Hindawi
Scientific Programming
Volume 2019, Article ID 5901087, 14 pages
https://doi.org/10.1155/2019/5901087

Research Article

Consensus Clustering-Based Undersampling Approach to Imbalanced Learning

Aytuğ Onan

İzmir Katip Çelebi University, Faculty of Engineering and Architecture, Department of Computer Engineering, 35620 İzmir, Turkey

Correspondence should be addressed to Aytuğ Onan; aytugonan@gmail.com

Received 11 November 2018; Revised 15 January 2019; Accepted 10 February 2019; Published 3 March 2019

Guest Editor: Vicente García-Díaz

Class imbalance is an important problem, encountered in machine learning applications, where one class (named as, the minority class) has extremely small number of instances and the other class (referred as, the majority class) has immense quantity of instances. Imbalanced datasets can be of great importance in several real-world applications, including medical diagnosis, malware detection, anomaly identification, bankruptcy prediction, and spam filtering. In this paper, we present a consensus clustering based-undersampling approach to imbalanced learning. In this scheme, the number of instances in the majority class was undersampled by utilizing a consensus clustering-based scheme. In the empirical analysis, 44 small-scale and 2 large-scale imbalanced classification benchmarks have been utilized. In the consensus clustering schemes, five clustering algorithms (namely, k-means, k-modes, k-means++, self-organizing maps, and DIANA algorithm) and their combinations were taken into consideration. In the classification phase, five supervised learning methods (namely, naïve Bayes, logistic regression, support vector machines, random forests, and k-nearest neighbor algorithm) and three ensemble learner methods (namely, AdaBoost, bagging, and random subspace algorithm) were utilized. The empirical results indicate that the proposed heterogeneous consensus clustering-based undersampling scheme yields better predictive performance.

1. Introduction

Class imbalance is an important research problem in machine learning, where the proportion of instances belonging to one class (referred as, the minority class) is extremely small, whereas the proportion of instances of the other class or classes (referred as, the majority class) is extremely high. Imbalanced datasets pose several challenges to the conventional supervised learning methods. Conventional supervised learning methods (such as support vector machines and decision trees) can build viable classification models for balanced datasets. Since imbalanced datasets suffer from outnumbering the instances of majority class and underrepresenting the instances of minority class, skewed distributions may lead to degradation of predictive performance [1, 2]. Supervised learning process is based on the use of global evaluation measures (such as classification accuracy). Hence, learning from imbalanced datasets can have bias towards the majority class, and classification models may tend to misclassify the instances of minority class [3]. Supervised learning algorithms may regard the instances of minority class as noise or outlier, and noisy data and outlier may be regarded as the instances of minority class [4]. In addition, classification models for datasets with skewed sample distributions may be challenging to learn due to the overlapping nature of the instances of minority class with the instances of other classes [5].

Imbalanced datasets can be encountered in several real-world problems and applications, including software fault identification [6], medical diagnosis [7], malware detection [8], anomaly identification [9], bankruptcy prediction [10], and spam filtering [11]. For data mining problems mentioned in advance, the number of instances for minority class is scarce. However, the identification of the instances of minority class may be more critical. For instance, the misclassification of cancerous (malignant) tumors as noncancerous (benign) in medical diagnosis can have severe effects. Similarly, the number of instances for fraudulent transactions can be scarce. However, it is critical to build prediction models that can identify fraudulent transactions in finance. Hence, handling imbalanced datasets properly is an important research problem in machine learning.

To deal efficiently with the datasets with imbalanced distribution and to build robust and efficient classification schemes, data preprocessing methods have been utilized in conjunction with machine learning algorithms. The methods utilized to tackle with class imbalance problem can be mainly divided into four categories as algorithm level approaches, data-level approaches, cost-sensitive approaches, and ensemble learning-based approaches [12]. Algorithm level approaches seek to adapt supervised learning algorithms to bias learning towards the instances of minority class [13]. Data-level approaches seek to rebalance the instances of the imbalanced dataset so that the effects of skewed distributions can be eliminated in the learning process [14]. In order to do so, data-level approaches utilize resampling on the training datasets. Cost-sensitive approaches aim at minimizing total cost of errors for minority and majority classes by defining misclassification costs [15]. In addition, ensemble learning-based approaches have been also utilized for class imbalance. Ensemble classifiers aim at enhancing the predictive performance of a single learning algorithm by combining the predictions of several learning algorithms. In ensemble approaches to imbalanced learning, several strategies (such as bagging and undersampling, undersampling and cost-sensitive learning, boosting and resampling) have been combined [12]. In data-level approaches, data preprocessing and learning process of supervised learning algorithm are handled independently. In addition, compared to the cost-sensitive approaches, which involve to set cost matrix for imbalanced datasets, data-level preprocessing (resampling) is a viable tool to apply for researchers who are not expert in the field [1]. Hence, regarding different approaches to imbalanced learning, data-level approaches, which are based on resampling the imbalanced datasets, are frequently employed. The two main directions on data-level approaches are undersampling and oversampling. In order to obtain a dataset with balanced class distribution, the original imbalanced dataset can be resampled by oversampling the minority class or undersampling the majority class [16, 17]. In addition, there are several hybrid approaches, which combine undersampling and oversampling methods, such as SMOTEBoost, OverBagging, and UnderBagging [18–20]. Compared to the oversampling, undersampling yields better predictive performance [21]. However, undersampling may result in elimination of some useful representative instances of majority class [22]. Hence, the identification of useful representative instances in undersampling is of great performance to the predictive performance of supervised learning algorithms on imbalanced learning. In response, clustering methods can be utilized to identify useful representative instances of majority class in undersampling for imbalanced learning [23–25].

In this paper, we present a consensus clustering-based undersampling approach to imbalanced learning. In this scheme, the number of instances in the majority class was undersampled by utilizing a consensus clustering-based scheme. There are a large number of clustering algorithms in the literature. However, there is no single clustering algorithm that can yield the best clustering results under all scenarios, as the no free lunch theorem claims [26]. In this regard, the presented scheme aims at combining the decisions of different clustering algorithms, to overcome the limitations of individual clustering algorithms to achieve more robust/efficient clustering results. In this way, the presented scheme aims at identifying better representative instances of majority class in undersampling for imbalanced learning. In the empirical analysis, 44 small-scale and 2 large-scale imbalanced classification (with imbalance ratios ranged between 1.8 and 163.19) were utilized. In the empirical analysis, the predictive performances of two clustering-based framework (namely, homogeneous and heterogeneous consensus clustering schemes) were compared with three data-level methods (namely, SMOTEBoost algorithm [16], RUSBoost [27], and underBagging algorithm [28, 29]). In the consensus clustering schemes, five clustering algorithms (namely, k-means, k-modes [30], k-means++ [31], self-organizing maps [32], and DIANA algorithm [33] and their combinations were taken into consideration. In the classification phase, five supervised learning methods (namely, naïve Bayes, logistic regression, support vector machines, random forests, and k-nearest neighbor algorithm) and three ensemble learner methods (namely, AdaBoost, bagging, and random subspace algorithm) were utilized. The empirical results indicate that the proposed heterogeneous consensus clustering-based undersampling scheme yields better predictive performance. To the best of our knowledge, the presented scheme is the first to use the paradigm of consensus clustering for imbalanced learning. The remainder of this paper is organized as follows. Section 2 briefly reviews the state of the art in imbalanced learning. Section 3 presents the proposed consensus clustering based-undersampling schemes. Section 4 presents the empirical analysis results, and Section 5 presents the concluding remarks.

2. Related Works

Imbalanced learning has attracted great research interest. As mentioned in advance, the methods to deal with imbalanced datasets can be broadly categorized as data-level methods, algorithm level methods, cost-sensitive methods, and ensemble learning-based methods. Compared to the other approaches, data-level approaches have greater potential use on imbalanced learning since they seek to improve the distribution of datasets, rather than relying on supervised learning-based enhancements [34]. This section briefly reviews the related work on imbalanced learning with emphasis on data-level approaches. Data-level approaches (sampling methods) can be mainly divided into two categories as undersampling and oversampling. Oversampling and undersampling approaches can be employed effectively for class imbalance.

