RESEARCH

Open Access

Feature selection strategies: a comparative analysis of SHAP-value and importance-based methods

Huanjing Wang[1*], Qianxin Liang[2], John T. Hancock[2] and Taghi M. Khoshgoftaar[2]

*Correspondence:
huanjing.wang@wku.edu

[1] Ogden College of Science and Engineering, Western Kentucky University, Bowling Green, USA
[2] College of Engineering and Computer Science, Florida Atlantic University, Boca Raton, USA

Abstract

In the context of high-dimensional credit card fraud data, researchers and practitioners commonly utilize feature selection techniques to enhance the performance of fraud detection models. This study presents a comparison in model performance using the most important features selected by SHAP (SHapley Additive exPlanations) values and the model's built-in feature importance list. Both methods rank features and choose the most significant ones for model assessment. To evaluate the effectiveness of these feature selection techniques, classification models are built using five classifiers: XGBoost, Decision Tree, CatBoost, Extremely Randomized Trees, and Random Forest. The Area under the Precision-Recall Curve (AUPRC) serves as the evaluation metric. All experiments are executed on the Kaggle Credit Card Fraud Detection Dataset. The experimental outcomes and statistical tests indicate that feature selection methods based on importance values outperform those based on SHAP values across classifiers and various feature subset sizes. For models trained on larger datasets, it is recommended to use the model's built-in feature importance list as the primary feature selection method over SHAP. This suggestion is based on the rationale that computing SHAP feature importance is a distinct activity, while models naturally provide built-in feature importance as part of the training process, requiring no additional effort. Consequently, opting for the model's built-in feature importance list can offer a more efficient and practical approach for larger datasets and more intricate models.

Keywords: Feature selection, Class imbalance, Credit card fraud, SHAP, Feature importance

Introduction

Detecting credit card fraud is crucial within the finance industry and heavily relies on the information stored in transaction datasets. However, the finance field and machine learning face a significant research challenge due to the quality of data, as it directly influences decisions made during modeling and analysis [1, 2]. To tackle this issue, we delve into the available feature space, extracting a pertinent set of features. This underscores the importance of feature selection as an essential data cleansing step before engaging in any modeling process. Feature selection has found application in various contexts within data mining and machine learning, with the goal of removing irrelevant or redundant features from the analysis. This not only results in expedited model training but also enhances classifier performance.

This study delves into a comparison between two feature selection methods: Shapley Additive exPlanation (SHAP)-value-based selection [3] and commonly used importance-based selection [4, 5]. SHAP leverages game theory concepts to compute feature importance in two steps: training a classification model using all features in the initial interaction and then computing SHAP values for each feature, subsequently ranking them to identify the most significant features for modeling the target problem. On the other hand, importance-based selection computes feature importance for all features during the model training process. Both methods are embedded since they involve the model-building process. In our feature selection process, we utilize five learners: Extreme Gradient Boosting (XGBoost) [6], Decision Tree (DT) [7], CatBoost [8], Extremely Randomized Trees (ET) [9], and Random Forest (RF) [10]. The selection of these five learners is based on their ability to generate an importance ranking list during the model-building process. LightGBM [11] was not included in our choices due to its poor performance, as indicated by our preliminary results in comparison to other learners. We have designated the SHAP-value-based methods as SHAP-XGBoost, SHAP-DT, SHAP-CatBoost, SHAP-ET, and SHAP-RF, while referring to the importance-based methods simply as XGBoost, DT, CatBoost, ET, and RF. In total, there are 10 feature selection methods, five from each category.

To conduct our study, we focus on the Credit Card Fraud Detection Dataset, a set of anonymized financial transactions available on Kaggle [12]. This dataset is the only publicly available large data for credit card fraud analysis. Hence the scope of the study is limited to one dataset. With 284,807 transactions and 30 independent features, only 492 (0.172%) records are labeled fraudulent. Using two different feature selection methods, we assess the performance of five sets of classifier models using different feature selection techniques (SHAP-XGBoost vs. XGBoost, SHAP-DT vs. DT, SHAP-CatBoost vs. CatBoost, SHAP-ET vs ET, and SHAP-RF vs RF) with their respective selected features. The top 3, 5, 7, 10, and 15 features are selected based on their respective scores. For classification, we build credit card fraud detection models using the five classifiers, the same models used in feature selection. The classifiers are evaluated using the Area Under the Precision Recall Curve (AUPRC) metric [13], and we additionally perform a statistical test with a significance level of $\alpha = 0.01$ to assess the statistical significance of our results.

To the best of our knowledge, this study is the first comprehensive empirical investigation comparing the performance of SHAP-value-based feature selection and importance-based feature selection in the context of fraud detection and potentially other application domains in machine learning.

The remainder of the paper is organized as follows. We begin with an overview of related work, which shows the novelty of the research work we exhibit here. Following that we present the methodology used in the experiment, including explanations of two feature methods, classifiers, cross-validation, and performance metric. We then describe the datasets, experimental design, and experimental results. Finally, we conclude the article with key highlights of this study, and offers suggestions for future work.

