Oversampling for Imbalanced Time Series Data

Tuanfei Zhu
zhutuanfei@hnu.edu.cn
Changsha University

Yaping Lin
yplin@hnu.edu.cn
Hunan University

Yonghe Liu
yonghe@cse.uta.edu
University of Texas at Arlington

ABSTRACT

Many important real-world applications involve time-series data with skewed distribution. Compared to conventional imbalance learning problems, the classification of imbalanced time-series data is more challenging due to high dimensionality and high inter-variable correlation. This paper proposes a structure preserving Oversampling method to combat the High-dimensional Imbalanced Time-series classification (OHIT). OHIT first leverages a density-ratio based shared nearest neighbor clustering algorithm to capture the modes of minority class in high-dimensional space. It then for each mode applies the shrinkage technique of large-dimensional covariance matrix to obtain accurate and reliable covariance structure. Finally, OHIT generates the structure-preserving synthetic samples based on multivariate Gaussian distribution by using the estimated covariance matrices. Experimental results on several publicly available time-series datasets (including unimodal and multi-modal) demonstrate the superiority of OHIT against the state-of-the-art oversampling algorithms in terms of F-value, G-mean, and AUC.

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning by classification**; • **Information systems** → *Clustering*.

KEYWORDS

imbalanced classification, oversampling, high-dimensional data, clustering

ACM Reference Format:
Tuanfei Zhu, Yaping Lin, and Yonghe Liu. 2019. Oversampling for Imbalanced Time Series Data. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The imbalanced learning problem appears when the distribution of samples is significantly unequal among different classes. The majority class with relatively more samples can overwhelm the minority class in sample size. Since standard machine learning methods usually seek the minimization of training errors, the resulting classifiers will be naturally biased towards the majority class, leading to the performance depreciation for important and interest minority samples [7, 8]. Over the past two decades, a large number of class imbalance techniques have been proposed to combat the imbalanced data learning [8, 21, 28, 31]. These existing solutions can be roughly divided into algorithm-level approaches and data-level approaches.

In algorithm-level approaches, traditional classification algorithms are improved to put more emphasis on the learning of minority class by adjusting training mechanism or prediction rule such as modification of loss function [10], introduction of class-dependant costs [7], and movement of output threshold [34]. Data-level approaches aim to establish class balance by adding new minority samples (i.e., oversampling) [33, 35, 36], deleting a part of original majority samples (i.e., undersampling) [31], or combining both of them. Compared to algorithm-level methods, data-level techniques have two main advantages. First, they are more universal, since the preprocessed data can be fed into various machine learning algorithms to boost their prediction capability in the minority class. Second, data-level approaches can flexibly integrate with other techniques such as kernel methods and ensemble learning to explore elaborate hybrid solutions [2, 28].

In this paper, we focus our attention on oversampling techniques in data-level approaches, since oversampling directly addresses the difficulty source of classifying imbalanced data by compensating the insufficiency of minority class information, and, unlike undersampling, does not suffer the risk of discarding informative majority samples.

1.1 Motivation

The problem of imbalanced time series classification is frequently encountered in many real-world applications, including medical monitoring [30], abnormal movement detection [29] and industrial hazards surveillance [19]. Although there are a large number of oversampling solutions in previous literature, few of them are exclusively designed to deal with imbalanced time-series data. Different from conventional data, time series data presents high dimensionality and high inter-variable correlation as time-series sample is an ordered variable set which is extracted from a continuous signal. As a result, addressing imbalanced time series classification exist some special difficulties as compared to classical class imbalance problems [4]. In terms of data oversampling, the designed oversampling algorithm should have the capability of coping with the additional challenges due to high dimensionality, and protect the original correlation among variables so as not to confound the learning.

Therefore, we want to develop an oversampling method for imbalanced time series data that can accurately acquire the correlation structure of minority class, and generate structure-preserving synthetic samples to maintain the main correlation implied in the minority class. The purpose is to greatly improve the performance of minority class without seriously damaging the classification accuracy on the majority class.