Oversampling approaches aim at obtaining a balanced dataset by generating synthetic instances for the minority class. In contrast, undersampling approaches aim at obtaining a balanced dataset by removing the instances of the majority class from the training set. For instance, Anand et al. [35] introduced a distance-based undersampling approach for class imbalance. Supervised learning methods can easily construct learning models for instances that are far from the decision boundaries. In response, the presented scheme aims at eliminating the instances of majority class that are far from decision boundaries, while preserving the instances near to the decision boundaries in the training set. In this way, the balanced training set was constructed and the balanced dataset was utilized in conjunction with the weighted support vector machines. Similarly, Li et al. [36] utilized vector quantization algorithm to decrease the instances of majority class. The presented scheme employed support vector machines for imbalanced learning. In another study, Kumar et al. [37] empirically examined the effect of undersampling on the performance of clustering algorithms. In another study, Sun et al. [22] presented an ensemble classification scheme based on undersampling for imbalanced learning. In the presented scheme, the instances of majority class were first divided into several partitions with similar number of instances with the minority class. In this way, balanced datasets were generated. The balanced datasets were trained on binary classifiers to build classification models. Finally, the predictions of binary classifiers were combined by an ensemble scheme to identify the final outcome. In another study, D'Addabbo and Maglietta [38] presented a selective sampling-based approach for imbalanced learning. Based on the observation that the instances near to decision boundaries are relevant/critical, the instances of majority class near to decision boundaries are preserved. In another study, Ha and Lee [39] presented an evolutionary undersampling scheme for class imbalance. In this scheme, genetic algorithm was utilized to select the informative instances of majority class by minimizing the loss between the distributions between original and balanced datasets. In another study, Lin et al. [24] introduced two clustering-based undersampling schemes for imbalanced learning. In this scheme, the number of clusters was determined based on the number of instances of minority class, and k-means algorithm was employed to undersample the instances of majority class. More recently, Shobana and Battula [40] presented an undersampling scheme based on diversified distribution and clustering for imbalanced learning. In this scheme, k-means algorithm was employed to identify and remove rare instances and outliers.

In a recent study, Guo and Wei [41] presented a hybrid scheme based on clustering and logistic regression for imbalanced learning. In the presented scheme, clustering was utilized to partition instances of the majority class into clusters. Similarly, Douzas et al. [42] integrated k-means clustering algorithm and synthetic minority oversampling technique to eliminate noisy data and to effectively obtain a balanced dataset within classes. Recently, Han et al. [43] presented a distribution-based approach for imbalanced learning. In the presented scheme, the instances of minority class were divided into groups as noisy instances, unstable instances, boundary instances, and stable instances based on the location information for the instances. The presented scheme has been utilized to improve the predictive performance on medical diagnosis. In another study, Tsai et al. [44] introduced an undersampling approach for imbalanced learning, which integrates clustering analysis and instance selection.

As mentioned in advance, undersampling is a simple resampling strategy to deal with class imbalance problem. However, undersampling may remove potentially useful/ informative instances of the majority class, which may lead to the degradation of the predictive performance of classification schemes. In this paper, a consensus clustering-based framework is presented to identify the informative instances of majority class through the use of a cluster ensemble method.

3. Proposed Consensus Clustering-Based Undersampling Framework

Undersampling and oversampling methods can be successfully employed for class imbalance. In order to obtain a robust classification scheme with high predictive performance, undersampling methods should retain useful and informative representative instances of the majority class in the training set. Clustering (cluster analysis) is an unsupervised technique which assigns similar instances (objects) into the same cluster in terms of their proximity or similarity. Hence, clustering algorithms can be employed to identify useful instances of majority class in undersampling. With the use of clustering on undersampling, the majority class yields a distribution of instances into clusters such that similar instances are grouped together within the same cluster. One of the main problems encountered in applying clustering algorithms is the selection of an appropriate algorithm for a given problem. Each clustering algorithm has strong and weak characteristics, and the results obtained by clustering algorithms are greatly influenced based on the characteristics of dataset, parameters of algorithm, etc. The clustering algorithms suffer from instability, and the same clustering algorithm can yield a particularly different partition for different parameter settings. One possible solution to this problem is to use multiple clustering algorithms on the same dataset and to combine the outputs of individual clustering algorithms. The process is referred as consensus clustering (or cluster ensembles). Consensus clustering aims at combining the clustering results of different clustering algorithms so that a final clustering with better clustering quality can be obtained [45]. In this paper, two ensemble generation schemes are presented to undersample the instances of majority class based on consensus clustering, namely, homogeneous and heterogeneous ensemble schemes are introduced.

3.1. Consensus Function. Consensus clustering involves a staged procedure: in Stage 1, cluster ensemble is generated, and in Stage 2, consensus function is utilized to obtain the final partition from the individual clustering algorithms. There are direct approaches (such as simple voting, incremental voting, and label correspondence search), feature -based approaches (such as iterative voting consensus, mixture model, clustering aggregation, and quadratic mutual information), pairwise similarity-based approaches (such as agglomerative hierarchical models), and graph-based approaches (such as cluster-based similarity partitioning algorithm and shared nearest neighbors-based combiner) [45]. Motivated by the success of clustering algorithms on imbalanced learning [24] and the enhanced clustering quality obtained by consensus clustering schemes [46], we seek to find an efficient consensus clustering-based scheme for imbalanced learning. In this regard, we have conducted an experimental analysis with several different consensus functions. Since the highest predictive performance is obtained by direct approaches, of the wide range of consensus functions available, three consensus functions were chosen for the study.

3.1.1. Simple Voting Function (SV). Let π_r denote the reference partition and let π_g denote to be relabelled partitions, a contingency matrix $\Omega \in R^{K\times K}$ is obtained, in which K corresponds to the number of clusters. The contingency matrix entries $(\Omega(l,\, l'))$ are filled by co-occurrence statistics computed based on the following equation [45,43]:

$$\Omega(l,l') = \sum_{\forall x_{i\in X}} w(x_i), \tag{1}$$

where $w(x_i) = 1$ if $(C^r(x_i) = l) \wedge (C^g(x_i) = l')$ and $w(x_i) = 0$ otherwise. Based on the label correspondence obtained based on equation (1), the aim of the simple voting consensus is to maximize the objective function, given by

$$\sum_{l=1}^{K}\sum_{l'=1}^{K} \Omega(l,l')\Theta(l,l'), \tag{2}$$

where $\Theta(l,l') \in R^{K\times K}$ is a label correspondence matrix amongst the labels of partitions π_r and π_g. First, the reference partition (π_r) is randomly selected among the partitions of the cluster ensemble. Then, the remaining partitions are relabelled based on the reference partition by following the procedure outlined above. Finally, a majority voting scheme is employed to identify the consensus label of each instance.

3.1.2. Incremental Voting Function (IV). In incremental voting scheme (IV), data partitions are repeatedly added to the cluster ensemble. Let $P_g \in R^{N\times K}$ denote gth partition $(\pi_g \in \Pi)$. $P_g(x_i, C_t^g)$ takes the value of 1 if a data point $x_i \in X$ belongs to cluster $C_t^g \in \pi_g$. Otherwise, it takes the value of 0. Let V_g denote the matrix of intermediate g partitions $(\pi_1, \ldots, \pi_g)$ and $V_g(x_i, L_j)$ denote the number of partitions in which label L_j is corresponds to data point x_i. The process of incremental voting-based consensus is initialized with the construction of contingency matrix $\Omega \in R^{K\times K}$. The contingency matrix entries $(\Omega(l,l'))$ are filled by the following equation [48]:

$$\Omega(l,l') = \sum_{\forall x_{i\in X}} w(x_i), \tag{3}$$

where $w(x_i) = 1$ if $(V_g(x_i, L_j) \geq 1) \wedge (P_g(x_i, l') = 1)$. Otherwise, it takes the value of 0. After obtaining the contingency matrix, the entries of matrix for the $(g+1)$th partition (denoted by V_{g+1}) are computed as given by

$$\mathrm{V}_{g+1}(x_i, l) = \mathrm{V}_g(x_i, l) + \mathrm{P}_{g+1}(x_i, l'). \tag{4}$$

Based on the incremental combinations of M data partitions, the consensus label of each data point $x_i \in X$ is determined based on following equation [45]:

$$C^*(x_i) = \mathrm{argmax}_l V_M(x_i, l). \tag{5}$$

3.1.3. Label Correspondence Search. In label correspondence search (LCS), the problem of correspondence is modelled as an optimization problem [49]. The aim of the method is to obtain a consensus partition such that overall agreement among the different partitions is maximized. Let $R_{\{c,s\}}$ denote the vector representation of cluster c of system s. The $k-$th element of $R_{\{c,s\}}$ represents the posterior probabilities of cluster c for the data points. The agreement between clusters $\{c,\, s\}$ and $\{c',\, s'\}$ can be defined as given by the following equation:

$$g\{c,s\},\{c',s'\} = R^T_{\{c,s\}} \cdot R_{\{c',s'\}}. \tag{6}$$

If a cluster c of system s is assigned to metacluster m, $\lambda^{\{m\}}_{\{c,s\}}$ takes the value of 1 and it takes the value of 0 otherwise. $r^{\{m\}}_{\{c,s\}}$ denotes the reward of assigning cluster c to metacluster m, and it can be defined as given by the following equation:

$$r^{\{m\}}_{\{c,s\}} = \frac{1}{|I(m)|} \sum_{\{c',s'\}\in I(m)} g\{c,s\}, \quad \{c',s'\} \in I(m) \leftrightarrow \lambda^{\{m\}}_{\{c,s\}} \neq 0. \tag{7}$$

Based on equations (6) and (7), the objective of label correspondence is to maximize the argument defined in the following equation [49]:

$$\lambda^* = \mathrm{argmax}_\lambda \sum_{m=1}^{M}\sum_{s=1}^{S}\sum_{c=1}^{C_s} \lambda^{\{m\}}_{\{c,s\}} r^{\{m\}}_{\{c,s\}}, \tag{8}$$

subject to

$$\sum_{m=1}^{M} \lambda^{\{m\}}_{\{c,s\}} = 1, \forall c, s. \tag{9}$$

3.2. Homogeneous Consensus Clustering-Based Undersampling Framework. Let D denote an imbalanced dataset with two classes, where there is one class (referred as, the minority class) containing the small number of instances and there is another class (referred as, the majority class) containing extremely high quantity of instances. Let us denote the number of instances corresponding to majority and minority classes as n and m, respectively. Initially, k-fold cross-validation scheme is utilized for dividing the imbalanced dataset into subsets as training and test sets. Then, the number of instances in the majority class (n) is undersampled so that it contains equal number of instances to the minority class (m). In the undersampling, homogeneous consensus clustering scheme is utilized to undersample the majority class. Clustering algorithms require the number of clusters as the input parameter. We adopted the clustering framework presented in [24]. Hence, the number of instances in the minority class (m) is taken as the number of clusters (k). In homogeneous consensus clustering scheme, the same clustering algorithm is utilized as the base clustering algorithm, with different parameter settings. In this scheme, five clustering algorithms (namely, k-means, k-modes, k-means++, self-organizing maps, and DIANA algorithm) are utilized as the base clustering algorithms.