Related work

Feature selection is a widely used technique in various data mining and machine learning applications. Its primary objective is to identify a subset of features that minimizes prediction errors for classifiers. In this study, we conducted a comprehensive literature review of research that employs either SHapley Additive exPlanations (SHAP) values or the model's built-in feature importance list for feature selection. While we found a limited number of studies that utilized the model's built-in feature importance list for feature selection in the context of the Credit Card Fraud Detection Dataset, we did not come across any studies that used SHAP for feature selection specifically in credit card fraud detection. Instead, we found a few studies that applied SHAP for feature selection in other application domains. Moreover, we did not encounter any studies that directly compared the performance of models built with features selected by SHAP feature importance versus models built with features selected by built-in feature importance. Therefore, our study presents a unique contribution to the field of credit card fraud detection, as it explores the comparison between SHAP and the model's built-in feature importance list for feature selection, a perspective that has not been extensively explored in the existing literature.

Rtayli and Enneya [14] applied a supervised feature selection method, Random Forest, to identify the most predictive features. Random Forest (RF) is an ensemble learning algorithm that is trained in parallel through bagging [15]. Recently, RF has been increasingly exploited as a feature selection method because it can handle complex, high-dimensional datasets and can detect interactions between features. It also reduces the risk of overfitting, which occurs when a model is too complex and fits the training data too closely. Moreover, RF calculates the feature importance by measuring the decrease in the impurity of the node when the feature is used for the split. The more the impurity decreases, the more important the feature is considered. By ranking the features based on their importance, RF can help select the most relevant features for the classification task. After selecting a feature subset from the Credit Card Fraud Detection Dataset, the authors ran Support Vector Machine to find fraudulent transactions. The model achieved an Accuracy of 95.12%, a Sensitivity of 87%, and an AUC of 0.91, outperforming three other models (Isolation Forest, Decision Tree, and Local Outlier Factor). The study does not provide clear information regarding the number of selected features. Additionally, the authors did not conduct a comparison of the performance between the selected features and the usage of all the available features. Furthermore, it is worth noting that the use of AUC as a metric for classification of imbalanced data has been found to be misleading [16].

In their study using the Credit Card Fraud Detection Dataset [12], Rosley et al. [17] first filtered out the data with a z-score greater than or equal to 3 and then normalized the remaining data using min-max scaling. Then they used Boruta to compute the importance score of each feature. Boruta [18] is a supervised feature selection algorithm that is designed as a wrapper around a Random Forest classifier to identify important features in a dataset. They kept the features with an importance score of 0.5 or higher to train the Autoencoder for each iteration. The model detected credit card fraud by defining a threshold in the reconstruction error to flag the transactions as legitimate or fraudulent. However, the number of features selected in the preprocessing step has not been specified by the authors. The authors evaluated the models using Accuracy, Precision, Recall, and F1 score. When working with datasets that exhibit significant class imbalance, these may not be suitable metrics due to the overwhelming size of the majority class.

Waspada et al. [4] use the RF classifier to calculate the importance score of each feature. Features with a low importance score are discarded. The paper lists the importance score of all features. The authors analyze several factors (dataset split ratio, the selection of top k features, the amount of fraud data on training data, and the setting of hyper-parameter values) that influence the performance of the Isolation Forest (IF) model to detect fraud on credit card transactions. Isolation Forest is a popular unsupervised outlier detection method. Their findings indicate that the best results can be obtained by setting training–testing ratio of 60:40, using the top five features ($V_{14}, V_4, V_{17}, V_{12}, V_{11}$), using only 60% of fraud data, and setting hyper-parameters with the number of trees 100, 128 sample maximum, and 0.001 contamination. The model shows impressive results obtaining precision of 80.7143%, recall of 76.3514%, F1 score of 78.4722%, Area Under the Receiver Operating Characteristic Curve (AUC) of 0.97371, and Area under the Precision-Recall Curve (AUPRC) of 0.759228. Waspada et al. utilized only a single importance-based feature selection method and did not incorporate SHAP for feature selection, which we have implemented in our study.

In their study, Liu et al. [19] utilized SHAP for feature selection on the UCI Parkinson's disease medical dataset [20]. They combined SHAP values with four classifiers: Deep Forest (gcForest), Extreme Gradient Boosting (XGBoost), Light Gradient Boosting machine (LightGBM), and Random Forest (RF). Each classifier was used to calculate the SHAP values of individual features. To assess the effectiveness of SHAP feature selection, they compared it with three filter-based feature selection methods: Fscore, analysis of variance (Anova-F), and Mutual Information. The experiments were conducted with a training and testing ratio of 70:30, and the feature selection was applied to the training dataset. The results showed that the gcForest model based on SHAP value feature selection achieved an impressive classification Accuracy of 91.78% and an F1-score of 0.945, with 150 features selected. This performance surpassed the outcomes of other feature selection methods considered in their study. While the authors specifically employed SHAP-value-based feature selection on the training dataset, we utilized the SHAP method across the entire dataset and subsequently conducted cross-validation following the feature selection procedure.

