Journal of Big Data

RESEARCH Open Access

Comparative analysis of binary and one-class classification techniques for credit card fraud data

Joffrey L. Leevy[1*], John Hancock[1] and Taghi M. Khoshgoftaar[1]

*Correspondence:
jleevy2017@fau.edu

[1] Florida Atlantic University, 777 Glades Road, Boca Raton 33431, FL, USA

Abstract

The yearly increase in incidents of credit card fraud can be attributed to the rapid growth of e-commerce. To address this issue, effective fraud detection methods are essential. Our research focuses on the Credit Card Fraud Detection Dataset, which is a widely used dataset that contains real-world transaction data and is characterized by high class imbalance. This dataset has the potential to serve as a benchmark for credit card fraud detection. Our work evaluates the effectiveness of two supervised learning classification techniques, binary classification and one-class classification, for credit card fraud detection. The performance of five binary-class classification (BCC) learners and three one-class classification (OCC) learners is evaluated. The metrics used are area under the precision-recall curve (AUPRC) and area under the receiver operating characteristic curve (AUC). Our results indicate that binary classification is a better approach for detecting credit card fraud than one-class classification, with the top binary classifier being CatBoost.

Keywords: Binary classification, One-class classification, Credit card fraud, High class imbalance

Introduction

It is a well-known fact that binary classification is a widely used technique for solving classification problems in machine learning. This technique involves training a model to differentiate between instances belonging to one of two classes. However, several factors such as class noise [1], class imbalance [2], and inadequate data [2] can significantly affect the performance of a binary classification model. Researchers have proposed various solutions to overcome these issues and improve the effectiveness of binary classification models. To achieve high classification performance, it is crucial for a binary classification model to perform well for both classes, not just the class of interest, which is typically the minority or positive class.

In some cases, machine learning algorithms may face difficulty in recognizing patterns within data when there is an insufficient number of instances present in one or both classes. While it is essential for practitioners to have an adequate number of instances in the class of interest, this is not always feasible in practice.

In contrast to binary classification, one-class classification deals with instances from only one class. Two popular tasks in one-class classification are outlier detection and novelty detection [3]. Outlier detection and novelty detection are similar concepts, but they differ in their objectives. Outlier detection identifies instances in a dataset that significantly deviate from the majority of the data due to various factors such as measurement errors, data entry errors, natural variability, and data diversity. Such outliers can have a substantial impact on the performance of a machine learning model, and detecting and removing them can help to enhance the accuracy of the model. On the other hand, novelty detection aims to identify new, unseen instances that are different from the training data, without requiring class labels. Applications of novelty detection include intrusion detection, fraud detection, and surveillance.

The number of credit card fraud incidents has been increasing annually due to the rapid growth of e-commerce [4]. As a result, the development of effective credit card fraud detection methods has become crucial. To improve these detection systems, access to datasets having both high volume and diversity of transactions is increasingly necessary. In our study, we focus on the Credit Card Fraud Detection Dataset, which is an anonymized set of financial transactions that can be downloaded from the Kaggle website [5]. In this dataset, fraudulent transactions account for only 0.172% of the total records. We chose this dataset due to its real-world transaction content, high class imbalance, and potential to become a benchmark for credit card fraud detection [6].

When working with a dataset that has a binary class label, class imbalance is present when there are significantly more instances in one class than the other. This creates a majority class, which is usually referred to as the negative class, and a minority class, which is usually referred to as the positive class. If the number of instances in the two classes differs greatly, such as in the case of high class imbalance, it can affect the outcome of a machine learning experiment. Researchers define high class imbalance as a condition where the ratio of minority to majority instances is between 1:100 and 1:10,000 [7, 8].

Our work assesses the effectiveness of two different supervised learning classification techniques, binary classification and one-class classification, on the Credit Card Fraud Detection Dataset. The goal is to determine which approach performs better in identifying fraudulent transactions. We employ eight classifiers during experimentation: CatBoost [9], Extremely Randomized Trees [10], Random Forest [11], XGBoost [12], Logistic Regression [13], One-Class Support Vector Machine (SVM) [14], One-Class Gaussian Mixture Model (GMM) [15], and One-Class Adversarial Nets (OCAN) [16]. The first five classifiers are used for Binary-class classification (BCC), with the first four being Decision Tree-based ensembles. The last three (One-Class SVM, One-Class GMM, and OCAN) are used for one-class classification (OCC). To assess the effectiveness of the classifiers, we use the area under the precision-recall curve (AUPRC) [17] and area under the receiver operating characteristic curve (AUC) metrics [18].