In this way, diversified partitions are obtained by the base clustering algorithms. The partitions obtained by the base clustering algorithms are combined by consensus function to obtain the final partition. For obtaining final partition with consensus function, three consensus functions (namely, simple voting function, incremental voting function, and label correspondence search algorithm) are utilized. The center of each cluster of the final partition is selected as the instance for the majority class. In this way, a balanced training set is obtained. The balanced training set is utilized to train supervised learning algorithms (namely, naïve Bayes, logistic regression, support vector machines, random forests, and k-nearest neighbor algorithm) and ensemble learning methods (namely, AdaBoost, bagging, and random subspace algorithm). The general stages of this scheme is depicted in Figure 1. In Figure 2, the general steps of homogeneous consensus clustering-based undersampling scheme (CONS1) are outlined.

3.3. Heterogeneous Consensus Clustering-Based Undersampling Framework. In heterogeneous consensus clustering scheme (CONS2), diversity among the clustering algorithms is achieved with the use of different clustering algorithms as the base clustering algorithms. As stated in advance, each clustering algorithm has its own strengths and weaknesses and can yield promising results on different datasets. The partitions obtained by different clustering algorithms may complement each other and can yield higher clustering quality. The heterogeneous consensus clustering-based undersampling framework follows the same stages as outlined in Figure 1. The only difference is that the heterogeneous consensus clustering framework utilizes 5 different clustering algorithms, as the base clustering algorithms, whereas the homogeneous consensus clustering framework utilizes the same clustering algorithm with different parameter settings, as the base clustering algorithms. The general structure of heterogeneous consensus clustering-based undersampling scheme is summarized in Figure 3. In the heterogeneous consensus clustering-based undersampling scheme, k-fold cross-validation is employed for dividing the imbalanced dataset into training set and test set. Then, the number of instances in the majority class is undersampled with the use of heterogeneous consensus clustering scheme. In this scheme, different clustering algorithms are utilized as the base clustering algorithms. The presented scheme can be configured with different clustering algorithms, yet, we have combined the five base clustering algorithms (namely, K-means, K-modes, K-means++, self-organizing maps, and DIANA algorithm). The partitions obtained by different clustering algorithms are combined by the consensus function. The center of each cluster of the final partition is selected as the instance for the majority class. In this way, a balanced training set is obtained. The predictive performance of undersampling scheme is examined with the use of supervised learning methods and ensemble learning methods.

4. Experimental Analysis and Results

This section presents the empirical analysis of the proposed consensus clustering-based undersampling schemes.

4.1. Datasets. To examine the effectiveness of the proposed undersampling approaches, we have utilized 44 small-scale and 2 large-scale imbalanced classification benchmarks. The imbalanced classification benchmarks were utilized in Galar et al. [12]. The imbalance ratios of small-scale benchmarks range from 1.8 to 129, and the number of instances ranges from 130 to 5500. The imbalance ratios of large-scale benchmarks range from 111.46 to 163.19, and the number of instances ranges from 102294 to 145751. For obtaining test and training sets for the supervised learning methods, we utilized k-fold cross-validation scheme, where we were partitioned the 80% and 20% training and testing sets with 5-fold cross-validation scheme. The basic descriptive information regarding the imbalanced classification benchmarks is presented in Table 1.

4.2. Experimental Procedure. In the empirical analysis, the presented consensus clustering-based undersampling schemes have been compared by seven state-of-the-art methods. The utilized methods in the analysis include UnderBagging4 (UB4), UnderBagging24 (UB24), RusBoost1 (Rus1), SMOTEBagging4 (SBAG4), UnderBagging1 (UB1), clustering-based undersampling based on cluster centers (Centers), and clustering-based undersampling based on the nearest neighbors of cluster centers (Centers_NN) [12, 24]. In order to examine the predictive performance changes obtained by data balancing strategies, the results obtained by C4.5 algorithm without data balancing have also been presented as the baseline results. In the consensus clustering schemes, five clustering algorithms (namely, k-means, k-modes, k-means++,

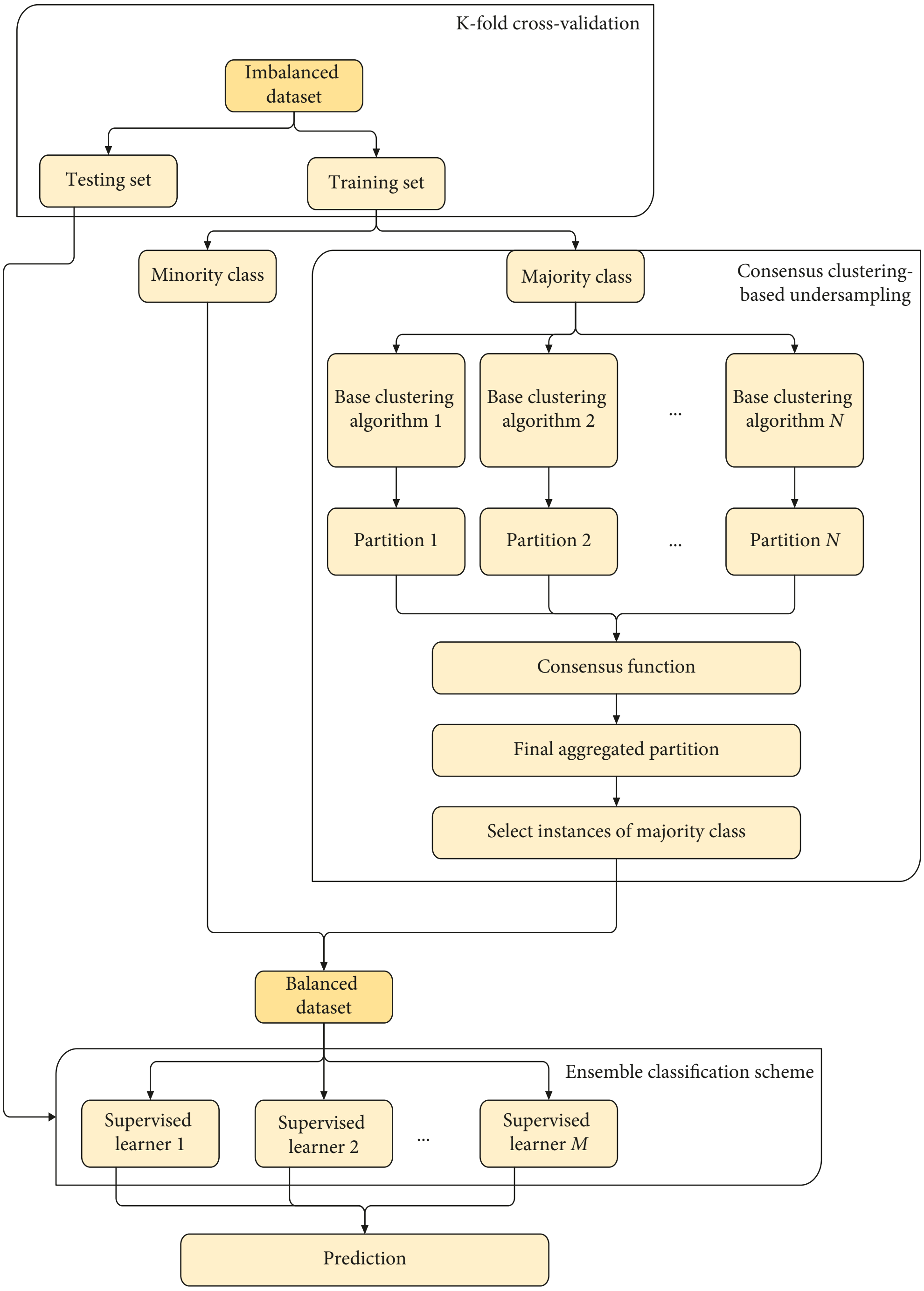


Figure 1: Homogeneous consensus clustering-based undersampling scheme (CONS1).

self-organizing maps, and DIANA algorithm) and their combinations were taken into consideration. In the classification phase, five supervised learning methods (namely, naïve Bayes, logistic regression, support vector machines, random forests, and k-nearest neighbor algorithm) and three ensemble learner methods (namely, AdaBoost, bagging, and random subspace algorithm) were utilized. In the empirical analysis, area under roc curve was utilized as the evaluation metric. For the supervised learning methods and state-of-the-art data preprocessing methods, the default parameters were employed. For the homogeneous consensus clustering-based undersampling scheme, i parameter (the number of base clustering algorithms) is taken as five.