Marcilio and Eler [21] employed the SHAP method as a feature selection technique and compared it against three widely used feature selection methods: Mutual Information, Recursive Feature Elimination, and ANOVA. The SHAP process involved utilizing XGBoost as the underlying model. They conducted experiments on five UCI datasets using the XGBoost classifier and three other UCI datasets using the XGBoost regressor. The results of their study revealed that SHAP outperformed the three commonly used methods in terms of the Area Under the Receiver Operating Characteristic Curve (AUC) metric. However, it was observed that SHAP required more computational time compared to the other feature selection methods. It is worth noting that the datasets used in Marcilio and Eler's experiments are not highly imbalanced, and not in the credit card fraud domain. In addition, the datasets are significantly smaller in size compared to the Kaggle Credit Card Fraud Detection Dataset, which caught our attention.

In our review of the literature, we discovered that only a single method of feature selection, either based on SHAP values or importance, was employed. Notably, no research has been identified that compares these two methods, particularly within the domain of credit card fraud detection. In order to fill this gap, our study undertook a comparative analysis of these two feature selection methods, employing five learners in each approach.

Methodology

Importance-based feature selection methods

Importance-based feature selection methods leverage decision trees to identify relevant features from a given dataset. These decision tree-based classifiers, such as Extreme Gradient Boosting (XGBoost) [6, 22], Extremely Randomized Trees (ET) [9], Random Forest (RF) [23], CatBoost [8], and Decision Tree [7], possess a built-in capability to determine feature importance during model fitting in supervised machine learning. Consequently, they can rank features based on their significance in classification tasks, making them valuable for feature selection. By discarding less relevant features and retaining the most important ones, more efficient and accurate models can be created.

In this study, five importance-based feature selection methods were employed: XGBoost [22], Decision Tree (DT) [7], CatBoost [8], Extremely Randomized Trees (ET) [9], and Random Forest (RF) [10].

XGBoost and CatBoost stand out as widely used gradient boosting algorithms, each employing distinct approaches to compute feature importance scores. While both algorithms construct ensembles of decision trees, their methodologies for deriving feature importance scores vary. In XGBoost, these scores are calculated using the “gain” method, evaluating the influence of each feature on model performance throughout the boosting process. In contrast, CatBoost’s ensemble of decision trees calculates feature importance based on the frequency of a feature being utilized for splitting and the subsequent improvement in model performance achieved through those splits.

A Decision Tree classifier is a type of machine learning algorithm used for classification tasks. It constructs a tree-like model of decisions and their potential outcomes by recursively splitting the data based on the most informative features at each node. Decision trees generate feature importance scores by evaluating their ability to reduce Gini impurity (or increase purity) within the data as the tree is built.

Extremely Randomized Trees and Random Forest, both rooted in decision tree ensembles, share common principles like Gini impurity and the Mean Decrease in Impurity to gauge feature importance. However, Extremely Randomized Trees introduce heightened randomness in the decision-making process during tree construction. This added stochasticity can result in divergent importance scores, potentially impacting the balance between model bias and variance.

SHAP-value-based feature selection methods

Shapley Additive exPlanation (SHAP), introduced by Lundberg and Lee [3], has gained popularity as a method for interpreting machine learning model predictions. By utilizing Game Theory techniques [24], SHAP provides insights into the contribution of each feature to specific predictions. It falls under a family of additive feature attribution techniques that remain model-agnostic, making them universally applicable to various machine learning and deep learning models. These techniques attribute significance to individual input features, facilitating better understanding of model behavior.

In the context of feature selection, SHAP-based methods work as follows: classification models, such as XGBoost and Decision Tree in this study, are trained on the entire dataset. Subsequently, SHAP values are computed for each instance, and these values are then aggregated across the dataset to derive average absolute values for each feature. The computation of SHAP values becomes computationally complex due to this process. The average SHAP value indicates the typical impact of each feature on model predictions across the entire dataset, while the absolute SHAP value represents the feature's importance, irrespective of its direction (positive or negative). By sorting features based on their average absolute SHAP values in descending order, features with higher SHAP values are identified as more influential in influencing the model's predictions.

Classification

In this study, credit card fraud detection models were built with five different classifiers, namely XGBoost [6], Decision Tree (DT) [7], CatBoost [8], Extremely Randomized Trees (ET) [9], and Random Forest (RF) [10]. Among these five learners, XGBoost, CatBoost, ET, and RF are ensemble of Decision Tree-based classifiers [25]. We select these learners on the basis that they are highly effective for dealing with complex, high-dimensional data and are known for their excellent performance in a wide range of classification tasks [25].