Conference'17, July 2017, Washington, DC, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1.2 Limitations of Existing Techniques

Interpolated techniques, probability distribution-based methods, and structure preserving approaches are three main types of oversampling.

In interpolation oversampling, the synthetic samples are randomly interpolated between the feature vectors of two neighboring minority samples [35, 36]. One of the most representative methods is SMOTE [8]. Because of the high dimensionality of time-series data, there may exist a considerable space between arbitrary two time-series minority samples. When the synthetic samples are allowed to create in such of the region, they seem to scatter in the whole feature space, which leads to severe over-generalization problem. In addition, interpolated oversampling methods can introduce a lot of random data variations since they only take the local characteristics of minority samples into account. It will weaken the inherent correlation of original time-series data.

For probability distribution-based methods, they first estimate the underlying distribution of minority class, then yield the synthetic samples according to the estimated distribution [6, 11]. However, accurate discrete probability distribution or probability density function is extremely hard to obtain due to the scarcity of minority samples, especially in high-dimensional space [17].

Structure-preserving oversampling methods generate the synthetic samples on the premise of reflecting the main structure of minority class. In paper [1], the authors proposed Mahalanobis Distance-based Oversampling (MDO). MDO produces the synthetic samples which obey the sample covariance structure of minority class by operating the value range of each feature in principal component space. The major drawback of MDO is that the sample covariance matrix can seriously deviate from the true covariance one for high-dimensional data, i.e., the smallest (/largest) eigenvalues of sample covariance matrix can be greatly underestimated (/overestimated) compared to the corresponding true eigenvalues [16]. Different from MDO, the structure-preserving oversampling methods SPO [4] and INOS [3] first divide the eigenspectrum of sample covariance matrix into the reliable and unreliable subspaces, then pull up the sample eigenvalues in unreliable subspace. However, both SPO and INOS assume the minority class is unimodal. This assumption often does not hold for real-life data, since the samples of a single class may imply multiple modes (e.g., the failure events of aircrafts exist multiple failure modes; a disease includes distinct subtypes). To handle the multi-modal minority class, Pang et al. developed a parsimonious Mixture of Gaussian Trees models (MoGT) which attempts to construct Gaussian graphical model for each mode [5]. However, MoGT only considers the correlations among pairs of nearest variables in order to reduce the number of estimated parameters. Besides, MoGT does not build the reliable mechanism to identify the modes of minority class. The authors, in fact, set the number of mixture components manually.

1.3 Our Method and Main Contributions

Based on the above analyses, existing oversampling algorithms cannot protect the structure of minority class well for imbalanced time series data, especially when the minority class is multi-modal. In this study, we propose a structure-preserving oversampling method OHIT which can accurately maintain the covariance structure of minority class and deal with the multi-modality simultaneously. OHIT leverages a Density-Ratio based Shared Nearest Neighbor clustering algorithm (DRSNN) to cluster the minority class samples in high-dimensional space. Each discovered cluster corresponds to the representative data of a mode. To overcome the problem of small sample and high dimensionality, OHIT for each mode use the shrinkage technique to estimate covariance matrix. Finally, the structure-preserving synthetic samples are generated based on multivariate Gaussian distribution by using the estimated covariance matrices.

The major contributions of this paper are as follows: 1) We design a robust DRSNN clustering algorithm to capture the potential modes of minority class in high-dimensional space. 2) We improve the estimate of covariance matrix in the context of small sample size and high dimensionality, through utilizing the shrinkage technique based on Sharpe's single-index model. 3) The proposed OHIT is evaluated on both the unimodal datasets and multi-modal datasets, the results show that OHIT has better performance than existing representative methods.

