Leevy *et al. Journal of Big Data* (2023) 10:157
https://doi.org/10.1186/s40537-023-00825-1

Journal of Big Data

RESEARCH Open Access

Investigating the effectiveness of one-class and binary classification for fraud detection

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

*Correspondence:
jleevy2017@fau.edu

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

Abstract

Research into machine learning methods for fraud detection is of paramount importance, largely due to the substantial financial implications associated with fraudulent activities. Our investigation is centered around the Credit Card Fraud Dataset and the Medicare Part D dataset, both of which are highly imbalanced. The Credit Card Fraud Detection Dataset is large data and contains actual transactional content, which makes it an ideal benchmark for credit card fraud detection. The Medicare Part D dataset is big data, providing researchers the opportunity to examine national trends and patterns related to prescription drug usage and expenditures. This paper presents a detailed comparison of One-Class Classification (OCC) and binary classification algorithms, utilizing eight distinct classifiers. OCC is a more appealing option, since collecting a second label for binary classification can be very expensive and not possible to obtain within a reasonable time frame. We evaluate our models based on two key metrics: the Area Under the Precision-Recall Curve (AUPRC)) and the Area Under the Receiver Operating Characteristic Curve (AUC). Our results show that binary classification consistently outperforms OCC in detecting fraud within both datasets. In addition, we found that CatBoost is the most performant among the classifiers tested. Moreover, we contribute novel results by being the first to publish a performance comparison of OCC and binary classification specifically for fraud detection in the Credit Card Fraud and Medicare Part D datasets.

Keywords: Binary Classification, One-Class Classification, Credit Card Fraud, Medicare Fraud, High Class Imbalance, Big Data

Introduction

Binary classification is a technique extensively applied in machine learning to distinguish between two classes. However, the performance of a classification model can be substantially impacted by issues such as class noise [1], class imbalance [2], and data scarcity [3]. To mitigate these challenges and augment the efficacy of binary classification models, many solutions have been suggested by researchers. In order to ensure optimal classification results, it is paramount that the binary classification model excels in identifying both classes, not merely the class of interest, which is often the main focus.

For a dataset with binary class labels, class imbalance occurs when there is a significant disparity in the number of instances between the classes. This discrepancy results in a majority class, typically labeled as the negative class, and a minority class, usually

identified as the positive class. When the discrepancy is prominent, which is a scenario referred to as high class imbalance, the outcome of a machine learning experiment can be affected. High class imbalance is specifically defined by researchers as a situation where the minority-to-majority instance ratio lies between 1:100 and 1:10,000 [4]. This imbalance can pose challenges for machine learning models as they tend to be biased towards the majority class during training, thereby neglecting the minority class. Addressing class imbalance is essential to ensure that the trained models can output accurate predictions for both classes. Although it is important for practitioners to have a sufficient quantity of instances in the class of interest, it is not always practical or achievable in real-world applications.

Unlike binary classification, One-Class Classification (OCC) concentrates on instances from a single class, with outlier detection and novelty detection being two prominent tasks within this field [5]. While there are similarities between outlier detection and novelty detection, their goals vary significantly. Outlier detection focuses on recognizing instances in a dataset that starkly differ from the bulk of the data, a variance potentially brought on by factors such as measurement errors, data entry mistakes, natural variability, and data diversity. These anomalies can significantly impact the performance of a machine learning model. Conversely, the objective of novelty detection is to discern, without the use of class labels, any previously unseen instances that diverge from the training data. This technique finds applications in areas like intrusion detection, fraud detection, and surveillance.

Investigating machine learning techniques for fraud detection is of utmost importance, primarily because of the significant financial consequences linked to fraudulent actions. This motivation for research is particularly poignant when considering the extent of credit card and Medicare fraud. Fraud, in various forms, can inflict severe financial damage on businesses and individuals alike. As our society becomes increasingly digital, fraudulent activities have become more complex and difficult to detect. Traditional methods often fall short, making it crucial to develop advanced machine learning methods that can accurately and efficiently identify fraudulent patterns. These innovative techniques could potentially save billions of dollars annually, and safeguard individuals, businesses, and even economies against the damaging impacts of financial fraud.

In our research, we utilize two binary class datasets to detect fraud. One of these datasets, the Credit Card Fraud Detection Dataset (large data), is a collection of anonymized financial transactions available on the Kaggle platform [6]. The original form of the second dataset, Medicare Part D (big data), is not ready for our machine learning data analytics. However, we have pre-processed and prepared Part D for this purpose. For more information, please see [7]. This preparation involves consolidating Part D [8] data from the Centers for Medicare and Medicaid Services (CMS) website and data from the List of Excluded Individuals and Entities (LEIE) [9]. The Credit Card Fraud Detection Dataset is particularly noteworthy because it comprises real-life transaction information and exhibits a pronounced class imbalance, making it a potential benchmark for credit card fraud detection efforts. This dataset can offer valuable insights into the intricacies of fraudulent credit card transactions, assisting in the development of more accurate detection models. The Medicare Part D dataset is big data that details prescription drug events for beneficiaries. It provides a wealth of information, including the types and

quantities of prescribed medications, the healthcare providers prescribing them, and the related costs. This extensive database allows researchers to study trends and patterns in prescription drug usage and spending on a national scale. The LEIE provides details about individuals and organizations that have been barred from federal healthcare programs like Medicare and Medicaid due to engagement in fraudulent or abusive practices.