XGBoost and CatBoost are all gradient boosting frameworks that are widely used for machine learning tasks, particularly for classification. These two algorithms are known to be highly effective and produce accurate predictions. However, the performance may vary depending on the specific dataset and problem at hand. XGBoost is an advanced refinement the Gradient Boosted Decision Tree (GBDT) ensemble method. GBDTs were initially introduced by Friedman in 2001 [26]. XGBoost enhances GBDTs in multiple ways. Firstly, it employs an improved loss function during training that includes an additional term for regularization, effectively preventing overfitting. Secondly, XGBoost introduces an "approximate algorithm" for calculating splits in the constituent decision trees, which is highly suitable for distributed environments and cases where the entire dataset cannot fit into main memory. Moreover, XGBoost incorporates a specialized algorithm for handling sparse data, where most values are nearly constant with occasional aberrations. The "sparsity aware split finding" feature enables XGBoost to capitalize on sparse data efficiently. CatBoost, on the other hand, is known for its robustness in handling categorical features and missing values, making it suitable for datasets with such characteristics. CatBoost's core algorithm is Ordered Boosting, which involves sorting the instances used by Decision

Trees. In contrast, XGBoost relies on a weighted quantile sketch and a function that takes into account sparsity. A weighted quantile sketch is an approximate tree learning [27] technique that is utilized for merging and pruning operations, while sparsity deals with values that are either zero or missing.

Breiman introduced the concept of Bagging in the domain of machine learning in a 1996 paper [28]. As our research revolves around binary classification, our focus is on Breiman's ideas about Bagging applied to binary classification. Extremely Randomized Trees (ET) and Random Forest (RF) are both ensemble learning algorithms that belong to the bagging family of decision tree-based methods. Random Forest, which was introduced by Breiman [10]. Random Forest builds upon the Bagging principle with an added improvement. In a Random Forest, each tree is constructed using a random subset of features and samples. This randomness helps to decorrelate the trees and reduce overfitting. Extremely Randomized Trees extends the concept of Random Forest by selecting values for Decision Tree splits at random, potentially making them more robust and computationally efficient in some scenarios. The choice between the two often depends on the specific characteristics of the data and the desired trade-off between bias and variance. We skip the detailed information about these learners and readers are referred to [25].

Decision Tree (DT) is a widely used supervised machine learning algorithm, prominently applied to classification and regression tasks. It is a non-linear model that recursively partitions input data into subsets based on feature values. Each node in the decision tree represents a decision based on a specific feature and threshold, facilitating predictions based on the input data's feature values. The resulting decision tree structure is highly interpretable, with each internal node representing a feature-based decision, edges signifying outcomes, and leaf nodes providing predictions.

To ensure the reproducibility of our results, we modified specific hyperparameter settings from their default values as listed in Table 1. Furthermore, we set random number generator seeds for all classifiers to ensure consistent and repeatable outcomes. All other settings were left at their default values. The determination of tree depths was guided by previous experimentation documented in [1], aiming to achieve a suitable trade-off between capturing complex patterns in the data and mitigating overfitting.

Table 1 Hyperparameter settings used in experiments

Classifier	Parameter name	Parameter setting
CatBoost	`task_type`	`GPU`*
	`max_ctr_complexity`	1
	`max_depth`	5
ET	`max_depth`	8
XGBoost	`max_depth`	3
	`tree_method`	`gpu_hist`*
Random Forest	`max_depth`	4

* Setting selects Graphics Processing Unit (GPU) implementation of the classifier

Performance metric

To assess the effectiveness of feature selection techniques, we constructed classification models subsequent to the feature selection process. The evaluation of these models in this study was based on the Area under the Precision-Recall Curve (AUPRC) metric.

In a two-class classification problem, such as distinguishing fraud (positive) and normal (negative) instances, we encounter four potential prediction outcomes: true positive (correctly classified positive instances), false positive (negative instance mistakenly classified as positive), true negative (correctly classified negative instances), and false negative (positive instance mistakenly classified as negative).

AUPRC represents the area under the Precision-Recall curve, which illustrates the trade-off between Recall (True Positive Rate) and Precision for specific classification thresholds. The definition of precision is

$$\frac{true\ positives}{true\ positives + false\ positives} \tag{1}$$

and the Recall or True Positive Rate is defined as

$$\frac{true\ positives}{true\ positives + false\ negatives} \tag{2}$$

To calculate AUPRC, we plot precision against recall for many classification thresholds and then determine the area under the curve. A higher AUPRC value indicates superior model performance. AUPRC ranges from a minimum of zero to a maximum of one.

Cross-validation

Cross-validation refers to a technique used to allow for the training and testing of machine learning models without resorting to using the same data [29]. The process involves dividing the dataset into a predetermined number of subsets or folds in a relatively balanced manner. In this study, we utilized five-fold cross-validation, where each fold served as the test data, while the remaining four folds were designated as the training data. To minimize any potential bias arising from a fortuitous or unfavorable split, we conducted ten independent runs of the five-fold cross-validation.

It is important to note, for reproducibility, that the feature selection process was conducted separately from the cross-validation step. In other words, the feature selection procedures were performed on the original dataset.

Experiments

Dataset

The experiments conducted in this study utilized the Credit Card Fraud Detection Dataset, which is available for download from the Kaggle website [12]. This dataset consists of anonymized financial transactions, specifically credit card transactions conducted by European cardholders over a two-day period in September 2013. As stated previously, out of a total of 284,807 transactions, 492 of them are fraudulent transactions, resulting in an imbalanced dataset with only 0.172% of transactions being fraudulent, while the rest are considered normal or non-fraudulent transactions.

