Journal of Big Data

RESEARCH Open Access

A multi-manifold learning based instance weighting and under-sampling for imbalanced data classification problems

Tayyebe Feizi[1], Mohammad Hossein Moattar[1*] and Hamid Tabatabaee[1]

*Correspondence:
moattar@mshdiau.ac.ir

[1] Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

Abstract

Under-sampling is a technique to overcome imbalanced class problem, however, selecting the instances to be dropped and measuring their informativeness is an important concern. This paper tries to bring up a new point of view in this regard and exploit the structure of data to decide on the importance of the data points. For this purpose, a multi-manifold learning approach is proposed. Manifolds represent the underlying structures of data and can help extract the latent space for data distribution. However, there is no evidence that we can rely on a single manifold to extract the local neighborhood of the dataset. Therefore, this paper proposes an ensemble of manifold learning approaches and evaluates each manifold based on an information loss-based heuristic. Having computed the optimality score of each manifold, the centrality and marginality degrees of samples are computed on the manifolds and weighted by the corresponding score. A gradual elimination approach is proposed, which tries to balance the classes while avoiding a drop in the F measure on the validation dataset. The proposed method is evaluated on 22 imbalanced datasets from the KEEL and UCI repositories with different classification measures. The results of the experiments demonstrate that the proposed approach is more effective than other similar approaches and is far better than the previous approaches, especially when the imbalance ratio is very high.

Keywords: Imbalanced data, Classification, Under-sampling, Multi-Manifold learning

Introduction

Imbalanced learning is one of the main challenges of classification in real-world problems. This challenge occurs when the number of examples from one class (called the majority class) is greater than the number of samples from the other class (called the minority class). Imbalance problem may be inevitable and happens when it is difficult to collect minority class samples, and the majority class samples are more abundant [1, 2]. Classification problems such as fraud detection [3], image segmentation [4, 5], intrusion detection [6, 7], disease detection [8–10], etc. are mostly imbalanced. Dealing with this issue is challenging because traditional classification approaches have presuppositions. Their default is that the training samples are equally distributed among the classes.

Therefore, the majority class prevails over the minority class, and the minority examples are ignored. The inherent characteristics of imbalanced problems, such as overlapping and inseparability of classes [11, 12], increase the complexity of such data and weaken the classification performance [1, 2, 13].

Big data is the term used to describe datasets that are very huge, complicated, and contain a tremendous amount of information. In this kind of enormous dataset, the minority class may nevertheless be represented by a sizable number of examples. Because there are a lot of minority class samples even if they only make up a small fraction of the total dataset, handling imbalance becomes more difficult. In large data analytics, dealing with class imbalance becomes essential because disregarding it may result in biased model training. Large dataset processing and machine learning model training can be time- and resource-intensive. By balancing the classes, the training process can be improved, becoming more efficient and controllable.

Many studies have been done on the imbalanced data problem, and various techniques have been proposed. These techniques are divided into four main categories, which are algorithm-driven, cost-sensitive approaches, data-driven, and ensemble approaches. Algorithm-based methods try to adapt classifiers to imbalanced problems. These approaches try to modify the learning stage and accept the issue of imbalance in the data. In cost-sensitive methods, higher penalties are imposed for misclassifying minority samples, and these methods try to minimize the final penalty [14, 15].

Unlike other techniques, data-driven methods do not depend on classification and operate completely independently. These methods are usually done in the preprocessing stage. These methods benefit from two under- or over-sampling techniques or both and try to create a relative balance on imbalanced data. It seems that under-sampling techniques are more popular than over-sampling techniques because over-sampling techniques cause over-fitting. In ensemble approaches, several classifications are used simultaneously, and learning is done with the help of a voting technique or by combining the scores of the classifications. There is still a fundamental challenge with this type of approaches. The challenge here is how to combine the optimal classifications to increases the learning time [1, 2, 13].

Due to justifications such as applicability, generalizability and classifier independence, in this paper, an under-sampling approach is proposed. Two basic problems of under-sampling techniques can be pointed out, which are still a challenge among researchers. The main problems are: how many and which samples should be removed from the majority class? In this research, it is tried to overcome these two problems by introducing a new under-sampling method. The proposed method is based on the hypothesis that manifolds are the structures that can reflect the density and neighborhood properties of data. But since we are not sure that which manifold best suits a specific problem and dataset, a multi-manifold learning is proposed in this paper, which assesses the optimality of manifolds based on a proposed information loss-based heuristic.

The optimality indexes are used in a weighted combination of centrality and marginality criteria for the samples. The proposed approach is supposed to assign weights that determine the degree of importance of the samples from the majority class. A sequence of weights is created according to the relative importance of the samples. Then, the most insignificant samples are gradually removed from the majority class.

Finally, combining the most important data of the majority class with the samples of the minority class, the training dataset is created. The proposed method is evaluated on 22 imbalanced datasets from KEEL and UCI repositories. The main contributions of this research are summarized as follows:

- The samples of the majority class are weighted according to the multi-manifold approach, based on a weighted combination of the centrality and marginality of the samples on each manifold.
- The weights are sorted in descending order. Less important samples are gradually removed from the majority class. The remaining samples can largely represent the distribution of the data.

The proposed approach reduces the overlap of minority and majority classes and increases class separability, which causes better classification performance. A simplified graphical abstract of the proposed method is shown in Fig. 1.

The rest of the paper is organized as follows: The next section gives a brief overview of under-sampling methods and related works. "Definitions and background" section introduces some definitions and the required background for better understanding the proposed approach. "Proposed method" section explains the proposed method in detail. The experimental results and discussions are in "Experimental Results and

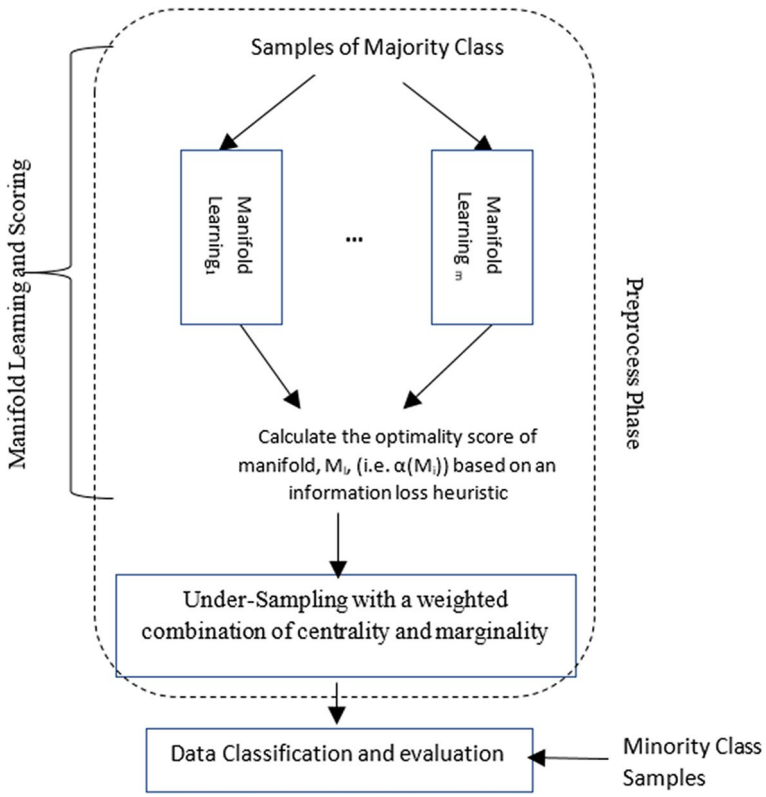

Fig. 1 A simplified flowchart of the proposed multi-manifold approach for under-sampling

discussion" section. Finally, in the last section, the conclusions and the future research directions are discussed.

Related works

Under-sampling and over-sampling methods are performed in the preprocessing stage. In oversampling methods, minority samples increase. In these methods, new samples are usually created around the minority samples, or the minority samples are repeated again. Increasing the sample can cause an overfitting problem [16].

Researchers introduced ensemble approaches based on bagging, such as the Over-bagging method [17]. Various versions of the SMOTE technique, which is an oversampling method, are presented [18]. The SMOTE technique synthesizes a number of new samples with the help of k nearest neighbors by randomly choosing a neighbor among the minority samples for each sample. Along with its advantages, the SMOTE algorithm will include problems such as overgeneralization and variation in convergence [19]. Researchers presented the SMOTE-Bagging [17] and SMOTE-Boost [20] methods, which are a combination of SMOTE, bagging, and boosting. These methods have a high level of computational complexity, although they are significant in terms of performance. In addition, these methods cause overfitting and have many parameters to adjust [21, 22].

There are numerous under-sampling methods in the literature. Random under-sampling deals with the random removal of majority class samples. This may remove useful instances. This method is combined with ensemble approaches [23]. The under-bagging method is a combination of the ensemble method based on bagging and the random under-sampling method [24]. Seiffert et al. presented the RUS-Boost method, which is a combination of random under-sampling and boosting approaches [25]. Researchers proposed de-under-sampling methods called Near Miss, which consider the elimination of majority samples according to their distance from minority samples [26].

Under-sampling methods can be divided into two main groups, including methods based on KNN (k nearest neighbors) [27–29] and methods based on k-means [30–32]. Some under-sampling methods eliminate the majority of samples based on the information obtained from the nearest neighbors of the samples. The purpose of these methods is to remove samples that are located in marginal areas or are noisy and redundant. Kubat and Matwin [27] presented an under-sampling method called the One Side Selection method (OSS), which is one of the applications of Tomek links [33]. The samples in Tomek's links are considered marginal or noise samples. In the condensed nearest neighbor (CNN) method, if the label of a sample is the same as the label of its nearest neighbor (1NN), this sample is considered redundant [34]. In the OSS method, a large number of majority samples that are borderline, noisy, or redundant are removed. Removing a large number of samples reduces the performance of the classifier.

Laurikkala proposed the Neighborhood Cleaning Rule (NCL) to remove the samples of the majority class [28]. This method uses the Edited Nearest Neighbor (ENN) method [35] to eliminate the samples. In ref. [28], samples whose marginal score is more than two are eliminated. Also, samples are removed if one of the three nearest neighbors is from the minority class. In ref. [13], an under-sampling method is proposed that uses the density of the data to progressively remove data points from the majority class. Two

factors are proposed to measure the degree of importance degree of each instance. Furthermore, the optimal under-sampling level is determined progressively.

In addition to eliminating the majority samples, Kang and his associates [29] also eliminated the noise in the minority class. They separated the minority samples into three groups: noisy, informative, and relatively informative. As a result, the classifier's performance will improve after the noisy minority samples have been eliminated. The exclusion of minority samples from the algorithm could lead to the failure of the classifier, which makes figuring out the value of the parameter k crucial.

Yang et al. proposed an under-sampling method that uses the natural neighborhood graph (NaNG). With the help of this graph, they are able to classify the training samples into central, marginal, and noise samples. They are able to under-sample by removing noisy and redundant samples. They called their sample reduction method NNGIR. One of the strengths of their methods is that they are non-parametric, increasing the reduction rate and improving prediction accuracy. The disadvantages of their method are the dependence on parameters and relatively low accuracy [36]. Hamidzadeh et al. [37] presented the LMIRA under-sampling method. They removed the non-marginal samples and kept the marginal samples. They considered their method a constrained binary optimization problem and used the filled function algorithm to solve it.

Pang and his colleagues [38] introduced a new secure under-sampling method called SIR-KMTSVM. In ref. [38], most redundant samples are removed from both the majority and minority classes. One of the advantages of their method is its use for large-scale problems. The disadvantages include high computational complexity and the removal of informative samples. Hamidzadeh and colleagues [39] proposed an under-sampling method that solves the instance reduction problem as an unconstrained multi-objective optimization problem. They designed a weighted optimizer and searched for the appropriate samples with the help of chaotic krill-herd evolutionary algorithm. The advantage of the method is the improvement in accuracy, geometric mean, and calculation time. The main weakness of the method is that it can only be applied to normal-sized datasets.

Another common under-sampling technique is clustering, which helps to have a logical training set [40]. Researchers named Chen and Shiu [30] put the majority of samples in k different clusters and used the k-means clustering method. Then, by combining each cluster from the majority class with the minority samples, they created new data groups that are more balanced. Each data group is trained separately and builds a classification model. Finally, all models are aggregated together to predict new samples. The weakness of this algorithm is that it has not determined how the value of the parameter k is determined.

Yen and Lee [32] used clustering to propose their under-sampling method. They identified representatives of the majority class to create new training data. They first divided the entire training data into k clusters. They performed clustering based on a ratio of majority samples to minority data. The weakness of the algorithm is that it does not specify how to value the parameters k and m. Lin and his colleagues [31] presented a new under-sampling method that clusters the majority samples with the k-means method, where the value of k is equal to the number of minority samples.

In addition to the mentioned methods, some researchers [41–43] used k-means clustering before applying the under-sampling method to determine the type of the majority

sample in terms of noise, redundancy, or marginality. Hamidzadeh and his colleagues [44] introduced an under-sampling method based on hyper-rectangle clustering and called their method IRAHC. They removed the central samples and kept the marginal and near-marginal samples.

Huang and colleagues [45] introduced a neural network algorithm (NN_HIDC) for the classification problem of highly imbalanced data. They proposed the generalized gradient descent algorithm. This algorithm is used in re-sampling and re-weighting methods in neural network. They extended the locally controllable bound to reduce the insufficient empirical representation of the positive class. The advantages of this algorithm can be mentioned in its use for any very imbalanced data, and the weaknesses of the algorithm are that the extended gradient of the positive class can only reach the local border and gradient measurement is required for all samples in each iteration. Koziarski [46] introduced a radial under-sampling (RBU) method in the classification problem of imbalanced data. RBU uses the concept of mutual class potential in the under-sampling method. The advantages of the method include reducing the time complexity compared to RBO, being effective on the difficult minority class that includes small disjunct, outliers and small number of samples, overcoming the limitations of neighborhood-based methods.

Sun et al. [47] introduced a radial under-sampling approach with adaptive under-sampling ratio determination. They called their algorithm RBU-AR. This method determines the appropriate under-sampling ratio according to the complexity of the class overlap and does not use the default value of one or trial and error. The advantages of their approach are better performance in high overlap. The weakness of their approach is the lack of application in multi-class problems. Mayabadi and colleagues [48] proposed two density-based algorithms to remove overlap, noise and balance between classes. The first algorithm uses under-sampling technique and the second algorithm uses under-sampling and over-sampling simultaneously. Their method removes high density samples from the majority class. The advantages of their algorithms are maintaining the class structure as much as possible and improving performance.

