A Novel Oversampling Technique Based on the Manifold Distance for Class Imbalance Learning

Yinan Guo, Botao Jiao and Lingkai Yang

School of Information and Control Engineering,
China University of Mining and Technology,
Xuzhou, China
E-mail: nanfly@126.com, 461375470@qq.com, yanglk@cumt.edu.cn

Jian Cheng*

China Coal Research Institute,
Beijing 100013, China
E-mail: chengjian@cumt.edu.cn
*Corresponding author

Shengxiang Yang

De Montfort University
Leicester LE1 9BH, United Kingdom
E-mail:syang@dmu.ac.uk

Fengzhen Tang

Shenyang Institute of Automation
Shenyang, China
E-mail:tangfengzhen@sia.cn

Abstract: Oversampling is a popular problem-solver for class imbalance learning by generating more minority samples to balance the dataset size of different classes. However, resampling in original space is ineffective for the imbalance datasets with class overlapping or small disjunction. Based on this, a novel oversampling technique based on manifold distance is proposed, in which a new minority sample is produced in terms of the distances among neighbors in manifold space, rather than the Euclidean distance among them. After mapping the original data to its manifold structure, the overlapped majority and minority samples will lie in areas easily being partitioned. In addition, the new samples are generated based on the neighbors locating nearby in manifold space, avoiding the adverse effect of the disjoint minority classes. Following that, an adaptive adjustment method is presented to determine the number of the newly generated minority samples according to the distribution density of the matched-pair data. The experimental results on 48 imbalanced datasets indicate that the proposed oversampling technique has the better classification accuracy.

Keywords: class imbalance learning; oversampling; manifold learning; overlapping; small disjunction.

Reference to this paper should be made as follows: Guo, Y., Jiao, B., Yang, L., Cheng, J., Yang, S., and Tang, F. (xxxx) 'A Novel Oversampling Technique Based on the Manifold Distance for Class Imbalance Learning', *International Journal of Bio-Inspired Computation* , Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Yinan Guo is a Professor in China University of Mining and Technology. Her research interests include class imbalance learning, computation intelligence in dynamic and uncertain optimization and their applications.

Jian Cheng is a associate professor in China Coal Research Institute. His research interests include imbalance learning and image processing.

Shengxiang Yang is a Professor in De Montfort University. His research interests include computation intelligence in dynamic and uncertain optimization and their applications.

Fengzhen Tang is an Assistant Researcher in Shenyang Institute of Automation. Her research interests include machine learning and computational neuroscience.

1 Introduction

The distribution of data is skewed in many real-world classification problems, such as fault diagnosis (Cai et al., 2017), face recognition (Soleymani et al., 2018), and fraud detection (Dal et al., 2018). That is, more data belongs to the majority class. However, the events corresponding to the minority samples, such as a failure of a space-shuttle (Shakeel et al., 2017), may result in a huge product cost. Hence, to accurately label the data in the minority class attracts the increasing interest. In particular, the small data size of the minority samples provide the limited information for class imbalance learning, causing the wrong classification boundary. Moreover, the implicit problems like class overlapping and small disjunction also bring severe hindrance to the performance of a classifier (Krawczyk, 2016). The cross-distributed minority and majority samples easily make the classifier overfitting or underfitting. The imbalanced data with small disjunction contains several minority subclusters lying far away each other, forming the complicated classification boundary.

Previous studies on class imbalance learning can be categorized into data-, algorithm-, or hybrid-level strategies (Loezer et al., 2020; Kang et al., 2018; Wang et al., 2016; Cheng et al., 2019). Among them, data-level strategies balance the distribution of imbalanced dataset by oversampling the minority samples or undersampling the majority samples. Synthetic minority oversampling technique (SMOTE) (Arafat et al., 2019) created the new synthetic samples along the line between a minority sample and its nearest neighbors. Following that, many improved SMOTE-based algorithms were proposed. Borderline-synthetic minority oversampling technique (Borderline-SMOTE)(Han et al., 2005) only oversampled the minority samples nearby the classification boundary. Adaptive synthetic sampling approach(ADASYN) (He et al., 2008) and majority-weighted minority oversampling technique (MWMOTE) (Barua et al., 2013) both assigned the weights for the hard-to-learn minority samples in terms of the Euclidean distances to the nearest majority samples, and then generated a new minority sample in any minority sub-cluster. In order to identify the data distribution with small disjunction in original space, some clustering-based data-level methods were presented (Lin et al., 2017). Lim (Lim et al., 2016) proposed evolutionary cluster-based synthetic oversampling ensemble (ECO-Ensemble), in which the clustering methods was introduced to oversample the synthetic minority samples in each sub-cluster. However, the above oversampling methods generate the minority samples in terms of the data structure in original space, which may be labelled incorrectly, especially for the imbalanced data with class overlapping and small disjunction.

Manifold learning maps high-dimensional data into the low-dimensional manifold space, which makes the cross-distributed imbalanced data in original space more easily being partitioned and creates the more qualified samples. Moreover, for the minority samples lying in disjoint sub-areas, a new sample created along the line connecting these sub-areas in original space may be incorrectly labelled. Different from it, the minority samples that produce based on the neighbors chosen from the disjoint sub-area by their manifold distances are close to one of the minority sub-clusters. Apparently, the manifold distance is good for retaining the original structure of the imbalanced data and provides a rational criterion for finding the rational neighbors of the minority samples lying in overlapping or disjoint area. Based on this, Bellinger, et al (Bellinger et al., 2017) found the neighbors in manifold space, and then mapped them back to original space. The new instances, subsequently, were generated by oversampling technique and employed to classify the highdimensional imbalanced datasets. However, the forward and backward mapping between manifold space and original space increases the computational cost.

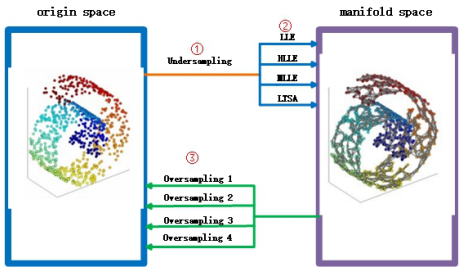
To address the problems, a novel oversampling technique based on the manifold distance (OTEMD) is proposed to produce a new minority sample in terms of the distances among the neighbors in manifold space. In order to decrease the computation complexity for extracting the neighbors, the redundant majority samples are firstly removed by undersampling. Following that, the nearest samples are selected as the neighbors in terms of their manifold distances, and finally the new minority samples are generated based on the matched-pair data by oversampling. In general, no matter how dense the matched-pair data is, the same number of the new samples are generated by oversampling, which makes the difference of the data density among the samples more serious, causing the weak classification performance in the sparse areas. Thus, a distance mechanism is introduced to produce more samples for the matched-pair data with the sparse distribution and close to the classification boundary. Based on this, four kinds of oversampling strategies derived from borderline-SMOTE-1 or borderline-SMOTE-2, expressed by OS_1, OS_2, OS_3 and OS_4, are presented, with the purpose of drawing the more accurate classification lines with the less computation complexity.

2 An Oversampling Technique Based On The Manifold Distance

OTEMD provides a generic framework for generating a synthetic minority sample on the basis of the neighbors selected by their manifold distance. Three key issues are undersampling, extracting the nearest neighbors based on manifold distance, and oversampling. Undersampling strategy is implemented before the data-mapping process, with the purpose of reserving more representative majority samples and reducing the computation complexity of manifold learning. After

Copyright © 201X Inderscience Enterprises Ltd.