Our study evaluates the performance of BCC and OCC techniques and presents a comparison between these approaches for credit card fraud detection. To the best of our knowledge, this research is the first to do this for both techniques, in relation to the Credit Card Fraud Detection Dataset. We utilize four ensembles of Decision Tree, three

OCC algorithms, and Logistic Regression. Our results may prove valuable for future research in this field.

The remaining sections of this paper are organized as follows: section Related work provides a review of related work; section Dataset presents an overview of the Credit Card Fraud Detection Dataset; section Classifiers discusses the classifiers being evaluated; section Training and testing outlines the training and testing methodology used; section Metrics describes the metrics used to assess classification performance; section Results and discussion presents the results and discusses their significance; and section Conclusion summarizes the main contributions of this work and provides recommendations for future research.

Related work

In this section, we examine research that focuses on detecting fraudulent instances from the Credit Card Fraud Detection Dataset. It is worth mentioning that the authors of these papers either employed their OCC algorithms in an unsupervised way, did not use the entire dataset, or did not indicate whether the entire dataset was used. In our methodology for the OCC learners, we use the entire dataset for training and testing, which aligns with the primary goal of one-class classification. Furthermore, we utilize the OCC algorithms as supervised learners.

Li et al. [19] proposed a dynamic weighted entropy hybrid approach to address class imbalance. To balance the data, they used Random Undersampling (RUS) prior to training and applied One-Class SVM and Isolation Forest for one-class classification. They assessed the models using F1 score and AUPRC. The authors combined minority and majority samples to train anomaly detection models, such as Isolation Forest, One-Class SVM, and an autoencoder, leaving a low imbalance ratio overlapping subset of leftover samples. They utilized Random Forest, deep neural networks (DNNs), and other nonlinear classifiers on this subset. They applied Dynamic Weighted Entropy to balance the exclusion of minority class outliers and the imbalance ratio of the overlapping subset. The authors evaluated their method using the Credit Card Fraud Detection Dataset and six private datasets, testing six hybrid and six non-hybrid models, with the hybrids outperforming the non-hybrids. The best performing model was a hybrid of an autoencoder and DNN, achieving F1 score and AUPRC values of 0.73 and 0.63, respectively. However, the authors did not provide information on the train-test procedure for the Random Forest classifier, nor did they explain their approach to converting One-Class SVM predictive scores to probabilities, which is necessary for obtaining AUPRC scores since One-Class SVM lacks a built-in probability output.

Jeragh and AlSulaimi [20] assessed the performance of their proposed hybrid model of an autoencoder and One-Class SVM for one-class classification, as well as other models, including an individual autoencoder, individual One-Class SVM, and a different hybrid of an autoencoder and One-Class SVM. They used Precision, Recall, F1 score, and Geometric Mean (G-mean) as metrics. The difference between the two hybrid models lies in the training of the One-Class SVM. In the authors'proposed model, the One-Class SVM is trained on the mean squared error between the input and output, while in the other approach, the One-Class SVM is trained directly on the output. In the authors'model, the input is first fed to the autoencoder, and then the input reconstruction error is

forwarded to the One-Class SVM. This model achieved the highest performance, with Precision, Recall, F1 score, and G-mean values of 0.938, 1, 0.9685, and 0.9998, respectively. However, the authors' paper would benefit from the inclusion of a Related Work section to better support their contribution.

Chandorkar [21] conducted a study to examine the effectiveness of three anomaly detection techniques: Local Outlier Factor, Isolation Forest, and One-Class SVM. Precision, Recall, and F1 score were the metrics used for evaluation. Only 10% of the Credit Card Fraud Detection dataset was used for the study. Local Outlier Factor and Isolation Forest both achieved perfect Precision, Recall, and F1 scores of 1. However, Isolation Forest had a higher accuracy of 0.9974 compared to Local Outlier Factor's 0.9966. The paper lacks detailed information, and some crucial data is missing. For instance, the results of One-Class SVM were not provided, and the use of Accuracy as a performance measure is potentially misleading.

Bodepudi [22] conducted a study to evaluate the performance of Local Outlier Factor, Isolation Forest, and One-Class SVM, experimenting with the same anomaly detection classifiers used by Chandorkar [21]. However, this study only used Accuracy as a performance metric. Isolation Forest was found to be the best-performing model, with an Accuracy score of 0.9974. We note that relying solely on Accuracy as a performance measure is problematic, especially when dealing with highly imbalanced data, as it can mask poor classification of the minority class. The author's paper is also lacking in specific details, such as information on their data preprocessing methods.