The Credit Card Fraud Detection Dataset has 30 numerical input features, out of which $V_1, V_2, ..., V_{28}$ have undergone numerical transformation using Principal Component Analysis (PCA) for data analysis and feature reduction purposes. However, the "Time" and "Amount" features were not transformed. The "Time" feature denotes the time in seconds since the first transaction, while the "Amount" feature represents the amount of the credit card transaction. The "Time" feature was excluded from the analysis to avoid influencing the reliability of the results since it is a unique feature that a model can memorize. As a result, there are 29 input features available for further experimentation. Prior to being input to the classifiers for training or classification, the features were normalized to fit within the [0, 1] range. The class feature is utilized to distinguish between legitimate and fraudulent transactions. In this context, a value of 1 represents a fraudulent transaction, while a value of 0 signifies a normal transaction.

Experimental design

In our experiments, we investigated two different feature selection techniques, SHAP-value-based feature selection and importance-based feature selection methods. To assess the efficacy of a feature selection method, we constructed classification models utilizing the subset of features chosen by the feature selection approach. Classification models were built with five classifiers, XGBoost, Decision Tree (DT), CatBoost, Extremely Randomized Trees (ET), and Random Forest (RF).

We conducted our experiments on a distributed computing platform consisting of nodes equipped with 16-core Intel Xeon CPUs, 256 GB RAM per CPU, and Nvidia V100 GPUs. All training and testing programs were implemented using the Python programming language. SHAP is publicly available as an open source library for the Python programming language [30]. In addition to the SHAP values for feature importance, this library also supplies several tools for visualizing SHAP feature importance values. The Python data science stack [31] was employed for experiment implementations.

First, we ranked the features using ten feature selection methods (SHAP-XGBoost, XGBoost, SHAP-DT, DT, SHAP-CatBoost, CatBoost, SHAP-ET, ET, SHAP-RF, and RF) separately. Following feature ranking, we chose the top 3, 5, 7, 10, and 15 features, including the class attribute, to construct the final training datasets. Subsequently, we applied classifiers to these training datasets, ensuring that the classifier used in the model-building process remained consistent with the one employed in feature selection. We used AUPRC to evaluate the performance of the classification models. For each feature selection method and classifier, we have a total of 5 (feature subset sizes) $\times$ 10 (runs) $\times$ 5 (folds) $=$ 250 AUPRC scores.

Results and discussion

As mentioned earlier, we have introduced ten feature selection methods, two feature selection techniques combined with five classifiers. We present the feature importance lists obtained from each method, where we focus on the top 15 most important features. The importance is determined either by SHAP values (for SHAP-XGBoost, SHAP-DT,

Table 2 Features selected by SHAP-XGBoost and XGBoost; the features are listed in order of their importance values from top to bottom

Ranking	SHAP-XGBoost	XGBoost
1	V_{14}	V_{17}
2	V_4	V_{14}
3	V_{12}	V_{10}
4	Amount	V_{27}
5	V_8	V_{12}
6	V_{11}	V_{26}
7	V_7	V_4
8	V_{10}	V_1
9	V_5	V_8
10	V_{19}	V_7
11	V_{26}	V_{16}
12	V_{27}	V_9
13	V_3	Amount
14	V_{16}	V_{13}
15	V_{18}	V_3

Table 3 Features selected by SHAP-DT and DT; the features are listed in order of their importance values from top to bottom

Ranking	SHAP-DT	DT
1	V_{14}	V_{17}
2	V_{17}	V_{14}
3	V_{12}	V_{27}
4	V_4	V_{12}
5	V_1	V_{10}
6	V_{20}	V_{26}
7	V_{19}	V_{24}
8	V_8	V_{16}
9	V_{10}	V_7
10	V_7	V_{20}
11	V_{21}	V_4
12	V_{26}	V_1
13	V_{27}	V_{23}
14	Amount	V_{19}
15	V_{22}	V_{15}

SHAP-CatBoost, SHAP-ET, and SHAP-RF) or built-in importance scores (for XGBoost, DT, CatBoost, ET, and RF). In Tables 2, 3, 4, 5, 6, we display the feature rankings, where rank 1 corresponds to the highest SHAP value or importance score. It's important to note that SHAP values may vary when different trained models are utilized. Notably, among all ten feature selection methods, feature V_{14} stood out as one of the top three features. Additionally, feature V_4 consistently appeared and held a ranking within the top 15 across all feature selection methods.