Figure 1: Flowchart of OTEMD



mapping the samples from original space to manifold space, the neighbors of each minority sample are selected to build the matched-pair data in terms of the manifold distance among the samples. Following that, a new minority sample is generated between the minority one and its nearest neighbors.

2.1 Undersampling

The goal of undersampling is to remove the data with redundant or useless information from the majority class. A valuable sample is the one that belongs to the different class or has the sparse distribution with its neighbors. To this end, a majority sample is removed if its neighbors all belong to the majority class and distribute densely.

Suppose the original imbalanced dataset expressed by O contains n_l majority samples and n_s minority data. The totally data size satisfies $n_o = n_l + n_s$. For each sample, the average distance between it and k_1 neighbors is called the distribution density, denoted as d_i. Let o_i and o_{ip} be ith sample and its pth neighbor.

$$d_i = \frac{1}{k_1}\sum_{p=1}^{k_1}||o_i - o_{ip}||_2 \tag{1}$$

Let $d = \frac{1}{n_o}\sum_{i=1}^{n_o} d_i$ be the average distribution density for all samples, and $\alpha \in (0,1)$ is a preset constant. The distribution of samples is dense if they are close to their neighbors. Based on this, the majority samples satisfying $d_i < d * \alpha$ are considered as the redundant ones.

Denote u_i and y_i^u are ith sample and its label after undersampling. The dataset after undersampling expressed by U is composed of u_l majority samples and u_s minority ones, and its data size is n_u. The detailed algorithm steps of undersampling are shown as Algorithm 1.

Algorithm 1 Undersampling

Input: The original dataset(O, Y^o), k_1, and α.
Output: The dataset after undersampling (U, Y^u).

For each sample $o_i \in O$,
  Extract its k_1 neighbors,
  Evaluate its distribution density d_i,
End
Calculate the sum of the average distance for all samples,
For $i = 1, 2, ..., n_o$
  if $y_i^o = 0$,
  Count the number of the neighbors that belong to the majority class and label as n_{maj},
  if $n_{maj} = k_1$ and $d_i < d * \alpha$,
  Delete ith sample.
End

2.2 Extracting the Neighbors Based on the Manifold Distance

Manifold learning focuses on the low-dimensional structure embedded in the original dataset (Lunga et al., 2013). After mapping the original data to manifold space, the manifold structure is explored and the neighbors are extracted to build the matched-pair data for generating the new minority samples. Based on this, four manifold learning methods, including locally linear embedding (LLE), Hessian-based LLE (HLLE), modified LLE (MLLE), and local tangent space alignment (LTSA) (Zhang et al., 2018), are employed to map the original imbalanced data to manifold space. Among them, LLE found the manifold embedding by preserving the distances unchanged within the neighbors of each sample (Roweis et al., 2000). Different from it, MLLE (Shen et al., 2016) and HLLE employed multiple weights and Hessian-based quadratic to address the regularization. LTSA prefers to find the embedding that aligns in the tangent space. In this following part, LLE that employed in OTEMD is illustrated in detail.

At first, each sample is reconstructed by its nearest neighbors, which has the minimum error $\varepsilon(W)$. Assume that there are k_2 neighbors. w_{ij} represents the relationship between ith sample and its neighbors, satisfying $\sum_j^{k_2} w_{ij} = 1$. $w_{ij} = 0$ if u_j isn't the nearest neighbor of u_i.

$$\varepsilon(W) = \sum_{i=1}^{n_u}(u_i - \sum_{j}^{k_2} w_{ij}u_j)^2 \tag{2}$$

Suppose M is the dataset in manifold space. Under the obtained W, the manifold structure is explored by minimizing $\phi(M) = \sum_{i=1}^{n_u}(m_i - \sum_j^{k_2} w_{ij}m_j)^2$. Based on M, the matched-pair data are built by exploring the neighbors of each minority sample. The detailed algorithm steps of extracting the neighbors in manifold space are shown in Algorithm 2.

For an imbalanced dataset, the overlapping samples that belong to different classes and locate closely may generate the new samples incorrectly labelled, resulting in the inaccurate classification boundary. To solve the problem, manifold learning that projects the original data to the low-dimensional space is introduced to find the more easily partitioned manifold structure for the inseparable data by the kernel trick in SVM or principal component analysis (Kang et al., 2014). In addition, the samples generated by two minority sub-clusters lying in the disjoint areas are normally grouped into

Algorithm 2 Extracting the neighbors in manifold space

Input: The dataset U; k_2 ; n_dim; the number of the neighbors in manifold space k_3

Output: The neighbors in manifold space.

1: Reconstruct each sample $u_i \in U$ by its nearest neighbors.
2: Explore the manifold structure and form the dataset M in manifold space.
3: Found the neighbors for each sample in manifold space.

the minority class. However, they may be mislabeled because the neighbors that selected in terms of the shortest Euclidean distance are located in the irrelevant areas. After reconstructing the data in manifold space, two nearest minority samples in the disjoint areas of original space may be located far away in manifold space. Therefore, only the minority samples belonging to the same sub-cluster can be selected as the neighbors in terms of the manifold distance.

2.3 Oversampling

Given that a minority sample far away from the classification boundary helps nothing to improve the accuracy, a distance mechanism is introduced to form four improved over-sampling strategies based on Borderline-SMOTE. To our best of knowledge, Borderline-SMOTE produces a minority sample based on the matched-pair data near the classification boundary, and the equal number of new samples are created for each matched-pair data. However, more samples may be generated from the neighbors with the dense distribution, which provide redundant, even useless information for the classification, and consume the extra computation resource. Thus, we present a distance mechanism to produce more samples in the sparse areas for the matched-pair data far away from each other, with the purpose of drawing the more accurate classification boundary.

Borderline-SMOTE-1 generates a sample from the minority matched-pair data near the classification boundary in original space. To address class overlapping and small disjunction, the neighbors are selected in terms of their manifold distances. Thereafter, a novel oversampling strategy based on Borderline-SMOTE-1, denoted as OS_1, is presented to create the same amount of minority samples for each minority matched-pair data. Taking the distribution density of the neighbors into account, OS_3 oversampling strategy produces less minority samples from the matched-pair data lying in the dense area. Different from them, OS_2 and OS_4 derived from Borderline-SMOTE-2 generate a new sample from not only the minority matched-pair data, but also the minority-majority ones. More specially, the minority-majority matched-pair data is employed only if the ratio of minority samples to their neighbors is larger than half of the imbalance ratio.

Suppose $IR = \frac{u_l}{u_s}$ is the imbalance ratio of a dataset and $g = u_l - u_s$ is the number of the generated minority samples. Let n_{maj} be the number of neighbors belonging to the majority class and n_G is the number of the matched-pair data. A minority sample satisfying $n_{maj} \in [0.5k_3, k_3)$ is considered as a valuable one near the classification boundary. The corresponding matched-pair data obtained based on the manifold distance is employed to produce a new minority sample. For OS_1 and OS_2, the number of the generated minority ones is same for all matched-pair data, expressed by n_{new}.

$$n_{new} = \frac{g}{n_G} \tag{3}$$

OS_3 and OS_4 generate the different number of the minority samples in terms of the distribution density of the matched-pair data, denoted as p_i. Let d_i^m be the manifold distances of the matched-pair data for ith minority sample.

$$p_i = \frac{d_i^m}{n_G} \sum_{i=1}^{n_G} (d_i^m) \tag{4}$$

There are g minority sample are generated by randomly sampling with replacement for all matched-pair data in terms of p_i. Because the matched-pair data with long distance is easier to be selected multiple times, more minority samples will be generated around them.