Ounacer et al. [23] conducted a performance comparison of three anomaly detection classifiers, namely Isolation Forest, Local Outlier Factor, and One-Class SVM, along with K-means clustering. The metric used for evaluation was AUC. Isolation Forest was found to be the top-performing model with an AUC score of 0.9168. However, using AUC as the only metric in a study can be misleading [24] when dealing with imbalanced data, as explained in Sect. 6. It is also unclear from the paper whether all 30 features were used in the experiment. Additionally, the authors did not mention their technique for converting One-Class SVM predictive scores to probabilities, which are necessary for deriving AUC scores.

Raza and Qayyum [25] compared the performance of their one-class classifier, a Variational Autoencoder model, with the performance of Decision Tree, SVM, and the Adaboost ensemble classifier. The metrics used were Recall, Precision, and F1 score. The autoencoder is ten layers deep, with normal transactions used for training. Abnormal (fraud) instances are detected by using the calculated reconstruction errors. Among the models, the autoencoder produced the highest Recall score (0.815) but the lowest Precision (0.742) and F1 score (0.776). In general, Adaboost produced the best performance results. The paper also shows the ROC curves for performances of the four methods. Adaboost has the highest AUC value (0.97) and Decision Tree has the lowest AUC value (0.89). We point out that no information is given on the train-test procedure for Decision Tree, SVM, and Adaboost.

Dornadula and Geetha [26] grouped customers according to their transactions and then profiled every card holder based on extracted patterns of behavior. The one-class classifiers used were Local Outlier Factor and Isolation Forest. In addition, the authors included the Logistic Regression, Decision Tree, and Random Forest classifiers. The

metrics used were Accuracy, Precision, and Matthews Correlation Coefficient. In addition to SMOTE [27], the authors determined that the use of Matthews correlation coefficient and One-Class classifiers were two effective means of addressing class imbalance in the dataset. Among the classifiers under evaluation, the authors observed that Logistic Regression, Decision Tree, and Random Forest were the top three performers. In this top group, Random Forest was the best performer, after the application of SMOTE, with best scores of 0.9998, 0.9996, and 0.9996 for Accuracy, Precision, and Matthews Correlation Coefficient. We point out that the train-test procedure for Logistic Regression, Decision Tree, and Random Forest was not provided.

Porwal and Mukund [28] proposed an approach for detecting instances of credit card fraud by assigning a consistency score to each data point using an ensemble of clustering methods. The AUC and AUPRC were the two metrics used in the study, with AUPRC being the primary metric. Using their approach, the authors calculated AUC and AUPRC scores, and compared them with scores obtained by one-class classifier Isolation Forest. The mean AUC score for the proposed approach was 0.8937 versus that of 0.9482 for Isolation Forest. The authors attribute the lower AUC score of their model to the fact that AUC presents an incomplete picture of performance for imbalanced datasets. The mean AUPRC score for the proposed approach was 0.2656 versus that of 0.1381 for Isolation Forest. In our view, the authors'proposed method does not involve a one-class algorithm.

Finally, Wu and Wang [29] proposed a model that uses an autoencoder as a generator for reconstructing transaction data and a fully connected neural network as a discriminator to for fraud detection. The metrics used were Accuracy, Precision, Recall, F1 score, and Matthews Correlation Coefficient. Local interpretability attempts to explain the decisions that a model makes about an instance, with the importance of input features either plotted or visualized. The explainers are built using LIME [30]. Other models evaluated include One-Class SVM, Object-based Convolutional Neural Network, Copular-based Outlier Detector, autoencoder, and One-Class Adversarial Nets. The authors'model was the top performer, with an Accuracy of 0.9061, Precision of 0.9216, Recall of 0.8878, F1 score of 0.9044, and Matthews Correlation Coefficient of 0.8128, respectively. Strictly speaking, the authors'proposed model is not a one-class classifier, but the approach can be used for one-class classification.

As far as we are aware, our study is the first to examine the combined use of one-class and binary classification for both AUC and AUPRC metrics in relation to the Credit Card Fraud Detection Dataset. We further note that previous research related to our work typically focused on either AUC or AUPRC alone, or do not use them at all. Moreover, we observed that many of these previous studies provide insufficient information, which can hinder the reproducibility of their experiments.