Our study aims to evaluate the performance of two different supervised learning classification methods, binary and OCC. Choosing OCC is more desirable, as the process of gathering a second label for binary classification can be quite costly and may not be achievable within a practical time frame. The primary objective is to discern which methodology is more effective in detecting fraudulent transactions for the Credit Card Fraud Detection Dataset and the Medicare Part D dataset. We engage eight classifiers for our experiments: CatBoost [10], Extremely Randomized Trees [11], Random Forest [12], XGBoost [13], Logistic Regression [14], One-Class Support Vector Machine (SVM) [15], One-Class Gaussian Mixture Model (GMM) [16], and One-Class Adversarial Nets (OCAN) [17]. The first five classifiers (CatBoost, Extremely Randomized Trees, Random Forest, XGBoost, Logistic Regression) are applied for binary classification, with the first four being Decision Tree-based ensembles. The last three classifiers (One-Class SVM, One-Class GMM, and OCAN) are OCC algorithms. To measure the performance and determine the effectiveness of these classifiers, we utilize Area Under the Precision-Recall Curve (AUPRC) and Area Under the Receiver Operating Characteristic Curve (AUC) metrics [18].

Our research examines the effectiveness of binary and OCC methods as applied to credit card fraud detection and Medicare fraud detection. To the best of our knowledge, this investigation is the first to do this for the Credit Card Fraud Detection Dataset (large data) and Medicare Part D dataset (big data). We utilize four ensembles of Decision Tree, three OCC algorithms, and Logistic Regression in our study. Our contribution is expected to be valuable for future studies in these domains.

The remaining sections of this paper are organized as follows: Section "Related work" provides a review of pertinent literature; Section "Datasets" presents a synopsis of the Credit Card Fraud Detection Dataset and the Medicare Part D dataset; Section "Classifiers" elaborates on the classifiers under evaluation; Section "Training and testing" lays out the strategy adopted for training and testing; Section "Metrics" describes the metrics used to assess classification performance; Section "Results and discussion" presents the results and their implications; and Section "Conclusion" summarizes the primary contributions of this study and proposes avenues for future research.

Related work

This section deals with related works based on the detection of fraudulent instances within the Credit Card Fraud Detection Dataset. To the best of our knowledge, there is no literature available on our Part D dataset with regard to OCC algorithms. It is important to note that the authors of the respective related works either applied their OCC algorithms in an unsupervised manner, did not utilize the whole dataset, or did not specify if the whole dataset was utilized. In contrast, our methodology for the OCC learners incorporates the complete dataset for both training and testing. This conforms to the principal objective of OCC. Additionally, we employ the OCC algorithms as supervised

learners. Finally, our study distinguishes itself from the related works by being the only one that deals with big data.

Li et al. [19] introduced a dynamic weighted entropy hybrid method to tackle the issue of class imbalance. To equalize the data, they implemented Random Undersampling (RUS) before training and employed One-Class SVM and Isolation Forest for OCC. F1 score and AUPRC were used as metrics to evaluate their models. The authors combined minority and majority samples to train anomaly detection models such as Isolation Forest, One-Class SVM, and an autoencoder, leaving a small subset of overlapping samples with a low imbalance ratio. On this subset, they applied Random Forest, Deep Neural Networks (DNNs), and other nonlinear classifiers. Dynamic Weighted Entropy was used to balance the omission of minority class outliers and the imbalance ratio of the overlapping subset. They tested their methodology using the Credit Card Fraud Detection Dataset and six private datasets, by assessing six hybrid and six non-hybrid models, with hybrid models demonstrating superior performance. The top-performing model was a hybrid of an autoencoder and DNN, achieving F1 score and AUPRC of 0.73 and 0.63, respectively. Nonetheless, the authors failed to provide details about the train-test procedure for the Random Forest classifier or how they converted One-Class SVM predictive scores to probabilities. This conversion is essential to obtain AUPRC scores since One-Class SVM does not inherently produce a probability output.

