IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 30, 2020, accepted January 10, 2021, date of publication January 18, 2021, date of current version January 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3051969

Feature Selection and Ensemble Learning Techniques in One-Class Classifiers: An Empirical Study of Two-Class Imbalanced Datasets

CHIH-FONG TSAI [1] AND WEI-CHAO LIN [2,3]
[1] Department of Information Management, National Central University, Taoyuan 320, Taiwan
[2] Department of Information Management, Chang Gung University, Taoyuan 333, Taiwan
[3] Department of Thoracic Surgery, Chang Gung Memorial Hospital, Linkou 333, Taiwan

Corresponding author: Wei-Chao Lin (viclin@gap.cgu.edu.tw)

This work was supported in part by the Ministry of Science and Technology of Taiwan under Grant MOST 109-2410-H-182-012, and in part by Chang Gung Memorial Hospital, Linkou, under Grant BMRPH13.

ABSTRACT Class imbalance learning is an important research problem in data mining and machine learning. Most solutions including data levels, algorithm levels, and cost sensitive approaches are derived using multi-class classifiers, depending on the number of classes to be classified. One-class classification (OCC) techniques, in contrast, have been widely used for anomaly or outlier detection where only normal or positive class training data are available. In this study, we treat every two-class imbalanced dataset as an anomaly detection problem, which contains a larger number of data in the majority class, i.e. normal or positive class, and a very small number of data in the minority class. The research objectives of this paper are to understand the performance of OCC classifiers and examine the level of performance improvement when feature selection is considered for pre-processing the training data in the majority class and ensemble learning is employed to combine multiple OCC classifiers. Based on 55 datasets with different ranges of class imbalance ratios and one-class support vector machine, isolation forest, and local outlier factor as the representative OCC classifiers, we found that the OCC classifiers are good at high imbalance ratio datasets, outperforming the C4.5 baseline. In most cases, though, performing feature selection does not improve the performance of the OCC classifiers in most. However, many homogeneous and heterogeneous OCC classifier ensembles do outperform the single OCC classifiers, with some specific combinations of multiple OCC classifiers, both with and without feature selection, performing similar to or better than the baseline combination of SMOTE and C4.5.

INDEX TERMS Data mining, one-class classifiers, class imbalance, machine learning, ensemble learning.

I. INTRODUCTION

Many real-world domain problem datasets are class imbalanced, meaning that the numbers of data in different classes are not the same. For example, for two-class classification problems, the imbalance ratio in cancer diagnosis datasets can range from 1.29 to 26.13 [1], [2] from 1.51 to 43.73 in software defect prediction datasets [3], 4565 in the malware detection dataset [4], and from 1.24 to 25 in a wide variety of behavior datasets [5].

In general, each class imbalanced dataset has at least one of the following three characteristics, small sample size, overlapping (or class separability), and small disjuncts [6], [7]. A small sample size means that there are not enough examples in the minority class, which can cause an imbalanced class distribution. In overlapping, examples in the minority and majority classes may be overlapped, which makes it hard to deduce discriminative rules. As a result, more general rules are deduced that may lead to the misclassification of many minority class instances. In small disjuncts, the concept represented by the minority class is formed of sub-concepts, which are located differently in the feature space and the amount of instances among them is not usually balanced. This also increases the complexity of the problem to be solved.

Due to the above characteristics in class imbalanced datasets, minority class instances are more often misclassified than those from the other classes. To this end, there are three types of solutions, which are the data level, algorithm level,

The associate editor coordinating the review of this manuscript and approving it for publication was Mehul S. Raval.

and cost-sensitive methods [6], [7]. The data level approaches as data pre-processing techniques focus on balancing the amount of data between different classes in order to decrease the skewed class distribution. The algorithm level approaches are based on creating or modifying the algorithms to bias the learning toward the minority class, such as non-iterative neural networks, SGTM (Successive Geometric Transformations Model) neural-like structure, and boosting classifier ensembles. Cost-sensitive methods consider different misclassification costs for each class in the learning phase. In particular, higher costs for the misclassification of the examples in the minority class are considered.

In related studies, novel approaches have been developed by considering one of three types of solutions, using a variety of two-class classification domain problem datasets for performance evaluation [8]–[16]. For example, Koziarski *et al.* propose Radial-Based Oversampling (RBO) method to find regions in which the synthetic objects from the minority class should be generated [8]. In Kumar *et al.*, Tomek-link undersampling-based boosting (TLUSBoost) algorithm is proposed, which uses Tomek-linked and redundancy-based undersampling (TLRUS) for data sampling and AdaBoost technique for boosting [9]. Piri *et al.* propose a new synthetic informative minority over-sampling (SIMO) algorithm, which are based on over-sampling the minority examples close to the support vector machine (SVM) decision boundary [12]. Tsai *et al.* combing clustering and instance selection to under-sample the majority class where noisy or outliers in the majority class are removed [14].

On the other hand, Liu and Zio combine a modified Feature Vector Selection (FVS) method with maximal between-class separability and an easy-tuning of SVM, where a small number of data points are selected to represent linearly all the dataset in the Reproducing Kernel Hilbert Space (RKHS) [10]. In Mahani and Baba-Ali, a three-phase-rule-based extraction process is introduced. Initially, the classification rules representing only majority instances are extracted, and then, the majority instances that are well classified by these rules are deleted to produce a balanced dataset. Finally, the balanced dataset is used to produce related rules to represent both majority and minority instances [11]. Sundar and Punniyamoorthy make two major modications to Wang's Boosted SVM (WBSVM) algorithm to improve the classification performance without increasing its existing time complexity [13]. In Wang *et al.*, a learning framework consisting of fisher kernel and Bi-Bagging is proposed, where the generated fisher vector contains better discriminatory information for Bi-Bagging to generate multi-view data and balanced training subsets [15]. Wang and Yang propose a novel loss function to combine with SVM, which results in a Bayes optimal classifier [16].

The two-class imbalance problem is similar to the anomaly detection (or outlier detection) in one-class classification problems, which focus on identifying rare items, events or observations from the majority of the data [17], [18]. For example, in bank fraud detection and machine fault detection, the anomalies can be regarded as the data in the minority class and the normal items are the data in the majority class. In particular, one-class classification (OCC) techniques are one major type of anomaly detection technique, where only data from the normal class (i.e. positive or majority class) are available for training the classifier and the trained one-class classifier is able to distinguish between the data from the majority and minority classes [19].

Despite OCC techniques having shown their potential for specific outlier detection and class imbalance problems [20]–[24], their applicability to various two-class imbalanced domain datasets with different imbalance ratios has not been fully explored. The OCC techniques are rarely considered in the above related works proposing novel approaches based on the data level, algorithm level, or cost-sensitive types of solutions. Moreover, ways to enhance the performance of one-class classifiers for two-class imbalanced domain datasets have not been examined. The performance of various classifiers can be improved by using feature selection to reduce feature dimensionality [25], [26] and ensemble learning by combining multiple classifiers [27], [28].