Vuttipittayamongkol and Elyan [49] proposed an under-sampling method to solve binary data imbalance problem by removing overlapping data. They focused on the detection and elimination of overlapping majority samples. The advantages of their algorithm are preventing information loss and improving the sensitivity criterion. The weak points of the algorithm are how to set the value of k in the k-NN law and the failure to examine multi-class problems. Nwe and colleagues [50] introduced an effective under-sampling method using k-nearest-neighbor-based overlapping samples filter to classify imbalanced and overlapping data. The advantage of their algorithm is to prevent information loss. The weak points of their algorithm are setting the value of k, not checking high dimensions and not checking multi-class problems.

Zhai and colleagues [51] proposed two diversity over-sampling methods, BIDC1 and BIDC2, which were based on generative models. BIDC1 and BIDC2 methods use extreme machine autoencoder and generative adversarial network, respectively. Among the advantages of their methods, we can mention the simple but effective idea, improving performance in data with low and high imbalance ratio, suitable for different practical scenarios, creating variety in over-sampling and preventing overlap of classes.

The weaknesses of their method are the lack of scalability in big data and the difference between the original and generated data distribution. Table 1 summarizes of the strengths and weaknesses of the some of the main approaches in this regard.

Definitions and background

For better explanation of the proposed approach, in this section some primary definitions and backgrounds are explained.

Definitions

Manifold: Manifold refers to any process, curve, or complex nonlinear shape. In fact, in the manifold learning method, the system's intrinsic parameters are identified, and the entire data set is placed on a manifold that expresses the intrinsic relationship between the data in a space with less dimension.

Multi-manifold learning: In pattern recognition, we often encounter situations where the data set is not on a manifold. In other words, if the dataset has several classes, the data for each class will have a separate manifold.

Traditional Degree of centrality: If the data is in the center of the class in such a way that its label is the same as the label of its K_c nearest neighbors, then it has a degree of centrality. The degree of centrality of a sample is greater when the number of neighbors with the same label is greater than the number of neighbors with the opposite label, or most of its neighbors are of the same class.

Traditional Degree of marginality: If a data point is on the edge or border of the class and its label is not the same as all the neighboring samples or some of its K_m nearest neighbors, then the degree of marginality can be considered for this data point. The degree of marginality of a sample is higher when the number of neighbors from the opposite class is greater than the number of neighbors from the same class.

Manifold learning

The purpose of manifold learning algorithms is to map a set of data with high dimensions to a set of data with smaller dimensions in such a way that the distance between samples in the lower dimensional subspace is close to the distance between samples in the original space. Assume that x_i is data in a high-dimensional space, and data set $X = (x_1, x_2, \ldots, x_n) \in R^{n \times D}$ represents n data in a space with D dimensions. Manifold learning methods seek to represent this set of data in a space with lower dimensions, d, which is much lower than the dimensions of the data in the original representation space, i.e. $d << D$. Supposing y_i as a data in the lower dimensional space, the corresponding data set in the low-dimensional space can be expressed as $Y = \{y_1, y_2, \ldots, y_n\} \in R^{d \times n}$. In this way, manifold learning is a process that calculates Y while maintaining the inherent connection of data, in such a way that the manifold resulting from Y in a low-dimensional space is the most similar to the manifold resulting from X in a high-dimensional space.

Manifold learning approaches are divided into different aspects of view but here, since we have a concern for computational complexity, we have limited the employed algorithms to linear unsupervised manifold learning methods. Among the main characteristics of linear manifold learning methods that make them appropriate for the approach,

Table 1 Summary of the strengths and weaknesses of the most important reviewed articles

Refs.	The proposed framework	Strengths	Weaknesses
[16]	A review of hybrid models in unbalanced data problem: approaches based on bagging, boosting and hybrid	- Increasing accuracy and precision - Improved performance	- Increasing complexity - Failure to examine multi-class issues
[21]	Application of automatic enhanced twin support vector machine for imbalanced data classification	- Better classifier performance - Less training time	- High computational complexity - Setting many parameters
[22]	Cost-sensitive multi-variate decision tree with hybrid feature measure on unbalanced data	- Performance improvement - Reducing the cost of misclassification	- Increasing the complexity - Setting many parameters
[23]	A new hybrid method for classification of imbalanced data	- Improved performance - Very unbalanced data fit	- Delete useful information - Wrong classification - Data distribution change - Increasing complexity
[29]	An under-sampling method with noise filtering for imbalanced data classification	- Performance improvement - Improvement of AUC, F-measure and G-means - Insensitivity to minority class noise	- Failure to build a learning model by removing the minority sample - Sensitive to the imbalance coefficient - Lack of efficiency in highly unbalanced data
[30]	Clustering-Based under-sampling for Imbalanced Data	- Runtime improvements - Data preprocessing - Better performance	- Remove useful examples - Determining the number of clusters
[36]	Parameter-free under-sampling algorithm based on natural neighborhood graph	- Being non-parametric - Increased reduction rate - Improvement of prediction accuracy	- Dependence on parameters - Relatively low accuracy
[37]	LMIRA: Large Margin Sample Reduction Algorithm	- Increase accuracy - Increasing the reduction rate	- Removing samples with information - Random selection of samples
[38]	A new secure under-sampling method called SIR-KMTSVM	- Increasing computing power - Speeding up the execution of the algorithm - Reduction of calculation time - Application in large-scale problems - Maintaining acceptable accuracy	- High computational complexity - Removal of informative examples
[39]	Unconstrained weighted multi-objective optimizer for under-sampling in binary imbalanced data problem	- Improved accuracy - Improved G-means - Improved calculation time - Effective on noise data	- Lack of efficiency by increasing the number of features - Lack of efficiency with increasing samples
[40]	Automatic clustering-based under-sampling for imbalanced data classification	- Improve accuracy - Improved performance - Increased stability	- Increasing complexity - Removal of informative examples
[41]	Fast-CBUS: a clustering-based under-sampling method to solve the imbalance problem	- Improved performance - Increasing the speed of prediction - Reducing time complexity	- Increasing computational complexity
[43]	Diverse sensitivity-based under-sampling for class imbalance	- Attention to data distribution - Increasing diversity in sampling - Increasing the sensitivity criterion	- Increasing computational complexity - Removal of informative examples
[44]	IRAHC: an under-sampling method based on hyper-rectangular clustering	- Increase accuracy - Increased reduction rate	- Delete examples with information - Random selection of samples

Table 1 (continued)

Refs.	The proposed framework	Strengths	Weaknesses
[45]	A neural network algorithm for highly unbalanced data classification problem	- Use for any very unbalanced data	- The extended gradient of the positive class can only reach the local edge - Gradient measurement is required for all samples in each repetition
[46]	Radial under-sampling method in unbalanced data classification problem	- It is effective on the difficult minority class - Overcoming the limitations of neighborhood-based methods	
[47]	Radial under-sampling approach by determining adaptive under-sampling ratio	- Better performance in high overlap	- Not applicable in multi-class problems
[48]	Two density-based sampling approaches for overlapping and unbalanced data problem	- Maintaining the class structure as much as possible - Improved performance	
[49]	A neighborhood-based under-sampling approach to solve the problem of unbalanced and overlapping data	- Prevent data loss - Improve the sensitivity criterion	- How to set the value of k in the k-NN law - Failure to examine multi-class issues
[50]	Overlapping samples filter method based on k-nearest neighbor to solve the imbalanced data problem	- Preventing information loss	- Setting the value of k - Failure to check the high dimensions - Failure to examine multi-class issues
[51]	Diversity over-sampling by generative models for unbalanced binary data classification	- Simple but effective idea - Diversity in prototyping - Improved performance in data with low and high imbalance ratio - Suitable for various practical scenarios	- Lack of scalability in big data - Difference between original and generated data distribution

is their out-of-sample mapping property. It means that they can map the test data to a low-dimensional space using the mapping matrix obtained from the training data. In the following, the manifold learning methods included in the proposed approach are briefly introduced.

Principal component analysis (PCA)

Principal component analysis is one of the most common global and linear methods of manifold learning and dimensionality reduction. The main idea of PCA is to find the linear subspace in the low-dimensional space that best fits the scatter of the data in the high-dimensional space. By defining the covariance matrix of the data in the high-dimensional space, Cov(X), and due to the non-negativity and symmetry of the covariance matrix we have:

$$\mathrm{Cov(x)} = \mathrm{U}_{PCA}\mathrm{D}\mathrm{U}_{PCA}{}^{\mathrm{T}} \tag{1}$$

In which $\mathrm{U}_{PCA} \in R^{D\times D}$ is an orthogonal identity matrix ($\mathrm{U}_{PCA}{}^{T}\mathrm{U}_{PCA} = I$) containing eigenvectors of Cov(X) and D is a diagonal matrix containing eigenvalues. Assuming U_{PCA}=[u_1,u_2, ..., u_d] as the matrix of eigenvectors corresponding to eigenvalues

$0 \leq \lambda_d \leq \lambda_{d-1} \leq \cdots \leq \lambda_1$, it is proved that λ_i represents the data scatter after a linear mapping by U_{PCA}. As a result, data in the lower dimensional subspace is as follows

$$Y = U_{PCA}^T X \tag{2}$$

Neighborhood preserving embedding (NPE)

The NPE algorithm is one of the popular local methods in manifold learning. This algorithm includes three steps. The first step is to determine the neighbors of each data point. The second step is to form the neighborhood graph matrix, W, and the third step is to calculate the transformation matrix, U_{NPE}, using W, after solving the following convex optimization problem.

$$\min trace\left(U_{NPE}{}^T X M X^T U_{NPE}\right) s.t. U_{NPE}{}^T X X^T U_{NPE} = I \tag{3}$$

where $M = (I_N - W)^T (I_N - W)$. After finding the optimal solutions for U_{NPE}, any data point x can be linearly mapped to the new subspace y using $y = U_{NPE}^T x$.

Locality preserving projection (LPP)

LPP manifold learning is a local method that, again includes the three main steps of neighbor finding, graph formation, and embedded data extraction. Determining the neighbor and how to form the LPP manifold graph are completely the same as the other local manifold learning methods and it differs from the other methods only in data extraction step. In fact, LPP manifold learning is a linear learning method in which the data mapping matrix from high-dimensional space to low-dimensional space is obtained from Eq. (4):

$$U_{LPP} = \underset{U}{\operatorname{argmin}} U_{LPP}^T X L X^T U_{LPP}, \text{s.t.} U_{LPP}^T X O X^T U_{LPP} = I \tag{4}$$

where U_{LPP} is the mapping matrix, L = O-W, W is the local manifold graph and O is the diagonal matrix with diagonal elements equal to $\sum_j w_{ij}$. In this method, the mapping matrix can be calculated as an eigenvalue problem. After calculating the mapping matrix, the data representation in the low-dimensional space will be $Y = U_{LPP}^T X$.

Proposed method

The logic of the proposed approach

As discussed, every subspace of data can be expressed by a manifold. The problem is that it is not possible to find out which manifolds the data points obey in each subspace or which manifolds the distribution of the data sample is based on. On the other hand, the data structure may be so complex that a concrete manifold is not appropriate.

Therefore, we use an alternative method. Instead of having multiple manifolds where each manifold represents a part of the data, we choose multiple manifolds to represent all the data samples, but according to the weight or the degree of suitability in maintaining the local neighborhood structure, optimality weights are assigned to each manifold. Consider Fig. 2. The graph in which we have a series of data points is the red dotted curve, and we don't know what their structure is. Instead of

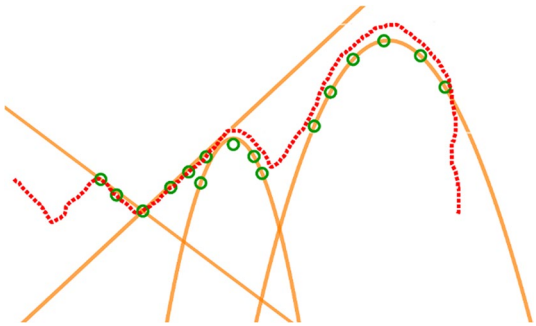

Fig. 2 The logic of the proposed multi-manifold learning method based on the weighting of linear manifolds

considering a complex non-linear function (manifold) for this data, we combine several linear functions (orange curves) and assign a combination weight to each manifold based on the degree of similarity to the structure of the whole data. We consider this linear combination of simpler manifolds as a suitable approximation of a more complex function.

In this paper, data importance weighting considers novel measures of the marginality and centrality of data on the data manifolds and then scores the data points based on a weighted combination of these measures. The algorithm gradually removes the data samples from the majority class until a definite termination condition is met. To measure and optimize the manifolds, a distance-based information loss heuristic is proposed.

In the proposed method, different manifolds of data are extracted and used to select the neighbors of the sample that belong to the majority class. For this purpose, several manifold learning approaches, namely principal component analysis (PCA), neighborhood preserving embedding (NPE), and locality preserving projection (LPP), are applied.

Mapped majority class data, X^N, on the extracted manifolds are denoted by Y^N. The three manifolds are trained in parallel. For each of the mentioned manifolds, M_i, a coefficient $\alpha(M_i)$ is calculated, which indicates the optimality of the manifold. Instance weighting is done based on two criteria of centrality and marginality on the extracted manifolds, separately. The final centrality and marginality criterion for the sample x_i is obtained from the $\alpha(M_i)$ weighted combination of centralities and marginalities obtained on each manifold M_i. In other words, $Centrality(x_i, M_i)$, which expresses the centrality degree of x_i on manifold M_i, and $Marginality(x_i, M_i)$, which expresses the marginality score of x_i on M_i, are weighted by the parameter $\alpha(M_i)$, to construct the final score. Then the samples are sorted based on their centrality and marginality degrees and the unnecessary samples are excluded using an iterative strategy. The following sections, explain the approach in more detail.

Multi-manifold learning approach

Assume that $X = \left(x_1^l, x_2^l, \ldots, x_n^l\right) \in R^{n \times D}$ refers to a set of n data points in a space with dimension D, where $l(x_i)$ is the class label of x_i and $i = \{1, 2, \ldots, n\}$. As stated before, a subset X^N from X which correspond to the larger class, is considered a the majority samples. In the multi-manifold learning approach, several manifolds are trained on X^N. A coefficient

$\alpha(M_i)$ is calculated for each manifold, M_i, which aims to indicate the optimality of that manifold for X^N.

In the initial experiments, supervised nonlinear manifold learning methods including Neighborhood Component Analysis (NCA), Maximally Collapsing Metric Learning (MCML) and Large-Margin Nearest Neighbor Metric Learning (LMNN) were used in the proposed approach, but due to the higher complexity and more execution time, the continuation of experiments with these manifolds was abandoned. The execution time of supervised manifolds, including NCA, MCML, and LMNN, increases greatly when the number of dataset samples is close to or more than 1000. Therefore, in this paper, unsupervised manifold learning approaches such as PCA, NPE, and LPP are investigated.