2 THE PROPOSED OHIT FRAMEWORK

OHIT involves three key issues: 1) clustering high-dimensional data; 2) estimating the large-dimensional covariance matrix based on limited data; 3) and yielding structure-preserving synthetic samples. In section 2.1, we introduce the clustering of high-dimensional data, where a new clustering algorithm DRSNN is presented. Section 2.2 describes the shrinkage estimation of covariance matrix. The shrinkage covariance matrix is a more accurate and reliable estimator than the sample covariance matrix in the context of limited data. Section 2.3 gives the generation of structure-preserving synthetic samples. Finally, the algorithm flow and complexity analysis of OHIT are together provided in Section 2.4

2.1 Clustering of High-dimensional Data

2.1.1 Preliminary. Two significant challenges exist in clustering high-dimensional data. First, the distances or similarities between samples tend to be more uniform, which can weaken the utility of similarity measures for discrimination, causing clustering more difficult. Second, clusters usually present different densities, sizes, and shapes.

Some research works developed Shared Nearest Neighbor similarity (SNN)-based density clustering methods to cluster high-dimensional data [12, 13]. In density clustering, the concept of core point can help to solve the problems of clusters with different sizes, shapes. In SNN similarity, the similarity between a pair of samples is measured by the number of the common neighbors in their nearest neighbor lists [20]. Since the rankings of the distances are still meaningful in high-dimensional space, SNN is regarded as a good secondary similarity measure for handling high-dimensional data [18]. Furthermore, given that SNN similarity only depends on the local configuration of the samples in the data space, the samples within dense clusters and sparse clusters will show roughly equal SNN similarities, which can mitigate the difficulty of clustering caused by the density variations of clusters.

The main phases of SNN clustering approaches can be summarized as follows: 1) defining the density of sample based on SNN

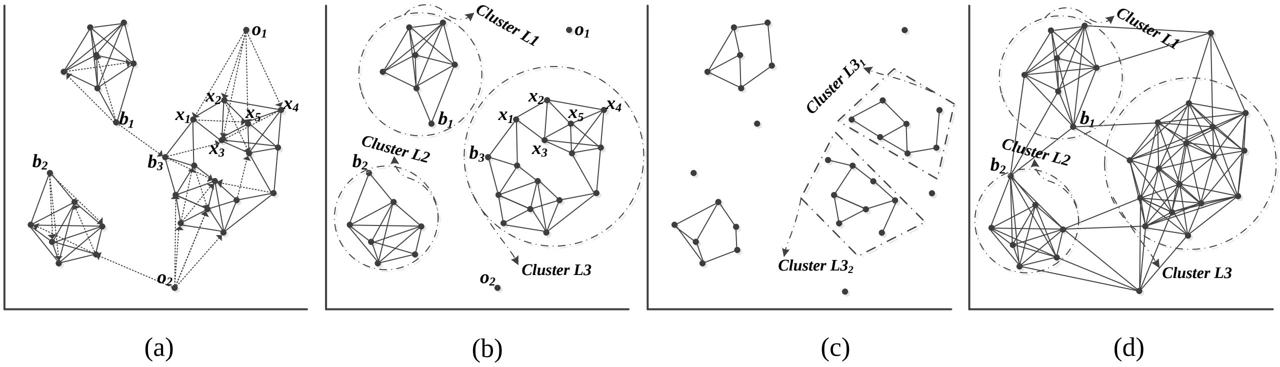

Figure 1: (a) Figures Illustrating a Nearest Neighbor Graph with $k = 5$. (b), (c) and (d) Figures Illustrating the Shared Nearest Neighbor Graph when k is 5, 3 and 10, respectively.

similarity; 2) finding the core points according to the densities of samples, and then defining directly density-reachable sample set for them; 3) building the clusters around the core points. Below, we describe the key concepts associated with these phases, respectively.

SNN similarity and the density of sample. For two samples x_i and x_j, their SNN similarity is given as follows,

$$SNN(x_i, x_j) = |N_k(x_i) \cap N_k(x_j)|, \tag{1}$$

where $N_k(x_i)$ and $N_k(x_j)$ are respectively the k-nearest neighbors of x_i and x_j, determined by certain primary similarity or distance measure (e.g., L_p norm).