Therefore, three research questions which have never been answered before are formulated in this paper. First, can one-class classifiers perform better than the conventional two-class classifiers over various two-class imbalanced datasets? Second, can performing feature selection improve the performance of one-class classifiers over class imbalanced datasets? Third, can one-class classifier ensembles outperform the single best one-class classifier?

Consequently, the contribution of this paper is two-fold. First, the research findings should allow us to understand whether the OCC techniques provide a suitable type of the algorithm level based solution for class imbalanced datasets. In other words, besides C4.5, which is a widely used baseline classifier, the optimal one-class classifier can be identified as another baseline classifier for future researches. Second, performing feature selection to filter out unrepresentative features in the majority class and employing ensemble learning techniques appropriately to combine multiple one-class classifiers can further enhance the performance of single one-class classifiers.

The rest of this paper is organized as follows. Section II overviews the concept of one-class classification and related techniques. Section III describes the experimental procedures used to develop the one-class classifiers related to the three research questions. Section IV presents the experimental results and Section VII concludes the paper.

II. ONE-CLASS CLASSIFICATION

A. THE BASIC IDEA

In one-class classification (OCC), in the classifier training stage, it is assumed that the objects only come from a single class, which is called as the target class (or positive class). In other words, the target class is well characterized by the training data, while the negative class has either no training instances or very few of them. As a result, a decision boundary enclosing all of the training data in the positive class as a data description can be created.

New objects need to be classified for the classification stage, which are either new instances in the target class or unknown instances outside the created decision boundary. In particular, a one-class classifier can assign an anomaly score to each new object, where a threshold is defined for the decision boundary to separate normal data from outliers [19], [29].

OCC is different from the conventional multi-class classification problem. In the traditional two-class or multi-class classification problem, the training data from two or more classes are available to create a decision boundary for each class. Classifiers can thus be developed to classify a new and unknown object into one of several pre-defined classes. However, since most conventional classifiers assume that the training data in each class are more or less equally balanced, they cannot perform well when the classes are extremely imbalanced [6]. In other words, OCC is very suitable for handling class imbalanced datasets with highly imbalance ratios [17], [18].

B. ONE-CLASS CLASSIFIERS

In the related literature, there are many types of outlier detection algorithms that can be used to construct one-class classifiers. Domingues *et al.* [30] divided them into probabilistic methods, distance-based methods, neighbor-based methods, information theory, neural networks, domain-based methods, and isolation methods. Specifically, they compared 14 well-known algorithms based on extent of scalability, memory consumption, and robustness testing. Of these methods, three good performing algorithms are chosen for examination in this paper, including local outlier factor (LOF), one-class support vector machine (OCSVM), and isolation forest (IFOREST) methods. They are overviewed below.

- LOF: The local outlier factor (LOF) is one of the neighbor-based methods, which focuses on searching the neighborhood of each data point to identify outliers. In other words, the degree to which a testing data point is identified as an outlier is computed based on the Euclidean distance between the testing data and its k closest neighbors. This degree is called the local outlier factor of the testing data. In other words, the degree is determined by how isolated the testing data point is in relation to the surrounding neighborhoods [31].
- OCSVM: The one-class support vector machine (OCSVM) is one of the domain-based methods that functions by constructing a decision boundary, so that any data falling outside of the boundary are regarded as outliers. OCSVM is a special application of SVM for one-class problems. It is based on the use of support vector data description (SVDD) to identify the smallest hypersphere (i.e. the minimum radius) consisting of all the data points. Consequently, the SVDD classifier can detect a testing data point as the outlier if it falls outside the hypersphere [32].
- IFOREST: Isolation forest (IFOREST) is one of the isolation methods, where an isolation score is computed for each teasing data point based upon the construction of an ensemble of trees (or random forests). The IFOREST algorithm recursively generates partitions on the data points by randomly selecting an attribute value, and then randomly selecting a split value for the attribute. As a result, multiple trees are built in order to isolate any testing data from the rest of the data. Specifically, the isolation score for the testing data is based on the average path length from the root of the tree to the node containing the single point. The testing data which have short average path lengths from multiple trees are regarded as outliers [33].

III. EXPERIMENTAL STUDIES AND PROCEDURE

In order to answer the three research questions described in Section 1, three corresponding experimental studies are designed.[1] All of the experiments are based on 55 different domain datasets, which are collected from KEEL-dataset repository.[2] These datasets are widely used for performance comparison between different approaches, techniques, and/or algorithms [6]–[8], [10], [11], [13]–[15], [34], [35]. The imbalance ratios of these datasets range from 1 to 130.

A. THE FIRST EXPERIMENTAL STUDY

The aim of the first experimental study is to compare the performance of OCC techniques, including LOF, OCSVM, and IFOREST, with conventional two-class classification techniques. Figure 1 shows the experimental procedure in the first study.

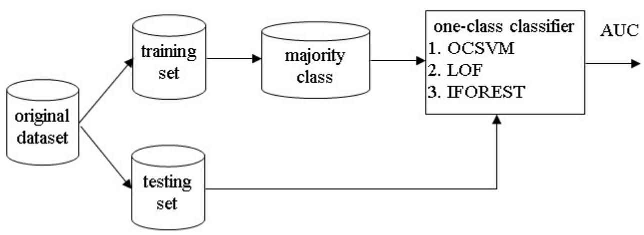

FIGURE 1. The experimental procedure for study one.

First of all, each dataset is divided into 80% training and 20% testing sets by the 5-fold cross validation method. Next, for the 80% training set of each fold, the data in the majority class are selected in order to construct the three one-class classifiers based on LOF, OCSVM, and IFOREST, respectively. Therefore, the numbers of training data to construct one-class classifiers are smaller than the ones to construct general two-class classifiers. Note that the imbalance ratios

[1]The computing environment is based on Intel(R) Core(TM) i7-3770CPU@3.4GHz with the 16GB memory, and the implementation software is based on Python scikit-learn package.

[2]https://sci2s.ugr.es/keel/datasets.php

of the training and testing sets are almost the same. Then, the testing set is used to test the one-class classifiers, where the performance of these classifiers is determined by examining the area under the curve (AUC) based on the receiver operating characteristic (ROC) curve [36]. This evaluation metric is widely used in related works to assess the performances of different classifiers over class imbalanced datasets [6].

According to Domingues *et al.* [30], these chosen algorithms have flexible parameters and perform very well without extensive tuning. More specifically, the parameters of LOF are $k = 10$ and the Euclidean distance function, OCSVM is based on the RBF kernel, $v = 0.5$, and gamma = 1/no. of features, and IFOREST is based on maximum samples = 100, numbers of trees = 100, and contamination = 0.5, respectively.

About the baseline two-class classifiers, C4.5 and SMT are used, where SMT is based on performing SMOTE[3] to pre-process the class imbalanced training set to become balanced, and then C4.5 is trained by the balanced training set. These two techniques have been widely used as the baseline classifiers in various class imbalanced domain datasets [3], [6], [12]–[14], [34], [35]. Different from OCSVM, LOF, and IFOREST, the construction of the two baseline classifiers are based on both the majority and minority classes in the training set are used.