Manifolds optimality determination

In the second step, the manifolds of the majority class are assessed to see if they fit the neighborhood structure of the class. The goal is to give higher score to the manifolds that best fit the data of the majority class. The idea for this manifold weighting is simple. After the mapping of the original data, X^N, there will be a distance between the original data and the mapped samples, Y^N. Here, an information loss criterion which is denoted as the distances between the initial data points and their mappings is used according to Eqs. (5) and (6) to score the manifolds.

Set of new data points Y^N in the latent space is obtained by mapping the majority samples X^N onto the manifold M_i. The set of Y^N will be obtained using a linear transformation like $Y^N = UX^N$, where U is the mapping matrix. Then, mapping distance, i.e., the distance between the points of X^N and their corresponding latent representation Y^N, is calculated for each manifold M_i according to Eq. (5). The smaller the distance, the better is the manifold. Suppose that the number of data samples equal n_c. If the sum of distances is divided by the number of samples, and the average distance is obtained. Each manifold M_i can be weighted according to the inverse value of the average distances according to Eq. (5). The higher the value of $\alpha(M_i)$, the better this manifold has preserved the neighborhood structure of the data.

$$InfoLost(M_i) = \frac{1}{n_c}\sum_{\forall(x_i, U^{-1}y_i)} d^2(x_i, y_i U^{-1}) = \frac{1}{n_c}\|x_i - y_i U^{-1}\|_2^2, \quad M_i \in \{PCA, NPE, LPP\}, \tag{5}$$

$$\alpha_i = \alpha(M_i) = 1/InfoLost(M_i) i \in \{1, 2, 3\} \tag{6}$$

Weighted combination of centrality and marginality in the multi-manifold approach

Instance selection is based on two criteria of centrality and marginality. The combinatorial criteria of centrality and marginality for each data sample x_i^N is calculated based on Eqs. (7) and (8). Equation (7) denotes the degree of centrality for sample x_i^N which is obtained from the weighted combination of centralities of the data point over the learned manifolds (i.e. PCA, NPE and LPP). Equation (8) shows the marginality degree for sample x_i^N which is obtained from the weighted combination of marginalities obtained over the mentioned manifolds.

$$Cent\left(x_i^N\right) = \alpha(PCA) * Centrality(x_i^N, PCA) + \alpha(LPP) * Centrality(x_i^N, LPP) + \alpha(NPE) * Centrality(x_i^N, NPE) \tag{7}$$

$$Marg\left(x_i^N\right) = \alpha(PCA) * Marginality(x_i^N, PCA) + \alpha(LPP) * Marginality(x_i^N, LPP) + \alpha(NPE) * Marginality\left(x_i^N, NPE\right) \tag{8}$$

Gradual under-sampling of data

In the sample reduction stage, first the marginal samples which may be outliers or noise samples, and then the central samples are gradually removed from the majority class with a specific reduction step. The relation for calculating the weight of each sample from the majority class can be written according to Eq. (9). This relationship means that the coefficient of sample x_i^N is obtained from the linear combination of centrality and marginality degrees.

$$W\left(x_i^N\right) = Marg\left(x_i^N\right) - Cent\left(x_i^N\right) \tag{9}$$

$$\text{if } W\left(x_i^N\right) < 0 x_i^N \textit{ is a central sample} \tag{10}$$

$$\text{if } W\left(x_i^N\right) > 0 x_i^N \textit{ is a marginal sample} \tag{11}$$

$$\text{f } W\left(x_i^N\right) = 0 x_i^N \textit{ is a noise sample} \tag{12}$$

After calculating the weight for all samples in the majority class, a sequence of weights is created. The sequence of the weights is sorted in descending order and gradually remove the majority samples with a specific step. A high value for the sample weight, $W\left(x_i^N\right)$, means that the sample tends to be an outlier and is a good choice to be removed. By removing a portion of the data (i.e., 5 or 10 percent in the experiments) as marginal data, the overlapping of the majority and minority classes will decrease. On the other hand, by removing marginal samples from the majority class, it helps to better separate the majority and minority classes. This process continues until the size of the minority and majority classes is equal or the F-measure on the validation set starts reducing. Figure 3 shows the algorithm of the proposed method. Figure 4 shows the flowchart of the proposed method along with all the calculation steps.

Experimental results and discussion

In this section, many experiments have been conducted with the aim of comparing the proposed multi-manifold approach with other methods. For example, the proposed multi-manifold approach is compared with the single-manifold approaches of PCA, NPE and LPP. Also, the proposed method is compared with RUS, NCL [28], OSS [27], CNN [34], ENN [35], CBU [31], and PUMD [13] under-sampling approaches using support vector machine (SVM), k nearest neighbors (kNN) and classification and regression trees (CART) with a 10-fold cross evaluation scheme and in 5 repetitions.

These evaluations are performed on KEEL and UCI datasets based on various efficiency criteria. The mentioned methods have been chosen for comparison because

Algorithm 1. Gradual multi-manifold learning based under-sampling approach

1. Calculate the primary F-measure of the classifier on the validation set F-measure$_{old}$ and save it.

2. Apply the multi manifold approach on the training examples of the majority class.

3. Assess the manifolds with the information loss based criterion according to equation (5).

4. Compute $\alpha(M_i)$ for each manifold M_i, according to Eqs. (6).

5. For each data sample of the majority class, the degree of centrality and the degree of marginality are computed according to (7) and (8), respectively.

6. Assign a weight for the majority data samples according to equation (9). Sort the weight sequence in descending order.

7. Repeat

    Remove a certain partition of the majority class (i.e., 5 or 10 percent) with the highest weight.

    Combine the reduced majority class with the minority class to create new training samples.

    Train the classifier.

    Calculate the F-measure of the classifier on the validation data samples, F-measure$_{new}$.

    Continue the loop as long as F-measure$_{new}$ is greater than or equal to the F-measure$_{old}$.

    Set F-measure$_{old}$= F-measure$_{new}$

Fig. 3 The algorithm of the proposed method

they are among the most common under-sampling methods in literature reviews. Also, a non-parametric Wilcoxon signed rank test is used for statistical evaluation of the results. The details are explained in the following sections.

Datasets

In this research, 22 datasets are used in the experiments. These datasets are standard datasets and are usually used in the evaluation of the imbalanced data problem. These datasets are taken from the KEEL and UCI repositories. The datasets are shown in Table 2 along with their attributes such as the number of features, the number of minority class samples, the number of majority class samples, the total number of samples and the imbalance ratio. Similar to other researches, the multi-class data are transformed into two-class data by the common one-versus-all technique. Fewer samples represent the minority class and more samples represent the majority class. As seen in Table 2, kddcup- buffer_overflow_vs_back and shuttle_2_vs_5 datasets are among the most imbalanced ones which are important to be monitored in the evaluation.

Figure 5 shows the data of ecoli1 and glass0 in three modes: the main data, the output of the single-manifold under-sampling method, and the output of the proposed method (multi-manifold). In this figure x_1 and x_2 denote features that increase the differentiation between classes. It can be seen that using the proposed method will reduce the overlap between the majority and minority classes. For this purpose, average number of opposite-label neighbors is considered as a class overlap criterion.

Consider K as the number of neighboring points of each data point. For each data sample, K nearest neighbors are found. Then, for each data sample, the ratio of neighbors

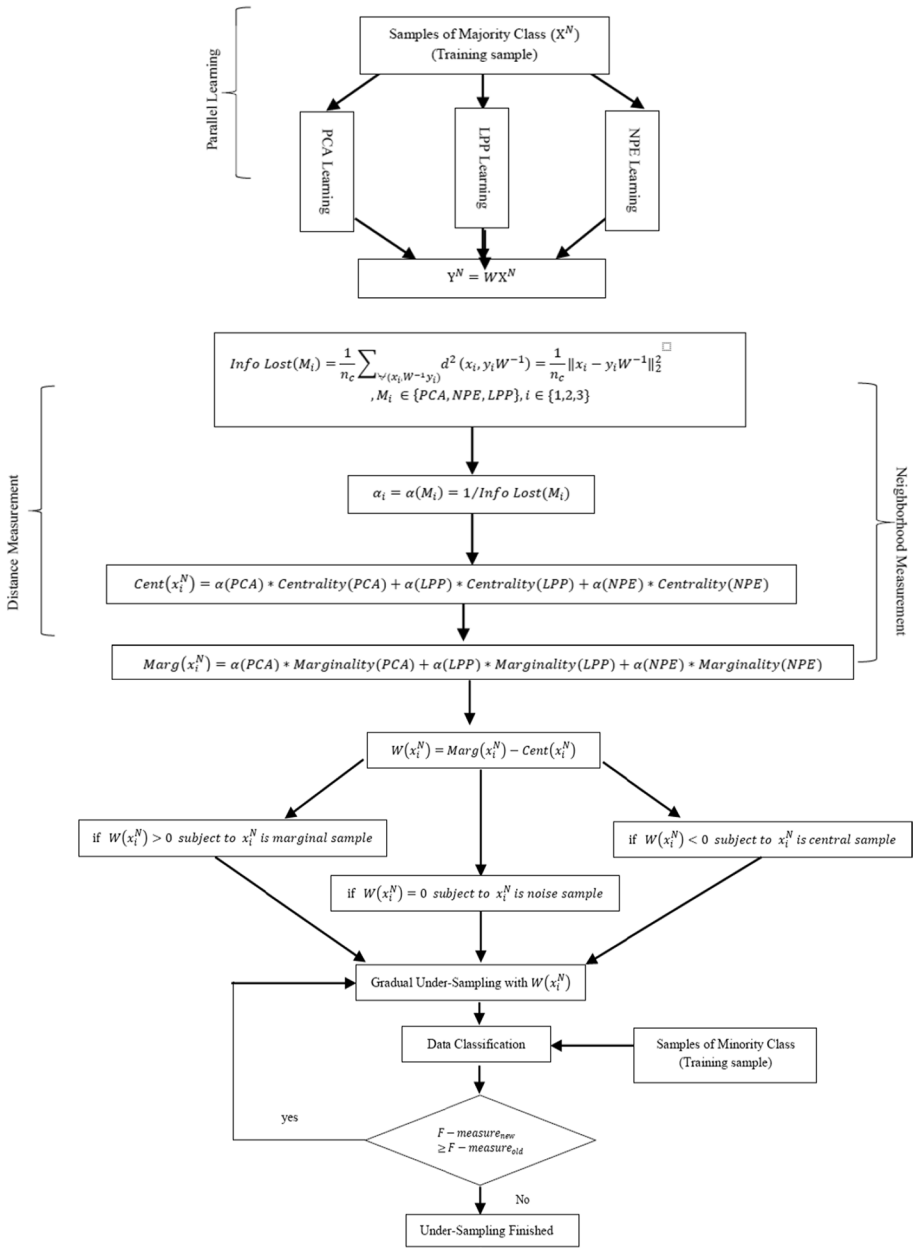

Fig. 4 The flowchart of the proposed method with the step-by-step calculations

belonging to the opposite class is calculated. This value is averaged for all data points. The smaller the value is, it suggests less overlap between two classes. For this purpose, experiments are conducted in three modes of original data, after single-manifold method and after multi-manifold method on a number of data sets. Table 3 denotes the results. The results of the experiments show that in all dataset, the amount of overlap after applying the proposed method, either single-manifold or multi-manifold, is less than the amount of overlap of the original data.

Table 2 Description of the experimental datasets

Name	#Attributes	#Minority Class	#Majority Class	#Examples	Imbalance Ratio
ecoli1	7	77	259	336	3.36
ecoli2	7	52	284	336	5.46
ecoli3	7	35	301	336	8.60
ecoli4	7	20	316	336	15.8
ecoli0147vs56	6	25	307	332	12.28
ecoli034_5	7	20	180	200	9
ecoli0147_2356	7	29	307	336	10.59
glass0	9	70	144	214	2.06
glass0123456	9	51	163	214	3.20
kddcup- buffer_overflow_ vs_back	41	30	2203	2233	73.43
kddcup-rootkit-imap_vs_ back	41	22	2203	2225	100.14
kddcup-guess_passwd_ vs_satan	41	53	1589	1642	29.98
kddcup-land_vs_portsweep	41	21	1040	1061	49.52
kddcup-land_vs_satan	41	21	1589	1610	75.67
new-thyroid1	5	35	180	215	5.14
page-blocks-1-3_vs_4	10	28	444	472	15.86
Pima	8	268	500	768	1.87
segment0	19	329	1979	2308	6.02
shuttle_2_vs_5	9	49	3267	3316	66.67
vehicle2–1	18	218	628	846	2.88
vowel0	13	90	898	988	9.98
Wisconsin	9	239	444	683	1.86

Experimental setup

The evaluations of the proposed under-sampling approach have been carried out with four scenarios, and they have been compared with the results of other articles. The evaluation criteria are precision, recall, F-measure, G-Mean and accuracy. In this research, a SVM classifier with an RBF kernel, a 3NN, and a CART with MaxNumSplits = 7 is used as the classifiers so that the results are comparable with those of other articles. Unsupervised manifold learning approaches such as PCA, NPE and LPP are used in the experiments. The proposed multi-manifold method can be implemented with supervised manifold learning approaches, but due to the high execution time, they are not used.