Dataset

The Credit Card Fraud Detection Dataset was jointly published by Worldline and the Université Libre de Bruxelles (ULB), and comprises transactions made by European credit cardholders. It consists of 284,807 instances and 30 input features, with 28 of the 30 input features transformed using Principal Component Analysis (PCA) [31]. Two features, namely "Time"and "Amount,"were not transformed. "Time"represents

the duration in seconds between a transaction and the first transaction in the dataset, while "Amount"indicates the transaction value. Although "Amount"was normalized, "Time"was excluded from the analysis since it behaves like a unique identifier, which could lead to overfitting and impact the reliability of the results.

The dataset has a binary label, where 1 denotes a fraudulent transaction and 0 indicates a non-fraudulent transaction. The dataset exhibits a high level of imbalance, with only 492 instances (0.172%) identified as fraudulent.

Classifiers

During experimentation, eight machine learning algorithms were utilized: CatBoost, XGBoost, Extremely Randomized Trees, Logistic Regression, Random Forest, One-Class GMM, One-Class SVM and OCAN. These classifiers belong to diverse families of machine learning algorithms, enabling robustness and generalizability of the results. The first five algorithms are BCC learners that assign data into one of two classes or labels, and are widely applied in various domains. Conversely, One-Class GMM, One-Class SVM, and OCAN are OCC learners that train on data associated with a single class, while disregarding or rejecting data from other classes.

CatBoost, Extremely Randomized Trees, Random Forest, and XGBoost are types of binary classifier ensembles made up of Decision Trees [32]. A Decision Tree predicts the class label of a data instance by traversing from the root to a leaf node. Ensemble learning combines the strengths of multiple models to overcome their individual weaknesses and make more accurate predictions. CatBoost and XGBoost are ensembles that are trained sequentially using boosting [33]. CatBoost uses Ordered Boosting [34], an algorithm that arranges the instances utilized by Decision Tree. XGBoost employs weighted quantile sketch and sparsity-aware function, where the former uses approximate tree learning [35] for merging and pruning processes and the latter exploits low-variance features. Random Forest and Extremely Randomized Trees are ensembles that are trained independently using bagging [36]. Random Forest determines the best split values for Decision Trees systematically, while Extremely Randomized Trees chooses these values randomly. Logistic Regression is a binary classifier that generates a score indicating the likelihood of belonging to a particular class. It is a linear model that employs a sigmoid function to produce a result between 0 and 1.

One-Class SVM is an algorithm that constructs a hypersphere in high-dimensional space to encompass as many data points as possible from a single class, while excluding those that do not belong, as explained in literature. The center of the hypersphere is calculated as the mean of the data points associated with the focused class, and the radius is set to enclose a specified percentage of data points. The algorithm then maximizes the distance between the hypersphere and the nearest data point that does not belong to the focused class, which becomes the decision boundary. One-Class SVM is popular because of its ability to handle high-dimensional data effectively and its robustness to noisy data [37].

One-Class GMM is an algorithm that is trained on a single class of data points and aims to identify outliers that deviate significantly from this class. It is a generative model that represents the data distribution as a weighted sum of multiple Gaussian distributions, where each Gaussian component represents a cluster of similar data

points. During training, the One-Class GMM learns the parameters of the Gaussian distributions, such as the mean and covariance, that best fit the focused class of data points. To classify a new data instance, the algorithm computes the likelihood that the instance belongs to the focused class, which is based on the probabilities of the instance being generated by each Gaussian component. If the likelihood falls below a certain threshold, the instance is considered an anomaly.

OCAN is an anomaly detection algorithm based on adversarial training [38]. It is designed to learn the underlying probability distribution of a single class of data points and distinguish between in-class and out-of-class samples. During training, the OCAN simultaneously trains a generator network [39] to produce realistic in-class samples and a discriminator network to classify whether a given sample is in-class or not. The generator network aims to generate in-class samples that are difficult for the discriminator network to distinguish from real in-class samples. The discriminator network, in turn, tries to differentiate between real in-class samples and generated samples. Through this adversarial process, the OCAN learns to identify the boundaries of the focused class and differentiate between in-class and out-of-class samples. To classify a new data instance, the algorithm computes the distance between the instance and the focused class in the feature space. If the distance falls below a certain threshold, the instance is considered in-class; otherwise, it is an anomaly. For the purpose of our experiments, instances from the majority class of the Credit Card Fraud Dataset are in-class, and the minority class instances are the anomalies.

Training and testing

For this study, the setup for the experiments involved utilizing a distributed computing platform comprised of nodes equipped with 16-core Intel Xeon CPUs, 256 GB RAM per CPU, and Nvidia V100 GPUs. Training and testing algorithms were implemented using the Python programming language. CatBoost, XGBoost and OCAN were used as standalone libraries. Random Forest, Extremely Randomized Trees, Logistic Regression, One-Class SVM, and One-Class GMM were implemented within the Scikit-learn [40] library.