B. THE SECOND EXPERIMENTAL STUDY

The aim of the second experimental study is to examine the effect of performing feature selection on the performance of one-class classifiers. Therefore, feature selection is only executed over the training data in the majority class before classifier training. Since feature selection algorithms can be divided into three types, namely filter, wrapper, and embedded methods, principal component analysis (PCA), genetic algorithms (GA), and C4.5 are chosen for each type of method, respectively.

As the given training set for feature selection only contains one class, it is not possible for the supervised learning based methods, i.e. GA and C4.5, to accomplish the feature selection task. According to the work of [14] that focusing on filtering out noisy data in the majority class by performing instance selection, supervised learning based instance selection algorithms also cannot directly handle the dataset containing only one class. Therefore, we followed the work of [14] to employ the k-means clustering algorithm to group the training data into different clusters, i.e. sub-concepts of the majority class. Then, these sub-concepts can be regarded as 'pseudo' classes, making the original one-class training set become a 'pseudo' multi-class training set. The maximum number of similar data to be grouped into one cluster by k-means clustering can be defined differently. As there is no ground truth answer for each dataset, in this work, 10, 50, and 100 are considered for performance comparison. Thus, given a one-class training set, three different clustering results are produced, which become three different 'pseudo' multi-class training sets. These are then used as the inputs for GA and C4.5 to perform the feature selection task. As a result, three different feature subsets are generated by GA and C4.5, respectively.

On the other hand, PCA and GA require some pre-defined parameters. In PCA, after computing the eigenvalues and eigenvectors of the principal components, each of the original features is associated with a level of variance. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. Thus the definition and computation of principal components are straightforward. Therefore, keeping the top N% feature variances lead to the representative features selected. In Chen *et al.* [37], the top 80%, 65%, and 50% of feature variances are kept and compared. They found that the best performance is obtained when keeping the top 80% of feature variance.

GA for feature selection is based on searching for the best feature subsets that can maximizes the predictive accuracy and minimizes irrelevant features according to a chosen fitness function. The parameters of GA including the number of iterations, population size, crossover rate, and mutation rate, are set to be 5, 10, 0.9, and 0.01 respectively. Note that we tested different population sizes and numbers of iterations, ranging from 10 to 30 and 5 to 100, respectively. The results show no significant differences between the nine different parameter settings in terms of the AUC. That is, the AUC differences between them are less than 0.025. In addition, the fitness function is based on logistic regression since it requires relatively lower computational cost if compared with other machine learning based classification techniques, such as SVM, neural networks, etc. Note that the implementation of GA is based on the DEAP (Distributed Evolutionary Algorithms in Python) framework.

For C4.5, since the nodes of the constructed tree over a given training set represent the attributes for the decision rules, they are used as the features to be selected from the training set.

Finally, the reduced feature subsets of the training data in the majority class produced by PCA, GA, and C4.5 are used to train the one-class classifiers. For classifier testing, the feature subsets of the testing data are reduced to be the same as their corresponding training data for successful testing of the constructed one-class classifiers. Consequently, the performance of the one-class classifiers with and without feature selection can be compared. Note that since performing feature selection over the 55 Keel datasets (which do not contain very high dimensional features) does not significantly affect the performance of the two-class baseline classifiers, i.e. C4.5 and SMT [6], their results based on feature selection are not compared in this paper.

C. THE THIRD EXPERIMENTAL STUDY

The third experimental study focuses on employing the ensemble learning techniques to construct one-class classifier

[3]SMOTE: Synthetic Minority Oversampling Technique

ensembles. The final classification result for the testing data point is based on the outputs produced by multiple classifiers, specifically by the voting method which is a simple and straightforward approach for combining multiple outputs.

Twelve one-class classifiers are chosen from previous studies as the candidates for constructing different types of one-class classifier ensembles. They include the three base classifiers, i.e. LOF, OCSVM, and IFOREST, and PCA, GA, and C4.5 combined with each base classifier. Note that for GA and C4.5, as there are three different settings for the k-means resulting in three different classifiers. For GA and C4.5, respectively, we only consider the best setting that provides the highest AUC rate.

From the twelve constructed one-class classifiers, two types of classifier ensembles are constructed, namely, homogeneous and heterogeneous ensembles. Homogeneous ensembles are based on combing the same base classifiers with PCA, GA, and C4.5 (c.f. Table 1). On the other hand, two to three different base classifiers are combined as the heterogeneous ensembles (c.f. Table 2). In addition, the combination of multiple OCC classifiers is based on the voting method. Note that the homogeneous and heterogeneous ensembles are all based on the bagging approach.

TABLE 1. Homogeneous Ensembles.

Base classifiers	Combinations
LOF	• Combinations of four classifiers: 1. LOF + PCA-LOF + GA-LOF + C4.5-LOF • Combinations of three classifiers: 1. LOF + PCA-LOF + C4.5-LOF, 2. LOF + PCA-LOF + GA-LOF, 3. LOF + GA-LOF + C4.5-LOF, 4. PCA-LOF + GA-LOF + C4.5-LOF • Combinations of two classifiers: 1. LOF + PCA-LOF, 2. LOF + GA-LOF, 3. LOF + C4.5-LOF, 4. PCA-LOF + C4.5-LOF, 5. PCA-LOF + GA-LOF, 6. GA-LOF + C4.5-LOF
OSVM	• Combinations of four classifiers: 1. OSVM + PCA-OSVM + GA-OSVM + C4.5-OSVM • Combinations of three classifiers: 1. OSVM + PCA-OSVM + C4.5- OSVM, 2. OSVM + PCA-OSVM + GA-OSVM, 3. OSVM + GA-OSVM + C4.5-OSVM, 4. PCA- OSVM + GA-OSVM + C4.5-OSVM • Combinations of two classifiers: 1. OSVM + PCA-OSVM, 2. OSVM + GA-OSVM, 3. OSVM + C4.5-OSVM, 4. PCA-OSVM + C4.5-OSVM, 5. PCA-OSVM + GA-OSVM, 6. GA-OSVM + C4.5-OSVM
IFOREST	• Combinations of four classifiers: 1. IFOREST + PCA-IFOREST + GA-IFOREST + C4.5-IFOREST • Combinations of three classifiers: 1. IFOREST + PCA-IFOREST + C4.5-IFOREST, 2. IFOREST + PCA-IFOREST + GA-IFOREST, 3. IFOREST + GA-IFOREST + C4.5-IFOREST, 4. PCA-IFOREST + GA-IFOREST + C4.5-IFOREST • Combinations of two classifiers: 1. IFOREST + PCA-IFOREST, 2. IFOREST + GA-IFOREST, 3. IFOREST + C4.5-IFOREST, 4. PCA-IFOREST + C4.5-IFOREST, 5. PCA-IFOREST + GA-IFOREST, 6. GA-IFOREST + C4.5-IFOREST

TABLE 2. Heterogeneous Ensembles.