In traditional density clustering, the density of a sample is defined as the number of the samples whose distances from this sample are not larger than the distance threshold Esp [14]. If this definition is extended to SNN clustering, the density of a sample x_i, $de(x_i)$, can be expressed as [12]:

$$de(x_i) = \textstyle\sum_{x \in SN_k(x_i)} SNN(x_i, x) * 1\{SNN(x_i, x) \geq Esp\}, \tag{2}$$

where $SN_k(x_i)$ is x_i's k-nearest neighbors according to SNN similarity, $1\{\cdot\}$ is an indicator function (it returns 1 if the relational expression is true, otherwise, 0 is returned). However, this kind of definition can make outliers and normal samples being non-discriminatory in density [13].

Consider Fig. 1a, the outliers o_1 and o_2 all have a considerable overlap degree of neighborhoods with their nearest neighbors. Hence, o_1 and o_2 are also high density according to Eqn. 2. To solve this problem, Eqn. 2 can be modified as

$$de(x_i) = \textstyle\sum_{x \in SN_k(x_i)} SNN(x_i, x) * \Im(x, x_i, Esp), \tag{3}$$

where $\Im(x, x_i, Esp)$ is $1\{x_i \in SN_k(x)\} * 1\{SNN(x_i, x) \geq Esp\}$. Eqn. 3 indicates only the samples occurred in the k nearest neighbor lists each other can contribute the similarity into their densities. Fig. 1b shows the corresponding shared nearest neighbor graph of Fig. 1a. From this figure, we can see that the outliers (o_1 and o_2) and their neighboring samples does not form pairs of shared nearest neighbors (i.e., there are no links), the densities of o_1 and o_2 tend to be 0; at the same time, the links of the border samples such as b_1, b_2 and b_3 are relatively sparse, their densities will be naturally lower than the densities of the samples within clusters. Eqn. 3 can benefit to obtain a reasonable distribution of sample density.

Core points and directly density-reachable sample set. In SNN clustering, the core points are the samples whose densities are higher the density threshold $MinPts$, and the directly density-reachable sample set of a core point is defined as those shared nearest neighbors which the similarities with this core point exceed Esp [13].

The creation of clusters. The core points, that are directly density-reachable each other, are put into the same clusters; all the samples that are not directly density-reachable with any core points are categorized as outliers (or noisy samples); and the non-core and non-noise points are assigned to the clusters in which their nearest core points are.

2.1.2 *DRSNN: A Density Ratio-based Shared Nearest Neighbor Clustering Method.*

$MinPts$ and Esp are two important parameters in SNN clustering, but, as we know, there is no general principle to set the "right" values for them [13]. In addition, SNN clustering is also sensitive to the neighbor parameter k [13, 18]. If k is set to be small, a complete cluster may be broken up into several small pieces, due to the limited density-reachable samples and the local variations in similarity. Consider Fig. 1c where the parameter k is set 3. The directly density-reachable sets of the points in the blocks $L3_1$ and $L3_2$ are restricted in the respective blocks, $L3_1$ and $L3_2$ cannot be combined into the integrated cluster $L3$. On the other hand, if k is too large such as being greater than the size of clusters, multiple clusters are prone to merge into a cluster, as the changes of density in transition regions will not have a substantially effect for separating different clusters. As shown in Fig. 1d where the shared nearest neighbor graph is presented when k is 10. The border points b_1 and b_2 contain the points from different clusters in their directly density-reachable sets, and show roughly equal densities with the points inside of the clusters (i.e., easy to be the core points). Hence, L_1, L_2, and L_3 tend to form a uniform cluster. In conclusion, the major drawback of SNN clustering is hard to set the appropriate values for the parameters, causing unsteady performance of clustering.

To solve the problem mentioned above, we propose a new clustering method based on Density Ratio and SNN similarity, DRSNN. We first present the key components in DRSNN, then summarize the algorithm process of DRSNN.