3 Experimental Results And Discussion

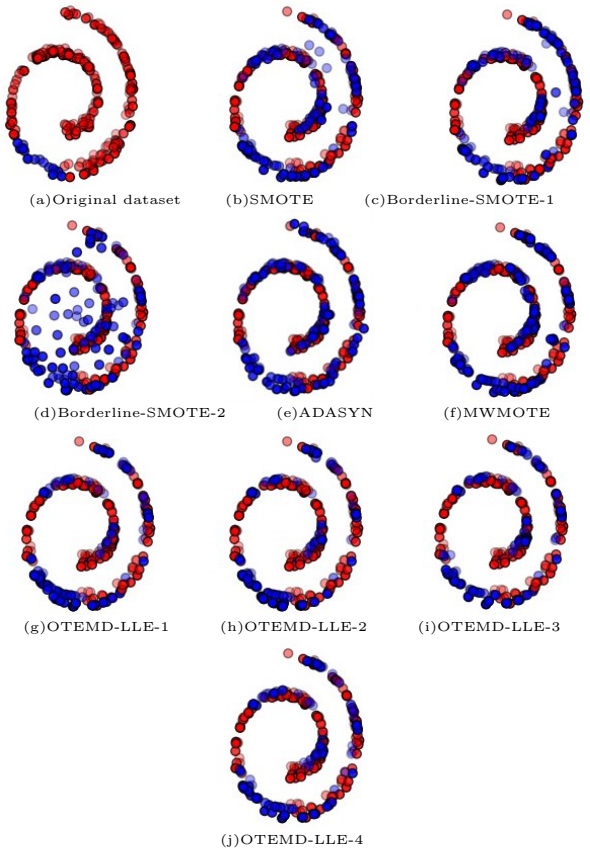
Nine synthetic and thirty-nine imbalanced datasets derived from the UCI machine learning repository are employed to compare the classification performances among SMOTE, Borderline-SMOTE, ADASYN, MWMOTE, ACOSampling, and the proposed oversampling strategy. Decision tree(DT) is used as the basic classifier. To synthetically evaluate the performances of class imbalance algorithms, Accuracy, G-mean, F-measure and the area under the curve (AUC) are employed as the metrics (Lim et al., 2016). Moreover, Wilcoxon paired signed-rank test is carried out to determine whether the proposed method is significantly better than others or not.

3.1 The classification performances on the synthetic Dataset

In order to intuitively compare the oversampling performance of various methods on the imbalanced dataset with small disjunction, 30 randomly generated samples are mixed with the Swiss-roll dataset, which simulate the real-world imbalanced dataset with noise and small disjunction. Fig.2(a) depicts the distribution of the mixed dataset. The red and blue points represent the majority and minority samples in original space,

respectively. The distribution of all samples in original space by various oversampling strategies are shown in Fig.2(b)-(j). Owing to the noise, some samples that newly developed by SMOTE, Borderline-SMOTE-1, and Borderline-SMOTE-2 may lie outside of the surface along the Swiss-roll curve. Apparently, all methods form the satisfied Swiss-roll after oversampling, except for SMOTE because it generates a new sample on the basis of the nearest minority samples by linear interpolation regardless of the structure of the whole dataset, resulting in the unexpected minority samples. More specially, the more appropriate neighbors can be found by four kinds of OTEMD-LLE in terms of the manifold distance and help to generate the satisfied samples that meet the need of the original data.

Figure 2: The performances of different oversampling techniques for the Swiss-roll dataset



Nine imbalanced datasets composed of the samples satisfying the normal distribution are constructed to further analyze the classification performance of OTEMD. The dataset contains two disjoint minority sub-clusters, namely, Min-A and Min-B. The margin between Min-A and the majority class retains [0,1,1] and [0,0,0], whereas the margin changes for Min-B. As shown in Table 1, IR and M_B represent the imbalance ratio and the margin between the Min-B and majority class, respectively. The AUC values obtained by different oversampling techniques on the artificial datasets shown in Figure.3 indicate that OTEMD-based methods have the superior classification accuracy. In particular, OTEMD-LLE-1 achieves the largest average AUC on the artificial datasets, as shown in Table 2.

Table 1. The attributes of synthetic datasets

Case	IR	Maj	Min-A	Min-B	M_B
C1	1.76	300	150	20	0,2,2
C2	1.76	300	150	20	0,1.5,1.5
C3	1.76	300	150	20	0,1.25,1.25
C4	2.91	350	100	20	0,2,2
C5	2.91	350	100	20	0,1.5,1.5
C6	2.91	350	100	20	0,1.25,1.25
C7	5.71	400	50	20	0,2,2
C8	5.71	400	50	20	0,1.5,1.5
C9	5.71	400	50	20	0,1.25,1.25

3.2 The classification performances on UCI Datasets

Thirty-nine datasets with the various distribution and the imbalance ratio are chosen from the UCI machine learning repository and employed to fully compare and analyze the classification performance of different oversampling techniques. However, several datasets are multiclass imbalance classification problems. Thus, we transformed them into binary-classification problems by combining the data in most of the classes into majority samples. In the experiments, the number of nearest neighbors for SMOTE, Borderline-SMOTE and ADASYN are set to 10. k_1, k_2 and k_3 of MWMOTE are set to 5, 3 and 0.5, respectively. The number of ants and the terminal iterations for ACOSampling are both 10, and the evaporation coefficient is 0.8. The hyper-parameters in OTEMD satisfy $k_1 = k_2 = k_3$, and the optimal k_2 and n_dim are chosen from [10, 15, 20, 25] and [3,5,7] by three-fold cross validation.

For OTWMD, the mapping method from original space to manifold space, as well as the oversampling strategies are the key issues. To further analyze their impact on the classification performances, F-measure, G-mean and AUC of different oversampling strategies are compared in Tables 3, 4 and 5, respectively. Apparently, the proposed oversampling strategies outperform the others for most of the imbalanced datasets, especially for the ones with higher imbalance ratio. More specially, OTEMD-LLE-2 obtains the best average AUC, whereas OTEMD-LLE-1 has the superior F-mean. By compared the performances of OTEMD with various mapping methods, including LLE, HLLE, MLLE, and LSTA, shown in Table 6, we see that the new minority samples generated by OS_2 provide more valuable information for the classification boundary.

Wilcoxon paired sign-rank test is applied to pairwise compare the performances of the proposed OTEMD method with the others on 39 datasets. The null hypothesis of Wilcoxon test H_0 indicates that OTEMD has the same classification accuracy as the other algorithms, and the alternative hypothesis H_α shows the significantly different performances between them.