Different base classifiers without feature selection	• Combinations of three classifiers: 1. OSVM + LOF + IFOREST • Combinations of two classifiers: 1. OSVM + LOF, 2. OSVM + IFOREST, 3. LOF + IFOREST
Different base classifiers with the same feature selection	• Combinations of three classifiers: 1. PCA-OSVM + PCA-LOF + PCA-IFOREST, 2. GA-OSVM + GA-OSVM + GA-IFORES, 3. C4.5-OSVM + C4.5-LOF + C4.5-IFOREST
Combinations of three best classifiers with and without feature selection	The combinations can be decided after the results from study I are obtained.
Combinations of top two, three, four, five, etc. with and without feature selection (out of 12)	The combinations can be decided after the results from study I are obtained.

It should be noted that since the outputs generated by different classification techniques are different, i.e. the probability scores used to decide whether the testing data belong to the majority and minority classes are different (they are normalized on a scale of 0 to 1). The final output of the classifier ensembles is based on the average of the normalized probability scores as follows:

$$output = \frac{\sum_{i=1}^{N} normalized\ probability\ scores}{N} \tag{1}$$

where N is the number of classifiers combined. If the output is equal to or larger than 0.5, then the testing data point is classified into the majority class; otherwise, it is classified into the minority class.

IV. RESULTS AND DISCUSSION

A. RESULTS FROM STUDY I

The imbalance ratios (IR) for the 55 chosen datasets range from 1.82 to 129.44. In order to understand the performance of the OCC techniques, the datasets are divided into 'low' and 'high' imbalance ratio groups. Specifically, the threshold imbalance ratio for these two groups is 27.34. As a result, there are 26 and 29 datasets for the low and high imbalance ratio groups, respectively, where the imbalance ratio of the former group ranges from 1.82 to 26.63, and the latter group ranges from 27.77 to 129.44. Table 3 shows the AUC rates for different classifiers over the 55 datasets. It should be noted that the average results are only used for comparing different techniques efficiently and making initial conclusions for the general circumstance.

The results are very interesting in that, for the low imbalance ratio datasets, the three OCC classifiers do not perform better than the baselines, not even the C4.5 classifier without data re-sampling. However, for the high imbalance ratio datasets, the three OCC classifiers significantly outperform

TABLE 3. AUC rates of different classifiers.

Datasets				C4.5	SMT	OCSVM	IFOREST	LOF
Name	*Domain*	*No. features/samples*	*IR*					
glass1	Physical sciences	9/214	1.82	**0.740**	0.737	0.528	0.496	0.704
Wisconsin	Life sciences	9/683	1.86	0.945	0.953	0.993	**0.994**	0.668
glass0	Physical sciences	9/214	2.06	**0.817**	0.775	0.427	0.358	0.578
yeast1	Life sciences	8/1484	2.46	0.664	**0.709**	0.521	0.535	0.617
Haberman	Life sciences	6/306	2.78	0.576	0.616	**0.674**	0.673	0.663
glass-0-1-2-3_vs_4-5-6	Physical sciences	9/214	3.20	0.916	0.923	0.922	0.942	**0.970**
new-thyroid1	Life sciences	5/215	5.14	0.914	**0.963**	0.922	0.929	0.937
new-thyroid2	Life sciences	5/215	5.14	0.937	**0.966**	0.931	0.935	0.937
ecoli2	Life sciences	7/336	5.46	0.864	**0.881**	0.488	0.732	0.769
segment0	Computer vision	19/2308	6.02	0.983	**0.993**	0.885	0.503	0.831
glass6	Physical sciences	9/214	6.38	0.813	0.884	0.896	**0.944**	0.880
yeast3	Life sciences	8/1484	8.10	0.860	**0.891**	0.570	0.673	0.813
ecoli3	Life sciences	7/336	8.60	0.728	**0.812**	0.730	0.747	0.607
page-blocks0	Computer sciences	10/5472	8.79	0.922	**0.950**	0.570	0.923	0.912
vowel0	Signal processing	13/988	9.98	0.971	0.951	**0.992**	0.970	0.931
glass-0-1-6_vs_2	Physical sciences	9/192	10.29	0.594	**0.606**	0.259	0.408	0.501
glass2	Physical sciences	9/214	11.59	**0.719**	0.639	0.252	0.352	0.519
shuttle-c0-vs-c4	Physical sciences	9/1829	13.87	**1.000**	**1.000**	0.996	**1.000**	0.994
glass4	Physical sciences	9/214	15.47	0.754	0.887	**0.916**	0.862	0.861
abalone9-18	Life sciences	8/731	16.40	0.598	0.628	0.599	0.758	**0.811**
dermatology-6	Life sciences	34/358	16.90	**0.999**	**0.999**	0.968	0.970	0.954
glass-0-1-6_vs_5	Physical sciences	9/184	19.44	**0.894**	0.813	0.689	0.789	0.891
glass5	Physical sciences	9/214	22.78	**0.898**	0.881	0.580	0.722	0.851
lymphography-normal-fibrosis	Life sciences	18/148	23.67	0.745	0.843	0.943	**0.979**	0.957
flare-F	Physical sciences	11/1066	23.79	0.499	**0.868**	0.797	0.795	0.675
kr-vs-k-zero-one_vs_draw	Game	6/2901	26.63	**0.996**	**0.996**	0.990	0.900	**0.996**
Avg.				0.821	**0.852**	0.732	0.765	0.774
kr-vs-k-one_vs_fifteen	Game	6/2244	27.77	**1.000**	**1.000**	**1.000**	0.998	**1.000**
yeast4	Life sciences	8/1484	28.10	0.614	0.712	**0.770**	0.737	0.681
winequality-red-4	Business	11/1599	29.17	0.515	**0.641**	0.466	0.598	0.521
poker-9_vs_7	Game	10/244	29.50	0.455	0.552	0.943	**0.945**	0.849
kddcup-guess_passwd_vs_satan	Intrusion detection	41/1642	29.98	0.990	0.990	0.995	**1.000**	0.896
abalone-3_vs_11	Life sciences	8/502	32.47	0.999	0.966	0.988	**1.000**	0.990
yeast5	Life sciences	8/1484	32.73	0.883	**0.934**	0.915	0.899	0.637
kr-vs-k-three_vs_eleven	Game	6/2935	35.23	0.967	**1.000**	0.989	0.945	0.977
winequality-red-8_vs_6	Business	11/656	35.44	0.517	**0.780**	0.534	0.700	0.658
abalone-17_vs_7-8-9-10	Life sciences	8/2338	39.31	0.679	**0.862**	0.596	0.726	0.839
abalone-21_vs_8	Life sciences	8/581	40.50	0.891	0.888	0.826	**0.901**	0.884
yeast6	Life sciences	8/1484	41.40	0.712	**0.829**	0.798	0.773	0.543
winequality-white-3_vs_7	Business	11/900	44.00	0.581	0.751	**0.917**	0.863	0.813
winequality-red-8_vs_6-7	Business	11/855	46.50	0.500	**0.721**	0.542	0.620	0.544
kddcup-land_vs_portsweep	Intrusion detection	41/1061	49.52	0.955	**1.000**	0.913	1.000	0.881
abalone-19_vs_10-11-12-13	Life sciences	8/1622	49.69	0.500	**0.674**	0.439	0.498	0.635
kr-vs-k-zero_vs_eight	Game	6/1460	53.07	0.994	**0.998**	0.913	0.853	0.941
winequality-white-3-9_vs_5	Business	11/1482	58.28	0.502	0.625	0.819	**0.832**	0.691
poker-8-9_vs_6	Game	10/1485	58.40	0.500	0.747	0.666	0.642	**0.797**
shuttle-2_vs_5	Physical sciences	9/3316	66.67	**1.000**	**1.000**	0.990	0.983	0.994
winequality-red-3_vs_5	Business	11/691	68.10	0.500	0.636	0.484	**0.772**	0.581
abalone-20_vs_8-9-10	Life sciences	8/1916	72.69	0.769	0.873	0.550	0.747	**0.889**
kddcup-buffer_overflow_vs_back	Intrusion detection	41/2233	73.43	**0.999**	0.983	0.989	0.994	0.872
kddcup-land_vs_satan	Intrusion detection	41/1610	75.67	0.975	**1.000**	0.992	0.999	0.775
kr-vs-k-zero_vs_fifteen	Game	6/2193	80.22	0.960	**1.000**	**1.000**	**1.000**	**1.000**
poker-8-9_vs_5	Game	10/2075	82.00	0.500	0.424	**0.838**	0.602	0.834
poker-8_vs_6	Game	10/1477	85.88	0.500	**0.749**	0.480	0.497	0.706
kddcup-rootkit-imap_vs_back	Intrusion detection	41/2225	100.14	**1.000**	0.980	0.989	0.993	0.889
abalone19	Life sciences	8/4174	129.44	0.500	0.521	0.467	0.489	**0.636**
Avg.				0.74	**0.822**	0.786	0.814	0.791
Average performance differences				-0.081	-0.031	**0.054**	0.049	-0.01