The density of sample. To avoid the use of *Esp*, DRSNN defines the density of a sample as the sum of the similarity between this sample and each of its shared nearest neighbors. Formally, the density of the considered sample x_i, $de(x_i)$, is

$$de(x_i) = \textstyle\sum_{x \in SN_k(x_i)} (SNN(x_i, x) * 1\{x_i \in SN_k(x)\}). \tag{4}$$

The density ratio of sample and the identification of core point. Instead of finding the core points based on the density estimate, DRSNN uses the estimate of density ratio [37]. Specifically, the density ratio of a sample is the ratio of the density of this sample to the average density value of κ-nearest neighbors of this sample,

$$dr(x_i) = \frac{de(x_i)}{\frac{1}{\kappa}\sum_{x \in SN_\kappa(x_i)} de(x)}. \tag{5}$$

The core points can be defined as the samples whose density ratios are not lower a threshold value drT. The use of density ratio has the following advantages. 1) It facilitates to identify core points. Given that the core points are the samples with local high densities, the density-ratio threshold drT can be set to around 1. 2) The parameter $MinPts$ can be eliminated. 3) The density ratios of samples are not affected by the variations of clusters in density.

Directly density-reachable sample set. In DRSNN, we define the directly density-reachable sample set for the core point as follows:

$$\begin{aligned} H_\kappa(x_i) = \{x_i\} \cup SN_\kappa(x_i) \cup \\ \{x_j \in RSN_\kappa(x_i) | x_j \ is\ a\ core\ sample\} \end{aligned} \tag{6}$$

where $RSN_\kappa(x_i)$ is x_i's reverse κ-nearest neighbors set. The directly density-reachable set $H_\kappa(x_i)$ mainly includes two parts, i.e., the κ-nearest neighbors of x_i and the core points in the reverse κ-nearest neighbors of x_i. The definition of $H_\kappa(\cdot)$ is based on two considerations. One is that the samples distributed closely around a core point should be directly density-reachable with this core point. The other one is to assure that $H_\kappa(\cdot)$ satisfies reflexivity and symmetry, which is a key condition that DRSNN can deterministically discover the clusters of arbitrary shape [25]. Note that the parameter κ can restrain the mergence of clusters by using a small value to shrink directly density-reachable sample set, and reduce the risk of splitting the clusters by employing a large value to augment the set of directly density-reachable samples.

The summary of DRSNN algorithm. DRSNN algorithm can be summarized as follows:

1) Find k-nearest neighbors of minority samples according to certain primary similarity or distance measure.
2) Calculate SNN similarity. For all pairs of minority samples, compute their SNN similarities as Eqn. 1.
3) Calculate the density of each sample as Eqn. 4.
4) Calculate the density ratio of each sample as Eqn. 5.
5) Identify the core points, i.e., all the samples that have a density ratio greater than drT.
6) Find the directly density-reachable sample set for each core point as Eqn. 6.
7) Build the clusters. The core points, that are directly density-reachable each other, are placed in the same clusters; the samples which are not directly density-reachable with any core points are treated as outliers; finally, all the other points are assigned to the clusters where their directly density-reachable core points are.

Although DRSNN also contains three parameters (i.e., drT, k and κ), it is capable of selecting the proper value for drT around 1. In addition, k and κ can be set in complementary way to avoid the mergence and dissociation of clusters, i.e., a large k, compared to the number of samples, with a relative low κ, while a small k accompanied by a relative high κ.

2.2 Shrinkage Estimation of Large-dimensional Covariance Matrix

In the setting of high dimensionality and small sample, the sample covariance matrix is not anymore an accurate and reliable estimate of the true covariance matrix Σ [16]. The shrinkage technique, as one of the most common methods improving the estimate of covariance matrix, aims to linearly combine the unrestricted sample covariance matrix $\mathbf{S}$ and a constrained target matrix $\mathbf{F}$ to yield a shrinkage estimator with less estimation error [22, 26], i.e.,

$$\mathbf{S}^* = \alpha\mathbf{F} + (1-\alpha)\mathbf{S}, \tag{7}$$

where $\alpha \in [0,1]$ is the weight assigned to the target matrix $\mathbf{F}$, called the shrinkage intensity. Since there are a lot of estimated parameters in $\mathbf{S}$ and a limited amount of data, the unbiased $\mathbf{S}$ will exhibit a high variance, whereas the preset $\mathbf{F}$ will have relatively low variance but potentially high bias as it is presumed to impose certain low-dimensional structure. The shrinkage technique can acquire more precise estimate for Σ by taking a properly trade-off between $\mathbf{S}$ and $\mathbf{F}$ [26].