The data was divided into training and testing sets using an 80:20 ratio. For the OCC algorithms, instances from the minority class were excluded, and only instances from the majority class were used for training. The training phase employed the k-fold cross-validation method, with the model being trained on k-1 folds and tested on the remaining fold in each iteration to maximize data utilization. Stratification was applied to the cross-validation process to ensure proportional representation of each class across the folds. A value of k=5 was selected, with 4 folds allocated for training and 1 fold for testing. To minimize the risk of data loss caused by random sampling of instances from the majority class, 10 iterations of cross-validation were executed. This approach resulted in 50 performance scores per classifier for each metric.

To prevent overfitting in the Decision-Tree based classifiers, the Maximum Tree Depth parameters indicated in Table 1 were utilized. These depths were determined through preliminary experimentation. We utilized default values for all other parameters.

Table 1 Maximum tree depths used in experiments

Classifier	Maximum tree depth
CatBoost	`max_depth=5`
Extremely Randomized Trees	`max_depth=8`
Random Forest	`max_depth=4`
XGBoost	`max_depth=1`

Table 2 Confusion matrix

	Predicted class	
	Positive	**Negative**
Actual class		
Positive	True Positive (TP)	False Negative (FN) (Type II error)
Negative	False Positive (FP)	True Negative (TN) (Type I error)

Metrics

We rely on a confusion matrix (Table 2) in our work, where the minority class is typically the class of interest, and the majority class is its opposite class, i.e., positives and negatives, respectively. The following are simple performance metrics [41], along with their definitions:

- *True Positive (TP):* the number of positive samples that are correctly identified as positive.
- *True Negative (TN):* the number of negative samples that are correctly identified as negative.
- *False Positive (FP)*, also known as Type I error: the number of negative instances that are incorrectly identified as positive.
- *False Negative (FN)*, also known as Type II error: the number of positive instances that are incorrectly identified as negative.

From these basic metrics, other performance metrics can be calculated as follows:

- *Recall*, also known as True Positive Rate (TPR) or sensitivity, can be computed as $TP/(TP + FN)$.
- *Precision*, also known as positive predictive value, can be computed as $TP/(TP + FP)$.
- *False Alarm Rate*, also known as False Positive Rate (FPR), can be computed as $FP/(FP + TN)$.

In order to gain a deeper understanding of the difficulties involved in evaluating machine learning models with severely imbalanced data, we utilized more than one performance metric. The two metrics we used, AUC and AUPRC, are described in detail below.

AUC, or the Area Under the Receiver Operating Characteristic (ROC) Curve, assesses the quality of a classifier by summarizing the balance between its TPR and FPR. The

ROC curve illustrates the relationship between the TPR and FPR metrics. AUC evaluates the classifier's performance by taking into account all potential classification thresholds along the ROC curve. This provides a comprehensive assessment of the model's performance, condensing it into a single value and enabling effective comparisons between different classifiers. AUC can take a value between 0 and 1, with a higher value indicating better classifier performance. A model that guesses randomly yields an AUC score of 0.5.

AUPRC, or the area under the precision-recall curve, quantifies the balance between Precision and Recall by plotting Precision against Recall for various classification thresholds. AUPRC can also take a value between 0 and 1, with a higher value indicating better classifier performance.

Based on the definition of AUC, it can be inferred that having a significantly large number of true negatives in a dataset would result in an inconsequentially small number of false positives. However, the number of true negatives is not taken into account in the AUPRC definition. Therefore, in the case of an extremely imbalanced dataset such as the Credit Card Fraud Detection Dataset, AUPRC offers a more precise measure of the number of false positives [42, 43]. As a result, for this study, AUPRC is deemed more important than AUC.

It is worth noting that both AUC and AUPRC require class probabilities as inputs. However, One-Class SVM does not provide probability estimates for its predictions by default and requires calibration. To address this, we used sigmoid calibration and isotonic regression as two different approaches. Sigmoid calibration involves adjusting the output predictive scores of the One-Class SVM algorithm using the Logistic Regression model. This enabled us to convert the One-Class SVM output scores into class probabilities, specifically for the positive class. The approach is based on the work of Platt [44]. Sigmoid calibration is more flexible than isotonic regression, as it can capture non-monotonic relationships between the scores and the probabilities. Isotonic regression is a non-parametric method that fits a piecewise-constant function to the output scores of the classifier. This function maps the scores to probabilities in a monotonic fashion, which means that a higher score always corresponds to a higher probability. Isotonic regression is generally computationally efficient and has the advantage of being model-agnostic, meaning that it can be applied to any type of classifier. In the context of classifier calibration, isotonic regression was first proposed by Zadrozny and Elkan [45].