Jeragh and AlSulaimi [20] evaluated the effectiveness of their proposed hybrid model of an autoencoder and One-Class SVM for OCC. They also tested other models, which included a standalone autoencoder, a standalone One-Class SVM, and a different hybrid of an autoencoder and One-Class SVM. Metrics such as Precision, Recall, F1 score, and Geometric Mean (G-mean) were used for evaluation. The difference between the two hybrid models is due to the training process of One-Class SVM. In the proposed model by the authors, One-Class SVM is trained using the mean squared error between the input and output, while in the alternate approach, One-Class SVM is trained directly on the output. In the authors' model, the input is initially processed by the autoencoder, after which the input reconstruction error is passed on to One-Class SVM. This model yielded the best performance with Precision, Recall, F1 score, and G-mean scores of 0.938, 1, 0.9685, and 0.9998 respectively. The authors noted that changing the hyperparameters of the autoencoder of the proposed model can affect the efficiency of the final prediction. However, they have not provided optimal values for the hyperparameters or indicated whether only default values were used.

Chandorkar [21] assessed the performance of three anomaly detection methods: Local Outlier Factor, Isolation Forest, and One-Class SVM. The evaluation metrics employed were Precision, Recall, and F1 score. Only a tenth of the Credit Card Fraud Detection dataset was utilized for the study. Both Local Outlier Factor and Isolation Forest achieved perfect Precision, Recall, and F1 scores of 1. However, Isolation Forest displayed a higher Accuracy score of 0.9974, in contrast to Local Outlier Factor's 0.9966. Their paper lacks comprehensive details, and some important information has been left out. For example, the performance results of One-Class SVM were not shown.

Bodepudi [22] evaluated the effectiveness of Local Outlier Factor, Isolation Forest, and One-Class SVM, employing the same anomaly detection classifiers as those utilized by Chandorkar [21]. However, this study used Accuracy as the sole

performance metric. Isolation Forest emerged as the top-performing model, achieving an Accuracy of 0.9974. We again stress that using Accuracy exclusively as a performance measure can be problematic. particularly when highly imbalanced data is involved, as it may conceal poor classification of the underrepresented class. The author's paper is also lacking in specific details, such as information on their data preprocessing methods.

Ounacer et al. [23] analyzed the performance of three anomaly detection classifiers, specifically, Isolation Forest, Local Outlier Factor, and One-Class SVM, along with K-means clustering. AUC was the only metric used for assessment. Isolation Forest was the best-performing model, achieving an AUC score of 0.9168. We caution that relying on AUC as the only evaluation metric may lead to false conclusions [24], especially when dealing with highly imbalanced data, as detailed in Sect. 6. Moreover, it is unclear from their paper if all 30 features of the dataset were utilized in the experiment. The authors also did not include their method for converting One-Class SVM predictive scores into probabilities, which is a necessary step for computing AUC scores.

Raza and Qayyum [25] compared and evaluated their Variational Autoencoder model, and other models including Decision Tree, SVM, and an Adaboost ensemble classifier. The metrics used were Recall, Precision, and F1 score. The autoencoder model, composed of ten layers, was trained on normal transactions. Abnormal (fraudulent) instances were identified via computed reconstruction errors. Among all the models, the autoencoder registered the highest Recall (0.815) but the lowest Precision (0.742) and F1 score (0.776). Overall, Adaboost yielded the best performance results. Their paper also shows ROC curves representing the performance of the four methods. Adaboost had the highest AUC (0.97), while Decision Tree registered the lowest (0.89). We note that their work does not provide details on the train-test procedure employed for Decision Tree, SVM, and Adaboost.

Lastly, Wu and Wang [26] introduced a model that employs an autoencoder as a generator to reconstruct transaction data and a fully connected neural network as a discriminator for fraud detection. They used Accuracy, Precision, Recall, F1 score, and Matthews Correlation Coefficient as metrics for evaluation. The authors also evaluated other models, including One-Class SVM, an Object-based Convolutional Neural Network, a Copular-based Outlier Detector, an autoencoder, and One-Class Adversarial Nets. Their proposed model performed best, achieving 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. It should be noted that while the authors' proposed model is not an OCC algorithm, the approach can be used for one-class classification.

Datasets

This section is divided into two parts, each focusing on one of the fraud datasets used in our study. The Credit Card Fraud Detection Dataset can be accessed online and was readily available for our analysis. For the Medicare Part D Dataset, although the data is also available online, we performed additional processing to transform the Medicare data into the desired format for our research purposes.

Credit card fraud detection dataset

The Credit Card Fraud Detection Dataset, a collaboration between Worldline and the Université Libre de Bruxelles (ULB), consists of transactions conducted by European credit cardholders. The dataset comprises 284,807 instances and includes 30 input features. Among these features, 28 underwent transformation using Principal Component Analysis (PCA) [27], while two features, namely "Time" and "Amount," were left untransformed. The "Time" feature represents the time duration in seconds between a transaction and the first transaction recorded in the dataset. On the other hand, the "Amount" feature indicates the monetary value associated with each transaction. Although the "Amount" feature was normalized, the "Time" feature was excluded from the analysis. This decision was made due to the "Time" feature acting more like a unique identifier, which could potentially lead to overfitting and impact the reliability of the results.