A key question is how to find the optimal shrinkage intensity. Once α is obtained, the shrinkage estimator $\mathbf{S}^*$ can be determined. A popular solution is to analytically choose the value of α by minimizing Mean Squared Error (MSE) [23]. The advantages of this way are that the resulting estimator is distribution-free and inexpensive in computational complexity. Specifically, the MSE can be expressed as the squared Frobenius norm of the difference between Σ and $\mathbf{S}^*$,

$$L(\alpha) = \|\alpha\mathbf{F} + (1-\alpha)\mathbf{S} - \Sigma\|^2, \tag{8}$$

which leads to the risk function

$$\begin{aligned} R(\alpha) = E(L(\alpha)) &= \sum_{i=1}^{d}\sum_{j=1}^{d} E(\alpha f_{ij} + (1-\alpha)s_{ij} - \sigma_{ij})^2 \\ &= \sum_{i=1}^{d}\sum_{j=1}^{d} Var(\alpha f_{ij} + (1-\alpha)s_{ij}) + [E(\alpha f_{ij} + (1-\alpha)s_{ij} - \sigma_{ij})]^2 \\ &= \sum_{i=1}^{d}\sum_{j=1}^{d} \alpha^2 Var(f_{ij}) + (1-\alpha)^2 Var(s_{ij}) + \\ &\quad 2\alpha(1-\alpha)Cov(f_{ij}, s_{ij}) + [\alpha E(f_{ij} - s_{ij}) + Bias(s_{ij})]^2. \end{aligned} \tag{9}$$

f_{ij}, s_{ij} and σ_{ij} are the elements of $\mathbf{F}$, $\mathbf{S}$ and Σ, respectively; d is the dimension of feature. Since $\mathbf{S}$ is an unbiased estimator of Σ, $Bias(s_{ij})$ is actually 0.

Computing the first and two derivatives of $R(\alpha)$ yields the following equations

$$\begin{aligned} R'(\alpha) = 2\sum_{i=1}^{d}\sum_{j=1}^{d} \alpha Var(f_{ij}) + (1-\alpha)Var(s_{ij}) + \\ (1-2\alpha)Cov(f_{ij}, s_{ij}) + \alpha(E(f_{ij} - s_{ij}))^2, \end{aligned} \tag{10}$$

Algorithm 1 OHIT(P, k, κ, drT, η)

```
Require: $P$: the minority sample set; $k, \kappa, drT$: three parameters in DRSNN clustering; $\eta$: the number of synthetic samples required to be generated;
Ensure: $Syn$: the generated synthetic sample set
Employ DRSNN to cluster the minority class samples, $C_i \leftarrow DRSNN(P, k, \kappa, drT), i = 1, 2, ...m$, where $m$ is the number of discovered clusters.
Compute the shrinkage covariance matrix $\mathrm{S}_\mathrm{i}^*$ for each cluster $C_i$ by combining Eqns. 7, 14, and 15.
Generate the synthetic sample set $Syn_i$ with size $\lceil \eta \frac{|C_i|}{|P|} \rceil$ for $C_i$ based on $\mathrm{N}(\mu_\mathrm{i}, \mathrm{S}_\mathrm{i}^*)$, then add $Syn_i$ into $Syn$.
```

$$R''(\alpha) = 2\sum_{i=1}^{d}\sum_{j=1}^{d} Var(f_{ij} - s_{ij}) + (E(f_{ij} - s_{ij}))^2. \qquad (11)$$