4.3. Experimental Results and Discussions. In Table 2, average AUC values of the state-of-the-art methods and conventional clustering algorithms (namely, K-means, K-means++, K-modes, self-organizing maps, and DIANA algorithm) are presented. As it can be observed from the results presented in Table 2, the application of data balancing strategies enhance the predictive performance in terms of

Input: An imbalanced dataset D

Output: A classification model obtained from a balanced dataset D'

Undersampling Phase

1. Let n denote the number instances of the majority class in the training set and let m denote the number of instances of the minority class in the training set.
2. Let i denote user-defined parameter for the number of base clustering algorithms in homogeneous clustering scheme.
3. Utilize k-fold cross validation scheme to divide D into training and test sets.
4. Set the number of clusters equal to m.
5. Apply i times the same clustering algorithm (K-means, K-modes, K-means++, self-organizing maps or DIANA algorithm) on the instances of majority class of training set to undersample the majority class.
6. Obtain the partitions of base clustering algorithms on the majority class.
7. Obtain the final partition from the individual clustering algorithms by employing consensus function (simple voting function, incremental voting function or label correspondence search algorithm).
8. Compute the center of each cluster of the final partition.
9. Take cluster centers of the final partition as the instances of the majority class.
10. Combine the instances of majority class and the instances of minority class to obtain the balanced training set D'.

Classification Phase

1. Employ supervised learning algorithms (Naïve Bayes, logistic regression, support vector machines, random forests and k-nearest neighbor algorithm) and ensemble learners (AdaBoost, Bagging and Random Subspace method) on the balanced training set D'.
2. Use test set to evaluate the predictive performance of supervised learning methods in terms of area under roc curve and classification accuracy.

Figure 2: The general structure of the homogeneous consensus clustering-based undersampling scheme (CONS1).

Input: An imbalanced dataset D

Output: A classification model obtained from a balanced dataset D'

Undersampling Phase

1. Let n denote the number instances of the majority class in the training set and let m denote the number of instances of the minority class in the training set.
2. Utilize k-fold cross validation scheme to divide D into training and test sets.
3. Set the number of clusters equal to m.
4. Apply five different clustering algorithm (K-means, K-modes, K-means++, self-organizing maps and DIANA algorithm) on the instances of majority class of training set to undersample the majority class.
5. Obtain the partitions of base clustering algorithms on the majority class.
6. Obtain the final partition from the individual clustering algorithms by employing consensus function (simple voting function, incremental voting function or label correspondence search algorithm).
7. Compute the center of each cluster of the final partition.
8. Take cluster centers of the final partition as the instances of the majority class.
9. Combine the instances of majority class and the instances of minority class to obtain the balanced training set D'.

Classification Phase

1. Employ supervised learning algorithms (Naïve Bayes, logistic regression, support vector machines, random forests and k-nearest neighbor algorithm) and ensemble learners (AdaBoost, Bagging and Random Subspace method) on the balanced training set D'.
2. Use test set to evaluate the predictive performance of supervised learning methods in terms of area under roc curve and classification accuracy.

Figure 3: The general structure of the heterogeneous consensus clustering-based undersampling scheme (CONS2).

AUC values. The lowest average AUC values obtained by C4.5 algorithm without data balancing have been applied. The highest average AUC values are generally obtained by UnderBagging4 algorithm, and the second highest average AUC values are generally obtained by UnderBagging24 algorithm. In the empirical analysis, five base clustering

Table 1: Descriptive information for the datasets [12, 24].

Dataset	Number of data samples	Number of features	Imbalance ratio
Small-scale datasets			
Abalone9-18	731	8	16.68
Abalone19	4174	8	128.87
Ecoli-0_vs_1	220	7	1.86
Ecoli-0-1-3-7_vs_2-6	281	7	39.15
Ecoli1	336	7	3.36
Ecoli2	336	7	5.46
Ecoli3	336	7	8.19
Ecoli4	336	7	13.84
Glass0	214	9	3.19
Glass0123vs456	192	9	10.29
Glass016vs2	184	9	19.44
Glass016vs5	214	9	1.82
Glass1	214	9	10.39
Glass2	214	9	15.47
Glass4	214	9	22.81
Glass5	214	9	22.81
Glass6	214	9	6.38
Haberman	306	3	2.68
Iris0	150	4	2
New-thyroid1	215	5	5.14
New-thyroid2	215	5	4.92
Page-blocks0	5472	10	8.77
Page-blocks13vs2	472	10	15.85
Pima	768	8	1.9
Segment	2308	19	6.01
Shuttle0vs4	1829	9	13.87
Shuttle2vs4	129	9	20.5
Vehicle0	846	18	3.23
Vehicle1	846	18	2.52
Vehicle2	846	18	2.52
Vehicle3	846	18	2.52
Vowel0	988	13	10.1
Wisconsin	683	9	1.86
Yeast05679vs4	528	8	9.35
Yeast1	1484	8	2.46
Yeast1vs7	459	8	13.87
Yeast1289vs7	947	8	30.56
Yeast1458vs7	693	8	22.1
Yeast2vs4	514	8	9.08
Yeast2vs8	482	8	23.1
Yeast3	1484	8	8.11
Yeast4	1484	8	28.41
Yeast5	1484	8	32.78
Yeast6	1484	8	39.15
Large-scale datasets			
Breast cancer	102294	117	163.19
Protein homology prediction	145751	74	111.46

algorithms have been taken into consideration. Among the base clustering algorithms, the highest average AUC values are obtained by DIANA clustering algorithm.

The homogeneous consensus clustering scheme utilizes a single clustering algorithm (of the same type) as the base clustering method. In the empirical analysis, five clustering algorithms (namely, k-means, k-modes, k-means++, self-organizing maps, and DIANA algorithm) are considered as the base clustering methods. For aggregating the clustering results of individual clustering results, we considered three consensus functions (namely, simple voting function, incremental voting function, and label correspondence search algorithm). In this way, 15 different homogeneous consensus clustering-based schemes are evaluated for imbalanced learning. In Table 3, average AUC values obtained by homogeneous consensus clustering schemes are presented. Compared to the results presented in Table 2 for conventional data-level methods and conventional clustering-based schemes, homogeneous consensus clustering schemes yield better predictive performance in terms of AUC values. Among the compared homogeneous consensus clustering schemes, the highest predictive performance is obtained by utilizing self-organizing map algorithm as the base clustering algorithm. In this scheme, simple voting function is employed as the consensus function.

For the heterogeneous consensus clustering scheme, k-means, k-modes, k-means++, self-organizing maps, and DIANA algorithm methods were utilized to identify individual partitions. Similar to the homogeneous scheme, we considered three consensus functions (namely, simple voting function, incremental voting function, or label correspondence search algorithm). In this way, 3 different heterogeneous consensus clustering-based schemes are taken into consideration. In Table 4, average AUC values obtained by heterogeneous consensus clustering schemes are presented. As it can be observed from the results listed in Table 4, heterogeneous consensus clustering schemes outperform homogeneous consensus clustering schemes, conventional data-level methods, and conventional clustering-based schemes. Regarding the average AUC values analyzed in the empirical analysis, the highest predictive performance is obtained by heterogeneous clustering scheme with label correspondence search-based consensus function. The second highest predictive performance is obtained by heterogeneous clustering scheme with simple voting-based consensus function.

In the classification phase, five supervised learning methods (namely, naïve Bayes, logistic regression, support vector machines, random forests, and k-nearest neighbor algorithm) and three ensemble learner methods (namely, AdaBoost, bagging, and random subspace algorithm) were utilized. In order to summarize the main findings of the empirical analysis, boxplots for undersampling methods and supervised learning methods are presented in Figures 4 and 5, respectively.

As it can be observed from Figure 4, average AUC values obtained from the presented heterogeneous clustering scheme is higher compared to the conventional data-level methods ($p < 0.05$). In Figure 5, the predictive performance analysis of conventional supervised learning methods and their ensembles are taken into consideration. As it can be observed, ensemble learning methods yield higher predictive performance in terms of AUC values compared to the conventional supervised learning methods. The highest predictive performance for supervised learning methods is achieved by random subspace ensemble of random forest, and the second highest predictive performance is obtained

Table 2: Average AUC values of state-of-the-art methods with C4.5 classifier.