C4.5 ($p < 0.05$) based on the Wilcoxon rank-sum test, in which IFOREST performs very similar to the SMT baseline with no significant difference in the level of performance. The results indicate that OCC classifiers are suitable for the datasets with high imbalance ratios. This finding is consistent with related works of anomaly detection where OCC classifiers usually perform better than traditional multi-class classifiers for highly imbalanced datasets [17].

TABLE 4. AUC rates of different classifiers with feature selection.

ID	OCSVM				IFOREST				LOF			
	base	*PCA*	*C4.5*	*GA*	*base*	*PCA*	*C4.5*	*GA*	*base*	*PCA*	*C4.5*	*GA*
1	0.528	0.533	0.797	0.602	0.496	0.525	**0.808**	0.774	0.704	0.704	0.675	0.795
2	0.993	0.992	0.990	0.943	0.994	0.987	0.888	0.964	0.668	0.756	**0.996**	0.873
3	0.427	0.417	**1.000**	**1.000**	0.358	0.370	0.995	0.999	0.578	0.577	**1.000**	0.996
4	0.521	0.524	**0.769**	0.656	0.535	0.542	0.737	0.768	0.617	0.627	0.681	0.727
5	**0.674**	0.668	0.466	0.503	0.673	0.661	0.612	0.486	0.663	0.646	0.521	0.567
6	0.922	0.925	0.943	0.751	0.942	0.952	0.943	0.934	**0.970**	**0.970**	0.849	0.943
7	0.922	0.918	0.995	0.900	0.929	0.909	1.000	0.994	0.937	0.934	0.896	**0.999**
8	0.931	0.926	0.988	0.990	0.935	0.929	0.998	0.988	0.937	0.937	0.990	**1.000**
9	0.488	0.480	0.915	0.666	0.732	0.782	0.904	**0.924**	0.769	0.760	0.637	0.904
10	0.885	0.886	0.989	0.979	0.503	0.719	0.936	**0.987**	0.831	0.831	0.977	0.881
11	0.896	0.902	0.534	0.643	**0.944**	0.935	0.690	0.512	0.880	0.880	0.658	0.616
12	0.570	0.566	0.596	0.833	0.673	0.757	0.727	0.590	0.813	**0.839**	**0.839**	0.723
13	0.730	0.735	0.826	0.901	0.747	0.760	0.914	0.825	0.607	0.614	0.884	**0.924**
14	0.570	0.565	0.798	0.592	**0.923**	0.898	0.782	0.813	0.912	0.912	0.543	0.783
15	0.992	**0.994**	0.917	0.826	0.970	0.937	0.860	0.934	0.931	0.922	0.813	0.884
16	0.259	0.276	0.542	**0.646**	0.408	0.451	0.611	0.569	0.501	0.497	0.544	0.635
17	0.252	0.257	0.892	0.853	0.352	0.358	0.977	0.972	0.519	0.509	0.250	**1.000**
18	0.996	0.996	0.439	0.605	**1.000**	0.995	0.484	0.436	0.994	0.994	0.635	0.482
19	0.916	0.917	0.913	0.934	0.862	0.882	0.870	0.913	0.861	0.863	**0.941**	0.883
20	0.599	0.601	0.819	0.703	0.758	**0.926**	0.829	0.830	0.811	0.815	0.691	0.824
21	**0.968**	0.877	0.666	0.747	0.970	0.671	0.626	0.678	0.954	0.887	0.797	0.601
22	0.689	0.743	0.990	0.964	0.789	0.803	0.983	0.991	0.891	0.889	**0.994**	0.933
23	0.580	0.666	0.484	0.686	0.722	0.732	0.789	0.480	**0.851**	0.846	0.581	0.757
24	0.943	0.904	0.550	0.864	**0.979**	0.961	0.747	0.556	0.957	0.922	0.889	0.749
25	0.797	0.773	0.989	0.812	0.795	0.775	**0.992**	0.988	0.675	0.717	0.879	0.990
26	0.990	0.688	0.992	0.836	0.900	0.539	**1.000**	0.986	0.996	0.845	0.775	0.999
Avg.	0.705	0.694	0.770	0.757	0.737	0.732	**0.804**	0.774	0.771	0.766	0.738	0.795
27	**1.000**	0.982	**1.000**	**1.000**	**1.000**	0.962	**1.000**	**1.000**	**1.000**	0.976	**1.000**	0.999
28	0.614	0.757	**0.838**	0.835	0.712	0.740	0.587	0.746	0.770	0.716	0.833	0.589
29	0.515	0.477	0.480	**0.729**	0.641	0.596	0.525	0.482	0.466	0.522	0.706	0.489
30	0.455	0.883	0.989	0.930	0.552	0.858	0.989	0.978	0.943	0.809	0.890	**0.991**
31	0.990	0.994	0.466	0.658	0.990	**1.000**	0.513	0.449	0.995	0.900	0.636	0.511
32	0.999	0.988	0.797	0.602	0.966	**1.000**	0.808	0.774	0.988	0.990	0.675	0.795
33	0.883	0.920	0.990	0.943	0.934	0.814	0.888	0.964	0.915	0.682	**0.996**	0.873
34	0.967	0.971	**1.000**	**1.000**	**1.000**	0.930	0.995	0.999	0.989	0.897	**1.000**	0.996
35	0.517	0.553	0.769	0.656	**0.780**	0.684	0.737	0.768	0.534	0.659	0.681	0.727
36	0.679	0.602	0.466	0.503	**0.862**	0.846	0.612	0.486	0.596	0.823	0.521	0.567
37	0.891	0.829	0.943	0.751	0.888	0.933	**0.943**	0.934	0.826	0.882	0.849	**0.943**
38	0.712	0.794	0.995	0.900	0.829	0.699	**1.000**	0.994	0.798	0.517	0.896	0.999
39	0.581	0.911	0.988	0.990	0.751	0.874	0.998	0.988	0.917	0.813	0.990	**1.000**
40	0.500	0.570	0.915	0.666	0.721	0.612	0.904	**0.924**	0.542	0.544	0.637	0.904
41	0.955	0.912	0.989	0.979	**1.000**	0.998	0.936	0.987	0.913	0.885	0.977	0.881
42	0.500	0.439	0.534	0.643	0.674	0.627	**0.690**	0.512	0.439	0.658	0.658	0.616
43	0.994	0.912	0.596	0.833	**0.998**	0.909	0.727	0.590	0.913	0.835	0.839	0.723
44	0.502	0.811	0.826	0.901	0.625	0.846	0.914	0.825	0.819	0.691	0.884	**0.924**
45	0.500	0.401	0.798	0.592	0.747	0.334	0.782	**0.813**	0.666	0.629	0.543	0.783
46	**1.000**	0.990	0.917	0.826	**1.000**	0.986	0.860	0.934	0.990	0.994	0.813	0.884
47	0.500	0.512	0.542	0.646	0.636	**0.746**	0.611	0.569	0.484	0.583	0.544	0.635
48	0.769	0.548	0.892	0.853	0.873	0.898	0.977	0.972	0.550	0.899	0.250	**1.000**
49	0.999	0.988	0.439	0.605	0.983	**1.000**	0.484	0.436	0.989	0.868	0.635	0.482
50	0.975	0.992	0.913	0.934	**1.000**	0.997	0.870	0.913	0.992	0.775	0.941	0.883
51	0.960	0.999	0.819	0.703	**1.000**	0.995	0.829	0.830	**1.000**	0.998	0.691	0.824
52	0.500	**0.852**	0.666	0.747	0.424	0.662	0.626	0.678	0.838	0.840	0.797	0.601
53	0.500	0.108	0.990	0.964	0.749	0.068	0.983	0.991	0.480	0.529	**0.994**	0.933
54	**1.000**	0.989	0.484	0.686	0.980	**1.000**	0.789	0.480	0.989	0.884	0.581	0.757
55	0.500	0.463	0.550	0.864	0.521	0.656	0.747	0.556	0.467	0.632	**0.889**	0.749
Avg.	0.740	0.764	0.779	0.791	**0.822**	0.802	0.804	0.778	0.786	0.773	0.771	0.795