Figure 3: The performances of different oversampling techniques for the artificial datasets

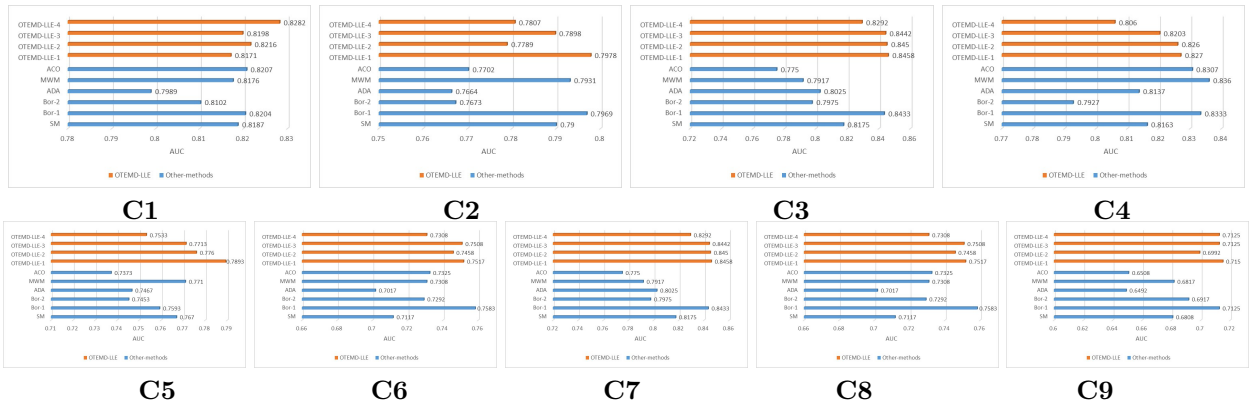


Table 2. The average AUC of different algorithms

	SM	Bor-1	Bor-1	ADA	MWM	ACO	OTEMD-LLE-1	OTEMD-LLE-2	OTEMD-LLE-3	OTEMD-LLE-4
Average	0.7701	0.7917	0.7622	0.7537	0.7716	0.7583	**0.7934**	0.7870	0.7893	0.7779
Rank	7	2	8	10	6	9	**1**	4	3	5

Table 3. Comparison of the AUC among different oversampling techniques

Dataset	SM	Bor-1	Bor-2	ADA	MWM	ACO	OTEMD-LLE-1	OTEMD-LLE-2	OTEMD-LLE-3	OTEMD-LLE-4
Abalone_18v9	0.6590	0.6936	0.6538	0.6115	0.6250	0.6160	0.6609	**0.7346**	0.6571	0.6500
Abalone_17v7	0.9429	0.9474	0.9191	0.9583	0.9409	**0.9685**	0.9288	0.9089	0.8794	0.9134
Abalone_19v5	**1.0000**	1.0000	0.9867	0.9933	0.9867	0.9867	0.9933	0.9933	0.9933	0.9933
CTG_PvN	0.9589	0.9643	0.9710	0.9649	0.9650	0.9561	0.9561	0.9721	0.9708	**0.9825**
CTG_SvN	0.9492	**0.9587**	0.9464	0.9475	0.9313	0.9492	0.9545	0.9522	0.9502	0.9467
Statlandsat_4v12	0.9746	0.9789	0.9732	**0.9889**	0.9775	0.9579	0.9846	0.9816	0.9761	0.9802
Statlandsat_5v12	0.9697	0.9767	0.9586	0.9731	0.9681	0.9611	0.9689	0.9703	0.9589	**0.9783**
Libra_123vAll	0.8758	0.9167	0.8840	0.9096	0.9069	0.9012	0.9122	0.9055	**0.9258**	0.8711
Libra_456vAll	0.9524	0.9557	0.9620	0.9622	**0.9694**	0.9182	0.9543	0.9505	0.9301	0.9361
Libra_789vAll	0.9368	0.8789	0.9270	0.9493	0.9421	0.9306	0.9524	0.9249	**0.9550**	0.9069
Yeast_ME1vCYT	**1.0000**	0.9984	0.9770	1.0000	0.9770	0.9841	0.9929	0.9984	0.9929	0.9929
Yeast_ME1vNUC	0.9673	0.9602	0.9673	0.9531	0.9714	0.9714	0.9786	0.9786	0.9786	**0.9908**
Yeast_ME2vCYT	0.9498	0.9332	0.8869	0.9498	**0.9782**	0.9564	0.9616	0.9253	0.9723	0.9766
Yeast_ME2vNUC	0.9141	0.9121	0.8937	0.9161	0.9373	0.9380	0.9646	**0.9666**	0.9555	0.9514
Yeast_ME3vCYT	0.9473	0.9489	0.9227	0.9371	0.9379	0.9556	0.9454	**0.9624**	0.9585	0.9208
Yeast_ME3vNUC	0.9349	0.9169	0.9267	0.9320	0.9190	**0.9364**	0.9114	0.8986	0.9114	0.9019
Robot_LvF	0.9986	**0.9997**	0.9913	0.9997	0.9987	0.9909	0.9986	0.9960	0.9984	0.9848
Robot_RvF	**0.9974**	0.9922	0.9745	0.9923	0.9960	0.9966	0.9960	0.9881	0.9965	0.9882
Ecoli_OMvCP	0.9203	0.8326	0.8993	0.8870	0.8659	0.9268	0.8993	0.8826	0.9080	**0.9609**
Ecoli_IMvAll	0.9772	0.9798	0.9713	0.9445	0.9772	0.9772	0.9772	**0.9824**	0.9798	0.9685
Ecoli_PPvCP	0.9322	0.9199	0.8728	0.9406	0.9319	0.8895	0.9326	0.9362	**0.9533**	0.9232
Ecoli_IMvCP	**0.9882**	0.9752	0.9563	0.9752	0.9795	0.9795	0.9839	0.9839	0.9795	0.9637
Glass_567vAll	0.9735	0.9731	**0.9957**	0.9644	0.9692	0.9281	0.9727	0.9822	0.9909	0.9870
Vehicle_VANvAll	0.9437	0.9339	0.9517	0.9429	0.9472	0.9307	0.9506	0.9490	**0.9575**	0.9426
Vehicle_OPELvAll	0.8272	0.8314	0.8204	0.8276	0.8246	0.8333	0.8211	**0.8561**	0.8193	0.8322
Vehicle_ASSBvAll	0.8424	0.8399	0.8021	0.8077	0.8313	**0.8543**	0.8345	0.8279	0.8417	0.8367
Vehicle_BUSvAll	0.9515	0.9444	0.9293	0.9434	0.9410	0.9564	0.9508	**0.9718**	0.9632	0.9506
Wine_3vAll	0.9462	**0.9769**	0.9669	0.9538	0.9385	0.9592	0.9462	0.9769	0.9538	0.9519
Wine_1vAll	0.9380	0.9432	**0.9771**	0.9771	0.9380	0.9523	0.9575	0.9541	0.9470	0.9594
Wine_2vAll	0.8898	0.8861	0.8963	0.8652	0.9401	0.8898	**0.9460**	0.9251	0.9096	0.9316
Breast-tissue_CFvAll	0.8100	0.7600	0.7700	0.7500	0.7400	0.7400	0.7500	0.7500	0.7400	**0.8500**
Breast-cancer_MvB	0.9323	0.9556	0.9292	**0.9583**	0.9011	0.9435	0.9414	0.9509	0.9509	0.9276
Ionosphere_BvG	0.8594	0.8368	0.8283	0.8631	0.8705	0.8205	0.8632	**0.8712**	0.8629	0.8591
PageBlocks_4v2	0.9667	0.9611	0.9431	0.9667	0.9577	**0.9778**	0.9667	0.9688	0.9688	0.9661
PageBlocks_5v2	**0.9830**	0.9798	0.9662	0.9630	0.9662	0.9729	0.9662	0.9763	0.9660	0.9361
Segment_4v123	0.9728	0.9584	0.9637	0.9684	0.9731	0.9692	0.9647	0.9702	**0.9738**	0.9676
Segment_5v123	**0.9458**	0.9237	0.9214	0.9365	0.9437	0.9280	0.9380	0.9445	0.9378	0.9397
Segment_6v123	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Segment_7v123	**1.0000**	**1.0000**	0.9995	**1.0000**	**1.0000**	0.9947	0.9989	0.9979	**1.0000**	0.9970
average value	0.9366	0.9319	0.9252	0.9327	0.9324	0.9307	0.9379	**0.9402**	0.9376	0.9363