	C4.5	UB4	UB24	Rus1	SBAG4	UB1	Centers	Centers_NN	KM	KM++	KMOD	SOM	DIANA
Abalone19	0.500	0.721	0.680	0.631	0.572	0.695	0.639	0.648	0.743	0.744	0.744	0.745	0.745
Abalone9-18	0.598	0.719	0.710	0.693	0.745	0.710	0.699	0.704	0.769	0.769	0.769	0.769	0.770
Breast cancer	0.867	0.927	0.929	0.929	0.925	0.922	0.889	0.914	0.839	0.847	0.854	0.845	0.857
Ecoli-0_vs_1	0.983	0.980	0.980	0.969	0.983	0.969	0.983	0.983	0.920	0.910	0.950	0.880	0.920
Ecoli-0-1-3-7_vs_2-6	0.748	0.745	0.781	0.794	0.828	0.726	0.715	0.726	0.774	0.774	0.775	0.775	0.775
Ecoli1	0.859	0.900	0.902	0.883	0.900	0.898	0.895	0.923	0.810	0.820	0.820	0.830	0.0.840
Ecoli2	0.864	0.884	0.881	0.899	0.888	0.870	0.864	0.878	0.800	0.810	0.820	0.820	0.830
Ecoli3	0.728	0.908	0.894	0.856	0.885	0.882	0.847	0.900	0.800	0.810	0.820	0.820	0.830
Ecoli4	0.844	0.888	0.899	0.942	0.933	0.891	0.905	0.862	0.800	0.810	0.810	0.820	0.820
Glass0	0.817	0.814	0.824	0.813	0.839	0.818	0.772	0.744	0.780	0.780	0.780	0.780	0.780
Glass0123vs456	0.916	0.904	0.917	0.930	0.946	0.894	0.914	0.902	0.810	0.810	0.820	0.830	0.840
Glass016vs2	0.594	0.754	0.625	0.617	0.559	0.636	0.645	0.708	0.773	0.773	0.773	0.773	0.774
Glass016vs5	0.894	0.943	0.943	0.989	0.866	0.943	0.943	0.943	0.810	0.820	0.830	0.840	0.850
Glass1	0.740	0.737	0.752	0.763	0.728	0.748	0.713	0.647	0.734	0.737	0.739	0.739	0.739
Glass2	0.719	0.769	0.706	0.780	0.779	0.758	0.658	0.756	0.783	0.783	0.783	0.783	0.783
Glass4	0.754	0.846	0.871	0.915	0.874	0.853	0.651	0.803	0.800	0.800	0.800	0.800	0.810
Glass5	0.898	0.949	0.949	0.943	0.878	0.949	0.888	0.949	0.820	0.830	0.840	0.840	0.850
Glass6	0.813	0.904	0.926	0.918	0.931	0.885	0.858	0.847	0.800	0.800	0.810	0.810	0.820
Haberman	0.576	0.664	0.668	0.655	0.656	0.658	0.620	0.595	0.715	0.715	0.716	0.717	0.718
Iris0	0.990	0.990	0.980	0.990	0.980	0.990	0.990	0.990	0.940	0.950	0.960	0.890	0.940
New-thyroid1	0.914	0.964	0.969	0.958	0.975	0.955	0.938	0.947	0.820	0.830	0.830	0.840	0.850
New-thyroid2	0.937	0.958	0.938	0.938	0.961	0.947	0.938	0.924	0.810	0.820	0.820	0.830	0.840
Page-blocks0	0.922	0.958	0.959	0.948	0.953	0.952	0.934	0.958	0.820	0.850	0.850	0.850	0.860
Page-blocks13vs2	0.998	0.978	0.975	0.987	0.988	0.975	0.911	0.992	0.980	0.980	0.980	0.930	0.950
Pima	0.701	0.760	0.753	0.726	0.751	0.758	0.753	0.727	0.776	0.776	0.776	0.776	0.777
Segmemt0	0.983	0.988	0.986	0.993	0.994	0.985	0.981	0.980	0.890	0.890	0.910	0.870	0.900
Shuttle0vs4	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.990	0.980	0.950
Shuttle2vs4	0.950	1.000	1.000	1.000	1.000	0.988	1.000	0.988	0.920	0.940	0.950	0.880	0.930
Vehicle0	0.930	0.952	0.954	0.958	0.965	0.945	0.942	0.948	0.820	0.830	0.840	0.840	0.850
Vehicle1	0.672	0.787	0.761	0.747	0.769	0.765	0.722	0.703	0.767	0.768	0.768	0.768	0.768
Vehicle2	0.956	0.964	0.964	0.970	0.966	0.957	0.942	0.956	0.820	0.840	0.840	0.850	0.860
Vehicle3	0.664	0.802	0.784	0.765	0.763	0.764	0.757	0.731	0.778	0.778	0.778	0.778	0.778
Vowel0	0.971	0.947	0.947	0.943	0.988	0.944	0.941	0.910	0.810	0.820	0.820	0.830	0.840
Wisconsin	0.945	0.960	0.971	0.964	0.960	0.957	0.945	0.945	0.820	0.820	0.830	0.840	0.850
Yeast05679vs4	0.680	0.794	0.814	0.803	0.818	0.782	0.756	0.769	0.826	0.826	0.826	0.826	0.826
Yeast1	0.664	0.722	0.721	0.719	0.734	0.716	0.741	0.738	0.779	0.779	0.779	0.779	0.779
Yeast1289vs7	0.616	0.734	0.689	0.721	0.658	0.675	0.632	0.700	0.754	0.755	0.755	0.755	0.755
Yeast1458vs7	0.500	0.606	0.617	0.567	0.623	0.563	0.559	0.603	0.727	0.727	0.728	0.728	0.730
Yeast1vs7	0.628	0.786	0.773	0.715	0.697	0.747	0.660	0.704	0.770	0.770	0.770	0.771	0.771
Yeast2vs4	0.831	0.936	0.929	0.933	0.897	0.940	0.914	0.882	0.800	0.810	0.820	0.820	0.830
Yeast2vs8	0.525	0.783	0.747	0.789	0.784	0.761	0.629	0.778	0.826	0.826	0.827	0.827	0.827
Yeast3	0.860	0.934	0.944	0.925	0.944	0.940	0.901	0.926	0.810	0.820	0.830	0.840	0.840
Yeast4	0.614	0.855	0.854	0.812	0.773	0.860	0.722	0.857	0.800	0.810	0.810	0.810	0.820
Yeast5	0.883	0.952	0.956	0.959	0.962	0.964	0.954	0.960	0.840	0.870	0.910	0.860	0.870
Yeast6	0.712	0.869	0.878	0.823	0.836	0.864	0.691	0.818	0.800	0.800	0.810	0.810	0.820
Protein homology prediction	0.922	0.956	0.961	0.956	0.945	0.952	0.928	0.947	0.820	0.828	0.835	0.840	0.850
Twitter-sentiment	0.962	0.979	0.978	0.980	0.981	0.976	0.966	0.979	0.903	0.914	0.927	0.888	0.909
Average	0.801	*0.870*	0.865	0.862	0.859	0.858	0.826	0.847	0.815	0.821	0.826	0.820	0.828

by random subspace ensemble of support vector machines ($p < 0.05$). Regarding the predictive performance of conventional clustering algorithms, naïve Bayes demonstrated the lowest predictive performance, whereas random forest algorithm demonstrated the best (the highest) predictive performance ($p < 0.05$).

In Figure 6, the confidence intervals for the mean values of average AUC values obtained by the compared algorithms for a confidence level of 95% are presented. Based on the statistical significances between the compared results, Figure 6 is divided into two regions denoted by red dashed line. As it can be observed from Figure 6, the predictive performance differences obtained by the proposed consensus clustering-based schemes are statistically significant.

5. Conclusion

Class imbalance is an important problem of machine learning. Imbalanced datasets can be seen in a wide variety of

Table 3: Average AUC values of homogeneous clustering schemes with C4.5 classifier.