We point out that the One-Class GMM algorithm inherently provides probability estimates for each data point by modeling the probability density function of the data using a mixture of Gaussian distributions. These probability estimates can be computed directly from the model parameters without the need for additional transformations or functions such as the sigmoid function. However, due to poor results for AUC and AUPRC in preliminary experiments, we used Logistic Regression (sigmoid calibration) with One-Class GMM. The OCAN classifier also natively produces output values that are probability estimates. It was not necessary to perform calibrations for OCAN.

Results and discussion

Table 3 presents classification results for experiments with the three OCC algorithms for the AUC and AUPRC metrics. Each score shown in the table is the mean of 50 performance scores on the test folds. Among these models, One-Class GMM

Table 3 Mean AUC and AUPRC scores by classifier for OCC type classifiers

Classifier	AUC	AUPRC
One-Class GMM (scores converted with Sigmoid Calibration)	0.9496	0.4971
OCAN	0.9409	0.3471
One-Class SVM (scores converted with Isotonic Regression)	0.9091	0.4165
One-Class SVM (scores converted with Sigmoid Calibration)	0.9084	0.3775

Table 4 Mean AUC and AUPRC scores by classifier for BCC type classifiers

Classifier	AUC	AUPRC
CatBoost	0.9751	0.8567
ET	0.9731	0.8097
Logistic Regression	0.9794	0.7490
Random Forest	0.9620	0.8069
XGBoost	0.9786	0.8549

achieved the highest AUC and AUPRC scores of 0.9496 and 0.4971, respectively. This indicates that One-Class GMM may be a promising algorithm for credit card fraud detection. In contrast, One-Class SVM with sigmoid calibration produced the lowest AUC score of 0.9084, while OCAN yielded the lowest AUPRC score of 0.3471.

Table 4 presents classification results for experiments with the five BCC algorithms for the AUC and AUPRC metrics. Each score shown in the table is the mean of 50 performance scores on the test folds. Among these models, Logistic Regression achieved the highest AUC score of 0.9794, while CatBoost achieved the highest AUPRC score of 0.8567. Conversely, Random Forest recorded the lowest AUC score of 0.9620, while Logistic Regression recorded the lowest AUPRC score of 0.7490.

As previously explained, we prioritize the AUPRC scores in this study due to the ability of this metric to provide more informative results compared to the AUC scores. The BCC learners listed in Table 4 demonstrate mean AUPRC scores ranging from 0.8567 to 0.7490. Based on these AUPRC scores, CatBoost is the best algorithm for credit card fraud detection. We note that Logistic Regression has a lower AUPRC score than any of the Decision Tree-based ensembles. The OCC learners listed in Table 3 display a range of mean AUPRC scores between 0.4975 and 0.3471, signifying a decline in classification performance. Specifically for credit card fraud detection, this decline could be due to the fact that one-class classification may struggle with identifying instances that deviate from the norm. This could include instances that are not fraud but are simply different from the majority of instances in the dataset.

Our findings imply that binary classification is a superior approach for detecting credit card fraud in comparison to one-class classification. However, we caution that our results pertain to one dataset, and more research will show whether our findings hold in general.

Conclusion

To the best of our knowledge, this paper is the first to assess one-class and binary classification techniques for AUC and AUPRC metrics, with regard to the Credit Card Fraud Detection Dataset. We employed several classifiers, including ensembles of Decision Tree, Logistic Regression, One-Class SVM, One-Class GMM, and OCAN, and evaluated their performance. The results indicated that binary classification is a better approach for detecting credit card fraud than one-class classification, with the BCC learners demonstrating mean AUPRC scores ranging from 0.8567 to 0.7490, while the OCC learners displayed a range of mean AUPRC scores between 0.4975 and 0.3471. CatBoost was the top performer, with an AUPRC score of 0.8567. The scores associated with the OCC learners point to a deterioration in classification performance. The larger disparity between their AUPRC scores shows that the OCC learners have more difficulty identifying the minority class. Our study provides a foundation for future work in the field of credit card fraud detection. We note that the results are specific to our analysis of the Medicare Part D dataset and require further investigation to determine their applicability for other datasets. Our methodology can be expanded by applying it to evaluate other fraud detection datasets and incorporating additional one-class classifiers.