Table 4. Comparison of F-measure among different oversampling techniques

Dataset	SM	Bor-1	Bor-2	ADA	MWM	ACO	OTEMD-LLE-1	OTEMD-LLE-2	OTEMD-LLE-3	OTEMD-LLE-4
Abalone_18v9	0.4374	0.4566	**0.5858**	0.5413	0.4704	0.5073	0.5058	0.4343	0.4865	0.5348
Abalone_17v7	0.9058	**0.9081**	0.8621	0.8901	0.9043	0.8738	0.9026	0.8899	0.8670	0.8652
Abalone_19v5	0.9703	0.9769	0.9846	0.9703	**0.9923**	0.9636	0.9846	0.9769	0.9769	0.9779
CTG_PvN	0.9512	0.9177	0.8929	0.9188	0.9449	0.8429	**0.9576**	0.9365	0.9338	0.8888
CTG_SvN	0.8727	0.8336	0.7585	0.8506	0.8801	0.7888	**0.8836**	0.8267	0.8715	0.7886
Statlandsat_4v12	0.9588	0.9616	0.9657	0.9673	0.9573	0.9643	0.9657	0.9614	**0.9829**	0.9775
Statlandsat_5v12	0.9353	**0.9684**	0.9086	0.9556	0.9297	0.9306	0.9560	0.9618	0.9635	0.9408
Libra_123vAll	0.8687	0.8751	0.8528	0.8689	0.8601	0.8714	**0.8757**	0.8574	0.8665	0.8331
Libra_456vAll	0.9411	0.9386	0.9110	0.9414	**0.9468**	0.8904	0.9441	0.9162	0.9392	0.9039
Libra_789vAll	0.9167	0.8605	0.8934	**0.9230**	0.9033	0.9001	0.9106	0.9215	0.9203	0.8912
Yeast_ME1vCYT	0.9751	**0.9815**	0.9700	0.9772	0.9684	0.8984	0.9786	0.9575	0.9786	0.9623
Yeast_ME1vNUC	0.9769	0.9550	0.9473	0.9769	0.9769	0.9340	0.9735	**0.9846**	0.9735	0.9684
Yeast_ME2vCYT	0.9044	0.8856	0.7536	0.9002	0.8920	0.7667	0.8836	0.8929	**0.9057**	0.8553
Yeast_ME2vNUC	**0.9284**	0.8835	0.7689	0.9157	0.9169	0.8120	0.9047	0.8938	0.8995	0.8396
Yeast_ME3vCYT	0.9217	0.9123	0.8459	0.9258	0.9231	0.9096	0.9238	**0.9393**	0.9312	0.8840
Yeast_ME3vNUC	0.9168	0.8954	0.8706	0.8992	**0.9189**	0.8939	0.8935	0.9116	0.9154	0.9054
Robot_LvF	0.9948	**0.9949**	0.9501	0.9936	0.9889	0.9635	0.9942	0.9515	0.9941	0.8847
Robot_RvF	0.9879	0.9805	0.9352	0.9797	**0.9895**	0.9874	0.9882	0.9607	0.9835	0.9056
Ecoli_OMvCP	0.7721	0.8535	**0.8545**	0.8324	0.8081	0.7657	0.8033	0.8163	0.8321	0.8394
Ecoli_IMvAll	0.9556	0.9693	0.9275	0.9423	0.9668	0.9504	0.9640	**0.9697**	0.9557	0.9286
Ecoli_PPvCP	0.8902	0.9081	0.8754	0.8747	0.9298	0.8540	0.9281	**0.9401**	0.9251	0.8859
Ecoli_IMvCP	0.9825	**0.9913**	0.963	0.9631	0.9854	0.9715	0.9882	0.9831	0.9773	0.9587
Glass_567vAll	0.9631	0.9821	0.9731	0.9778	**0.9861**	0.9603	0.9639	0.9492	0.9492	0.9268
Vehicle_VANvAll	0.9117	0.9204	0.9016	0.9209	0.9174	0.9188	0.9129	0.9128	0.9142	**0.9238**
Vehicle_OPELvAll	0.8135	0.7669	0.7934	0.7973	0.8048	**0.8504**	0.7867	0.7785	0.8026	0.8286
Vehicle_ASSBvAll	0.8305	0.7958	0.8067	0.7928	0.8143	**0.8940**	0.8076	0.8179	0.8154	0.8286
Vehicle_BUSvAll	0.9643	0.9591	0.9553	0.9663	0.9619	**0.9696**	0.9603	0.9606	0.9629	0.9547
Wine_3vAll	0.9720	0.9408	0.9539	0.9615	0.9679	0.9190	0.9606	**0.9763**	0.9646	0.9532
Wine_1vAll	**0.9706**	0.9600	0.9484	0.9706	0.9561	0.9343	0.9632	0.9674	0.9534	0.9445
Wine_2vAll	0.8920	0.8955	0.8809	0.8729	**0.9099**	0.8556	0.8921	0.8935	0.8967	0.8816
Breast-tissue_CFvAll	0.7233	0.7297	0.7197	0.7588	0.7490	0.7430	**0.7962**	0.7191	0.7842	0.7751
Breast-cancer_MvB	0.9065	0.9085	0.8903	0.8995	0.8979	**0.9263**	0.9086	0.9136	0.9118	0.9051
Ionosphere_BvG	0.8615	0.8598	0.8585	0.8403	0.8638	**0.8707**	0.8562	0.8416	0.8429	0.8274
PageBlocks_4v2	**0.9748**	0.9625	0.9487	0.9686	0.9601	0.9598	0.9663	0.9598	0.9717	0.9327
PageBlocks_5v2	0.9671	0.9805	0.9257	0.9724	0.9629	0.9802	0.9604	0.9709	0.9694	**0.9834**
Segment_4v123	0.9650	0.9592	0.9449	0.9675	**0.9695**	0.9589	0.9640	0.9606	0.9668	0.9327
Segment_5v123	**0.9194**	0.9014	0.8918	0.9142	0.9155	0.9006	0.9188	0.9173	**0.9194**	0.9017
Segment_6v123	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9988
Segment_7v123	0.9950	**0.9981**	0.9962	0.9968	0.9975	0.9944	0.9944	0.9931	0.9950	0.9908
average value	0.9059	0.8965	0.8980	0.9122	0.9055	0.8931	**0.9203**	0.9002	0.9138	0.8919