The dataset has a binary label, where the value 1 represents a fraudulent transaction, and the value 0 represents a non-fraudulent transaction. Notably, the dataset is highly imbalanced, with only 492 instances (0.172%) identified as fraudulent transactions.

Medicare part D dataset

The Part D Dataset is derived from physician claims associated with the Medicare Part D Prescription Drug Program. This dataset is available for download from the CMS. In this dataset, physicians are identified by their unique National Provider Identifier (NPI), while drugs are labeled with both their brand and generic names. The dataset provides additional information, including average payments and charges, drug quantities prescribed, and medical specialty. The CMS has aggregated the data based on the combination of NPI and drug name. Each row of the dataset corresponds to a unique combination of NPI, provider type, and drug name for a physician. It includes various features such as gender and related information. To ensure the privacy of Medicare beneficiaries, aggregated records with less than 11 claims are excluded from the dataset.

Part D, in its original form, is not suitable for our machine learning data analytics. However, we have modified and readied the dataset for this task. We rely on two distinct resources for our Part D data and follow the technique described in [7] to process the original dataset. The initial resource is "Medicare Part D Prescribers – by Provider and Drug" [28], followed by "Medicare Part D Prescribers – by Provider" [29]. The primary distinction between these two resources lies in the degree of detail provided in the data. The "Medicare Part D Prescribers – by Provider and Drug" source provides a unique record for each healthcare provider, medication prescribed, and year - this is often referred to as the "provider-drug-level Part D data". In contrast, the second source, "Medicare Part D Prescribers – by Provider", is less granular, offering a record for each healthcare provider per year. This is what we term as the "provider-level Part D data".

The provider-drug-level Part D data consists of 22 attributes, not all of which are relevant for machine learning applications. These include attributes linked to the provider's name and address, which we opt to exclude as they could act as unique identifiers that a machine learning model might memorize rather than effectively generalize the data. However, we do maintain the NPI for labeling purposes later on. This dataset features two categorical attributes that indicate the type of medication prescribed. During our dataset aggregation process, we choose to discard these categorical features. The claim

year is another attribute that, while useful for processing and aggregation by year, is not used as an attribute for supervised machine learning and is therefore discarded.

As we compile data at the provider level, we do maintain a categorical attribute for the provider type. The numerical attributes in the provider-drug-level Part D data, including total volume and frequency of provider's claims, the number of patients, and the total cost of the claims, are particularly useful for supervised machine learning. Additional similar features are also included for patients aged 65 and over. The collection of provider-drug-level Part D data files comprises approximately 174 million records.

The provider-level Part D data includes an additional 51 attributes related to the claims that a provider submits to Medicare throughout the year for all prescribed medications. This data includes ten summary statistical features concerning the beneficiaries of the provider's claims, as well as an average beneficiary risk score. This risk score is derived from a model that adjusts risk according to Hierarchical Condition Categories (HCC) [30]. Following the methodology prescribed by the CMS, beneficiaries with risk scores exceeding the average HCC score of 1.08 are anticipated to incur Medicare expenditures above the average.

Additionally, the provider-level Part D data provides features for the total number of claims, the total number of 30-day prescription orders, the overall drug cost, the total day's supply dispensed, and the total number of beneficiaries seen. These figures are provided as subtotals within various claim categories such as Low-Income Subsidy claims, Medicare Advantage Prescription Drug Plan coverage claims, and Medicare Prescription Drug Plan claims. Furthermore, these statistics are categorized by several drug types, including claims for opiate drugs, long-acting opiate drugs, antibiotic drugs, and anti-psychotic drugs.

Our Medicare Part D dataset is sourced from prescription drug data spanning from 2013 to 2019. This dataset was annotated using LEIE data [9], which includes information about doctors who have committed fraudulent activities. The LEIE also offers additional information, such as the reason and date of exclusion. The establishment and oversight of the LEIE was carried out by the Office of Inspector General (OIG) [31], whose duty is to bar individuals and organizations from participating in healthcare programs funded by the federal government. However, it should be noted that only a portion of provider fraud in the U.S. is represented by NPI values within the LEIE.

We integrated the Medicare Part D dataset with the LEIE, using NPI and year attributes to execute the join operation. Physicians who were practicing within a year before their exclusion end year are tagged as fraudulent. In this dataset, a binary label is used: a value of 1 signifies a fraudulent physician, while a value of 0 denotes a physician without any fraudulent record. The final version of our processed dataset comprises 5,344,106 records and 82 attributes. However, it is important to note that the dataset is significantly skewed, as only 3,700 instances (representing just 0.0692% of the total) are identified as instances of physicians having committed fraud.