The classification performance results in terms of AUPRC are shown in Tables 7, 8, 9, 10, 11. The reported values represent averages across ten rounds of five-fold cross-validation outcomes. The results were obtained by creating new datasets using the 3,

Table 4 Features selected by SHAP-CatBoost and CatBoost; the features are listed in order of their importance values from top to bottom

Ranking	SHAP-CatBoost	CatBoost
1	V_1	V_1
2	V_{14}	V_4
3	V_4	V_{14}
4	V_8	Amount
5	V_{26}	V_{11}
6	V_6	V_{26}
7	Amount	V_{13}
8	V_{24}	V_8
9	V_{12}	V_{17}
10	V_{13}	V_3
11	V_{11}	V_{20}
12	V_{18}	V_{18}
13	V_{10}	V_{24}
14	V_{17}	V_{15}
15	V_{19}	V_{28}

Table 5 Features selected by SHAP-ET and ET; the features are listed in order of their importance values from top to bottom

Ranking	SHAP-ET	ET
1	V_{14}	V_{14}
2	V_{17}	V_{17}
3	V_{12}	V_{12}
4	V_4	V_{10}
5	V_{10}	V_{11}
6	V_{11}	V_{16}
7	V_{16}	V_{18}
8	V_3	V_4
9	V_9	V_9
10	V_{18}	V_3
11	V_7	V_7
12	V_{19}	V_2
13	V_1	V_{21}
14	V_2	V_{19}
15	V_{15}	V_{26}

5, 7, 10, and 15 highest-ranked features along with the class attribute to form the final training data. We conducted statistical z-tests [32] on pairs of models (same classifier but different feature selection methods), where each pair consists of one model built with n of the most important features selected by SHAP or the model's built-in feature importance list. The value of n ranges from 3 to 15. The null hypothesis is that there is no significant difference between the mean AUPRC scores of the two models. In Tables 7, 8, the Winner column indicates whether the SHAP or built-in feature selection method has a higher mean AUPRC value based on the outcome of a z-test with a significance level of $\alpha = 0.01$. If the difference in means is not significant, we report a tie.

Table 6 Features selected by SHAP-RF and RF; the features are listed in order of their importance values from top to bottom

Ranking	SHAP-RF	RF
1	V_{14}	V_{17}
2	V_{17}	V_{12}
3	V_{12}	V_{14}
4	V_{10}	V_{10}
5	V_4	V_{16}
6	V_1	V_{11}
7	V_{11}	V_9
8	V_{16}	V_4
9	V_2	V_7
10	V_7	V_{18}
11	V_{19}	V_{26}
12	V_3	V_{21}
13	V_5	V_1
14	Amount	V_8
15	V_{18}	V_3

Table 7 Comparison of SHAP and XGBoost feature selection methods in terms of their AUPRC scores

Size	SHAP-XGBoost	XGBoost	*p*-value	Winner
3	0.7247	0.7727	0.0000	XGBoost
5	0.8165	0.7978	0.0121	Tie
7	0.8302	0.8255	0.5005	Tie
10	0.8446	0.8350	0.0041	SHAP-XGBoost
15	0.8535	0.8557	0.7097	Tie

Table 8 Comparison of SHAP and DT feature selection methods in terms of their AUPRC scores

Size	SHAP-DT	DT	*p*-value	Winner
3	0.7421	0.7323	0.4968	Tie
5	0.7493	0.7414	0.6293	Tie
7	0.7594	0.7666	0.7013	Tie
10	0.7380	0.7686	0.2429	Tie
15	0.7664	0.7564	0.5058	Tie

Table 9 Comparison of SHAP and CatBoost feature selection methods in terms of their AUPRC scores

Size	SHAP-CatBoost	CatBoost	*p*-value	Winner
3	0.6106	0.7235	0.0000	CatBoost
5	0.7266	0.7745	0.0000	CatBoost
7	0.7897	0.8279	0.0000	CatBoost
10	0.8333	0.8472	0.0000	CatBoost
15	0.8506	0.8491	0.7502	Tie

Table 10 Comparison of SHAP and ET feature selection methods in terms of their AUPRC scores

Size	SHAP-ET	ET	*p*-value	Winner
3	0.7796	0.7843	0.6756	Tie
5	0.8172	0.8118	0.4243	Tie
7	0.8143	0.8137	0.9179	Tie
10	0.8175	0.8168	0.9152	Tie
15	0.8086	0.8048	0.7238	Tie

Table 11 Comparison of SHAP and RF feature selection methods in terms of their AUPRC scores

Size	SHAP-RF	RF	*p*-value	Winner
3	0.8097	0.8137	0.5673	Tie
5	0.8396	0.8248	0.0133	Tie
7	0.8416	0.8382	0.6126	Tie
10	0.8447	0.8479	0.6399	Tie
15	0.8544	0.8512	0.6693	Tie

Table 12 ANOVA for Size, Classifier and Technique as factors of performance in terms of AUPRC

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Size	4	1.90	0.48	237.44	less than 10^{-4}
Classifier	4	2.11	0.53	262.91	less than 10^{-4}
Technique	1	0.05	0.05	24.77	less than 10^{-4}
Residuals	2490	4.99	0.00		

Table 7 shows a tie for XGBoost models built on feature subset sizes of 5, 7, and 15. However, for feature subset size 3, the *p*-value is less than the significance level of 0.01, indicating a significant difference in the AUPRC scores. Therefore, XGBoost outperforms SHAP-XGBoost for feature count 3. On the other hand, for feature subset size 10, SHAP-XGBoost outperforms XGBoost.