Further examination of the average differences in performance from low to high imbalance ratio datasets shows degradation in performance when the imbalance ratio increases. This indeed occurs in the C4.5 and SMT baselines. In contrast, the OCC classifiers, especially OCSVM and IFOREST, provide better and more stable performance than the baselines over high imbalance ratio datasets. Among the three OCC classifiers, IFOREST is the better choice, which performs significantly better than OCSVM and LOF ($p < 0.05$). This indicates that tree-based learning techniques are good candidate for class imbalanced datasets. Related works of using multi-class classifiers also show that C4.5 perform reasonably well, which are widely used as the one representative baseline classifier [6].

In short, the OCC classifiers are very suitable for the class imbalanced datasets with high imbalance ratios. More specifically, without the help of data re-sampling to balance the class imbalanced datasets, the OCC classifiers, especially IFOREST, have the potential to outperform the C4.5 baseline with a performance similar to the SMT baseline. This motivates us to further study the possibility of improving the performances of the OCC classifiers, for example, by performing feature selection over the training data in the majority class (c.f. Study II) or constructing one-class classifier ensembles (c.f. Study III).

B. RESULTS FROM STUDY II

The focus in this study is on performing feature selection by PCA, GA, and C4.5 on the training data in the majority class. For GA and C4.5, the clustering analysis by k-means is based on setting the number of clusters at 10, 50, and 100, i.e. sub-classes. Only the best clustering settings for GA and C4.5 are provided here. Table 4 shows the performance results for the OCC classifiers with feature selection. Note that due to the limitations of space; the 55 datasets are listed based on the imbalance ratios from 1.82 to 129.44, and they are divided into low and high imbalance ratio groups, which are the same as Table 3. In addition, 'base' means the baseline OCC classifiers without performing feature selection.

As we can see, the combination of feature selection with the OCC classifiers does not necessarily lead to better performance than for the baseline OCC classifiers. For the example of the datasets with low and high imbalance ratios, after performing PCA, the AUC rates provided by the three OCC classifiers are slightly worse than for the baseline classifiers.

One possible reason for these results is that the dimensions of the 55 datasets only range from 7 to 41, where the average dimension is 12. This indicates that performing feature selection over relatively lower dimension datasets does not produce certain performance improvement. Particularly, after performing feature selection by PCA, GA, and C4.5, the average feature reduction rates are 20%, 53%, and 45%, respectively.

Although the results show that consideration of a data pre-processing step, i.e. feature selection, for the majority class does not always have a positive impact on the performance of OCC classifiers, it should be noted that the feature dimensions of these 55 datasets are not high, ranging from 5 to 41. This may be a major reason for the lack of obvious performance improvement after performing feature selection.

C. RESULTS FROM STUDY III

The aim of the third experimental study is to examine the performance of OCC classifier ensembles constructed by combining multiple classifiers, as in previous studies. As described in Section 3.3, there are homogeneous and heterogeneous classifier ensembles, with each type of ensemble including different combinations. Tables 5 and 6 show the average AUC rates obtained with the homogeneous and heterogeneous classifier ensembles over the 55 datasets. Note that in Table 5, the performance of the classifier ensembles, which is better than the corresponding base classifiers, is underlined. In Table 6, the best performance for each type of combinations is underlined.

TABLE 5. Average AUC rates of homogeneous classifier ensembles.