In the first scenarios (i.e. "Multi-manifold approach with reduction step of 5 percent" and "Multi-manifold approach with reduction step of 10 percent" sections), the effect of the proposed multi-manifold approach for gradual elimination of the majority samples is investigated separately with steps of 5% and 10% respectively, and the efficiency criteria are reported along with the standard deviation. In "Comparison of single-manifold and multi-manifold approaches" section, the results of the proposed multi-manifold approach and the best single-manifold results for gradual elimination with a step of 5% are compared based on the three criteria of recall, precision and F-measure, and the results together with standard deviation are reported. In "Comparison with other under-sampling approaches" section, the proposed multi-manifold approach is compared with RUS, NCL [28], OSS [27], CNN [34], ENN [35], CBU [31], and PUMD [13]

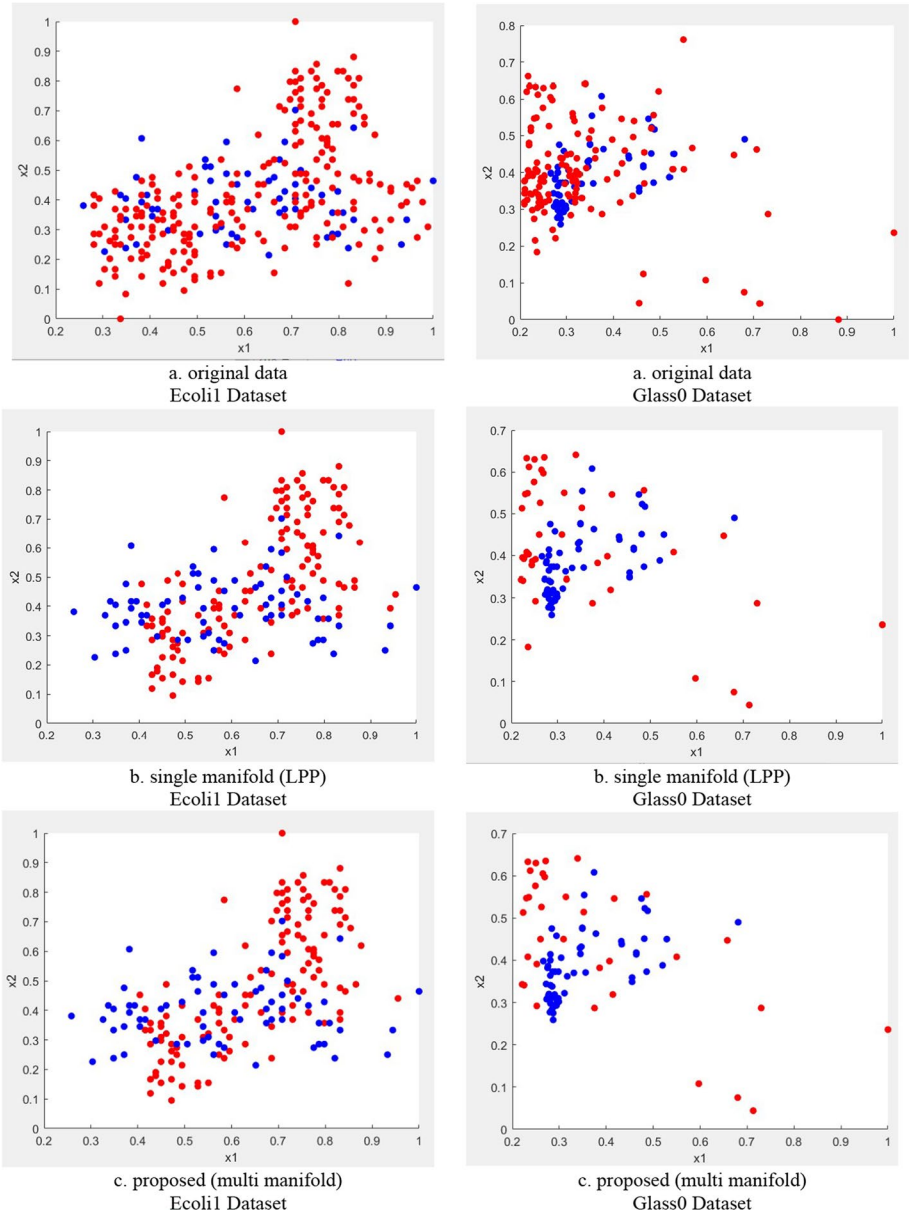

Fig. 5 Data distribution of Ecoli1, and Glass0 in three modes, original data, single manifold under-sampling method and the proposed multi manifold approach. Red dots denote the majority class

Table 3 The average number of opposite class neighbors in three modes (i.e. original data, after using the single-manifold method, and after using the multi-manifold method)

Dataset	K	Original	Single-manifold	Multi-manifold
ecoli1	11	0.030	0	0
page-blocks-1-3_vs_4	7	0.035	0	0
Pima	35	0.004	0	0
vehicle2–1	5	0.158	0.055	0

under-sampling methods. The simulation results show that our proposed method has better results than other methods on most datasets.

R2018b MatLab and DRToolbox are used for evaluations. For simplicity, the optimal parameters used in the simulation, like the number of nearest neighbors to calculate the centrality (K_c) and marginality (K_m), are 5.

Evaluation criteria

In this research, common criteria such as F-measure and G-Mean are used to measure the classification quality. To calculate these criteria, it is necessary to count the number of TP, FN, FP, TN. The confusion matrix is illustrated in Table 4. In imbalanced problems, examples with positive labels represent the minority class, and examples with negative labels represent the majority class.

Precision, Recall, F-measure, and accuracy criteria can be calculated by Eqs. (13) to (19).

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \tag{13}$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \tag{14}$$

$$\text{Sensitivity} = TP/(TP + FN) \tag{15}$$

$$\text{Specificity} = TN/(TN + FP) \tag{16}$$

$$\text{F} - \text{measure} = 2 * (\text{Recall} * \text{Precision})/(\text{Recall} + \text{Precision}) \tag{17}$$

$$\text{G} - \text{Mean} = \sqrt[2]{\text{Sensitivity} \times \text{Specificity}} \tag{18}$$

$$\text{Accuracy} = (\text{TN} + \text{TP})/(\text{TN} + \text{TP} + \text{FN} + \text{FP}) \tag{19}$$

Simulation results

Multi-manifold approach with reduction step of 5 percent

In this section, the effect of the proposed multi-manifold approach for gradual elimination of the majority samples with a reduction step of 5% is investigated. Performance criteria are reported in Tables 5, 6, 7 along with the standard deviation. These evaluations are performed for All three selected classifiers.

Table 4 Confusion matrix

	Predict positive class	**Predict negative class**
Actual positive class	TP	FN
Actual negative class	FP	TN

Table 5 The performance of the SVM classifier with the multi-manifold approach with step of 5 percent

Dataset	Recall	Precision	G-means	F-measure	Accuracy
ecoli1	0.90 ± 0.170	0.71 ± 0.228	0.88 ± 0.141	0.83 ± 0.159	0.87 ± 0.131
ecoli2	0.90 ± 0.202	0.88 ± 0.192	0.92 ± 0.131	0.90 ± 0.142	0.96 ± 0.062
ecoli3	0.93 ± 0.160	0.76 ± 0.270	0.93 ± 0.091	0.86 ± 0.195	0.94 ± 0.072
ecoli4	0.90 ± 0.200	0.78 ± 0.307	0.93 ± 0.137	0.92 ± 0.170	0.96 ± 0.071
ecoli0147vs56	0.85 ± 0.240	0.75 ± 0.311	0.80 ± 0.303	0.89 ± 0.119	0.88 ± 0.266
ecoli034_5	0.85 ± 0.320	0.63 ± 0.394	0.75 ± 0.387	0.88 ± 0.151	0.86 ± 0.261
ecoli0147_2356	0.78 ± 0.279	0.64 ± 0.327	0.75 ± 0.307	0.78 ± 0.211	0.86 ± 0.264
glass0	0.80 ± 0.165	0.67 ± 0.137	0.77 ± 0.099	0.79 ± 0.116	0.78 ± 0.098
glass0123456	0.98 ± 0.060	0.85 ± 0.244	0.93 ± 0.148	0.92 ± 0.166	0.91 ± 0.171
kddcup-buffer_overflow_vs_back	1 ± 0	0.88 ± 0.256	0.99 ± 0.014	0.95 ± 0.161	0.99 ± 0.027
new-thyroid1	0.91 ± 0.204	0.83 ± 0.274	0.93 ± 0.151	0.89 ± 0.206	0.94 ± 0.098
page-blocks-1-3_vs_4	0.57 ± 0.372	0.69 ± 0.440	0.66 ± 0.352	0.79 ± 0.262	0.94 ± 0.089
Pima	0.71 ± 0.163	0.58 ± 0.130	0.65 ± 0.131	0.63 ± 0.086	0.68 ± 0.122
segment0	0.97 ± 0.039	0.98 ± 0.023	0.98 ± 0.020	0.98 ± 0.017	0.99 ± 0.006
shuttle_2_vs_5	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
vehicle2–1	0.97 ± 0.035	0.94 ± 0.079	0.98 ± 0.026	0.97 ± 0.033	0.98 ± 0.028
vowel0	0.95 ± 0.113	0.84 ± 0.198	0.96 ± 0.063	0.90 ± 0.120	0.97 ± 0.031
Wisconsin	0.98 ± 0.040	0.88 ± 0.079	0.90 ± 0.085	0.95 ± 0.053	0.91 ± 0.065
Average	0.881	0.793	0.872	0.879	0.912

Table 6 The performance of the 3NN classifier with the multi-manifold approach with step of 5 percent

Dataset	Recall	Precision	G-means	F-measure	Accuracy
ecoli1	0.93 ± 0.130	0.70 ± 0.201	0.89 ± 0.110	0.85 ± 0.153	0.87 ± 0.111
ecoli2	0.95 ± 0.111	0.78 ± 0.188	0.94 ± 0.074	0.91 ± 0.105	0.94 ± 0.063
ecoli3	0.93 ± 0.225	0.60 ± 0.337	0.90 ± 0.152	0.75 ± 0.231	0.90 ± 0.097
ecoli4	0.95 ± 0.150	0.75 ± 0.334	0.93 ± 0.178	0.88 ± 0.279	0.91 ± 0.199
ecoli0147vs56	0.88 ± 0.183	0.72 ± 0.251	0.92 ± 0.106	0.84 ± 0.159	0.96 ± 0.036
ecoli034_5	0.90 ± 0.200	0.78 ± 0.269	0.92 ± 0.122	0.86 ± 0.180	0.95 ± 0.067
ecoli0147_2356	0.82 ± 0.240	0.66 ± 0.253	0.87 ± 0.149	0.81 ± 0.159	0.94 ± 0.047
glass0	0.89 ± 0.124	0.64 ± 0.116	0.79 ± 0.070	0.78 ± 0.076	0.78 ± 0.083
glass0123456	0.96 ± 0.80	0.86 ± 0.227	0.94 ± 0.117	0.92 ± 0.153	0.93 ± 0.133
kddcup-buffer_overflow_vs_back	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
new-thyroid1	0.93 ± 0.200	0.93 ± 0.155	0.95 ± 0.125	0.93 ± 0.160	0.97 ± 0.048
page-blocks-1-3_vs_4	0.93 ± 0.133	0.75 ± 0.293	0.94 ± 0.079	0.87 ± 0.210	0.95 ± 0.069
Pima	0.62 ± 0.208	0.56 ± 0.148	0.64 ± 0.114	0.58 ± 0.141	0.68 ± 0.103
segment0	0.99 ± 0.027	0.92 ± 0.056	0.99 ± 0.014	0.97 ± 0.023	0.99 ± 0.010
shuttle_2_vs_5	1 ± 0	0.91 ± 0.274	0.99 ± 0.025	0.97 ± 0.100	0.98 ± 0.047
vehicle2–1	0.97 ± 0.042	0.82 ± 0.098	0.94 ± 0.030	0.92 ± 0.051	0.93 ± 0.036
vowel0	0.90 ± 0.213	0.67 ± 0.330	0.89 ± 0.181	0.87 ± 0.157	0.89 ± 0.153
Wisconsin	0.98 ± 0.027	0.93 ± 0.083	0.94 ± 0.074	0.97 ± 0.050	0.94 ± 0.061
Average	0.918	0.767	0.91	0.871	0.917

The tables show two observations. The first one is that using the proposed approach, the recall rate is much higher than the precision rate. This means that the classifiers are more successful at remembering the positive class, which is the minority class. This can

Table 7 The performance of the CART classifier with the multi-manifold approach and step of 5 percent

Dataset	Recall	Precision	G-means	F-measure	Accuracy
ecoli1	0.80 ± 0.334	0.61 ± 0.247	0.72 ± 0.328	0.78 ± 0.221	0.80 ± 0.206
ecoli2	0.69 ± 0.324	0.60 ± 0.354	0.72 ± 0.334	0.72 ± 0.260	0.83 ± 0.245
ecoli3	0.73 ± 0.361	0.32 ± 0.160	0.70 ± 0.267	0.58 ± 0.155	0.81 ± 0.083
ecoli4	0.85 ± 0.229	0.58 ± 0.364	0.78 ± 0.287	0.79 ± 0.193	0.83 ± 0.276
ecoli0147vs56	0.72 ± 0.365	0.34 ± 0.357	0.57 ± 0.392	0.79 ± 0.138	0.68 ± 0.341
ecoli034_5	0.85 ± 0.229	0.79 ± 0.335	0.87 ± 0.172	0.88 ± 0.212	0.91 ± 0.156
ecoli0147_2356	0.77 ± 0.300	0.44 ± 0.322	0.71 ± 0.261	0.67 ± 0.255	0.75 ± 0.275
glass0	0.71 ± 0.268	0.57 ± 0.216	0.62 ± 0.191	0.66 ± 0.198	0.66 ± 0.174
glass0123456	0.90 ± 0.132	0.78 ± 0.245	0.88 ± 0.143	0.87 ± 0.157	0.88 ± 0.159
kddcup-buffer_overflow_vs_back	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
new-thyroid1	0.93 ± 0.133	0.89 ± 0.231	0.92 ± 0.126	0.93 ± 0.146	0.94 ± 0.148
page-blocks-1-3_vs_4	1 ± 0	0.90 ± 0.200	1 ± 0.009	0.96 ± 0.080	0.99 ± 0.018
pima	0.66 ± 0.104	0.61 ± 0.148	0.69 ± 0.075	0.65 ± 0.071	0.71 ± 0.095
segment0	0.95 ± 0.069	0.92 ± 0.156	0.96 ± 0.044	0.97 ± 0.032	0.97 ± 0.047
shuttle_2_vs_5	1 ± 0	0.91 ± 0.284	0.99 ± 0.044	0.91 ± 0.261	0.97 ± 0.080
vehicle2–1	0.85 ± 0.133	0.79 ± 0.159	0.87 ± 0.061	0.87 ± 0.081	0.89 ± 0.049
vowel0	0.84 ± 0.206	0.85 ± 0.185	0.89 ± 0.125	0.90 ± 0.104	0.97 ± 0.028
wisconsin	0.90 ± 0.128	0.86 ± 0.130	0.82 ± 0.182	0.91 ± 0.073	0.85 ± 0.111
Average	0.841	0.708	0.817	0.824	0.857

denote that the approach has lowered the effect of imbalanced classes on the minority class, although at the cost of more false alarms and lower precision. The other observation is that the performances of the SVM and 3NN classifiers are similar, but the CART performance is degraded. Therefore, to avoid excessive evaluations and tables and due to the fact that KNN is the most common classifier in this regard, future evaluations will only concern the 3NN as the experimental classifier.

Multi-manifold approach with reduction step of 10 percent

In this section, the effect of the proposed new multi-manifold approach with a reduction step of 10% is investigated. The other experimental settings are the same as the previous experiments. The results of Table 8 on 3NN classifier does not show much different from the results indicated in Table 6. Therefore, we can conclude that the approach is not much dependent on the step size. Therefore, in the following experiment reduction step size of 5 is selected to avoid divergence of the proposed reduction method while maintaining a good reduction speed.