Classifiers

In our experimentation, we employed eight machine learning algorithms: CatBoost, XGBoost, Extremely Randomized Trees, Logistic Regression, Random Forest, One-Class GMM, One-Class SVM, and OCAN. These classifiers encompass a diverse range of

machine learning algorithm families, contributing to the robustness and generalizability of the results. The first five algorithms, namely CatBoost, XGBoost, Extremely Randomized Trees, Logistic Regression, and Random Forest, are binary classifiers that assign data to one of two classes or labels. These algorithms are widely utilized across various domains. On the other hand, One-Class GMM, One-Class SVM, and OCAN are OCC algorithms 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 binary classifier ensembles that consist 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 improve prediction accuracy. CatBoost and XGBoost are ensembles that are trained sequentially using boosting [33]. CatBoost utilizes Ordered Boosting [34], an algorithm that organizes the instances used by the Decision Trees. XGBoost employs techniques like weighted quantile sketch and sparsity-aware functions. The former uses approximate tree learning [35] for merging and pruning processes, while the latter leverages low-variance features. Random Forest and Extremely Randomized Trees are ensembles trained independently using bagging [36]. Random Forest systematically determines the best split values for Decision Trees, whereas Extremely Randomized Trees selects these values randomly. Logistic Regression, on the other hand, is a binary classifier that generates a score indicating the likelihood of belonging to a particular class. It is a linear model that utilizes a sigmoid function to produce a result ranging between 0 and 1.

One-Class SVM is a well-known algorithm that constructs a hypersphere in high-dimensional space with the aim of enclosing as many data points as possible from a single class, while excluding those that do not belong [37]. According to the literature, the center of the hypersphere is determined by calculating the mean of the data points associated with the focused class, and the radius is set to enclose a specified percentage of the data points. The algorithm further maximizes the distance between the hypersphere and the nearest data point that does not belong to the focused class, thereby establishing the decision boundary. One-Class SVM is highly regarded for its effectiveness in handling high-dimensional data and its resilience to noisy data. In Section 6, we discuss how to convert the outputs of these classifiers to probabilities.

One-Class GMM is an algorithm designed to identify outliers that significantly deviate from a particular class of data points. It operates as a generative model, representing the data distribution as a weighted sum of multiple Gaussian distributions, with each Gaussian component representing a cluster of similar data points. During the training phase, One-Class GMM learns the parameters of the Gaussian distributions, such as mean and covariance, that best capture the characteristics of the focused class of data points. When classifying a new data instance, the algorithm calculates the likelihood that the instance belongs to the focused class. This likelihood is derived from the probabilities of the instance being generated by each Gaussian component. If the likelihood falls below a predefined threshold, the instance is classified as an anomaly or outlier, indicating that it deviates significantly from the focused class.

OCAN is an anomaly detection algorithm that utilizes adversarial training techniques [38]. Its objective is to learn the underlying probability distribution of a single class

of data points and differentiate between in-class and out-of-class samples. During the training process, OCAN employs a generator network [39] and a discriminator network simultaneously. The generator network generates realistic in-class samples, while the discriminator network is responsible for classifying whether a given sample belongs to the in-class or out-of-class category. The generator network aims to produce in-class samples that are challenging for the discriminator network to distinguish from real in-class samples. Conversely, the discriminator network attempts to differentiate between genuine in-class samples and the generated samples. Through this adversarial interplay, OCAN learns to identify the boundaries of the focused class and effectively discriminate between in-class and out-of-class samples. To classify a new data instance, OCAN computes the distance between the instance and the focused class in the feature space. If the distance falls below a predefined threshold, the instance is classified as in-class, indicating that it belongs to the focused class. Otherwise, if the distance exceeds the threshold, the instance is considered an anomaly. In the context of our experiments, instances from the majority class of the Credit Card Fraud Dataset are considered in-class, while instances from the minority class are regarded as anomalies.

Training and testing

In this research, we conducted experiments using a distributed computing platform. This setup consisted of nodes furnished with 16-core Intel Xeon CPUs, each having 256 GB RAM, and Nvidia V100 GPUs. The algorithms for training and testing were implemented using the Python programming language. We used standalone libraries for CatBoost, XGBoost, and OCAN. In contrast, Random Forest, Extremely Randomized Trees, Logistic Regression, One-Class SVM, and One-Class GMM were implemented using the Scikit-learn library [40].

Both datasets were divided into training and testing sets using an 80:20 ratio. To ensure balanced training for the OCC algorithms in the training set, instances belonging to the minority class were excluded, and only instances from the majority class were utilized. The training phase employed the k-fold cross-validation method, where the model was trained on k-1 folds and tested on the remaining fold in each iteration. This approach maximized the utilization of available data. To maintain a proportional representation of each class across the folds, stratification was applied during the cross-validation process. For our experiments, we chose a value of $k = 5$, allocating 4 folds for training and 1 fold for testing. To minimize potential data loss resulting from random sampling of instances from the majority class, we executed 10 iterations of cross-validation. As a result, this methodology generated 50 performance scores per classifier for each metric.