Table 5. Comparison of G-meam among different oversampling techniques

Dataset	SM	Bor-1	Bor-2	ADA	MWM	ACO	OTEMD-LLE-1	OTEMD-LLE-2	OTEMD-LLE-3	OTEMD-LLE-4
Abalone_18v9	**0.5447**	0.4081	0.5304	0.4600	0.4268	0.4594	0.4056	0.4098	0.4826	0.4938
Abalone_17v7	**0.9236**	0.8954	0.8884	0.9036	0.9032	0.8580	0.8714	0.9018	0.8911	0.9026
Abalone_19v5	0.9703	0.9626	0.9692	0.9692	0.9703	**0.9779**	0.9703	0.9626	0.9626	0.9626
CTG_PvN	0.9662	0.9137	0.9009	0.9320	0.9444	0.8227	0.9589	**0.9665**	0.9483	0.9259
CTG_SvN	0.8801	0.8480	0.7780	0.8489	0.8736	0.8041	**0.8902**	0.8508	0.8585	0.7976
Statlandsat_4v12	0.9528	0.9515	0.9620	0.9557	0.9546	0.9407	0.9601	0.9528	**0.9700**	0.9699
Statlandsat_5v12	0.9363	**0.9724**	0.9099	0.9697	0.9363	0.9341	0.9433	0.9486	0.9487	0.9401
Libra_123vAll	0.8621	0.8717	0.8430	**0.8724**	0.8548	0.8327	0.8645	0.8562	0.8498	0.8653
Libra_456vAll	0.9417	0.9383	0.9238	0.9390	**0.9442**	0.8955	0.9250	0.9149	0.9077	0.9171
Libra_789vAll	0.9279	0.9089	0.9218	0.9219	0.9240	0.9006	0.9213	0.9146	**0.9320**	0.9040
Yeast_ME1vCYT	0.9714	0.9666	0.9118	0.9748	**0.9783**	0.9070	**0.9783**	0.9714	0.9783	0.9679
Yeast_ME1vNUC	0.9503	0.9586	0.9567	0.9572	**0.9632**	0.9210	0.9494	0.9609	0.9535	0.9585
Yeast_ME2vCYT	0.9061	0.9162	0.7472	**0.9417**	0.9281	0.8596	0.9003	0.9075	0.9203	0.8464
Yeast_ME2vNUC	0.8797	0.8651	0.7660	0.8536	0.8566	0.7792	0.8812	**0.8849**	0.8523	0.8429
Yeast_ME3vCYT	0.9191	0.9136	0.8749	0.9102	0.9103	0.9131	0.9204	**0.9221**	0.9207	0.8899
Yeast_ME3vNUC	0.8921	0.8955	0.8526	0.8980	0.8979	0.9039	**0.9084**	0.9070	0.9045	0.8974
Robot_LvF	0.9974	**0.9981**	0.9484	0.9968	0.9974	0.9523	0.9955	0.9608	0.9968	0.8756
Robot_RvF	0.9855	0.9763	0.9341	0.9827	0.9850	0.9871	**0.9924**	0.9612	0.9843	0.9402
Ecoli_OMvCP	0.8659	0.8659	0.8302	0.8576	**0.8911**	0.8653	0.8632	0.8802	0.8788	0.8565
Ecoli_IMvAll	0.9556	0.9756	0.9738	0.9469	0.9613	0.9627	0.9729	**0.9820**	0.9670	0.9417
Ecoli_PPvCP	**0.8949**	0.8559	0.8161	0.8364	0.8566	0.8548	0.8816	0.8735	0.8911	0.8437
Ecoli_IMvCP	0.9675	0.9759	0.9539	0.9561	0.9710	0.9637	**0.9823**	0.9768	0.9823	0.9545
Glass_567vAll	0.9564	0.9842	0.9458	**0.9957**	0.9870	0.9353	0.9660	0.9660	0.9617	0.9624
Vehicle_VANvAll	0.9226	0.9365	0.9242	0.9406	0.9356	0.9432	0.9216	0.9408	0.9274	**0.9457**
Vehicle_OPELvAll	0.7811	0.7810	0.7819	0.7882	0.7843	**0.8636**	0.7458	0.7811	0.8071	0.7903
Vehicle_ASSBvAll	0.8137	0.7769	0.8079	0.7819	0.7858	**0.8880**	0.8189	0.8030	0.7814	0.8098
Vehicle_BUSvAll	**0.9610**	0.9446	0.9492	0.9581	0.9489	0.9520	0.9471	0.9461	0.9475	0.9253
Wine_3vAll	**0.9719**	0.9395	0.9565	0.9609	0.9559	0.956	0.9639	0.9445	0.9609	0.9572
Wine_1vAll	0.9754	0.9657	0.9466	**0.9791**	0.9652	0.9332	0.9625	0.9754	0.9721	0.9721
Wine_2vAll	0.8917	0.9014	0.8892	0.8994	0.9079	0.8977	**0.9261**	0.9113	0.9084	0.9000
Breast-tissue_CFvAll	0.7587	0.7083	0.7315	0.7436	0.7262	0.7628	0.7643	0.7616	**0.8122**	0.7599
Breast-cancer_MvB	0.9224	0.9182	0.9186	**0.9360**	0.9268	0.9171	0.9169	0.9141	0.9189	0.9188
Ionosphere_BvG	**0.8711**	0.8546	0.8446	0.8535	0.8685	0.8532	0.8616	0.8550	0.8617	0.8453
PageBlocks_4v2	0.9635	0.9539	0.9464	0.9628	0.9576	0.9621	0.9547	0.9606	**0.9636**	0.9610
PageBlocks_5v2	0.9662	**0.9756**	0.9728	0.9657	0.9726	0.9666	0.9632	0.9613	0.9595	0.9640
Segment_4v123	**0.9706**	0.9611	0.9532	0.9672	0.9669	0.9426	0.9629	0.9646	0.9651	0.9505
Segment_5v123	**0.9269**	0.9111	0.8907	0.9163	0.9112	0.8933	0.9248	0.9099	0.9164	0.9039
Segment_6v123	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9994
Segment_7v123	**0.9969**	0.9944	0.9957	0.9957	0.9957	0.9931	0.9950	0.9963	0.9951	0.9944
average value	0.9200	0.9145	0.8981	0.9200	0.9218	0.9091	0.9236	0.9210	**0.9269**	0.9087

Table 6. The performances of OTEMD with various mapping methods

Algorithm	Metric	OS1	OS2	OS3	OS4
HLLE	AUC	0.9343	0.9355	0.9316	0.9243
	F-measure	0.9064	0.9056	0.9183	0.8896
	G-mean	0.9245	0.9016	0.9134	0.9200
MLLE	AUC	0.9341	0.9344	0.9320	0.9316
	F-measure	0.9023	0.9039	0.9060	0.8946
	G-mean	0.9215	0.9188	0.9145	0.9245
LTSA	AUC	0.9372	0.9368	0.9341	0.9262
	F-measure	0.9156	0.9064	0.8912	0.9056
	G-mean	0.9113	0.8997	0.9245	0.9212

$p < 5\%$ means that the experimental results provide the significant evidence against H_0. Therefore, we reject the null hypothesis and accept the alternative hypothesis. By taking win-lose-draw as the evaluation criterion, a win method is better than the others at the 95% confidence level. The performances of the proposed oversampling techniques are compared with six traditional data-level strategies, as shown in Table 7. The statistical results show that LLE-based OTEMD has the better and more stable classification performances, except for OTEMD-LLE-4. By analysis, a minority sample generated by OS_4 based on the minority-minority or minority-majority matched-pair data may be mislabeled, resulting in the wrong classification boundary. In addition, the Wilcoxon paired sign-rank comparison listed in Table 8 also shows that OTEMD-LLE can capture the ground-truth structure of the dataset accurately and generate the valuable minority samples, thus, improving the classification performance in most of the cases.