Abbreviations

AUC	Area under the receiver operating characteristic curve
AUPRC	Area under the precision-recall curve
BCC	Binary-class classification
DNN	Deep neural network
ET	Extremely Randomized Trees
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
GBDT	Gradient-Boosted Decision Tree
GMM	Gaussian Mixture Model
k-NN	k-Nearest Neighbor
OCAN	One-Class Adversarial Nets
OCC	One-class classification
PCA	Principal Component Analysis
ROC	Receiver Operating Characteristic
ROS	Random Oversampling
RUS	Random Undersampling
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
ULB	Université Libre de Bruxelles

Acknowledgements

We would like to thank all reviewers of this manuscript.

Author contributions

JLL searched for relevant papers and drafted the manuscript. All authors provided feedback to JLL and helped shape the work. JLL and JH prepared the manuscript. TMK introduced this topic to JLL and helped to complete and finalize the work. All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

Not applicable.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 15 March 2023 Accepted: 21 June 2023
Published online: 17 July 2023

References

1. Pandey SK, Tripathi AK. An empirical study toward dealing with noise and class imbalance issues in software defect prediction. Soft Computing. 2021;25(21):13465–92.
2. Al-Stouhi S, Reddy CK. Transfer learning for class imbalance problems with inadequate data. Knowl Inf Syst. 2016;48:201–28.
3. Seliya N, Abdollah Zadeh A, Khoshgoftaar TM. A literature review on one-class classification and its potential applications in big data. J Big Data. 2021;8(1):1–31.
4. Alharbi A, Alshammari M, Okon OD, Alabrah A, Rauf HT, Alyami H, Meraj T. A novel text2img mechanism of credit card fraud detection: a deep learning approach. Electronics. 2022;11(5):756.
5. Kaggle: Credit Card Fraud Detection. 2018. https://www.kaggle.com/mlg-ulb/creditcardfraud.
6. Leevy JL, Johnson JM, Hancock J, Khoshgoftaar TM. Threshold optimization and random undersampling for imbalanced credit card data. J Big Data. 2023;10(1):1–22.
7. He H, Garcia EA. Learning from imbalanced data. IEEE Trans Knowl Data Eng. 2009;21(9):1263–84.
8. Kennedy R.K, Salekshahrezaee Z, Khoshgoftaar T.M. A novel approach for unsupervised learning of highly-imbalanced data. In: 2022 IEEE 4th International Conference on Cognitive Machine Intelligence (CogMI), IEEE; 2022. pp. 52–58.
9. Sanjeetha R, Raj A, Saivenu K, Ahmed MI, Sathvik B, Kanavalli A. Detection and mitigation of botnet based ddos attacks using catboost machine learning algorithm in sdn environment. Int J Adv Technol Eng Exploration. 2021;8(76):445.
10. Acosta MRC, Ahmed S, Garcia CE, Koo I. Extremely randomized trees-based scheme for stealthy cyber-attack detection in smart grid networks. IEEE Access. 2020;8:19921–33.
11. Dileep M, Navaneeth A, Abhishek M. A novel approach for credit card fraud detection using decision tree and random forest algorithms. In: 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), IEEE; 2021. pp. 1025–1028.
12. Priscilla C.V, Prabha D.P. Influence of optimizing xgboost to handle class imbalance in credit card fraud detection. In: 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), IEEE; 2020. pp. 1309–1315.
13. Chiramdasu R, Srivastava G, Bhattacharya S, Reddy P.K, Gadekallu T.R. Malicious url detection using logistic regression. In: 2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), IEEE; 2021. pp. 1–6.
14. Mhamdi L, McLernon D, El-Moussa F, Zaidi S.A.R, Ghogho M, Tang T. A deep learning approach combining autoencoder with one-class svm for ddos attack detection in sdns. In: 2020 IEEE Eighth International Conference on Communications and Networking (ComNet), IEEE; 2020. pp. 1–6.
15. Hayashi T, Fujita H. One-class ensemble classifier for data imbalance problems. Appl Intell. 2022;52(15):17073–89.
16. Zheng P, Yuan S, Wu X, Li J, Lu A. One-class adversarial nets for fraud detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019. pp. 1286–1293.
17. Boyd K, Eng K.H, Page C.D. Area under the precision-recall curve: point estimates and confidence intervals. Joint European conference on machine learning and knowledge discovery in databases, Springer; 2013. 451–466.
18. Bekkar M, Djemaa H.K, Alitouche T.A. Evaluation measures for models assessment over imbalanced data sets. J Inf Eng Appl. 2013;3(10).
19. Li Z, Huang M, Liu G, Jiang C. A hybrid method with dynamic weighted entropy for handling the problem of class imbalance with overlap in credit card fraud detection. Expert Syst Appl. 2021;175:1–10.
20. Jeragh M, AlSulaimi M. Combining auto encoders and one class support vectors machine for fraudulant credit card transactions detection. In: 2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), IEEE; 2018. pp. 178–184.
21. Chandorkar A. Credit card fraud detection using machine learning. Int Res J Modern Eng Technol Sci. 2022;4:42–50.
22. Bodepudi H. Credit card fraud detection using unsupervised machine learning algorithms. Int J Comput Trends Technol. 2021;69:1–13.
23. Ounacer S, El Bour HA, Oubrahim Y, Ghoumari MY, Azzouazi M. Using isolation forest in anomaly detection: the case of credit card transactions. Period Eng Nat Sci. 2018;6(2):394–400.
24. Hancock J, Khoshgoftaar T.M, Johnson J.M. Informative evaluation metrics for highly imbalanced big data classification. In: 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE; 2022.
25. Raza M, Qayyum U. Classical and deep learning classifiers for anomaly detection. In: 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), IEEE;2019. pp. 614–618.
26. Dornadula VN, Geetha S. Credit card fraud detection using machine learning algorithms. Proc Comput Sci. 2019;165:631–41.

27. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. Smote: synthetic minority over-sampling technique. J Artif Intell Res. 2002;16:321–57.
28. Porwal U, Mukund S. Credit card fraud detection in e-commerce. In: 2019 18th IEEE international conference on trust, security and privacy in computing and communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), IEEE;2019. pp. 280–287.
29. Wu T.-Y, Wang Y.-T. Locally interpretable one-class anomaly detection for credit card fraud detection. In: 2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), IEEE;2021. pp. 25–30.
30. Ribeiro MT, Singh S, Guestrin C. "why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144 (2016)
31. Salekshahrezaee Z, Leevy J.L, Khoshgoftaar T.M. Feature extraction for class imbalance using a convolutional autoencoder and data sampling. In: 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI), IEEE;2021. pp. 217–223.
32. Kushwah JS, Kumar A, Patel S, Soni R, Gawande A, Gupta S. Comparative study of regressor and classifier with decision tree using modern tools. Mater Today Proc. 2022;56:3571–6.
33. Basha SM, Rajput DS, Vandhan V. Impact of gradient ascent and boosting algorithm in classification. Int J Intell Eng Syst (IJIES). 2018;11(1):41–9.
34. Prokhorenkova L, Gusev G, Vorobev A, Dorogush A.V, Gulin A. Catboost: unbiased boosting with categorical features. In: Advances in Neural Information Processing Systems, 2018. pp. 6638–6648.
35. Gupta A, Nagarajan V, Ravi R. Approximation algorithms for optimal decision trees and adaptive tsp problems. Math Oper Res. 2017;42(3):876–96.
36. González S, García S, Del Ser J, Rokach L, Herrera F. A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. Inf Fusion. 2020;64:205–37.
37. Kassab R, Alexandre F. Incremental data-driven learning of a novelty detection model for one-class classification with application to high-dimensional noisy data. Mach Learn. 2009;74:191–234.
38. Sriramanan G, Addepalli S, Baburaj A, et al. Towards efficient and effective adversarial training. Adv Neural Inf Process Syst. 2021;34:11821–33.
39. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial networks. Commun ACM. 2020;63(11):139–44.
40. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al: Scikit-learn: Machine learning in python. J Mach Learn Res. 2011;12:2825–2830.
41. Seliya N, Khoshgoftaar TM, Van Hulse J. A study on the relationships of classifier performance metrics. In: ICTAI'09. 21st International Conference On Tools with Artificial Intelligence, IEEE;2009. pp. 59–66.
42. Hancock JT, Khoshgoftaar TM, Johnson JM. Evaluating classifier performance with highly imbalanced big data. J Big Data. 2023;10(1):1–31.
43. Davis J, Goadrich M. The relationship between precision-recall and roc curves. In: Proceedings of the 23rd International Conference on Machine Learning, 2006. pp. 233–240.
44. Platt J, et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Adv Large Margin Classifiers. 1999;10(3):61–74.
45. Zadrozny B, Elkan C. Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 694–699 (2002)

Publisher's Note