To prevent overfitting in the Decision-Tree based classifiers, we employed the Maximum Tree Depth parameters as specified in Table 1. These depths were determined based on preliminary experimentation, which is conducted to identify optimal parameters, potential challenges and viable methodologies before starting the main study. For all other parameters, we utilized the default values.

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

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

Metrics

In our work, we utilize a confusion matrix (Table 2) as a fundamental tool. Within this matrix, the class that holds less representation is usually the focal class of interest, while the opposite class, consisting of the larger portion, serves as the majority class. These classes are referred to as positives and negatives, respectively. The following are simple performance metrics [41], along with their definitions:

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

By building upon these fundamental metrics, additional performance measures can be computed in the following manner:

- *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 $FP/(FP + TN)$.

To acquire a comprehensive understanding of the challenges associated with evaluating machine learning models trained on highly imbalanced data, we employed more than one performance metric. The metrics used, AUC and AUPRC, are described in detail below.

AUC, also known as the Area Under the Receiver Operating Characteristic (ROC) Curve, serves as a measure to evaluate the effectiveness of a classifier. It summarizes

the balance between the TPR and FPR. The ROC curve visually illustrates the relationship between these two metrics. By considering all possible classification thresholds along the ROC curve, AUC provides a comprehensive evaluation of the classifier's performance. This condensed single value allows for effective comparisons between different classifiers. The AUC score ranges between 0 and 1, where a higher value signifies better performance of the classifier. A random guessing model achieves an AUC score of 0.5.

The AUPRC (Area under the Precision-Recall Curve) measures the trade-off between Precision and Recall by graphing Precision against Recall for different classification thresholds. AUPRC is a numerical value ranging from 0 to 1, where a higher score indicates superior performance of the classifier.

Given the definition of AUC, it can be deduced that when a dataset contains a significantly large number of true negatives, the number of false positives becomes negligible. However, the AUPRC definition does not consider the number of true negatives. Hence, in the context of an extremely imbalanced dataset such as the Credit Card Fraud Detection Dataset and the Medicare Part D dataset, AUPRC provides a more accurate assessment of the false positive count [42]. Consequently, for this study, AUPRC is considered more important than AUC.

Both AUC and AUPRC evaluation metrics require class probability inputs. However, One-Class SVM does not inherently generate probability estimates for its predictions, necessitating calibration. To accommodate this, we utilized sigmoid calibration and isotonic regression as two distinct methods. Sigmoid calibration modifies the predictive scores produced by One-Class SVM by using a Logistic Regression model. This transformation allows us to translate the output scores from One-Class SVM into class probabilities, focusing particularly on the positive class. This technique is rooted in Platt's work [43]. Sigmoid calibration demonstrates greater flexibility than isotonic regression, as it can capture non-monotonic relationships between the scores and the probabilities. On the other hand, isotonic regression is a non-parametric method that constructs a piecewise-constant function based on the classifier's output scores. This function establishes a monotonic relationship between the scores and probabilities, meaning that a higher score always equates to a higher probability. Isotonic regression is usually computationally efficient and has the benefit of model-agnosticism, meaning that it can be applied to any classifier type. Zadrozny and Elkan [44] first proposed the use of isotonic regression for classifier calibration.

We emphasize that One-Class GMM inherently offers probability estimates for individual data points by representing the data's probability density function through a mixture of Gaussian distributions. These probability estimates can be directly calculated from the model parameters, without requiring additional transformations or functions like the sigmoid function. However, based on unsatisfactory outcomes for AUC and AUPRC observed in preliminary experiments, we opted to utilize Logistic Regression (sigmoid calibration) in conjunction with One-Class GMM. The OCAN classifier also provides output values that serve as probability estimates, eliminating the need for further calibration.

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

Table 5 One-Class Classification Algorithms

Classifier	AUC	AUPRC
One-Class GMM (Sigmoid)	0.7289	0.1481
OCAN	0.5533	0.0015
One-Class SVM (Isotonic)	0.5127	0.0031

Table 6 Binary Classification Algorithms

Classifier	AUC	AUPRC
Catboost	0.9693	0.8124
Extremely Randomized Trees	0.9348	0.6480
Random Forest	0.9627	0.7859
XGBoost	0.9682	0.7803
Logistic Regression	0.8600	0.2901

Results and discussion

In this section, we present and analyze results for the Credit Card Fraud Detection Dataset and the Medicare Part D dataset. Tables 3 and 4 pertain to the former while Tables 5 and 6 pertain to the latter.

Table 3 displays the outcome of using three different OCC algorithms, assessed using AUC and AUPRC metrics. Each score presented in this table represents the average value derived from 50 test-fold results. Out of all the models, One-Class GMM (Sigmoid) registered the most impressive scores, with AUC and AUPRC values reaching 0.9496 and 0.4971 respectively, which implies its potential for effective credit card fraud detection. Conversely, One-Class SVM (Sigmoid) logged the lowest AUC value of 0.9084, while OCAN recorded the lowest AUPRC score of 0.3471.