4 Conclusion

To address the class imbalance problems, a novel oversampling strategy based on manifold distance is proposed, in which a new minority sample is generated in terms of the distances among neighbors in manifold space. The redundant majority data are firstly removed by undersampling to decrease the computation complexity of calculating the manifold distance among the samples. Thereafter, the nearest ones are chosen as neighbors and form the matched-pair data. Based on them, four improved oversampling strategies are presented to create a new minority sample nearby the classification boundary. The experiments on the 39 UCI datasets and 9 synthetic datasets indicate that OTEMD has better classification accuracy in most cases, especially for the datasets with class overlapping and small disjunction. Moreover, the oversampling strategies have more significant effect on OTEMD performance. The proposed oversampling technique provides a generic framework for data-based class imbalance learning. The combination of advanced optimizing techniques (Guo et al., 2020, 2019) with OTEMD is our future work. Moreover, it's interesting to investigate more efficient manifold learning methods for class imbalance learning.

5 Acknowledgement

This work was jointly supported by National Natural Science Foundation of China (61973305, 61573361 and 61803369), and Natural Science Foundation of Liaoning Province for the State Key Laboratory of Robotics(2020-KF-22-02). Also, thank you for the support from the State Key Laboratory of Robotics (2019-O12).

References

Arafat, Y., Hoque, S., Xu, S. and Farid, D. M. (2019) 'Machine learning for mining imbalanced data.', *IAENG Internaitonal journal of computer science*, Vol. 46, pp.332–348

Barua, S., Islam, M. M., Yao, X. and Murase, K. (2013) 'Mwmote-majority weighted minority oversampling technique for imbalanced data set learning.', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, No. 2, pp.405–425

Bellinger, C., Drummond, C. and Japkowicz, N. (2017) 'Manifold-based synthetic oversampling with manifold conformance estimation.', *Machine Learning*, Vol. 107, No. 1, pp.605–637

Cai, B., Zhao, Y., Liu, H. and Min, X. (2017) 'A data-driven fault diagnosis methodology in three-phase inverters for pmsm drive systems.', *IEEE Transactions on Power Electronics*, Vol. 32, No. 99, pp.5590–5600

Cheng, J. , Chen, J. , Guo, Y. N. , Cheng, S. , Yang, L. , and Zhang, P. (2018) 'Adaptive CCR-ELM with variable-length brain storm optimization algorithm for class-imbalance learning.', *Natural Computing*,[online] https://doi.org/10.1007/s11047-019-09735-9

Dal, P. A., Boracchi, G., Caelen, O., Alippi, C. and Bontempi, G. (2018) 'Credit card fraud detection: A realistic modeling and a novel learning strategy.', *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 29, No. 8, pp.3784–3797

Guo, Y., Zhang, X., Gong, D.-W., Zhang, Z. and Yang, J.-J. (2020) 'Novel interactive preference-based multi-objective evolutionary optimization for bolt supporting networks.', *IEEE Transactions on Evolutionary Computation*, Vol. 24, No. 4, pp.750–764

Guo, Y., Yang, H., Chen, M., Cheng, J. and Gong, D. (2019) 'Ensemble prediction-based dynamic robust multi-objective optimization methods.', *Swarm and Evolutionary Computation*, Vol. 48, pp.156–171

Han, H., Wang, W. Y. and Mao, B. H. (2005) 'Borderline-smote: A new over-sampling method in imbalanced data sets learning.', In *Advances in Intelligent Computing*, pp.878–887

Table 7. Win-Lose-Draw results of Wilcoxon pairwise test

	SM	Bor-1	Bor-2	ADA	MWM	ACO
OTEMD-LLE-1	3-8-28	9-4-26	16-2-21	6-5-28	6-3-30	10-4-25
OTEMD-LLE-2	9-8-22	9-6-24	15-4-20	7-8-24	10-6-23	13-6-20
OTEMD-LLE-3	7-6-26	11-7-21	15-4-20	12-7-20	9-2-28	9-3-27
OTEMD-LLE-4	8-11-20	10-13-16	14-7-18	10-14-15	7-10-22	10-11-18
OTEMD-HLLE-1	2-4-33	11-4-24	15-1-23	10-6-23	4-4-31	13-4-22
OTEMD-HLLE-2	4-8-27	10-10-19	15-3-21	9-8-22	4-7-28	10-10-19
OTEMD-HLLE-3	2-4-33	11-5-23	10-2-27	8-10-21	4-10-25	10-7-22
OTEMD-HLLE-4	1-16-22	8-16-15	10-10-19	5-17-17	3-17-19	5-13-21
OTEMD-MLLE-1	3-6-30	10-0-29	15-4-20	7-10-22	7-6-26	9-6-24
OTEMD-MLLE-2	5-4-30	8-0-31	16-2-21	5-7-27	9-4-26	9-7-23
OTEMD-MLLE-3	3-4-32	13-2-24	18-2-19	10-4-25	7-3-29	8-6-25
OTEMD-MLLE-4	3-8-28	7-5-27	13-4-22,	4-8-27,	4-9-26,	9-11-19
OTEMD-LTSA-1	7-5-27	14-2-23	21-1-17	9-7-23	9-6-24	11-3-25
OTEMD-LTSA-2	5-9-25	11-7-21	16-1-22	5-6-28	7-9-23	10-8-21
OTEMD-LTSA-3	4-6-29	12-5-22	14-0-25	9-5-25	3-6-30	9-5-25
OTEMD-LTSA-4	2-14-23	5-9-25	9-5-25	2-15-22	4-15-20	5-11-23

Table 8. Significance Test of Averaged AUC between OTEMD-LLE-1, -2, -3, -4 and MWMOTE