Comparison of single-manifold and multi-manifold approaches

In this section, the results of the proposed multi-manifold approach and the single-manifold approaches compared and shown in Tables 9 and 10. The numbers in parentheses show the rank of each approach for each the corresponding data separately. The average rank and efficiency of each approach are shown separately in the last row of the tables. According to Table 9, the multi-manifold approach has the best average rank considering the recall performance measure and other single-manifold approaches have won the second to fourth ranks.

Table 8 The Average performance measures of the 3NN classifier with the multi-manifold approach with reduction step of 10 percent

Name	Recall	Precision	G-means	F-measure	Accuracy
ecoli1	0.98 ± 0.050	0.70 ± 0.208	0.89 ± 0.126	0.85 ± 0.152	0.86 ± 0.152
ecoli2	0.95 ± 0.110	0.75 ± 0.200	0.93 ± 0.076	0.89 ± 0.116	0.93 ± 0.066
ecoli3	0.93 ± 0.225	0.56 ± 0.359	0.88 ± 0.155	0.73 ± 0.250	0.87 ± 0.114
ecoli4	0.95 ± 0.150	0.74 ± 0.350	0.92 ± 0.178	0.88 ± 0.279	0.91 ± 0.199
ecoli0147vs56	0.88 ± 0.183	0.71 ± 0.258	0.92 ± 0.106	0.84 ± 0.159	0.95 ± 0.039
ecoli034_5	0.90 ± 0.200	0.78 ± 0.269	0.92 ± 0.122	0.86 ± 0.180	0.95 ± 0.067
ecoli0147_2356	0.82 ± 0.240	0.64 ± 0.261	0.87 ± 0.147	0.81 ± 0.159	0.93 ± 0.049
glass0	0.89 ± 0.124	0.61 ± 0.121	0.77 ± 0.093	0.78 ± 0.076	0.75 ± 0.101
glass0123456	0.96 ± 0.080	0.83 ± 0.226	0.93 ± 0.115	0.91 ± 0.151	0.92 ± 0.131
kddcup-buffer_overflow_vs_back	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
new-thyroid1	1 ± 0	0.93 ± 0.155	0.99 ± 0.026	0.98 ± 0.075	0.98 ± 0.043
page-blocks-1-3_vs_4	0.93 ± 0.133	0.71 ± 0.319	0.94 ± 0.084	0.87 ± 0.210	0.95 ± 0.071
Pima	0.70 ± 0.162	0.57 ± 0.142	0.66 ± 0.095	0.63 ± 0.094	0.68 ± 0.110
segment0	0.99 ± 0.027	0.90 ± 0.066	0.98 ± 0.014	0.97 ± 0.023	0.98 ± 0.012
shuttle_2_vs_5	1 ± 0	0.91 ± 0.269	0.99 ± 0.021	0.97 ± 0.100	0.98 ± 0.040
vehicle2–1	0.98 ± 0.040	0.80 ± 0.110	0.94 ± 0.034	0.92 ± 0.048	0.92 ± 0.047
vowel0	0.97 ± 0.245	0.62 ± 0.301	0.84 ± 0.176	0.87 ± 0.212	0.81 ± 0.117
Wisconsin	0.98 ± 0.028	0.93 ± 0.079	0.94 ± 0.086	0.97 ± 0.039	0.94 ± 0.061
Average	0.933889	0.761667	0.906111	0.873889	0.906111

Table 9 The average recall of the 3NN classifier in the multi-manifold and single manifold approaches

Dataset	PCA	LPP	NPE	Multi-Manifold
ecoli1	0.89 ± 0.139 (4)	0.93 ± 0.130 (1)	0.93 ± 0.130 (1)	0.93 ± 0.130 (1)
ecoli2	0.95 ± 0.110 (1)	0.95 ± 0.110 (1)	0.95 ± 0.110 (1)	0.95 ± 0.111 (1)
ecoli3	0.90 ± 0.229 (2)	0.85 ± 0.229 (3)	0.85 ± 0.229 (3)	0.93 ± 0.225 (1)
ecoli4	0.90 ± 0.200 (4)	0.95 ± 0.150 (1)	0.95 ± 0.150 (1)	0.95 ± 0.150 (1)
ecoli0147vs56	0.85 ± 0.189 (3)	0.85 ± 0.189 (3)	0.88 ± 0.183 (1)	0.88 ± 0.183 (1)
ecoli034_5	0.90 ± 0.200 (1)	0.90 ± 0.200 (1)	0.90 ± 0.200 (1)	0.90 ± 0.200 (1)
ecoli0147_2356	0.82 ± 0.240 (1)	0.82 ± 0.240 (1)	0.82 ± 0.240 (1)	0.82 ± 0.240 (1)
glass0	0.89 ± 0.106 (1)	0.89 ± 0.124 (1)	0.87 ± 0.134 (4)	0.89 ± 0.124 (1)
glass0123456	0.96 ± 0.080 (1)	0.96 ± 0.080 (1)	0.96 ± 0.080 (1)	0.96 ± 0.80 (1)
kddcup-buffer_overflow_vs_back	1 ± 0 (1)	1 ± 0 (1)	1 ± 0 (1)	1 ± 0 (1)
new-thyroid1	0.93 ± 0.200 (1)	0.93 ± 0.200 (1)	0.93 ± 0.200 (1)	0.93 ± 0.200 (1)
page-blocks-1-3_vs_4	0.93 ± 0.133 (1)	0.93 ± 0.133 (1)	0.93 ± 0.133 (1)	0.93 ± 0.133 (1)
Pima	0.63 ± 0.167 (1)	0.62 ± 0.177 (2)	0.61 ± 0.153 (4)	0.62 ± 0.208 (2)
segment0	0.99 ± 0.027 (1)	0.99 ± 0.020 (1)	0.99 ± 0.020 (1)	0.99 ± 0.027 (1)
shuttle_2_vs_5	1 ± 0 (1)	1 ± 0 (1)	1 ± 0 (1)	1 ± 0 (1)
vehicle2–1	0.96 ± 0.041 (2)	0.95 ± 0.061 (3)	0.95 ± 0.061 (3)	0.97 ± 0.042 (1)
vowel0	0.88 ± 0.245 (4)	0.90 ± 0.213 (1)	0.90 ± 0.213 (1)	0.90 ± 0.213 (1)
Wisconsin	0.98 ± 0.027 (1)	0.98 ± 0.028 (1)	0.98 ± 0.028 (1)	0.98 ± 0.027 (1)
Average Ratings	1.72 (4)	1.39 (2)	1.56 (3)	1.06 (1)
Average Recall	0.91 (2)	0.91 (2)	0.91 (2)	0.92 (1)

Table 10 The average F measure of the 3NN classifier in the multi-manifold and single manifold approaches

Dataset	PCA	LPP	NPE	Multi-Manifold
ecoli1	0.84 ± 0.160 (4)	0.85 ± 0.152 (1)	0.85 ± 0.152 (1)	0.85 ± 0.152 (1)
ecoli2	0.88 ± 0.132 (4)	0.90 ± 0.104 (2)	0.90 ± 0.104 (2)	0.91 ± 0.105 (1)
ecoli3	0.73 ± 0.223 (2)	0.71 ± 0.191 (3)	0.71 ± 0.191 (3)	0.86 ± 0.195 (1)
ecoli4	0.86 ± 0.296 (4)	0.89 ± 0.272 (1)	0.89 ± 0.272 (1)	0.92 ± 0.170 (1)
ecoli0147vs56	0.82 ± 0.150 (3)	0.82 ± 0.150 (3)	0.84 ± 0.159 (1)	0.89 ± 0.119 (1)
ecoli034_5	0.86 ± 0.180 (1)	0.86 ± 0.180 (1)	0.86 ± 0.180 (1)	0.88 ± 0.151 (1)
ecoli0147_2356	0.81 ± 0.159 (1)	0.81 ± 0.159 (1)	0.81 ± 0.159 (1)	0.81 ± 0.159 (1)
glass0	0.78 ± 0.077 (3)	0.79 ± 0.080 (1)	0.78 ± 0.076 (3)	0.79 ± 0.116 (1)
glass0123456	0.92 ± 0.153 (1)	0.92 ± 0.153 (1)	0.92 ± 0.153 (1)	0.92 ± 0.153 (1)
kddcup-buffer_overflow_vs_back	1 ± 0 (1)	1 ± 0 (1)	1 ± 0 (1)	1 ± 0 (1)
new-thyroid1	0.93 ± 0.160 (2)	0.93 ± 0.160 (2)	0.93 ± 0.160 (2)	0.98 ± 0.075 (1)
page-blocks-1-3_vs_4	0.87 ± 0.210 (2)	0.87 ± 0.210 (2)	0.87 ± 0.210 (2)	096 ± 0.080 (1)
pima	0.59 ± 0.113 (2)	0.58 ± 0.094 (3)	0.58 ± 0.092 (3)	0.65 ± 0.067 (1)
segment0	0.97 ± 0.023 (3)	0.98 ± 0.023 (1)	0.98 ± 0.023 (1)	0.98 ± 0.017 (1)
shuttle_2_vs_5	0.97 ± 0.100 (2)	0.97 ± 0.100 (2)	0.97 ± 0.100 (2)	1 ± 0 (1)
vehicle2–1	0.92 ± 0.050 (2)	0.92 ± 0.052 (2)	0.92 ± 0.052 (2)	0.97 ± 0.030 (1)
vowel0	0.85 ± 0.174 (4)	0.87 ± 0.157 (2)	0.87 ± 0.157 (2)	0.90 ± 0.104 (1)
wisconsin	0.97 ± 0.050 (1)	0.97 ± 0.050 (1)	0.97 ± 0.050 (1)	0.97 ± 0.050 (1)
Average Ratings	2.00 (4)	1.33 (2)	1.28 (2)	1 (1)
Average F-measure	0.87 (2)	0.87 (2)	0.87 (2)	0.90 (1)

The experimental results which are denoted in Table 9 show a marginal superiority of the proposed multi-manifold approach over each single manifold approaches. It denotes the classification recall of the reduction approach using each manifold learning method alone is approximately the same, but using the multi-manifold approaches, we have a slightly better measure for dropping the instances.

The results of evaluation based to the average F-measure with 3NN classifier are reported in Table 10. According to Table 10, the multi-manifold approach has the best average rank and other single-manifold approaches have obtained a lower average efficiency and average rank. As seen, the effectiveness and superiority of the proposed multi-manifold approach is obvious compared to single manifold learning approaches. The main strength of the manifold based approach either single or multiple, is their impressing F measure on the highly imbalanced datasets (i.e., kddcup-buffer_overflow_vs_back and shuttle_2_vs_5). This observation can approximately be seen for both single manifold and multi-manifold approaches. This will be discussed in the next experiments which concerns comparisons with the other state-of-the-art approaches.

Comparison with other under-sampling approaches

In this section, the results of the proposed multi-manifold approach are compared with under-sampling models such as RUS, NCL [28], OSS [27], CNN [34], ENN [35], CBU [31], and PUMD [13], and illustrated in Tables 11, 12, 13. Comparisons are based on recall, precision and F-measure criteria. The results of the simulation show that the F-measure of the proposed method outperforms the other under-sampling methods by a wide margin.

Table 11 The average recall of different under-sampling methods

Dataset	Original	RUS	NCL [28]	OSS [27]	CNN [34]	ENN [35]	CBU [31]	PUMD [13]	Proposed
ecoli1	0.72 (9)	0.82 (4)	0.79 (5)	0.78 (6)	0.77 (7)	0.73 (8)	0.83 (3)	0.89 (2)	0.98 ± 0.050 (1)
ecoli2	0.65 (6)	0.62 (7)	0.66 (4)	0.66 (4)	0.62 (7)	0.61 (9)	0.76 (3)	0.83 (2)	0.95 ± 0.111 (1)
ecoli3	0.50 (8)	0.72 (4)	0.56 (6)	0.43 (9)	0.51 (7)	0.58 (5)	0.77 (3)	0.86 (2)	0.95 ± 0.100 (1)
ecoli4	0.67 (6)	0.68 (4)	0.64 (7)	0.64 (7)	0.68 (4)	0.64 (7)	0.82 (3)	0.90 (2)	0.95 ± 0.150 (1)
ecoli0147vs56	0.67 (7)	0.79 (3)	0.70 (6)	0.71 (5)	0.73 (4)	0.65 (8)	0.83 (2)	0.83 (2)	0.88 ± 0.183 (1)
ecoli034_5	0.68 (7)	0.77 (3)	0.73 (5)	0.73 (5)	0.68 (7)	0.68 (7)	0.82 (2)	0.77 (3)	0.90 ± 0.200 (1)
ecoli0147_2356	0.50 (8)	0.79 (3)	0.55 (5)	0.55 (5)	0.50 (8)	0.52 (7)	0.79 (3)	0.87 (1)	0.82 ± 0.240 (2)
glass0	0.43 (9)	0.78 (3)	0.61 (5)	0.57 (6)	0.46 (8)	0.57 (6)	0.78 (3)	0.83 (2)	0.89 ± 0.124 (1)
glass0123456	0.75 (4)	0.57 (9)	0.73 (5)	0.68 (8)	0.73 (5)	0.71 (7)	0.76 (3)	0.87 (2)	0.98 ± 0.060 (1)
kddcup-buffer_over-flow_vs_back	0.67 (5)	0.85 (3)	0.61 (8)	0.67 (5)	0.64 (7)	0.64 (7)	0.82 (4)	0.90 (2)	1 ± 0 (1)
new-thyroid1	0.74 (7)	0.80 (5)	0.71 (8)	0.78 (6)	0.86 (3)	0.67 (9)	0.86 (3)	0.87 (2)	1 ± 0 (1)
page-blocks-1-3_vs_4	0.74 (9)	0.91 (3)	0.77 (8)	0.85 (5)	0.85 (5)	0.79 (7)	0.91 (3)	0.90 (2)	1 ± 0 (1)
Pima	0.51 (9)	0.63 (3)	0.58 (6)	0.61 (5)	0.52 (8)	0.54 (7)	0.63 (3)	0.86 (1)	0.78 ± 0.160 (2)
segment0	0.88 (7)	0.88 (7)	0.87 (7)	0.87 (7)	0.86 (9)	0.88 (4)	0.90 (2)	0.89 (3)	0.99 ± 0.027 (1)
shuttle_2_vs_5	0.91 (2)	0.89 (9)	0.91 (2)	0.91 (2)	0.91 (2)	0.91 (2)	0.91 (2)	0.90 (3)	1 ± 0 (1)
vehicle2–1	0.70 (7)	0.85 (2)	0.70 (7)	0.70 (7)	0.73 (4)	0.72 (5)	0.84 (3)	0.71 (6)	0.98 ± 0.022 (1)
vowel0	0.83 (5)	0.89 (3)	0.79 (9)	0.81 (7)	0.83 (5)	0.81 (7)	0.89 (3)	0.90 (2)	0.99 ± 0.033 (1)
wisconsin	0.83 (9)	0.85 (7)	0.84 (8)	0.88 (4)	0.89 (3)	0.88 (4)	0.87 (6)	0.90 (2)	0.98 ± 0.028 (1)
Average Ratings	6.83 (9)	4.56 (4)	6.28 (7)	5.83 (5)	5.89 (6)	6.61 (8)	3.28 (3)	2.27 (2)	1.11 (1)
Average Recall	0.69 (9)	0.78 (4)	0.71 (5)	0.71 (5)	0.71 (5)	0.70 (8)	0.82 (3)	0.86 (2)	0.95 (1)