The scores in Table 4 represent the average of 50 performance scores derived from the test folds. Among these models, Logistic Regression excelled by obtaining the highest AUC score of 0.9794, while CatBoost outperformed others with the highest AUPRC score of 0.8567. Random Forest had the lowest AUC score of 0.9620, and Logistic Regression, despite its high AUC score, registered the lowest AUPRC score of 0.7490.

For Table 5 of the Medicare Part D dataset, we note that results for One-Class SVM (Sigmoid) are not available due to long running times. Hence, we find that this algorithm is not an appropriate classifier for big data. In this table, we also note that scores for One-Class SVM (Isotonic) are the mean value of only 1 iteration of 5-fold cross-validation (5 performance scores) due to long running times. Moreover, the scores for One-Class GMM (Sigmoid) and OCAN denote the average value taken from 10 iterations of 5-fold cross-validation (totaling 50 performance scores). In reference to Table 6, each displayed score corresponds to the average from 10 rounds of 5-fold cross-validation (50 performance scores).

In Table 5, One-Class GMM (Sigmoid) recorded the best AUC and AUPRC scores of 0.7289 and 0.1481 respectively. In Table 6, CatBoost delivered the best AUC and AUPRC results with scores of 0.9693 and 0.8124, respectively.

As mentioned before, our study places greater emphasis on the AUPRC scores due to its capability to generate more insightful results compared to the AUC scores. The binary classification learners, as presented in Table 4, show an average AUPRC score range from 0.8567 to 0.7490. From this perspective, CatBoost emerges as the top-performing algorithm for detecting credit card fraud. The One-Class Classification (OCC) learners displayed in Table 3 manifest a set of average AUPRC scores between 0.4975 and 0.3471, indicating a drop in their classification performance.

In Table 5, the mean AUPRC scores achieved by the OCC algorithms vary, from a high of 0.1481 by One-Class GMM (Sigmoid), to a low of 0.0015 by OCAN. The relatively low AUPRC scores of OCAN and One-Class SVM (Isotonic), when compared to One-Class GMM (Sigmoid), suggest that only One-Class GMM (Sigmoid) is efficient in training on the dominant class. This is another reason why we do not recommend One-Class SVM for big data applications. The mean AUPRC scores obtained by the binary class algorithms in Table 6 span from 0.8124 for CatBoost to 0.2901 for Logistic Regression. As in the case of the Credit Card Fraud Detection Dataset, CatBoost stands out as the top performer for the Medicare Part D Dataset.

When focusing on the AUPRC performance scores of the binary classification algorithms, it is worth noting that the OCC algorithms display substantially inferior performance. This reduced efficiency might be attributed to the potential difficulties faced by OCC algorithms in recognizing 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 results suggest that binary classification models perform better than OCC models when it comes to identifying credit card and Medicare fraud. However, it is essential to note that these conclusions are based on datasets for two different application domains. Further research is required to ascertain if these observations generally hold true across datasets from other domains.

Conclusion

As far as we are aware, this paper is the first to evaluate one-class and binary classification methods for AUC and AUPRC metrics with respect to both Medicare and credit card fraud. We used various classifiers and assessed their performance, including ensembles of Decision Tree, Logistic Regression, One-Class SVM, One-Class GMM, and OCAN. Our results show that binary classification is more effective than one-class classification at identifying instances of fraud in highly imbalanced data. With AUPRC scores of 0.8567 and 0.8124 for the Credit Card Fraud detection dataset and the Medicare Part D dataset, respectively, CatBoost yielded the best performance for binary classification. The significant difference in AUPRC results between the binary and OCC algorithms suggests that OCC learners have a harder time recognizing the minority class. If both class labels can be obtained without difficulty, we recommend binary classification. However, if only one class label is easily available, we recommend OCC. Future work should consider big data from other application domains.

Abbreviations

AUC	Area Under the Receiver Operating Characteristic Curve
AUPRC	Area Under the Precision-Recall Curve
BCC	Binary-Class Classification
CMS	Centers for Medicare and Medicaid Services
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
HCC	Hierarchical Condition Categories
k-NN	k-Nearest Neighbor
LEIE	List of Excluded Individuals and Entities
NPI	National Provider Identifier
OCAN	One-Class Adversarial Nets
OCC	One-Class Classification
OIG	Office of Inspector General
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, JH, and AA 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: 30 June 2023 Accepted: 12 September 2023
Published online: 12 October 2023