Table 8 indicates that there is no significant difference in the AUPRC scores between SHAP-DT and DT for any of the feature counts tested (3, 5, 7, 10, and 15). As a result, we cannot declare a winner between the two feature selection methods based on the AUPRC scores. Tables 10 and 11 are similar to Table 8. The results suggest that, for the given dataset and evaluation metric, there is no consistent superior performance between the SHAP feature selection methods and the traditional importance-value based decision tree, extra tree, or random forest methods across different feature sizes.

Table 9 presents a comparison between SHAP-CatBoost and CatBoost feature selection methods in terms of their AUPRC scores for different feature sizes. In summary, for feature sizes 3–10, CatBoost consistently outperforms SHAP-CatBoost in terms of AUPRC, and the differences are statistically significant with p-values of 0.0000. However, for size 15, there is no statistically significant difference between the two methods, resulting in a tie.

In general, the performance of the two feature selection methods is comparable across various scenarios. However, there are specific instances, such as with certain XGBoost and CatBoost models, where distinctions arise. Notably, XGBoost demonstrates superior performance over SHAP-XGBoost when the feature subset size is 3, while CatBoost outperforms SHAP-CatBoost for feature sizes 3, 5, 7, and 10. Moreover, SHAP-XGBoost surpasses XGBoost when the feature subset size is 10.

An analysis of variance (ANOVA) [33] was performed on AUPRC performance metrics, and the results are reported in Table 12. Three factors, Size, Classifier, and Technique, were considered in the analysis. The Size Factor included feature subset sizes of 3, 5, 7, 10, and 15, the Classifier Factor included five classifiers, while the Technique factor included two feature selection methods, SHAP-value based (Represented with SHAP) and Importance-value based (represented with Importance). The statistical test used a significance level of $\alpha = 1\%$. The ANOVA results indicate that there were significant differences among the groups in each of the main factors in terms of the AUPRC metric, as all Pr(>F) or p-values in the last column of the table were less than the cutoff of 0.01.

Since the ANOVA test results revealed that all factors had a significant impact on AUPRC scores, we conducted Tukey's Honestly Significant Difference (HSD) tests [34] to rank the Technique and Classifier based on their impact on AUPRC scores. The performance was ranked alphabetically, with group 'a' having the highest AUPRC scores. Items in the same performance group indicate no statistically significant difference between them. The HSD test results are presented in Tables 13, 14, 15.

Based on the HSD tests, it is evident that feature selection with a subset size of 15 and 10 yields superior performance in AUPRC compared to smaller subset sizes. This suggests that constructing models with a feature subset size of 15 or 10 is advantageous. The reduced size leads to faster model training times and improved outcomes. Among the five classifiers, RF demonstrated the highest AUPRC, followed by XGBoost and ET,

Table 13 HSD test groupings after ANOVA of AUPRC for the Size factor

Group a consists of: 15
Group ab consists of: 10
Group b consists of: 7
Group c consists of: 5
Group d consists of: 3

Table 14 HSD test groupings after ANOVA of AUPRC for the Classifier factor

Group a consists of: RF
Group b consists of: XGBoost, ET
Group c consists of: CatBoost
Group d consists of: DT

Table 15 HSD test groupings after ANOVA of AUPRC for the Technique factor

Group a consists of: Importance
Group b consists of: SHAP

while DT showed relatively poorer performance. Table 15 indicates that the importance-value-based feature selection method significantly outperforms the SHAP-value-based feature selection method, across all feature subsets sizes, and learners.

As mentioned earlier, SHAP is an external tool, and the computational time for SHAP feature selection depends on several factors, including the model's complexity, the number of features, the dataset size, and the number of instances for which SHAP values need to be computed. The complexity of computing SHAP values is generally higher than other feature importance methods like decision-tree-based classifiers. Therefore, we conclude that using the built-in feature importance to select feature subsets may be more suitable for models with a large number of features and a large dataset.

Conclusion

The challenge of dealing with high dimensionality in machine learning significantly affects the evaluation of model performance. This study specifically concentrates on the comparison of two feature selection techniques: identifying the most crucial features through SHAP values and relying on the model's intrinsic feature importance list. Using the Credit Card Fraud Detection Dataset, we generate multiple training datasets. We employ five classifiers with distinct feature subset sizes, applying both feature selection methods to each classifier. Our results indicate that, on the whole, feature selection methods based on importance values outperform those based on SHAP values across the classifiers used in this study and various feature subset sizes.

However, notable variations arise in XGBoost models. XGBoost surpasses SHAP-XGBoost for a feature subset size of 3, while SHAP-XGBoost outperforms XGBoost for a feature subset size of 10. In the case of CatBoost, CatBoost outperforms SHAP-CatBoost for feature sizes less than 15. It is important to note that calculating SHAP feature importance introduces an additional step in the experimental methodology. According to our findings, the return on investment for implementing SHAP may be relatively low, particularly when built-in feature selection methods are available, especially for large datasets. Additionally, the considerable computational expenses associated with SHAP may render it impractical for handling Big Data. For future research, our plan is to explore these two feature selection methods across diverse application domains.

Acknowledgements
The authors would like to thank the various members of the Data Mining and Machine Learning Laboratory, Florida Atlantic University, for their assistance with the reviews.