Consensus function	Simple voting	Simple voting	Simple voting	Simple voting	Simple voting	Incremental voting	Incremental voting	Incremental voting	Incremental voting	LCS	Incremental voting	LCS	LCS	LCS	LCS
Method	CONS1 (KM)	CONS1 (KM++)	CONS1 (KMOD)	CONS1 (SOM)	CONS1 (DIANA)	CONS1 (KM)	CONS1 (KM++)	CONS1 (KMOD)	CONS1 (DIANA)	CONS1 (KM)	CONS1 (SOM)	CONS1 (KM++)	CONS1 (KMOD)	CONS1 (SOM)	CONS1 (DIANA)
Abalone19	0.746	0.746	0.746	0.766	0.747	0.747	0.747	0.748	0.766	0.766	0.766	0.766	0.766	0.746	0.766
Abalone9-18	0.770	0.770	0.770	0.794	0.770	0.792	0.792	0.793	0.793	0.793	0.793	0.793	0.793	0.770	0.811
Breast cancer	0.855	0.867	0.870	0.940	0.882	0.879	0.891	0.887	0.903	0.909	0.921	0.926	0.918	0.888	0.931
Ecoli-0_vs_1	0.870	0.880	0.920	0.970	0.900	0.910	0.930	0.930	0.950	0.950	0.960	0.960	0.960	0.940	0.980
Ecoli-0-1-3-7_vs_2-6	0.775	0.775	0.775	0.782	0.775	0.778	0.779	0.780	0.780	0.780	0.781	0.781	0.782	0.775	0.788
Ecoli1	0.850	0.850	0.850	0.950	0.870	0.870	0.880	0.880	0.900	0.910	0.920	0.930	0.930	0.870	0.950
Ecoli2	0.830	0.840	0.850	0.930	0.860	0.860	0.870	0.860	0.870	0.890	0.900	0.910	0.910	0.860	0.910
Ecoli3	0.840	0.850	0.850	0.940	0.870	0.870	0.870	0.870	0.890	0.900	0.900	0.910	0.920	0.860	0.940
Ecoli4	0.830	0.840	0.840	0.930	0.860	0.860	0.870	0.860	0.870	0.890	0.890	0.910	0.850	0.850	0.850
Glass0	0.780	0.781	0.781	0.823	0.784	0.822	0.822	0.822	0.822	0.823	0.823	0.823	0.823	0.781	0.824
Glass0123vs456	0.840	0.850	0.850	0.950	0.870	0.870	0.880	0.880	0.900	0.900	0.900	0.910	0.930	0.860	0.940
Glass016vs2	0.774	0.774	0.774	0.789	0.774	0.786	0.787	0.787	0.788	0.788	0.789	0.789	0.789	0.774	0.790
Glass016vs5	0.850	0.860	0.860	0.960	0.880	0.880	0.890	0.890	0.910	0.920	0.940	0.940	0.950	0.890	0.960
Glass1	0.740	0.740	0.740	0.765	0.741	0.742	0.743	0.743	0.764	0.765	0.765	0.765	0.765	0.741	0.765
Glass2	0.784	0.784	0.784	0.842	0.784	0.842	0.842	0.842	0.842	0.842	0.842	0.842	0.842	0.784	0.842
Glass4	0.800	0.810	0.800	0.840	0.840	0.800	0.840	0.810	0.830	0.800	0.820	0.850	0.800	0.840	0.810
Glass5	0.850	0.870	0.880	0.960	0.890	0.890	0.900	0.910	0.920	0.930	0.940	0.950	0.950	0.900	0.970
Glass6	0.820	0.820	0.840	0.900	0.860	0.860	0.860	0.820	0.870	0.820	0.880	0.890	0.810	0.850	0.820
Haberman	0.718	0.722	0.722	0.759	0.725	0.725	0.725	0.727	0.757	0.757	0.758	0.758	0.759	0.724	0.759
Iris0	0.900	0.960	0.930	0.980	0.930	0.910	0.950	0.940	0.950	0.950	0.960	0.970	0.970	0.960	0.990
New-thyroid1	0.850	0.860	0.870	0.960	0.890	0.880	0.900	0.910	0.910	0.930	0.940	0.950	0.950	0.890	0.970
New-thyroid2	0.850	0.850	0.850	0.950	0.880	0.870	0.880	0.880	0.900	0.920	0.930	0.930	0.930	0.870	0.960
Page-blocks0	0.860	0.880	0.890	0.970	0.890	0.900	0.910	0.920	0.930	0.940	0.950	0.950	0.960	0.920	0.970
Page-blocks13vs2	0.960	0.960	0.950	0.990	0.970	0.940	0.950	0.940	0.950	0.970	0.990	0.980	0.970	0.960	0.990
Pima	0.777	0.777	0.777	0.792	0.777	0.790	0.790	0.791	0.791	0.791	0.792	0.792	0.792	0.777	0.792
Segmemt0	0.870	0.880	0.890	0.970	0.900	0.900	0.920	0.930	0.940	0.940	0.950	0.960	0.960	0.920	0.980
Shuttle0vs4	0.980	0.990	0.980	1.000	0.980	0.990	0.970	0.940	0.970	1.000	1.000	0.980	0.990	0.990	1.000
Shuttle2vs4	0.890	0.950	0.920	0.970	0.910	0.910	0.940	0.930	0.950	0.950	0.960	0.960	0.960	0.950	0.980
Vehicle0	0.850	0.870	0.880	0.960	0.890	0.890	0.900	0.910	0.920	0.930	0.940	0.950	0.950	0.890	0.970
Vehicle1	0.768	0.768	0.768	0.766	0.769	0.760	0.761	0.762	0.762	0.763	0.763	0.765	0.765	0.768	0.766
Vehicle2	0.860	0.880	0.880	0.970	0.890	0.900	0.900	0.910	0.920	0.940	0.950	0.950	0.950	0.900	0.970
Vehicle3	0.779	0.779	0.779	0.801	0.779	0.799	0.799	0.800	0.800	0.800	0.801	0.801	0.801	0.779	0.803
Vowel0	0.840	0.850	0.850	0.950	0.870	0.870	0.880	0.880	0.900	0.910	0.910	0.920	0.930	0.870	0.940
Wisconsin	0.850	0.860	0.870	0.960	0.880	0.880	0.890	0.890	0.910	0.930	0.940	0.950	0.950	0.890	0.960
Yeast05679vs4	0.826	0.826	0.826	0.842	0.826	0.842	0.842	0.842	0.842	0.842	0.842	0.842	0.842	0.826	0.842
Yeast1	0.779	0.780	0.780	0.811	0.780	0.809	0.810	0.810	0.810	0.810	0.811	0.811	0.811	0.780	0.813
Yeast1289vs7	0.756	0.756	0.756	0.767	0.756	0.757	0.757	0.757	0.767	0.767	0.767	0.767	0.767	0.756	0.770
Yeast1458vs7	0.730	0.731	0.731	0.762	0.732	0.732	0.733	0.734	0.760	0.762	0.762	0.762	0.762	0.732	0.762
Yeast1vs7	0.771	0.772	0.772	0.787	0.772	0.782	0.783	0.784	0.784	0.785	0.785	0.786	0.786	0.772	0.787
Yeast2vs4	0.840	0.840	0.850	0.940	0.870	0.870	0.870	0.870	0.890	0.900	0.900	0.910	0.910	0.860	0.920
Yeast2vs8	0.827	0.827	0.827	0.851	0.827	0.850	0.850	0.850	0.850	0.851	0.851	0.851	0.851	0.827	0.851
Yeast3	0.850	0.850	0.860	0.950	0.880	0.870	0.890	0.890	0.900	0.920	0.930	0.930	0.940	0.880	0.960
Yeast4	0.830	0.840	0.840	0.910	0.860	0.860	0.870	0.860	0.870	0.880	0.890	0.900	0.840	0.850	0.840
Yeast5	0.870	0.880	0.890	0.970	0.890	0.900	0.910	0.920	0.930	0.940	0.950	0.950	0.960	0.920	0.980

Table 3: Continued.

Consensus function	Simple voting	Simple voting	Simple voting	Simple voting	Simple voting	Incremental voting	Incremental voting	Incremental voting	Incremental voting	LCS	Incremental voting	LCS	LCS	LCS	LCS
Yeast6	0.810	0.820	0.830	0.850	0.850	0.800	0.840	0.810	0.850	0.810	0.870	0.870	0.810	0.850	0.810
Protein homology prediction	0.850	0.865	0.875	0.960	0.888	0.885	0.898	0.905	0.915	0.930	0.940	0.950	0.950	0.893	0.968
Twitter-sentiment	0.896	0.918	0.917	0.977	0.918	0.918	0.931	0.929	0.943	0.953	0.963	0.962	0.964	0.940	0.982
Average	0.826	0.835	0.837	<u>0.893</u>	0.845	0.848	0.856	0.854	0.867	0.871	0.879	0.883	0.878	0.849	0.888

Table 4: Average AUC values of heterogeneous clustering schemes with C4.5 classifier.

Consensus function	IV	SV	LCS
Method	CONS2	CONS2	CONS2
Abalone19	0.766	0.767	0.782
Abalone9-18	0.812	0.812	0.812
Breast cancer	0.945	0.946	0.954
Ecoli-0_vs_1	0.990	0.990	1.000
Ecoli-0-1-3-7_vs_2-6	0.789	0.789	0.797
Ecoli1	0.970	0.980	0.980
Ecoli2	0.920	0.920	0.940
Ecoli3	0.960	0.960	0.980
Ecoli4	0.900	0.880	0.890
Glass0	0.824	0.824	0.826
Glass0123vs456	0.960	0.960	0.980
Glass016vs2	0.790	0.791	0.791
Glass016vs5	0.970	0.980	0.990
Glass1	0.765	0.765	0.782
Glass2	0.842	0.842	0.842
Glass4	0.820	0.800	0.800
Glass5	0.970	0.980	1.000
Glass6	0.870	0.860	0.850
Haberman	0.760	0.762	0.772
Iris0	0.990	1.000	1.000
New-thyroid1	0.970	0.980	0.990
New-thyroid2	0.970	0.980	0.990
Page-blocks0	0.980	0.990	1.000
Page-blocks13vs2	0.990	1.000	1.000
Pima	0.793	0.793	0.793
Segmemt0	0.990	0.990	1.000
Shuttle0vs4	1.000	1.000	1.000
Shuttle2vs4	0.990	1.000	1.000
Vehicle0	0.970	0.980	0.990
Vehicle1	0.767	0.767	0.768
Vehicle2	0.980	0.990	1.000
Vehicle3	0.803	0.804	0.806
Vowel0	0.970	0.980	0.980
Wisconsin	0.970	0.980	0.990
Yeast05679vs4	0.843	0.843	0.843
Yeast1	0.813	0.813	0.815
Yeast1289vs7	0.770	0.770	0.782
Yeast1458vs7	0.762	0.763	0.781
Yeast1vs7	0.787	0.788	0.812
Yeast2vs4	0.950	0.940	0.940
Yeast2vs8	0.851	0.851	0.851
Yeast3	0.970	0.980	0.990
Yeast4	0.880	0.860	0.890
Yeast5	0.980	0.990	1.000
Yeast6	0.820	0.800	0.810
Protein homology prediction	0.970	0.980	0.993
Twitter-sentiment	0.988	0.994	1.000
Average	0.897	0.898	0.906

applications, including medical diagnosis, malware detection, anomaly identification, bankruptcy prediction, and spam filtering. In order to build efficient and robust classification schemes, data preprocessing methods can be utilized in conjunction with supervised learning methods. Undersampling- and oversampling-based methods can be successfully utilized for class imbalance. However, the identification of informative instances to be included in the training set is a critical issue for undersampling. In this