References

1. Salekshahrezaee Z, Leevy JL, Khoshgoftaar TM. A reconstruction error-based framework for label noise detection. J Big Data. 2021;8:1–16.
2. Bauder RA, Khoshgoftaar TM, Hasanin T. Data sampling approaches with severely imbalanced big data for medicare fraud detection. In: 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 137–142 2018;. IEEE
3. Hasanin T, Khoshgoftaar TM, Leevy JL, Bauder RA. Investigating class rarity in big data. J Big Data. 2020;7(1):1–17.
4. He H, Garcia EA. Learning from imbalanced data. IEEE Trans Knowl Data Eng. 2009;21(9):1263–84.
5. 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.
6. Kaggle: Credit Card Fraud Detection. https://www.kaggle.com/mlg-ulb/creditcardfraud (2018).
7. Johnson JM, Khoshgoftaar TM. Data-centric ai for healthcare fraud detection. SN Comp Sci. 2023;4(4):389.
8. of Enterprise Data, C.O., Analytics: Medicare Fee-For Service Provider Utilization & Payment Data Part D prescriber public use file: a methodological overview. https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Downloads/Prescriber_Methods.pdf.
9. Herland M, Khoshgoftaar TM, Bauder RA. Big data fraud detection using multiple medicare data sources. J Big Data. 2018;5(1):29.
10. Hancock J, Khoshgoftaar TM. Medicare fraud detection using catboost. In: 2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI), pp. 97–103 2020;. IEEE Computer Society
11. Hancock J, Khoshgoftaar TM, Johnson JM. The effects of random undersampling for big data medicare fraud detection. In: 2022 IEEE International Conference on Service-Oriented System Engineering (SOSE), pp. 141–146 2022;. IEEE.
12. Kumar MS, Soundarya V, Kavitha S, Keerthika E, Aswini E. Credit card fraud detection using random forest algorithm. In: 2019 3rd International Conference on Computing and Communications Technologies (ICCCT), pp. 149–153 2019;. IEEE
13. Hancock J, Khoshgoftaar TM. Performance of catboost and xgboost in medicare fraud detection. In: 19th IEEE International Conference On Machine Learning And Applications (ICMLA) 2020;. IEEE.
14. Alenzi HZ, Aljehane NO. Fraud detection in credit cards using logistic regression. International Journal of Advanced Computer Science and Applications. 2020. **11**(12).
15. Najafabadi MM, Khoshgoftaar TM, Calvert C, Kemp C. A text mining approach for anomaly detection in application layer ddos attacks. In: The Thirtieth International Flairs Conference 2017.
16. Hayashi T, Fujita H. One-class ensemble classifier for data imbalance problems. Appl Intell. 2022;52(15):17073–89.
17. Leevy JL, Hancock J, Khoshgoftaar TM. Comparative analysis of binary and one-class classification techniques for credit card fraud data. J Big Data. 2023;10(1):118.
18. Hancock JT, Khoshgoftaar TM, Johnson JM. Evaluating classifier performance with highly imbalanced big data. J Big Data. 2023;10(1):1–31.
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), pp. 178–184 2018;. IEEE.
21. Chandorkar A. Credit card fraud detection using machine learning. Int Res J Moderniz 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. Periodic Eng Nat Sci. 2018;6(2):394–400.
24. Hancock J, Khoshgoftaar TM, Johnson JM. Informative evaluation metrics for highly imbalanced big data classification. In: 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA) 2022; IEEE.
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), pp. 614–618 2019; IEEE.
26. 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), pp. 25–30 2021;. IEEE.

27. Salekshahrezaee Z, Leevy JL, Khoshgoftaar TM. Feature extraction for class imbalance using a convolutional autoencoder and data sampling. In: 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI), pp. 217–223 2021; IEEE.
28. The Centers for Medicare and Medicaid Services: Medicare Part D Prescribers – by Provider and Drug. https://data.cms.gov/provider-summary-by-type-of-service/medicare-part-d-prescribers/medicare-part-d-prescribers-by-provider-and-drug (2021).
29. The Centers for Medicare and Medicaid Services: Medicare Part D Prescribers - by Provider. https://data.cms.gov/provider-summary-by-type-of-service/medicare-part-d-prescribers/medicare-part-d-prescribers-by-provider (2021).
30. Chamoun GF, Li L, Chamoun NG, Saini V, Sessler DI. Comparison of an updated risk stratification index to hierarchical condition categories. Anesthesiology. 2018;128(1):109–16.
31. OIG: Office of Inspector General Exclusion Authorities US Department of Health and Human Services. https://oig.hhs.gov/.
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. Mat 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 AV, Gulin A. Catboost: unbiased boosting with categorical features. In: Advances in Neural Information Processing Systems, pp. 6638–6648 2018.
35. Gupta A, Nagarajan V, Ravi R. Approximation algorithms for optimal decision trees and adaptive tsp problems. Mathemat Operat 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. Inform 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 Inform Proc 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 Learning Res. 2011;12:2825–30.
41. Seliya N, Khoshgoftaar TM, Van Hulse J. A study on the relationships of classifier performance metrics. In: Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International Conference On, pp. 59–66 2009;. IEEE.
42. Davis J, Goadrich M. The relationship between precision-recall and roc curves. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 233–240 2006.
43. Platt J, et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Adv Large Margin Classif. 1999;10(3):61–74.
44. 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