Author contributions
HJW contributed to the manuscript. QXL and JTH conducted experiments, and JTH contributed to the manuscript. TMK provided oversight of experiments, coordinated research, and contributed to the 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: 18 December 2023 Accepted: 14 March 2024
Published online: 26 March 2024

References

1. Hancock JT, Khoshgoftaar TM, Johnson JM. A comparative approach to threshold optimization for classifying imbalanced data. In: The International Conference on Collaboration and Internet Computing (CIC), Atlanat, GA, USA, 2022. pp. 135–142. IEEE.
2. Wang H, Liang Q, Hancock JT, Khoshgoftaar TM. Enhancing credit card fraud detection through a novel ensemble feature selection technique. In: 2023 IEEE International Conference on Information Reuse and Integration (IRI), Bellevue, WA, USA, 2023. pp. 121–126.
3. Lundberg S.M, Lee S.-I. A unified approach to interpreting model predictions. Adv Neural Inf Process Syst. 2017;30.
4. Waspada I, Bahtiar N, Wirawan PW, Awa BDA. Performance analysis of isolation forest algorithm in fraud detection of credit card transactions. Khazanah Informatika Jurnal. 2022.
5. Wang H, Hancock JT, Khoshgoftaar TM. Improving medicare fraud detection through big data size reduction techniques. In: 2023 IEEE International Conference on Service-Oriented System Engineering (SOSE), Athens, Greece; 2023. pp. 208–217.
6. Chen T, Guestrin C. Xgboost: a scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD '16. 2016.
7. Breiman L. Classification and regression trees. 2017.
8. Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A. Catboost: unbiased boosting with categorical features. Adv Neural Inf Process Syst. 2018;31.
9. Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. Mach Learn. 2006;63(1):3–42.
10. Breiman L. Random forests. Mach Learn. 2001;45(1):5–32.
11. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T-Y. Lightgbm: a highly efficient gradient boosting decision tree. Adv Neural Inf Process Syst. 2017;30:3146–54.
12. Kaggle: Credit card fraud detection. https://www.kaggle.com/mlg-ulb/creditcardfraud. 2018.
13. Leevy JL, Khoshgoftaar TM, Hancock JT. Evaluating performance metrics for credit card fraud classification. In: 2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI), 2022. pp. 1336–1341.
14. Rtayli N, Enneya N. Selection features and support vector machine for credit card risk identification. Procedia Manuf. 2020;46:941–8.
15. González S, García S, Ser JD, 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.
16. Hancock JT, Khoshgoftaar TM, Johnson JM. Evaluating classifier performance with highly imbalanced big data. J Big Data. 2023;10(42).
17. Rosley N, Tong G-K, Ng K-H, Kalid SN, Khor K-C. Autoencoders with reconstruction error and dimensionality reduction for credit card fraud detection. J Syst Manag Sci. 2022;12(6):70–80.
18. Kursa MB, Rudnicki WR. Feature selection with the Boruta package. J Stat Softw. 2010;36(11):1–13.
19. Liu Y, Liu Z, Luo X, Zhao H. Diagnosis of Parkinson's disease based on SHAP value feature selection. Biocybern Biomed Eng. 2022;42(3):856–69.
20. Sakar CO, Serbes G, Gunduz A, Tunc H, Nizam H, Sakar B, Tütüncu M, Aydin T, Isenkul M, Apaydin H. A comparative analysis of speech signal processing algorithms for Parkinson's disease classification and the use of the tunable q-factor wavelet transform. Appl Soft Comput. 2019;74:255–63.
21. Marcilio WE, Eler DM. From explanations to feature selection: assessing SHAP values as feature selection mechanism. In: 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Los Alamitos, CA, USA, 2020. pp. 340–347.
22. Hancock JT, Khoshgoftaar TM. Gradient boosted decision tree algorithms for Medicare fraud detection. SN Comput Sci. 2021;2(4):268.
23. Muaz A, Jayabalan M, Thiruchelvam V. A comparison of data sampling techniques for credit card fraud detection. Int J Adv Comput Sci Appl (IJACSA). 2020;11(6):477–85.
24. Shapley L. A value for n-person games. Contributions to the Theory of Games, 1953. pp. 307–317.
25. 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(6):3571–6.
26. Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat. 2001;1189–1232.
27. Gupta A, Nagarajan V, Ravi R. Approximation algorithms for optimal decision trees and adaptive tsp problems. Math Oper Res. 2017;42(3):876–96.
28. Breiman L. Bagging predictors. Mach Learn. 1996;24(2):123–40.
29. Witten IH, Frank E, Hall MA. Data mining: practical machine learning tools and techniques. 2011.
30. S. Lundberg and others: SHAP. https://github.com/slundberg/shap/tree/v0.41.0, accessed: 2023-07-09.
31. Oliphant T. Python for scientific computing. Comput Sci Eng. 2007;9(3):10–20.
32. Jain R. The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling. 1991.
33. Iversen GR, Norpoth H. Analysis of Variance, vol. 1. Newbury Park: Sage; 1987.
34. Tukey JW. Comparing individual means in the analysis of variance. Biometrics. 1949;99–114.

Publisher's Note