Homogeneous OCC classifier ensembles: *Base classifier*	*Combinations*	Avg. AUC rates
OCSVM (0.761)	OCSVM + OCSVM_PCA + OCSVM_C4.5 + OCSVM_GA	0.764
	OCSVM + OCSVM_PCA + OCSVM_C4.5	0.761
	OCSVM + OCSVM_PCA + OCSVM_GA	0.761
	OCSVM + OCSVM_C4.5 + OCSVM_GA	0.767
	OCSVM_PCA + OCSVM_C4.5 + OCSVM_GA	0.763
	OCSVM + OCSVM_PCA	0.755
	OCSVM + OCSVM_C4.5	0.764
	OCSVM + OCSVM_GA	0.765
	OCSVM_PCA + OCSVM_C4.5	0.758
	OCSVM_PCA + OCSVM_GA	0.757
	OCSVM_C4.5 + OCSVM_GA	**0.768**
IFOREST (0.791)	IFOREST + IFOREST_PCA + IFOREST_C4.5 + IFOREST_GA	0.798
	IFOREST + IFOREST_PCA + IFOREST_C4.5	**0.799**
	IFOREST + IFOREST_PCA + IFOREST_GA	0.797
	IFOREST + IFOREST_C4.5 + IFOREST_GA	0.794
	IFOREST_PCA + IFOREST_C4.5 + IFOREST_GA	0.797
	IFOREST + IFOREST_PCA	0.797
	IFOREST + IFOREST_C4.5	0.794
	IFOREST + IFOREST_GA	0.791
	IFOREST_PCA + IFOREST_C4.5	0.798
	IFOREST_PCA + IFOREST_GA	0.794
	IFOREST_C4.5 + IFOREST_GA	0.793
LOF (0.796)	LOF + LOF_PCA + LOF_C4.5 + LOF_GA	0.796
	LOF + LOF_PCA + LOF_C4.5	0.795
	LOF + LOF_PCA + LOF_GA	0.795
	LOF + LOF_C4.5 + LOF_GA	**0.797**
	LOF_PCA + LOF_C4.5 + LOF_GA	0.795
	LOF + LOF_PCA	0.795
	LOF + LOF_C4.5	0.796
	LOF + LOF_GA	**0.797**
	LOF_PCA + LOF_C4.5	0.795
	LOF_PCA + LOF_GA	0.793
	LOF_C4.5 + LOF_GA	**0.797**

Some specific homogeneous OCC classifier ensemble combinations can provide slightly better performances than the baselines. In particular, the IFOREST series include the most combinations, which significantly perform better than the IFOREST baseline ($p < 0.05$). Moreover, among the 33 combinations, the best homogeneous OCC classifier ensembles are based on IFOREST + IFOREST_PCA + IFOREST_C4.5 for an average AUC rate of 0.799.

On the other hand, most of the heterogeneous OCC classifier ensembles combinations outperform the best homogeneous OCC classifier ensembles, i.e. 15 out of 19, with the significant level of performance difference ($p<0.05$). These results are consistent with those in the ensemble learning

TABLE 6. Average AUC rates of heterogeneous classifier ensembles.

Heterogeneous OCC classifier ensembles: *Types*	*Combinations*	Avg. AUC rates
Ensembles without feature selection	OSVM + LOF + IFOREST	0.804
	OSVM + LOF	0.796
	OSVM + IFOREST	0.785
	LOF + IFOREST	**0.814**
Ensembles with feature selection	OSVM_PCA + LOF_PCA + IFOREST_PCA	0.793
	OSVM_C4.5 + LOF_C4.5 + IFOREST_C4.5	**0.806**
	OSVM_GA + LOF_GA + IFOREST_GA	0.802
The best base classifier with/without feature selection	OCSVM_C4.5 + LOF + IFOREST_C4.5	**0.806**
The top 2, 3, 4, … combinations of all twelve OCC classifiers	Top 2	0.796
	Top 3	**0.819**
	Top 4	0.816
	Top 5	**0.819**
	Top 6	0.817
	Top 7	0.818
	Top 8	**0.819**
	Top 9	0.817
	Top 10	0.814
	Top 11	0.811
	All of the twelve OCC classifiers	0.808

TABLE 7. Performance comparison of baselines, OCC classifiers, and their ensembles.

C4.5	SMT	Single OCC classifiers (OSVM/IFOREST/LOF)			Best OCC classifier ensembles (Homo./Hete.)	
Imbalance ratios < 27.34						
0.821	**0.852**	0.732	0.765	0.801	0.774	0.808
Imbalance ratios > 27.34						
0.740	0.822	0.786	0.814	0.791	0.821	**0.829**

literature, where it can be seen that the combined classifiers should be as diversified as possible in order to obtain certain performance improvement over single classifiers (Rokach, 2010; Wozniak *et al.*, 2014). In this category, the best combinations are based on combining the top 3, 5, and 8 OCC classifiers out of 12, which all provide an average AUC rate of 0.819.

As we can see, for the datasets having low class imbalance ratios, the best OCC classifier ensembles do not provide much performance improvement over the best single OCC classifier, i.e. LOF, specifically 0.808 vs. 0.801. Moreover, the performance of the OCC classifiers and their ensembles cannot competitive with the C4.5 and SMT baselines. However, the OCC classifier ensembles show their potential for high class imbalance ratio datasets. In other words, for the datasets with very high class imbalance ratio, IFOREST as the single OCC classifier, is a good candidate for a baseline classifier, whereas a carefully combined heterogeneous OCC classifier ensemble can perform even better than IFOREST.

Table 7 shows comparative performance of the C4.5 and SMT baselines, the three single OCC classifiers, and the best homogeneous and heterogeneous OCC classifier ensembles. Note that the best homogeneous and heterogeneous OCC classifier ensembles are IFOREST + IFOREST_PCA + IFOREST_C4.5 and LOF + LOF_C4.5 + IFOREST_C4.5, respectively. For the best heterogeneous OCC classifier ensembles, we only consider the combination of the top 3 classifiers due to the model complexity.

V. CONCLUSION

Many real-world domain problem datasets used for classification problems suffer from the class imbalance problem. While many related works focus on the data level, algorithm level, and cost sensitivity solutions, very few consider one-class classification (OCC) techniques, which have been widely used in anomaly or outlier detection where only the normal class data are available.

In this paper, we conduct an empirical study of the performance of three representative OCC classifiers, i.e. OCSVM, IFOREST, and LOF, and their ensembles based on 55 different two-class datasets containing different imbalance ratios ranging from 1.82 to 129.44. Unlike the conventional two-class classifiers, OCC classifiers only take the training data in the majority class during the classifier training stage.

The research findings can be summarized as follows. First, the OCC classifiers are especially suitable for datasets having very high class imbalance ratios. For lower class imbalance ratio datasets, traditional approaches, such as C4.5 and SMOTE + C4.5, can perform better than the OCC classifiers. Second, using different algorithms to perform feature selection to filter out unrepresentative or noisy features from the training data in the majority class does not improve the final performance of the OCC classifiers for most datasets. Third, OCC classifier ensembles constructed by combining different OCC classifiers, with and without feature selection, have shown their potential to improve the performances, especially for high class imbalance ratio datasets.

There are several issues that can be considered in the future. First, although the feature selection results are not promising, this may be because the feature dimensions of the chosen datasets are not high, ranging from 5 to 41. It is still worth investigating the feature selection effect on the performance of OCC classifiers for some specific domain problem datasets or larger scale datasets that contain higher dimensional features and larger numbers of data samples. On the other hand, using the k-means algorithm to group the data in the majority class may provide some bias clustering result for feature selection. Further examinations should be considered, such as using different clustering algorithms, such as affinity propagation, or filtering out some noisy data (or outliers) from the majority class first and then performing feature selection. Second, instance selection is another important data preprocessing step aimed at filtering out noisy data from a given training set. It would be useful to perform instance selection over the training data in the majority class to note the level of performance improvement of OCC classifiers. Third, for the problem of multi-class class imbalanced datasets, multiple OCC classifiers may be constructed, in which each classifier is trained by one specific class. In this case, the combination method becomes critical, such as what classifiers are combined, the final output generated from multiple outputs obtained by multiple classifiers, etc. Fourth, other supervised learning based classifiers, such as random forests, naïve bayes, logistic regression, multilayer perceptron, etc. and further modifications of the classifiers based on such as SGTM (Successive Geometric Transformations Model) neural-like structure can be considered for performance comparison.