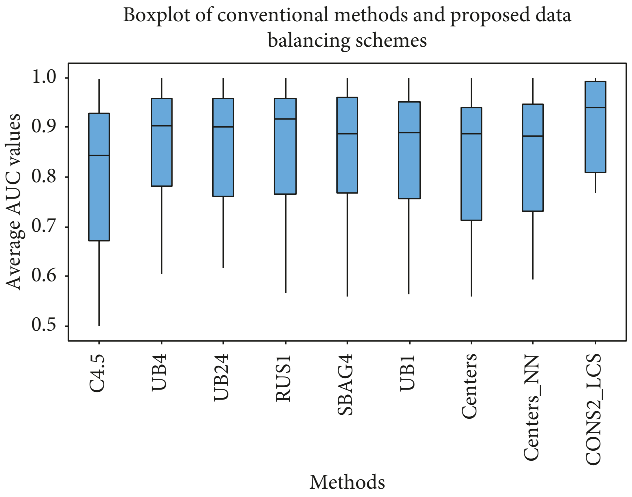


Figure 4: Boxplot distributions of AUC values for conventional data balancing methods and the proposed scheme.

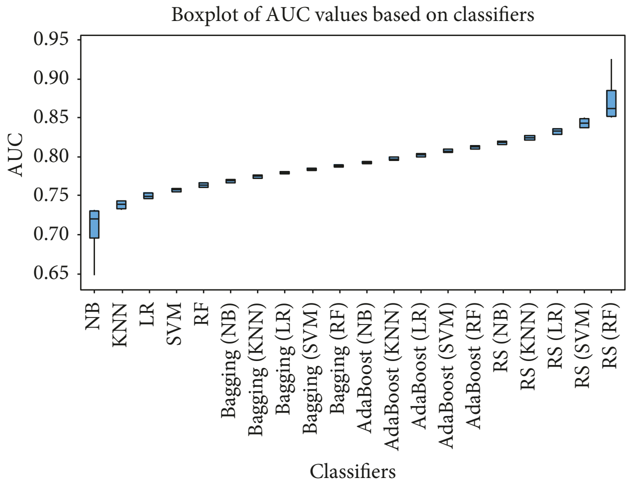


Figure 5: Boxplot distributions of AUC values for supervised learning methods and ensemble methods.

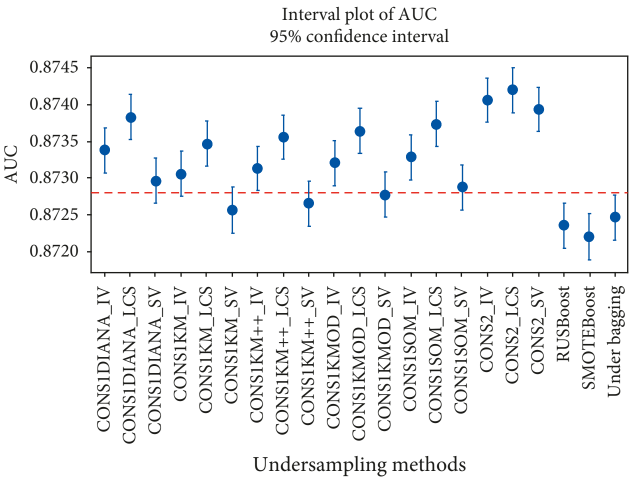


Figure 6: Interval plots for the compared algorithms.

regard, this paper empirically examines the predictive performance of two consensus clustering-based undersampling schemes for imbalanced learning. In the empirical analysis, 44 small-scale and 2 large-scale imbalanced classification benchmarks (with imbalance ratios ranged between 1.8 and 163.19) were utilized. The experimental analysis indicates that clustering-based undersampling schemes can outperform conventional data-level preprocessing methods for class imbalance. In addition, consensus clustering, which aggregates the partitions of individual clustering algorithms, can further enhance the predictive performance of clustering-based undersampling schemes.

There are a number of issues that should be beneficial to extend in the future. The presented consensus clustering based undersampling scheme utilizes five clustering algorithms (namely, *k*-means, *k*-modes, *k*-means++, self-organizing maps, and DIANA algorithm). The clustering algorithms have been integrated with the use of three consensus functions, namely, simple voting-based consensus function, incremental voting function, and label correspondence search. Hence, the predictive performance of other conventional and swarm-based clustering algorithms (such as ant clustering, particle swarm-based clustering, firefly clustering) can be examined for imbalanced learning. In addition, recent proposals on the field indicate that imbalancing schemes which integrate instance selection and clustering may yield higher predictive performance. Hence, the performance of consensus clustering-based undersampling scheme should be taken into consideration in conjunction with conventional instance selection methods.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

The study was performed as part of the employment of the author at Izmir Katip Celebi University.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220–239, 2017.

[2] 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," *Information Sciences*, vol. 250, pp. 113–141, 2013.

[3] G. M. Weiss, "Mining with rarity," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 7–19, 2004.

[4] C. Beyan and R. Fisher, "Classifying imbalanced data sets using similarity based hierarchical decomposition," *Pattern Recognition*, vol. 48, no. 5, pp. 1653–1672, 2015.

[5] M. Denil and T. Trappenberg, "Overlap versus imbalance," in *Proceedings of Canadian Conference on Artificial Intelligence*, pp. 220–231, Springer, Ottawa, Canada, May 2010.

[6] D. Rodriguez, I. Herraiz, R. Harrison, J. Dolado, and J. C. Riquelme, "Preliminary comparison of techniques for dealing with imbalance in software defect prediction," in *Proceedings of 18th International Conference on Evaluation and Assessment in Software Engineering*, p. 43, ACM, London, UK, May 2014.

[7] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Proceedings of European Conference on Machine Learning ECML 2004*, pp. 39–50, Prague, Czech Republic, September 2004.

[8] N. Peiravian and X. Zhu, "Machine learning for android malware detection using permission and api calls," in *Proceedings of IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 300–305, IEEE, Herndon, VA, USA, November 2013.

[9] W. Khreich, E. Granger, A. Miri, and R. Sabourin, "Iterative Boolean combination of classifiers in the ROC space: an application to anomaly detection with HMMs," *Pattern Recognition*, vol. 43, no. 8, pp. 2732–2752, 2010.

[10] M.-J. Kim, D.-K. Kang, and H. B. Kim, "Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1074–1082, 2015.

[11] T. R. Hoens, R. Polikar, and N. V. Chawla, "Learning from streaming data with concept drift and imbalance: an overview," *Progress in Artificial Intelligence*, vol. 1, no. 1, pp. 89–101, 2012.

[12] 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 Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2012.

[13] B. Liu, Y. Ma, and C. K. Wong, "Improving an association rule based classifier," in *Proceedings of European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 504–509, Springer, Lyon, France, September 2000.

[14] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[15] N. V. Chawla, D. A. Cieslak, L. O. Hall, and A. Joshi, "Automatically countering imbalance and its empirical relationship to cost," *Data Mining and Knowledge Discovery*, vol. 17, no. 2, pp. 225–252, 2008.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority oversampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[17] N. Japkowicz, "The class imbalance problem: significance and strategies," in *Proceedings of International Conference on Artificial Intelligence*, Las Vegas, NV, USA, June 2000.

[18] R. Barandela, R. M. Valdovinos, and J. S. Sánchez, "New applications of ensembles of classifiers," *Pattern Analysis and Applications*, vol. 6, no. 3, pp. 245–256, 2003.

[19] N. V. Chawla, N. Japkowicz, and A. Kolcz, "Workshop learning from imbalanced data sets II," in *Proceedings of International Conference on Machine Learning*, Washington, DC, USA, August 2003.

[20] S. Wang and X. Yao, "Diversity analysis on imbalanced data sets by using ensemble models," in *Proceedings of IEEE Symposium on Computational Intelligence and Data Mining*

CIDM'09, pp. 324–331, IEEE, Nashville, TN, USA, March 2009.

[21] J. Błaszczyński and J. Stefanowski, "Neighbourhood sampling in bagging for imbalanced data," *Neurocomputing*, vol. 150, pp. 529–542, 2015.