First, the results of the evaluations related to the average recall are reported in Table 11. The numbers in parenthesis, show the rank of the method on that dataset. On all the data, the recall of the proposed approach ranks first and other under-sampling methods rank second to eighth. The average rank and average efficiency of each under-sampling methods are shown separately in the last row of Table 11. The multi-manifold approach has the first average rank compared to other approaches. Also, it can clearly be seen that the recall of the proposed approach is considerably higher than the other approaches specially when the IR increases (refer to the rows corresponding to ecoli1, ecoli2, ecoli3, ecoli4, ecoli034_5, kddcup, page-block, vowel0 and shuttle dataset), the recall increases by a wide margin of 10 percent. Since, minority class is the positive class,

Table 12 The average precision of different under-sampling methods

Dataset	Original	RUS	NCL [28]	OSS [27]	CNN [34]	ENN [35]	CBU [31]	PUMD [13]	Proposed
ecoli1	0.78 (4)	0.83 (2)	0.83 (2)	0.78 (4)	0.77 (6)	0.77 (6)	0.77 (6)	0.89 (1)	0.72 ± 0.212 (9)
ecoli2	0.80 (7)	0.86 (3)	0.82 (6)	0.76 (8)	0.76 (8)	0.86 (3)	0.86 (3)	0.90 (1)	0.88 ± 0.192 (2)
ecoli3	0.66 (6)	0.77 (2)	0.55 (9)	0.58 (8)	0.70 (5)	0.65 (7)	0.76 (3)	0.88 (1)	0.76 ± 0.270 (3)
ecoli4	0.73 (7)	0.79 (4)	0.76 (6)	0.71 (9)	0.82 (3)	0.73 (7)	0.88 (2)	0.90 (1)	0.78 ± 0.307 (5)
ecoli0147vs56	0.84 (2)	0.78 (8)	0.84 (2)	0.83 (5)	0.80 (6)	0.84 (2)	0.80 (6)	0.90 (1)	0.75 ± 0.311 (9)
ecoli034_5	0.83 (3)	0.80 (7)	0.82 (5)	0.83 (3)	0.86 (2)	0.82 (5)	0.80 (7)	0.90 (1)	0.79 ± 0.335 (9)
ecoli0147_2356	0.73 (7)	0.77 (2)	0.76 (4)	0.77 (2)	0.71 (8)	0.74 (6)	0.75 (5)	0.90 (1)	0.66 ± 0.253 (9)
glass0	0.51 (8)	0.67 (2)	0.60 (5)	0.60 (5)	0.51 (8)	0.59 (7)	0.67 (2)	0.83 (1)	0.67 ± 0.137 (2)
glass0123456	0.79 (8)	0.60 (9)	0.82 (5)	0.81 (6)	0.83 (3)	0.80 (7)	0.83 (3)	0.89 (1)	0.86 ± 0.227 (2)
kddcup-buffer_over-flow_vs_back	0.15 (9)	0.89 (3)	0.82 (5)	0.80 (7)	0.80 (7)	0.81 (6)	0.89 (3)	0.9 (2)	1 ± 0 (1)
new-thyroid1	0.78 (8)	0.78 (8)	0.82 (5)	0.81 (6)	0.86 (3)	0.86 (3)	0.81 (6)	0.81 (2)	0.93 ± 0.155 (1)
page-blocks-1-3_vs_4	0.88 (4)	0.83 (8)	0.88 (4)	0.88 (4)	0.91 (1)	0.82 (9)	0.86 (5)	0.90 (2)	0.90 ± 0.200 (2)
Pima	0.62 (8)	0.67 (5)	0.68 (3)	0.72 (2)	0.63 (7)	0.66 (6)	0.68 (3)	0.75 (1)	0.61 ± 0.148 (8)
segment0	0.87 (8)	0.87 (8)	0.88 (5)	0.88 (5)	0.89 (3)	0.88 (5)	0.89 (3)	0.90 (1)	0.98 ± 0.023 (2)
shuttle_2_vs_5	0.91 (3)	0.91 (3)	0.91 (3)	0.91 (3)	0.91 (3)	0.91 (3)	0.89 (9)	0.9 (2)	1 ± 0 (1)
vehicle2–1	0.74 (5)	0.80 (3)	0.73 (6)	0.72 (8)	0.75 (4)	0.73 (6)	0.83 (2)	0.71 (9)	0.94 ± 0.079 (1)
vowel0	0.82 (6)	0.85 (4)	0.81 (8)	0.86 (3)	0.82 (6)	0.81 (8)	0.89 (2)	0.9 (1)	0.85 ± 0.185 (4)
wisconsin	0.86 (5)	0.84 (8)	0.87 (3)	0.85 (6)	0.87 (3)	0.84 (8)	0.85 (6)	0.9 (2)	0.93 ± 0.083 (1)
Average Ratings	6.06 (9)	5.00 (6)	4.83 (4)	5.28 (7)	4.89 (5)	5.83 (8)	4.39 (3)	1.72 (1)	3 (2)
Average Precision	0.74 (9)	0.80 (4)	0.79 (6)	0.78 (8)	0.79 (6)	0.78 (5)	0.82 (3)	0.87 (1)	0.83 (2)

it denotes that the proposed approach is successful in reducing the impact of majority (negative) class specially when there is a high imbalance.

Table 12 illustrates the results of evaluations related to the average precision criterion. As seen, the multi-manifold approach relatively degrades on this measure and has the second average rank compared to other under-sampling models. This lower precision is the observation we have seen previously in the initial experiments. This shows that the approach is tending to focus more on the positive class (minority class) and increase the recall rate with the cost of decreasing the precision. The degradation is not favorable, but the main measure that we have to focus on is F measure which is the harmonic mean of these metrics and compromise between recall and precision. The evaluations based on this measure are denoted in Table 13.

The results of evaluations of the approaches based on the average F-measure are reported in Table 13. As seen, this time, the proposed multi-manifold approach has the

Table 13 The average F measure of different under-sampling methods

Dataset	Original	RUS	NCL [28]	OSS [27]	CNN [34]	ENN [35]	CBU [31]	PUMD [13]	Proposed
ecoli1	0.73 (9)	0.82 (3)	0.81 (4)	0.77 (6)	0.76 (7)	0.75 (8)	0.79 (5)	0.89 (1)	0.85 ± 0.152 (2)
ecoli2	0.70 (6)	0.71 (5)	0.72 (4)	0.69 (8)	0.66 (9)	0.70 (6)	0.80 (3)	0.86 (2)	0.91 ± 0.105 (1)
ecoli3	0.54 (4)	0.70 (3)	0.54 (4)	0.46 (8)	0.54 (4)	0.53 (7)	0.72 (2)	0.86 (1)	0.86 ± 0.195 (1)
ecoli4	0.70 (5)	0.69 (6)	0.66 (8)	0.64 (9)	0.73 (4)	0.67 (7)	0.83 (3)	0.90 (2)	0.92 ± 0.170 (1)
ecoli0147vs56	0.72 (8)	0.77 (4)	0.74 (6)	0.75 (5)	0.74 (6)	0.70 (9)	0.80 (3)	0.86 (2)	0.89 ± 0.119 (1)
ecoli034_5	0.72 (9)	0.77 (4)	0.76 (5)	0.76 (5)	0.73 (7)	0.73 (7)	0.80 (3)	0.81 (2)	0.88 ± 0.151 (1)
ecoli0147_2356	0.56 (9)	0.75 (3)	0.60 (5)	0.60 (5)	0.57 (8)	0.59 (7)	0.75 (3)	0.89 (1)	0.81 ± 0.159 (2)
glass0	0.46 (9)	0.72 (3)	0.60 (5)	0.57 (6)	0.47 (8)	0.57 (6)	0.72 (3)	0.82 (1)	0.79 ± 0.116 (2)
glass0123456	0.75 (6)	0.57 (9)	0.77 (4)	0.73 (8)	0.76 (5)	0.74 (7)	0.78 (3)	0.88 (2)	0.92 ± 0.153 (1)
kddcup-buffer_over-flow_vs_back	0.24 (9)	0.86 (3)	0.68 (8)	0.71 (7)	0.75 (5)	0.74 (6)	0.84 (4)	0.90 (2)	1 ± 0 (1)
new-thyroid1	0.73 (7)	0.79 (5)	0.72 (9)	0.76 (6)	0.85 (3)	0.73 (7)	0.82 (4)	0.84 (2)	0.98 ± 0.075 (1)
page-blocks-1-3_vs_4	0.79 (9)	0.87 (5)	0.80 (8)	0.86 (7)	0.87 (5)	0.88 (3)	0.88 (3)	0.90 (2)	096 ± 0.080 (1)
Pima	0.56 (9)	0.65 (3)	0.63 (6)	0.66 (2)	0.57 (8)	0.59 (7)	0.65 (3)	0.80 (1)	0.65 ± 0.067 (3)
segment0	0.88 (3)	0.88 (3)	0.87 (7)	0.88 (3)	0.87 (7)	0.88 (3)	0.89 (2)	0.89 (2)	0.98 ± 0.017 (1)
shuttle_2_vs_5	0.91 (2)	0.91 (2)	0.91 (2)	0.91 (2)	0.91 (2)	0.91 (2)	0.90 (3)	0.90 (3)	1 ± 0 (1)
vehicle2–1	0.71 (6)	0.82 (3)	0.71 (6)	0.71 (6)	0.74 (4)	0.72 (5)	0.83 (2)	0.72 (5)	0.97 ± 0.030 (1)
vowel0	0.81 (6)	0.87 (3)	0.79 (8)	0.83 (4)	0.82 (5)	0.80 (7)	0.89 (2)	0.90 (1)	0.90 ± 0.104 (1)
Wisconsin	0.84 (8)	0.85 (7)	0.85 (7)	0.86 (4)	0.88 (3)	0.86 (4)	0.86 (4)	0.90 (2)	0.97 ± 0.050 (1)
Average Ratings	7.17 (9)	4.39 (4)	6.17 (7)	5.89 (6)	5.83 (5)	6.28 (8)	3.33 (3)	1.77 (2)	1.27 (1)
Average F-measure	0.69 (9)	0.78 (4)	0.73 (5)	0.73 (5)	0.73 (5)	0.73 (5)	0.81 (3)	0.86 (2)	0.90 (1)

first average rank by a wide margin and the best average performance compared to other under-sampling models.

Apart from the intrinsic effectiveness of the proposed approach in reducing the samples from the majority class, a very interesting observation is seen in these experiments. As discussed previously in Sect. "Datasets" and Table 2, there are two highly imbalance datasets in the experiments, namely kddcup-buffer_overflow_vs_back and shuttle_2_vs_5. Fortunately, the proposed approach shows a significant performance on these datasets which is more than 10 percent better than the next competing approach. This shows the effectiveness, scalability and generalization of the proposed approach on highly imbalanced data.

Table 14 Average F-measure of different state-of-the-art under-sampling and over-sampling methods as compared with the proposed approach

Name	BIDC1 [51]	BIDC2 [51]	K-US [50]	NB-Rec [49]	DB_US [48]	Proposed
ecoli2	0.81	0.83	0.59	0.91	**0.96**	0.90 ± 0.142
ecoli3	NA	NA	0.71	0.63	0.71	**0.86 ± 0.195**
glass0	0.63	0.66	0.61	0.61	0.61	**0.79 ± 0.116**
new-thyroid1	NA	NA	0.73	0.92	**1.00**	0.89 ± 0.206
Pima	NA	NA	**0.73**	0.58	0.71	0.63 ± 0.086
segment0	**1.00**	**1.00**	0.83	**1.00**	**1.00**	0.98 ± 0.017
wisconsin	NA	NA	**0.97**	**0.97**	**0.97**	0.95 ± 0.053
Average F-measure	0.81(4)	0.83(3)	0.74(6)	0.80(5)	0.85(2)	**0.86(1)**

Table 15 The Wilcoxon test on the proposed multi-manifold method compared with other methods

	Original	RUS	NCL [28]	OSS [27]	CNN [34]	ENN [35]	CBU [31]	PUMD [13]
p-value	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.01

Comparison with state-of-the-art under/over sampling approaches

It should be noted that in Tables 11, 12, 13, the PUMD method is one of the recent under-sampling methods. However, in this section, the results of the proposed multi-manifold approach are compared with some other state-of-the-art under-sampling methods such as DB_US [48], NB-Rec [49], K-US [50], state-of-the-art over-sampling methods such as BIDC1 [51] and BIDC2 [51] on KEEL and UCI data based on the F-measure and reported in Table 14. Other settings are the same as the previous experiments described. The average performance of each under-sampling and over-sampling model are shown separately in the last row of Table 14. The simulation results show that the F-measure of the proposed method has the best average rank compared to the other mentioned methods. As mentioned in the previous experiments, the proposed method can obtain significant results on very imbalanced data.

Statistical analysis by Wilcoxon test

In this research, Wilcoxon's non-parametric signed rank test is used for statistical evaluation of results. The mentioned test investigates the significant difference of F-measure between the proposed multi-manifold method and other under-sampling approaches according to Table 15. In this test, the hypotheses H0 and H1 are defined as follows:

H0: There is no significant difference between the two methods.

H1: There is a significant difference between the two methods.

The p-value of the Wilcoxon test is reported for each pair of methods and can be seen based on the F-measure evaluation criteria according to Table 15.

As it is clear in Table 15, all p-value values are so lower than $\alpha = 0.05$ and the H0 condition is rejected. Therefore, there is a significant difference between the proposed

multi-manifold method and other under-sampling methods. This means that other methods have not performed better than the proposed method, and the proposed method is significantly superior.

Evaluation and discussions on kddcup network intrusion detection dataset

One of the applications that can show the effectiveness of the proposed method specially on highly imbalanced data is the problem of network intrusion detection. For this purpose, kddcup datasets are incorporated in the research. Different versions of kddcup data are shown in Table 2. The highest imbalanced ratios of these datasets which are included in Table 2 are 73, 75 and 100, which are very significant.

