tree classifier, were employed and trained to evaluate their performance in identifying credit card fraud. While all algorithms have an ACC of 99.9%, which demonstrate that tree models can indeed show their relatively excellent performance as predicted in the presence of large sample data and unbalanced sample data. It also can discover that the extra tree classifier, a model that combines various decision trees, performs the best in all aspects. To further minimize the vulnerability of models to data imbalance, and boost prediction accuracy, a part of advanced dataset balancing techniques may be employed in the future study.

References

1. Quah, & Sriganesh, M. Real-time credit card fraud detection using computational intelligence. Expert Systems with Applications, 2008, 35(4), 1721–1732.
2. Wikipedia contributors. Credit card fraud. In Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Credit_card_fraud&oldid=1121448316, 2022.
3. Taha, & Malebary, S. J. An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine. IEEE Access, 2020, 8, 25579–25587.
4. Bhattacharyya, Jha, S., Tharakunnel, K., & Westland, J. C. Data mining for credit card fraud: A comparative study. Decision Support Systems, 2011, 50(3), 602–613.
5. Wikipedia contributors. Machine learning. In Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Machine_learning&oldid=1125574586, 2022.
6. Liu, & Liang, L.-Y. Credit Card Fraud Detection Based on KM-SVMSMOTE-CNN. Jisuanji Xitong Yingyong = Computer Systems and Applications, 2022, 6, 361–.
7. Ghosh, & Reilly. Credit card fraud detection with a neural-network. 1994 Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences, 1994, 3, 621–630.
8. Maes, S., Tuyls, K., Vanschoenwinkel, B., & Manderick, B. Credit card fraud detection using Bayesian and neural networks. In Proceedings of the 1st international naiso congress on neuro fuzzy technologie, 2002, Vol. 261, p. 270.

9. Srivastava, A., Kundu, A., Sural, S., & Majumdar, A. Credit card fraud detection using hidden Markov model. IEEE Transactions on dependable and secure computing, 2008, 5(1), 37-48.
10. Bahnsen, Stojanovic, A., Aouada, D., & Ottersten, B. Cost Sensitive Credit Card Fraud Detection Using Bayes Minimum Risk. 2013 12th International Conference on Machine Learning and Applications, 2013, 1, 333–338.
11. Sahin YG, Duman E. Detecting credit card fraud by decision trees and support vector machines. Proceedings of the International Multi Conference of Engineers and Computer Scientists. Hong Kong: International Association of Engineers, 2011, 442–447.
12. ULB, M. L. G.-. Credit Card Fraud Detection. Kaggle. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud, 2018.
13. Myles, Anthony J., et al. An introduction to decision tree modeling. Journal of Chemometrics: A Journal of the Chemometrics Society 18.6, 2004, 275–285.
14. Pedregosa et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12, 2011, 2825-2830.
15. Biau, Gérard, and Erwan Scornet. A random forest guided tour. Test 25.2, 2016, 197-227.
16. Susmaga, Robert. Confusion matrix visualization. Intelligent information processing and web mining. Springer, Berlin, Heidelberg, 2004. 107–116.