[22] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, and Y. Zhou, "A novel ensemble method for classifying imbalanced data," *Pattern Recognition*, vol. 48, no. 5, pp. 1623–1637, 2015.

[23] J. Kwak, T. Lee, and C. O. Kim, "An incremental clustering-based fault detection algorithm for class-imbalanced process data," *IEEE Transactions on Semiconductor Manufacturing*, vol. 28, no. 3, pp. 318–328, 2015.

[24] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Information Sciences*, vol. 409-410, pp. 17–26, 2017.

[25] V. Vigneron and H. Chen, "A multi-scale seriation algorithm for clustering sparse imbalanced data: application to spike sorting," *Pattern Analysis and Applications*, vol. 19, no. 4, pp. 885–903, 2016.

[26] D. H. Wolpert and W. G. Macready, "No free lunch theorems for search," vol. 10, Santa Fe Institute, Santa Fe, NM, USA, 1995, Technical Report SFI-TR-95-02-010.

[27] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: a hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, 2010.

[28] S. Wang, K. Tang, and X. Yao, "Diversity exploration and negative correlation learning on imbalanced data sets," in *Proceedings of International Joint Conference on Neural Networks, IJCNN 2009*, pp. 3259–3266, IEEE, Atlanta, GA, USA, June 2009.

[29] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: improving prediction of the minority class in boosting," in *Proceedings of European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 107–119, Springer, Cavtat-Dubrovnik, Croatia, September 2003.

[30] Z. Huang, "A fast clustering algorithm to cluster very large categorical data sets in data mining," *DMKD*, vol. 3, no. 8, pp. 34–39, 1997.

[31] D. Arthur and S. Vassilvitskii, "*k*-means++: the advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, January 2007.

[32] T. Kohonen, *Self-Organising Maps Berlin*, Springer, Berlin, Germany, 2001.

[33] H. Chipman and R. Tibshirani, "Hybrid hierarchical clustering with applications to microarray data," *Biostatistics*, vol. 7, no. 2, pp. 286–301, 2005.

[34] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE—majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 405–425, 2014.

[35] A. Anand, G. Pugalenthi, G. B. Fogel, and P. N. Suganthan, "An approach for classification of highly imbalanced data using weighting and undersampling," *Amino Acids*, vol. 39, no. 5, pp. 1385–1391, 2010.

[36] Q. Li, B. Yang, Y. Li, N. Deng, and L. Jing, "Constructing support vector machine ensemble with segmentation for imbalanced datasets," *Neural Computing and Applications*, vol. 22, no. S1, pp. 249–256, 2013.

[37] N. S. Kumar, K. N. Rao, A. Govardhan, K. S. Reddy, and A. M. Mahmood, "Undersampled K-means approach for handling imbalanced distributed data," *Progress in Artificial Intelligence*, vol. 3, no. 1, pp. 29–38, 2014.

[38] A. D'Addabbo and R. Maglietta, "Parallel selective sampling method for imbalanced and large data classification," *Pattern Recognition Letters*, vol. 62, pp. 61–67, 2015.

[39] J. Ha and J. S. Lee, "A new under-sampling method using genetic algorithm for imbalanced data classification," in *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication*, p. 95, January 2016.

[40] G. Shobana and B. P. Battula, "An under sampled *k*-means approach for handlingimbalanced data using diversified distribution," *International Journal of Engineering and Technology (UAE)*, vol. 7, no. 1.8, pp. 113–117, 2018.

[41] H. Guo and T. Wei, "Logistic regression for imbalanced learning based on clustering," *International Journal of Computational Science and Engineering*, vol. 18, no. 1, pp. 54–64, 2019.

[42] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on *k*-means and SMOTE," *Information Sciences*, vol. 465, pp. 1–20, 2018.

[43] W. Han, Z. Huang, S. Li, and Y. Jia, "Distribution-sensitive unbalanced data oversampling method for medical diagnosis," *Journal of medical Systems*, vol. 43, no. 2, p. 39, 2019.

[44] 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," *Information Sciences*, vol. 477, pp. 47–54, 2019.

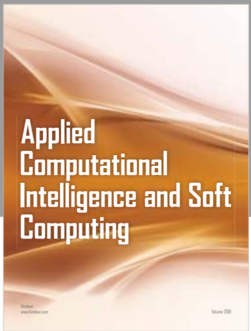
[45] T. Boongoen and N. Iam-On, "Cluster ensembles: a survey of approaches with recent extensions and applications," *Computer Science Review*, vol. 28, pp. 1–25, 2018.

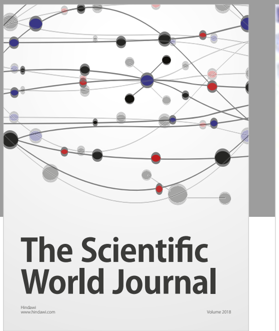
[46] N. Nguyen and R. Caruana, "Consensus clusterings," in *Proceedings of Seventh IEEE International Conference on Data Mining ICDM 2007*, pp. 607–612, IEEE, Omaha, NE, USA, October 2007.

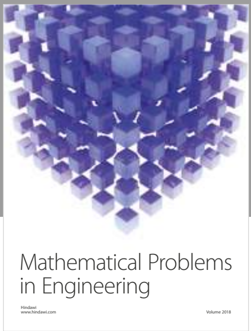
[47] A. P. Topchy, M. H. Law, A. K. Jain, and A. L. Fred, "Analysis of consensus partition in cluster ensemble," in *Proceedings of Fourth IEEE International Conference on Data Mining ICDM'04*, pp. 225–232, IEEE, Brighton, UK, November 2004.

[48] H. G. Ayad and M. S. Kamel, "Cumulative voting consensus method for partitions with variable number of clusters," *IEEE Transactions on Pattern Analysis and Machine Intellzigence*, vol. 30, no. 1, pp. 160–173, 2008.

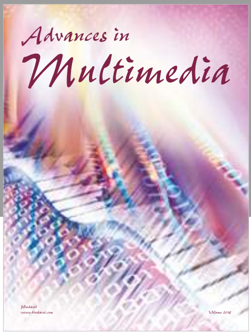
[49] C. Boulis and M. Ostendorf, "Combining multiple clustering systems," in *Proceedings of European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 63–74, Springer, Cavtat-Dubrovnik, Croatia, September 2003.

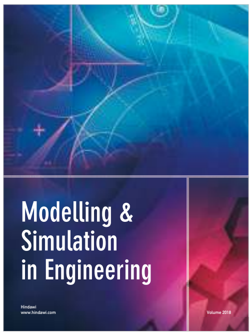

Applied Computational Intelligence and Soft Computing


The Scientific World Journal

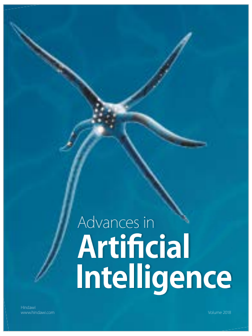

Mathematical Problems in Engineering

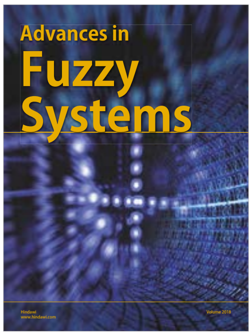
Journal of Engineering

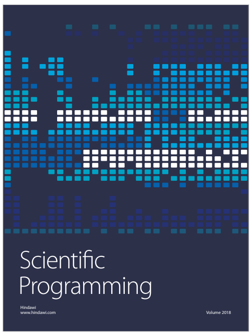

Advances in Multimedia


Modelling & Simulation in Engineering

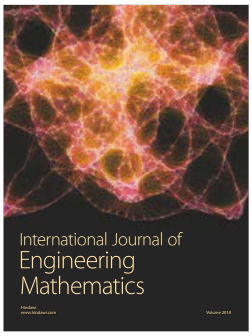
International Journal of Reconfigurable Computing

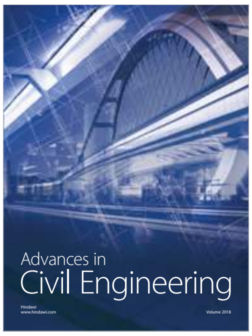

Advances in Artificial Intelligence

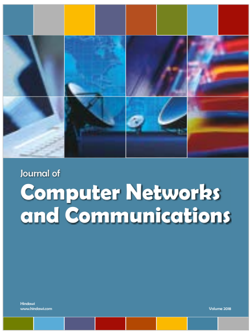

Advances in Fuzzy Systems


Scientific Programming

Advances in Human-Computer Interaction


International Journal of Engineering Mathematics


Advances in Civil Engineering

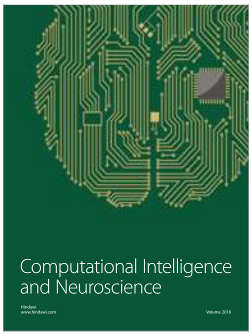

Journal of Computer Networks and Communications

Journal of Robotics

International Journal of Computer Games Technology

International Journal of Biomedical Imaging

Journal of Electrical and Computer Engineering


Computational Intelligence and Neuroscience