In Table 16, the average F-measure of different under-sampling methods and the proposed approach on these data can be seen. According to these evaluations, the proposed method has considerably performed better than other NN_HIDC [45], NBUS [13], CRIUS [1] and RBUS [48] methods. The average efficiency of the proposed method is in the first place compared to other methods.

It should be noted that the proposed multi-manifold-based under-sampling method cannot be implemented when the number of minority class samples in a data set is less than or equal to the number of features of that class. This constraint is imposed by LPP and NPE manifold learning approaches. Therefore, in this situation, we are forced to use the single-manifold method on that dataset. Therefore, in Table 16, a column titled "manifold model" is added, which shows the type of manifold learning approach (i.e. multi-manifold/single-manifold). According to Table 16, the proposed single-manifold method is evaluated on kddcup-land_vs_portsweep, kddcup-land_vs_satan and kddcup-rootkit-imap_vs_back datasets. PCA manifold learning is used for this purpose. The multi-manifold-based under-sampling method is applied on other kddcup datasets.

Evaluations on artificial datasets

In addition to the evaluations performed on the KEEL and UCI datasets, some experiments are performed on some imbalanced artificially created datasets. These evaluations can show the stability of the proposed method on datasets with different levels of

Table 16 Average F-measure of different under-sampling methods on kddcup datasets

Dataset	Manifold model	RBUS [48]	CRIUS [1]	NBUS [13]	NN_HIDC [45]	Proposed
kddcup-buffer_overflow_vs_back	multi-manifold	0.66	0.83	0.43	1	**1 ± 0**
kddcup-rootkit-imap_vs_back	single-manifold	0.83	0.86	0.78	1	**1 ± 0**
kddcup-guess_passwd_vs_satan	multi-manifold	0.00	0.00	0.00	0.99	**0.99 ± 0.027**
kddcup-land_vs_portsweep	single-manifold	0.96	0.23	0.23	0.23	**1 ± 0**
kddcup-land_vs_satan	single-manifold	0.97	0.97	0.97	1	**1 ± 0**
Average F-measure		0.685(3)	0.578(4)	0.482(5)	0.844(2)	**0.998(1)**

imbalance. Two synthetic datasets are generated. The first model uses uniform distribution function in a specific interval [47]. Figure 6 shows 4 synthetic datasets generated using the first model. The second model uses the synthetic dataset of Two Moons [1]. Figure 7 shows 4 synthetic data sets generated using the second model. Each data contains two features, denoted by x_1 and x_2. In the first and second models of imbalanced synthetic data generation, the imbalance ratio is from the set {1, 5, 10, 20}. In the following, the generation process of both models of imbalanced artificial data is described.

The first model: The process of generating synthetic data sets from the first model is as follows: The minority class includes 100 data samples, which are shown with blue circles in Fig. 6. The values of the first feature (x_1) and the values of the second feature (x_2) are randomly extracted from a uniform distribution. The values of x_1 are selected from the interval [50,100] while the values of x_2 are selected from the interval [0,100]. The majority class includes $N_{majority}$ of data samples, which are indicated by red circles in Fig. 6. The $N_{majority}$ variable takes values from the set {100, 500, 1000, 2000}. The values of the first attribute (x_1) and the values of the second attribute (x_2) of the majority class are created in the same way as the minority class, with the difference that the values of x_1 are extracted from the interval [0,50], while the values of x_2 are

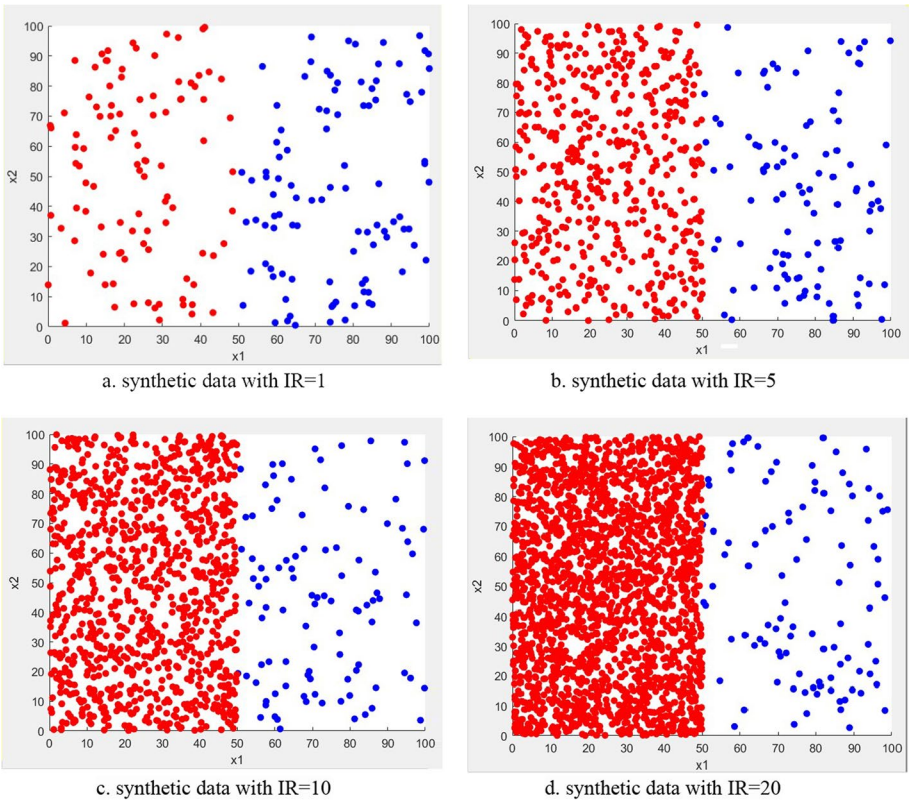

Fig. 6 Synthetic datasets generated using the uniform model

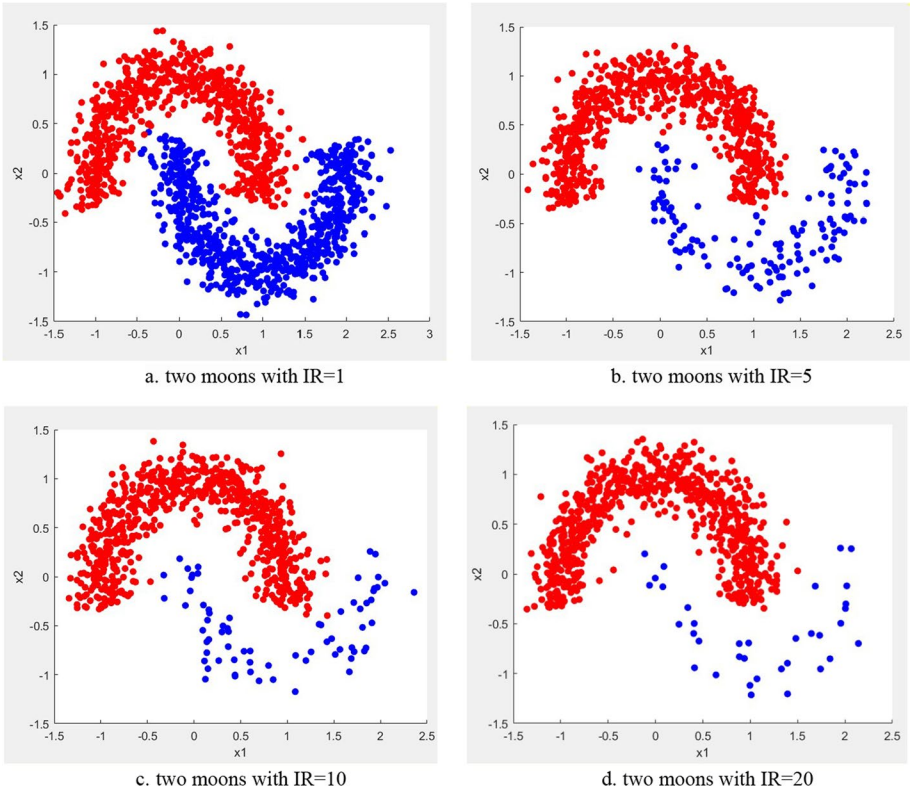

Fig. 7 Synthetic datasets generated using the second model

extracted from the interval of values [0,100]. The number of samples in the majority class controls the imbalance ratio (IR) in the generated imbalanced dataset.

a. If $N_{majority} = 100$, Fig. 6-a, IR = 1.
b. If $N_{majority} = 500$, Fig. 6-b, IR = 5.
c. If $N_{majority} = 1000$, Fig. 6-c, IR = 10.
d. If $N_{majority} = 2000$, Fig. 6-d, IR = 20.

The second model: The process of generating the Two Moons data set is as follows: the majority class contains 700 data samples, which are marked with red circles in Fig. 7. The samples of the majority class that make up the upper moon are created with the center (0, 0). Minority class includes $N_{minority}$ data samples, which are shown with blue circles in Fig. 7. The $N_{minority}$ variable takes values from the set {35, 70, 140, 700}. Minority class samples that form the lower moon are created with center (0, 1). The number of minority class samples controls the imbalance ratio (IR) in the generated imbalanced Two Moons dataset.

a. If $N_{minority} = 700$, Fig. 7-a, the IR = 1.
b. If $N_{minority} = 140$, Fig. 7-b, the IR = 5.

Table 17 Average F-measure of SVM classification on the uniformly created artificial datasets

Dataset	Original data	Proposed
synthetic_data_IR = 1	0.98 ± 0.050	0.98 ± 0.050
synthetic_data_IR = 5	0.91 ± 0.243	**0.93 ± 0.219**
synthetic_data_IR = 10	0.95 ± 0.135	0.95 ± 0.127
synthetic_data_IR = 20	0.87 ± 0.258	**0.88 ± 0.260**

Table 18 Average F-measure of 3NN classification on the uniformly created artificial datasets

Dataset	Original data	Proposed
synthetic_data_IR = 1	0.98 ± 0.050	0.98 ± 0.050
synthetic_data_IR = 5	0.88 ± 0.295	**0.95 ± 0.065**
synthetic_data_IR = 10	0.90 ± 0.246	**0.96 ± 0.090**
synthetic_data_IR = 20	0.83 ± 0.311	**0.90 ± 0.162**

Table 19 Average F-measure of CART classification on the uniformly created artificial datasets

Dataset	Original data	Proposed
synthetic_data_IR = 1	0.97 ± 0.068	0.97 ± 0.056
synthetic_data_IR = 5	0.91 ± 0.185	0.91 ± 0.184
synthetic_data_IR = 10	0.90 ± 0.217	0.90 ± 0.217
synthetic_data_IR = 20	0.85 ± 0.302	0.85 ± 0.302

Table 20 Average F-measure of SVM classification on the Two Moons artificial datasets

Dataset	Original data	Proposed
imb_two_moons_IR = 1	0.98 ± 0.027	0.98 ± 0.027
imb_two_moons_IR = 5	0.98 ± 0.029	**0.99 ± 0.025**
imb_two_moons_IR = 10	0.98 ± 0.034	**0.99 ± 0.029**
imb_two_moons_IR = 20	0.91 ± 0.182	**1 ± 0**

c. If $N_{minority} = 70$, Fig. 7-c, the IR = 10.
d. If $N_{minority} = 35$, Fig. 7-d, the IR = 20.

The results of the evaluations (by SVM, 3NN and CART classifiers) on the imbalanced artificially created datasets with uniform distribution are shown in Tables 17, 18, 19. The average F-measure is reported in two situations of the original data and the under-sampled data. The results of experiments with 3NN classification show that when the imbalance coefficient increases, the proposed method is more effective, and the average F-measure of the proposed method is increased compared to the original data state.

The results of the propose approach on Two Moons imbalanced data are illustrated in Tables 20, 21, 22. The average F-measure is reported in two situations of

Table 21 Average F-measure of 3NN classification on the Two Moons artificial datasets

Dataset	Original data	Proposed
imb_two_moons_IR = 1	0.99 ± 0.019	0.99 ± 0.017
imb_two_moons_IR = 5	0.99 ± 0.024	0.99 ± 0.024
imb_two_moons_IR = 10	0.98 ± 0.034	0.98 ± 0.034
imb_two_moons_IR = 20	0.98 ± 0.043	**0.99 ± 0.043**

Table 22 Average F-measure of CART classification on the Two Moons artificial datasets

Dataset	Original data	Proposed
imb_two_moons_IR = 1	0.80 ± 0.198	**0.88 ± 0.120**
imb_two_moons_IR = 5	0.90 ± 0.165	**0.96 ± 0.069**
imb_two_moons_IR = 10	0.95 ± 0.074	**0.96 ± 0.075**
imb_two_moons_IR = 20	0.79 ± 0.318	**0.84 ± 0.265**

Table 23 Average F-measure of the proposed method with three different sample weighting models

Dataset	Model 1 weight = marg	Model 2 weight = -cent	Model 3 weight = marg-cent
ecoli1	0.85 ± 0.153	0.85 ± 0.153	**0.85 ± 0.152**
ecoli2	0.90 ± 0.105	0.90 ± 0.123	**0.91 ± 0.105**
ecoli3	0.86 ± 0.196	0.86 ± 0.196	**0.86 ± 0.195**
ecoli4	0.92 ± 0.171	0.92 ± 0.171	**0.92 ± 0.170**
ecoli0147vs56	0.89 ± 0.119	0.88 ± 0.141	**0.89 ± 0.119**
ecoli034_5	0.88 ± 0.151	0.88 ± 0.151	**0.88 ± 0.151**
ecoli0147_2356	0.81 ± 0.159	0.81 ± 0.159	**0.81 ± 0.159**
glass0	0.78 ± 0.076	0.78 ± 0.076	**0.79 ± 0.116**
glass0123456	0.92 ± 0.154	0.92 ± 0.154	**0.92 ± 0.153**
kddcup-buffer_overflow_vs_back	1.00 ± 0.000	1.00 ± 0.000	**1 ± 0**
new-thyroid1	0.93 ± 0.146	0.93 ± 0.146	**0.98 ± 0.075**
page-blocks-1-3_vs_4	0.96 ± 0.080	0.96 ± 0.080	**096 ± 0.080**
Pima	0.65 ± 0.071	0.65 ± 0.071	**0.65 ± 0.067**
segment0	0.98 ± 0.017	**0.99 ± 0.018**	0.98 ± 0.017
shuttle_2_vs_5	1.00 ± 0.000	1.00 ± 0.000	**1 ± 0**
vehicle2–1	**0.98 ± 0.032**	0.97 ± 0.033	0.97 ± 0.030
vowel0	0.90 ± 0.120	0.90 ± 0.120	**0.90 ± 0.104**
wisconsin	0.97 ± 0.050	0.97 ± 0.050	**0.97 ± 0.050**

the original data and the under-sampled data. When IR = 20, the average F-measure has increased from 91 to 100%. The results of experiments with CART classification also show that the results of the proposed method are better than the original data.