REFERENCES

[1] S. Fotouhi, S. Asadi, and M. W. Kattan, "A comprehensive data level analysis for cancer diagnosis on imbalanced data," *J. Biomed. Informat.*, vol. 90, Feb. 2019, Art. no. 103089.

[2] Z. Liu, D. Tang, Y. Cai, R. Wang, and F. Chen, "A hybrid method based on ensemble WELM for handling multi class imbalance in cancer microarray data," *Neurocomputing*, vol. 266, pp. 641–650, Nov. 2017.

[3] Q. Song, Y. Guo, and M. Shepperd, "A comprehensive investigation of the role of imbalanced learning for software defect prediction," *IEEE Trans. Softw. Eng.*, vol. 45, no. 12, pp. 1253–1269, Dec. 2019.

[4] Z. Chen, Q. Yan, H. Han, S. Wang, L. Peng, L. Wang, and B. Yang, "Machine learning based mobile malware detection using highly imbalanced network traffic," *Inf. Sci.*, vols. 433–434, pp. 346–364, Apr. 2018.

[5] J. Vanhoeyveld and D. Martens, "Imbalanced classification in sparse and large behaviour datasets," *Data Mining Knowl. Discovery*, vol. 32, no. 1, pp. 25–82, Jan. 2018.

[6] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern., C, Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2012.

[7] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Inf. Sci.*, vol. 250, pp. 113–141, Nov. 2013.

[8] M. Koziarski, B. Krawczyk, and M. Woźniak, "Radial-based oversampling for noisy imbalanced data classification," *Neurocomputing*, vol. 343, pp. 19–33, May 2019.

[9] S. Kumar, S. K. Biswas, and D. Devi, "TLUSBoost algorithm: A boosting solution for class imbalance problem," *Soft Comput.*, vol. 23, no. 21, pp. 10755–10767, Nov. 2019.

[10] J. Liu and E. Zio, "Integration of feature vector selection and support vector machine for classification of imbalanced data," *Appl. Soft Comput.*, vol. 75, pp. 702–711, Feb. 2019.

[11] A. Mahani and A. R. Baba-Ali, "A new rule-based knowledge extraction approach for imbalanced datasets," *Knowl. Inf. Syst.*, vol. 61, no. 3, pp. 1303–1329, Dec. 2019.

[12] S. Piri, D. Delen, and T. Liu, "A synthetic informative minority oversampling (SIMO) algorithm leveraging support vector machine to enhance learning from imbalanced datasets," *Decis. Support Syst.*, vol. 106, pp. 15–29, Feb. 2018.

[13] R. Sundar and M. Punniyamoorthy, "Performance enhanced boosted SVM for imbalanced datasets," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105601.

[14] C.-F. Tsai, W.-C. Lin, Y.-H. Hu, and G.-T. Yao, "Under-sampling class imbalanced datasets by combining clustering analysis and instance selection," *Inf. Sci.*, vol. 477, pp. 47–54, Mar. 2019.

[15] Z. Wang, Y. Zhu, Z. Chen, J. Zhang, and W. Du, "Multi-view learning with Fisher kernel and bi-bagging for imbalanced problem," *Int. J. Speech Technol.*, vol. 49, no. 8, pp. 3109–3122, Aug. 2019.

[16] Y. Wang and L. Yang, "A robust loss function for classification with imbalanced datasets," *Neurocomputing*, vol. 331, pp. 40–49, Feb. 2019.

[17] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1-15:58, 2009.

[18] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, Oct. 2004.

[19] S. S. Khan and M. G. Madden, "One-class classification: Taxonomy of study and review of techniques," *Knowl. Eng. Rev.*, vol. 29, no. 3, pp. 345–374, Jun. 2014.

[20] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognit.*, vol. 58, pp. 121–134, Oct. 2016.

[21] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4393–4402.

[22] L. Swersky, H. O. Marques, J. Sander, R. J. G. B. Campello, and A. Zimek, "On the evaluation of outlier detection and one-class classification methods," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2016, pp. 1–10.

[23] Q. Wei, Y. Ren, R. Hou, B. Shi, J. Y. Lo, and L. Carin, "Anomaly detection for medical images based on a one-class classification," *Proc. SPIE*, vol. 10575, Feb. 2018, Art. no. 105751M.

[24] P. Wu, J. Liu, and F. Shen, "A deep one-class neural network for anomalous event detection in complex scenes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2609–2622, Jul. 2020.

[25] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[26] Y. Li, T. Li, and H. Liu, "Recent advances in feature selection and its applications," *Knowl. Inf. Syst.*, vol. 53, no. 3, pp. 551–577, Dec. 2017.

[27] L. Rokach, "Ensemble-based classifiers," *Artif. Intell. Rev.*, vol. 33, nos. 1–2, pp. 1–39, 2010.

[28] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Inf. Fusion*, vol. 16, pp. 3–17, Mar. 2014.

[29] B. Krawczyk, I. Triguero, S. Garcia, M. Wozniak, and F. Herrera, "Instance selection for one-class classification," *Knowl. Inf. Syst.*, vol. 59, pp. 601–628, Jun. 2019.

[30] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognit.*, vol. 74, pp. 406–421, Feb. 2018.

[31] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, 2000.

[32] D. M. J. Tax and R. P. W. Duin, "Support vector domain description," *Pattern Recognit. Lett.*, vol. 20, nos. 11–13, pp. 1191–1199, Nov. 1999.

[33] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422.

[34] Z. Chen, T. Lin, X. Xia, H. Xu, and S. Ding, "A synthetic neighborhood generation based ensemble learning for the imbalanced data classification," *Int. J. Speech Technol.*, vol. 48, no. 8, pp. 2441–2457, Aug. 2018.

[35] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Inf. Sci.*, vols. 409–410, pp. 17–26, Oct. 2017.

[36] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[37] C. Chen, Y. Tsai, F. Chang, and W. Lin, "Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results," *Expert Syst.*, vol. 37, no. 5, Oct. 2020, Art. no. e12553.

CHIH-FONG TSAI received the Ph.D. degree from the School of Computing and Technology, University of Sunderland, U.K., in 2005. He is currently a Distinguish Professor with the Department of Information Management, National Central University, Taiwan. He has published more than 80 professional publications. His current research interests include data mining and machine learning.

WEI-CHAO LIN received the Ph.D. degree from the University of Sunderland, U.K. He is currently an Associate Professor with the Department of Information Management, Chang Gung University, Taiwan. He has published over 20 journal articles, such as *Information Sciences*, the *Journal of Systems and Software*, *Applied Soft Computing*, *Neurocomputing*, and so on. His current research interests include data mining, machine learning, and information retrieval.