Dataset	MWMOTE	OTEMD-LLE-1 vs. MWM			OTEMD-LLE-2 vs. MWM			OTEMD-LLE-3 vs. MWM			OTEMD-LLE-4 vs. MWM		
		OTEMD	R+	R-	OTEMD	R+	R-	OTEMD	R+	R-	OTEMD	R+	R-
Abalone_18v9	0.625	0.6608	38	-	0.7346	39	-	0.657	35	-	0.65	31	-
Abalone_17v7	0.9409	0.9287	-	30	0.9089	-	35	0.8794	-	39	0.9133	-	33
Abalone_19v5	0.9866	0.9933	17	-	0.9933	15	-	0.9933	15	-	0.9933	9	-
CTG_PvN	0.9649	0.956	-	25	0.9721	17	-	0.9708	12	-	0.9825	27	-
CTG_SvN	0.9312	0.9544	35	-	0.9521	29	-	0.9502	29	-	0.9466	22	-
Statlandsat_4v12	0.9774	0.9845	18	-	0.9816	11	-	0.976	-	9	0.9802	3	-
Statlandsat_5v12	0.968	0.9688	7	-	0.9702	7	-	0.9588	-	20	0.9783	17	-
Libra_123vAll	0.9069	0.9122	14	-	0.9055	-	4	0.9258	28	-	0.871	-	37
Libra_456vAll	0.9693	0.9543	-	31	0.9504	-	27	0.9301	-	36	0.9361	-	35
Libra_789vAll	0.9421	0.9523	29	-	0.9248	-	26	0.955	24	-	0.9069	-	36
Yeast_ME1vCYT	0.9769	0.9928	32	-	0.9984	30	-	0.9928	26	-	0.9928	24	-
Yeast_ME1vNUC	0.9714	0.9785	19	-	0.9785	16	-	0.9785	16	-	0.9908	29	-
Yeast_ME2vCYT	0.9782	0.9616	-	33	0.9252	-	38	0.9722	-	13	0.9766	-	2
Yeast_ME2vNUC	0.9372	0.9645	36	-	0.9666	32	-	0.9554	27	-	0.9513	21	-
Yeast_ME3vCYT	0.9378	0.9454	21	-	0.9623	31	-	0.9584	30	-	0.9207	-	25
Yeast_ME3vCYT	0.9189	0.9113	-	22	0.8985	-	28	0.9113	-	18	0.9018	-	26
Robot_LvF	0.9987	0.9985	-	5	0.996	-	8	0.9984	-	6	0.9848	-	20
Robot_RvF	0.996	0.996	4	2.0	0.988	-	18	0.9964	7	-	0.9882	-	11
Ecoli_OMvCP	0.8659	0.8992	37	-	0.8826	25	-	0.9079	37	-	0.9608	38	-
Ecoli_IMvAll	0.9772	0.9772	0.5	0.5	0.9823	14	-	0.9797	10	-	0.9684	-	15
Ecoli_PPvCP	0.9318	0.9326	6	-	0.9362	12	-	0.9532	31	-	0.9231	-	14
Ecoli_IMvCP	0.9795	0.9838	13	-	0.9838	13	-	0.9795	1.0	1.0	0.9636	-	23
Glass_567vAll	0.9691	0.9727	12	-	0.9822	22	-	0.9909	32	-	0.9869	28	-
Vehicle_VANvAll	0.9471	0.9506	10	-	0.949	5	-	0.9575	21	-	0.9426	-	6
Vehicle_OPELvAll	0.8246	0.8211	-	11	0.856	34	-	0.8193	-	11	0.8321	10	-
Vehicle_ASSBvAll	0.8313	0.8344	9	-	0.8278	-	10	0.8416	22	-	0.8367	7	-
Vehicle_BUSvAll	0.9409	0.9507	27	-	0.9718	33	-	0.9631	33	-	0.9505	16	-
Wine_3vAll	0.9384	0.9461	23	-	0.9769	36	-	0.9538	25	-	0.9519	19	-
Wine_1vAll	0.9379	0.9575	34	-	0.9541	24	-	0.9469	19	-	0.9593	30	-
Wine_2vAll	0.9401	0.9459	16	-	0.9251	-	23	0.9096	-	34	0.9315	-	13
Breast-tissue_CFvAll	0.74	0.75	28	-	0.75	19	-	0.74	2.0	2.0	0.85	39	-
Breast-cancer_MvB	0.9011	0.9413	39	-	0.9508	37	-	0.9508	38	-	0.9275	32	-
Ionosphere_BvG	0.8704	0.8632	-	20	0.8712	3	-	0.8629	-	17	0.859	-	18
PageBlocks_4v2	0.9576	0.9666	26	-	0.9687	21	-	0.9687	23	-	0.966	12	-
PageBlocks_5v2	0.9662	0.9662	1.0	1.0	0.9763	20	-	0.9659	-	5	0.936	-	34
Segment_4v123	0.973	0.9647	-	24	0.9701	-	9	0.9738	8	-	0.9676	-	8
Segment_5v123	0.9437	0.9379	-	15	0.9444	2	-	0.9377	-	14	0.9396	-	5
Segment_6v123	1.0	1.0	1.5	1.5	1.0	0.5	0.5	1.0	1.5	1.5	1.0	0.5	0.5
Segment_7v123	1.0	0.9989	-	8	0.9978	-	6	1.0	0.5	0.5	0.9969	-	4
		R+ = 553.0 and R- = 229.0			R+ = 547.5 and R- = 232.5			R+ = 553.0 and R- = 227.0			R+ = 414.5 and R- = 365.5		
		$p_{value} = 0.0256$			$p_{value} = 0.0290$			$p_{value} = 0.0209$			$p_{value} = 0.7223$		

He, H., Yang, B., Garcia, E. A. and Li, S. (2008) 'Adasyn: Adaptive synthetic sampling approach for imbalanced learning', In *IEEE International Joint Conference on Neural Networks*, pp.1322–1328

Hsu, C. C., Wang, K. S. and Chang, S. H. (2011) 'Bayesian decision theory for support vector machines: Imbalance measurement and feature optimization.', *Expert Systems with Applications*, Vol. 38, No. 5, pp.4698–4704

Kang, Q., Shi, L., Zhou, M. C., Wang, X. S., Wu, Q. D. and Wei, Z. (2018) 'A distance-based weighted undersampling scheme for support vector machines and its application to imbalanced classification.', *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 29, pp.4152–4165

Kang, Q., Wang, K., Huang, B. and An, J. (2014) 'Kernel optimisation for kpca based on gaussianity estimation.', *International Journal of Bio Inspired Computation*, Vol. 6, No. 2, pp.91–107

Krawczyk, B. (2016) 'Learning from imbalanced data: open challenges and future directions.', *Progress in Artificial Intelligence*, Vol. 5, No. 4, pp.221–232

Lim, P., Goh, C. K. and Tan, K. C. (2016) 'Evolutionary cluster-based synthetic oversampling ensemble (eco-ensemble) for imbalance learning.', *IEEE Transactions on Cybernetics*, Vol. 47, No. 99, pp.2850–2861

Lin, W. C., Tsai, C. F., Hu, Y. H. and Jhang, J. S. (2017) 'Clustering-based undersampling in class-imbalanced data.', *Information Sciences*, Vol. 409, pp.17–26

Loezer, L., Enembreck, F., Barddal, J. P. and de Souza Britto, A. (2020) 'Cost-sensitive learning for imbalanced data streams.', In *The 35th ACM/SIGAPP Symposium on Applied Computing*, pp.498–504

Lunga, D., Prasad, S., Crawford, M. M. and Ersoy, O. (2013) 'Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning.', *IEEE Signal Processing Magazine*, Vol. 31, No. 1, pp.55–66

Roweis, S. T. and Saul, L. K. (2000) 'Nonlinear dimensionality reduction by locally linear embedding.', *Science*, Vol. 290, No. 5500, pp.2323–2326

Shakeel, F., Sabhitha, A. S. and Sharma, S. (2017) 'Exploratory review on class imbalance problem: An overview.', In *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp.1–8

Shen, Y., Wang, G. and Gao, H. (2016) 'Data-driven process monitoring based on modified orthogonal projections to latent structures.', *IEEE Transactions on Control Systems Technology*, Vol. 24, No. 4, pp.1480–1487

Soleymani, R., Granger, E. and Fumera, G. (2018) 'Progressive boosting for class imbalance and its application to face re-identification.', *Expert Systems with Applications*, Vol. 101, pp.271–291

Wang, Z., Xing, H., Li, T., Yan, Y. and Yi, P. (2016) 'A modified ant colony optimization algorithm for network coding resource minimization.', *IEEE Transactions on Evolutionary Computation*, Vol. 20, No. 3, pp.325–342

Zhang, S., Ma, Z. and Tan, H. (2018) 'On the equivalence of hlle and ltsa.', *IEEE Transactions on Cybernetics*, Vol. 48, No. 99, pp.742–753