Discussion on marginality and centrality criteria

To discuss the effect of the proposed marginality and centrality degrees, the average F-measure of the proposed method is compared in three different weighting models

in Table 23. In order to weight the samples, in the first model (i.e. the first column), only the marginality degree is used. In the second model (i.e. the second column), only the degree of centrality is applied. In the third model (i.e. third column), the linear combination of marginality and centrality is experimented. The results of the experiments show that in most of the datasets, the use of the linear combination of marginality and centrality is more effective than other methods of weighting. Only for segment0 dataset, the results of the second model are better than other models, and for the vehicle2-1 dataset, the results of the first model are the best.

Computational complexity analysis

The proposed approach includes mapping stages, traditional centrality and marginality calculation, weighted centrality and marginality calculation and gradual removal of samples. Assume that n is the number of samples and D is the dimension of data. Since manifolds are trained in parallel in the proposed method, the computational complexity of the mapping part is equal to the highest computational complexity of the manifolds used. Therefore, we assume that in the worst case, the computational complexity of the mapping part is equal to that of PCA which is $O(D^3)$ [52]. On the other hand, the complexity of traditional centrality and marginality calculation section depends on the complexity of k nearest neighborhoods selection. The

Table 24 The average execution time of the proposed method in 5 experimental repetitions of 10-fold CV

Name	#Attributes	#Examples	Run time (second)
ecoli1	7	336	1.52
ecoli2	7	336	1.53
ecoli3	7	336	1.43
ecoli4	7	336	1.44
ecoli0147vs56	6	332	1.59
ecoli034_5	7	200	0.85
ecoli0147_2356	7	336	1.55
glass0	9	214	0.77
glass0123456	9	214	0.79
kddcup-buffer_overflow_vs_back	41	2233	6.72
kddcup-rootkit-imap_vs_back	41	2225	1.00
kddcup-guess_passwd_vs_satan	41	1642	8.40
kddcup-land_vs_portsweep	41	1061	0.52
kddcup-land_vs_satan	41	1610	0.82
new-thyroid1	5	215	0.75
page-blocks-1-3_vs_4	10	472	1.92
pima	8	768	3.34
segment0	19	2308	8.87
shuttle_2_vs_5	9	3316	43.09
vehicle2–1	18	846	2.34
vowel0	13	988	6.25
wisconsin	9	683	2.27

computational complexity of the the selection is generally O(nDk) where $k << n$. Therefore, the complexity of calculating traditional centrality and marginality becomes $3 \times O(nDk)$, in which 3 indicates the number of mappings. The computational complexity of computing weighted centrality and marginality depends on the number of samples, so the order becomes $n \times 3 \times O(nDk)$. The complexity of the gradual under-sampling part is O(1) because it does not depend on the size of the problem. Finally, the computational complexity of the proposed multi-manifold approach is $O(D^3) + O(n \times 3 \times nDk) + O(1)$ in the worst case.

In Table 24, the average execution time of the proposed method in 5 experimental repetitions of 10-fold CV is shown. The average execution time indicates that the execution time of the proposed method is consistent with the theoretical analysis of computational complexity and follows the polynomial time order.

Conclusions and future work

Class imbalance is an important issue that is tried to be handled in this paper. This issue can be solved via under-sampling, and there are many under-sampling strategies in the literature. This paper introduces a multi-manifold learning-based technique to evaluate the importance of the data points. Different manifold learning strategies are used and assessed using a criterion based on information loss. Three linear unsupervised manifold learning methods are used in order to avoid high computing complexity. The traditional centrality and marginality degrees of the samples are computed on the manifolds and weighted by the corresponding score after computing the optimality score of each manifold. The suggested method of gradual removal attempts to balance the classes without causing the F measure to decrease on the validation dataset. The proposed approach is assessed on 22 imbalanced datasets from the KEEL and UCI repositories with different and considerable imbalance ratios using various classification metrics. The findings show that the proposed method outperforms other comparable approaches, especially on highly imbalanced problems.

The weakness of the proposed method is that if the number of minority class examples in a dataset is less than or equal to the number of features of that class, the multi-manifold-based approach cannot be implemented. LPP and NPE manifold learning methods must have a minimum number of samples to be applicable. Therefore, it may be desirable to use other mapping approaches that do not have this constraint. Also, the proposed multi-manifold method performs poorly when the overlap of the classes increases. On the other hand, when the number of samples (i.e. n) increases, for example, $n > 5000$, the execution time increases dramatically. Therefore, for a large number of samples, more powerful hardware may be required. Supervised nonlinear manifold learning methods including Neighborhood Components Analysis (NCA), Maximally Collapsing Metric Learning (MCML) and Large-Margin Nearest Neighbor Metric Learning (LMNN) were omitted due to computational complexity and more execution time. Some other limitations of the proposed method are:

– Manifold learning methods such as LLE and IsoMap, do not produce out-of-sample mapping matrix.
– Manifold learning methods such as Factor Analysis (FA) produce NaN values in the transformation matrix.
– Manifold learning methods such as Locally Linear Embedding (LLE) and IsoMap reduce the number of samples after mapping.

All these limitations led us to use unsupervised manifold learning methods such as PCA, LPP, and NPE.

However, the approach has some costs that are not negligible. The first weakness is the relative loss of precision. Precision reduction is unavoidable when we target to increase the classification rate of minority classes, but some future research is required to mitigate this reduction and maybe improve the classification measures much more. The second weakness is the relatively higher computational costs due to applying several manifold learning approaches and computing the degrees of centrality and marginality on each manifold. Although the experiment tries to reduce the computational costs and increase the applicability of the approach by applying three linear unsupervised manifold learning approaches, further improvements are necessary in this regard.

Acknowledgements
Not applicable.

Author contributions
TF implemented the approach and did the analyses and also prepared the initial draft. MHM proposed the original idea and supervised the process and proofread the manuscript. HT is the advisor and helped completion of the manuscript. All authors read and approved the final manuscript.

Funding
The authors declare that no funding was received for this research.

Availability of data and materials
The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Received: 13 May 2023 Accepted: 26 September 2023
Published online: 06 October 2023

References

1. Hoyos-Osorio J, et al. Relevant information undersampling to support imbalanced data classification. Neurocomputing. 2021;436:136–46.
2. Koziarski M. CSMOUTE: Combined Synthetic Oversampling and Undersampling Technique for Imbalanced Data Classification. in 2021 International Joint Conference on Neural Networks (IJCNN). 2021.
3. Tran TC, Dang TK. Machine Learning for Prediction of Imbalanced Data: Credit Fraud Detection. in 2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM). 2021.

4. Yan M, et al. A lightweight weakly supervised learning segmentation algorithm for imbalanced image based on rotation density peaks. Knowl-Based Syst. 2022;244: 108513.
5. Yeung M, et al. Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation. Comput Med Imaging Graph. 2022;95: 102026.
6. Lin YD, et al. Machine Learning With Variational AutoEncoder for Imbalanced Datasets in Intrusion Detection. IEEE Access. 2022;10:15247–60.
7. Shahraki A, et al. A comparative study on online machine learning techniques for network traffic streams analysis. Comput Netw. 2022;207: 108836.
8. Ghorbani M, et al. RA-GCN: Graph convolutional network for disease prediction problems with imbalanced data. Med Image Anal. 2022;75: 102272.
9. Ning Z, et al. BESS: Balanced evolutionary semi-stacking for disease detection using partially labeled imbalanced data. Inf Sci. 2022;594:233–48.
10. Zhao H, et al. Severity level diagnosis of Parkinson's disease by ensemble K-nearest neighbor under imbalanced data. Expert Syst Appl. 2022;189: 116113.
11. Xu Z, et al. A cluster-based oversampling algorithm combining SMOTE and k-means for imbalanced medical data. Inf Sci. 2021;572:574–89.
12. Liu J. A minority oversampling approach for fault detection with heterogeneous imbalanced data. Expert Syst Appl. 2021;184: 115492.
13. Xie X, et al. A novel progressively undersampling method based on the density peaks sequence for imbalanced data. Knowl-Based Syst. 2021;213: 106689.
14. Fattahi M, et al. Improved cost-sensitive representation of data for solving the imbalanced big data classification problem. J Big Data. 2022;9(1):1–24.
15. Fattahi M, et al. Locally alignment based manifold learning for simultaneous feature selection and extraction in classification problems. Knowl-Based Syst. 2023;259:110088.
16. Galar M, et al. A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. IEEE Trans Syst Man Cybern. 2012;42(4):463–84.
17. Wang S, Yao X. Diversity analysis on imbalanced data sets by using ensemble models. In: 2009 IEEE Symposium on Computational Intelligence and Data Mining. 2009.
18. Chawla NV. Philip Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique. J Artif Intell Res. 2002;16:21–357.
19. Wang B. Imbalanced data set learning with synthetic samples. in: Proc. IRIS Machine Learning Workshop, 2004. 19.
20. Chawla NV, Hall LO, Bowyer KW. Smoteboost: improving prediction of the minority class in boosting. in: European Conference on Principles of Data Mining and Knowledge Discovery, Springer. 2003: p. 107–119.
21. Jimenez-Castaño C, Orozco-Gutierrez A. Enhanced automatic twin support vector machine for imbalanced data classification. Pattern Recogn. 2020;89:107442.
22. Li F, Zhang X, Du C, Xu Y, Tian Y-C. Cost-sensitive and hybrid-attribute measure multi-decision tree over imbalanced data sets. Inf Sci. 2018;422:242–56.
23. Sun Z, et al. A novel ensemble method for classifying imbalanced data. Pattern Recogn. 2015;48(5):1623–37.
24. Barandela R, Sánchez JS. New applications of ensembles of classifiers. Pattern Anal Appl. 2003;6(3):245–56.
25. Seiffert C, Van Hulse J, Napolitano A. Rusboost: a hybrid approach to alleviating class imbalance. IEEE Trans Syst Man Cybern A Syst Hum. 2010;40(1):185–97.
26. Mani I. Knn approach to unbalanced data distributions: a case study involving information extraction. In: Proc. of International Conference on Machine Learning, Workshop Learning from Imbalanced Data Sets, 2003. 126.
27. Kubat M. Addressing the curse of imbalanced training sets:one-sided selection. in: Proceedings of the 14th International Conference on Machine Learning, Nashville, TN, USA; 1997: p. 179–186.
28. Laurikkala J, Barahona P, Andreassen S (Eds). Improving identification of difficult small classes by balancing class distribution. In: Artificial Intelligence in Medicine, 2001: p. 63–66.
29. Kang Q, Chang X, Li S, Zhou M. A noise-filtered under-sampling scheme for imbalanced classification. IEEE Trans Cybern. 2017;47(12):4263–74.
30. Chen C. Clustering-based binary-class classification for imbalanced data sets. in: Proceedings of 2011 IEEE International Conference on Information Reuse and Integration, IEEE, Las Vegas, NV, USA, 2011: p. 384–389.
31. Lin WC, Hu YH, Jhang JS. Clustering-based undersampling in class-imbalanced data. Inform Sci. 2017;409–410:17–26.
32. Yen SJ. Cluster-based under-sampling approaches for imbalanced data distributions. Expert Syst Appl. 2009;36(3):5718–27.
33. Tomek I. Two modifications of CNN. IEEE Trans Syst Man Cybern A Syst Hum. 1976;6(11):769–72.
34. Hart P. The condensed nearest neighbor rule. IEEE Trans Inform Theory. 1968;14(3):515–6.
35. Tomek I. An experiment with the edited nearest-neighbor rule. IEEE Trans Syst Man Cybern A Syst Hum. 1976;6(6):448–52.
36. Yang L, et al. Natural neighborhood graph-based instance reduction algorithm without parameters. Appl Soft Comput. 2018;70:279–87.
37. Hamidzadeh J, Monsefi R, Yazdi HS. LMIRA: Large Margin Instance Reduction Algorithm. Neurocomputing. 2014;145:477–87.
38. Pang X, Xu C, Xu Y. Scaling KNN multi-class twin support vector machine via safe instance reduction. Knowl-Based Syst. 2018;148:17–30.
39. Hamidzadeh J, Kashefi N, Moradi M. Combined weighted multi-objective optimizer for instance reduction in two-class imbalanced data problem. Eng Appl Artif Intell. 2020;90: 103500.
40. Deng X. IEEE 35th International Performance Computing and Communications Conference. IPCCC, IEEE. 2016;2016:1–8.

41. Ofek N, Stern R, Shabtai A. Fast-CBUS: A fast clusteringbased undersampling method for addressing the class imbalance problem. Neurocomputing. 2017;243:88–102.
42. Zhang X. Unbalanced data classification algorithm based on clustering ensemble under-sampling. Comput Sci. 2015;42(11):63–6.
43. Ng WWY, Yeung DS, Yin S, Roli F. Diversified sensitivity-based undersampling for imbalance classification problems. IEEE Trans Cybern. 2015;45(11):2402–12.
44. Hamidzadeh J, Monsefi R, Sadoghi Yazdi H. IRAHC: Instance reduction algorithm using hyperrectangle clustering. Pattern Recogn. 2015;48(5):1878–89.
45. Huang ZA, et al. A neural network learning algorithm for highly imbalanced data classification. Inform Sci. 2022;612:496–513.
46. Koziarski M. Radial-based undersampling for imbalanced data classification. Pattern Recogn. 2020;102:107262.
47. Sun B, et al. Radial-based undersampling approach with adaptive undersampling ratio determination. Neurocomputing. 2023;553: 126544.
48. Mayabadi S, Saadatfar H. Two density-based sampling approaches for imbalanced and overlapping data. Knowl-Based Syst. 2022;241: 108217.
49. Vuttipittayamongkol P, Elyan E. Neighbourhood-based undersampling approach for handling imbalanced and overlapped data. Inf Sci. 2020;509:47–70.
50. Nwe MM, Lynn KT. KNN-Based Overlapping Samples Filter Approach for Classification of Imbalanced Data. In: Lee R, editor. Software Engineering Research, Management and Applications. Cham: Springer International Publishing; 2020. p. 55–73.
51. Zhai J, Qi J, Shen C. Binary imbalanced data classification based on diversity oversampling by generative models. Inf Sci. 2022;585:313–43.
52. Chen HE, Weiqi L, Jane W. A Low complexity quantum principal component analysis algorithm. arXiv, 2021.
53. Shi-Jie Pan L-CW, Hai-Ling L, Yu-Sen W, Su-Juan Q, Qiao-Yan W, Fei G. Quantum algorithm for Neighborhood Preserving Embedding. arXiv, 2021.

Publisher's Note