By setting $R'(\alpha) = 0$, we can obtain

$$\alpha^* = \frac{\sum_{i=1}^{d}\sum_{j=1}^{d} Var(s_{ij}) - Cov(f_{ij}, s_{ij})}{\sum_{i=1}^{d}\sum_{j=1}^{d} E[(f_{ij} - s_{ij})^2]}. \qquad (12)$$

$R''(\alpha)$ is positive according to Eqn. 11. Hence, α^* is a minimum solution of $R(\alpha)$. Following [26], we replace the items of expectations, variances, and covariances in Eqn. 12 with their unbiased sample counterparts, which gives rise to

$$\hat{\alpha}^* = \frac{\sum_{i=1}^{d}\sum_{j=1}^{d} \hat{Var}(s_{ij}) - \hat{Cov}(f_{ij}, s_{ij})}{\sum_{i=1}^{d}\sum_{j=1}^{d} (f_{ij} - s_{ij})^2}. \qquad (13)$$

For the preset **F**, we use the covariance matrix implied by Sharpe's single-index model [27]. The single-index model is used to forecast stock returns from time-series stock exchange data. In this case, **F** can be expressed by S as follows,

$$f_{ij} = \begin{cases} s_{ij} & if \quad i = j \\ \sqrt{s_{ii}s_{jj}} & otherwise. \end{cases} \qquad (14)$$

Putting Eqn. 14 into Eqn. 13, an expression of $\hat{\alpha}^*$, that only contains the elements of sample covariance matrix, can be obtain finally

$$\hat{\alpha}^* = \frac{\sum_{i \neq j} \hat{Var}(s_{ij}) - t_{ij}}{\sum_{i \neq j}(\sqrt{s_{ii}s_{jj}} - s_{ij})^2}, \qquad (15)$$

where $t_{ij} = \frac{1}{2}[\sqrt{s_{jj}/s_{ii}}\hat{Cov}(s_{ii}, s_{ij}) + \sqrt{s_{ii}/s_{jj}}\hat{Cov}(s_{jj}, s_{ij})]$.

Given that the value of $\hat{\alpha}^*$ may be greater (/samller) than 1 (/0) due to limited samples, $\hat{\alpha}^{**} = \max(0, \min(1, \hat{\alpha}^*))$ is often adopted in practice.

2.3 Generation of Structure-preserving Synthetic Samples

The generation of synthetic samples of OHIT is simple. For a cluster i discovered by DRSNN, we first compute its mean of cluster (μ_i) and shrinkage covariance matrix (S_i^*), then the synthetic samples are yielded based on the Gaussian distribution $N(\mu_\mathrm{i}, \mathrm{S}_\mathrm{i}^*)$. In this way, the synthetic samples can maintain the covariance structure of each mode.

2.4 OHIT Algorithm and Complexity Analysis

Algorithm 1 summarizes the process of OHIT. Note that the actual data may not follow Gaussian distribution, but separately treating each mode is analogous to approximating the underlying distribution of minority class by the mixture of multiple Gaussian distribution, which alleviates the negative impacts from the violation of assumption to some degree.

The computational complexity of OHIT primarily consists of performing DRSNN clustering and estimating covariance matrix. Once the similarities are calculated for all pairs of samples (complexity–$O(nd^2)$), DRSNN only requires $O(n^2)$ to accomplish the process of clustering [13], while computing shrinkage covariance estimator has equal time complexity with the calculation of sample covariance matrix [26]. Hence, the complexity of OHIT can be finally simplified to $O(nd^2)$ in the case of high dimensionality and small sample. This time requirement is same with that of simple SMOTE, which shows that OHIT is very efficient in computation.

3 EXPERIMENTAL RESULTS AND DISCUSSION

3.1 Experimental setting

Experimental Datasets. We construct two groups of binary imbalanced datasets from the UCR time series repository [9]. For the first group, the minority class of each dataset is the smallest original class in the data, and the majority class consists of all the remaining original classes. We call this group the "unimodal" data group, where "unimodal" refers to that the minority class is formed by only one original class [5]. Table 1 presents the data characteristics of this group. It is worth pointing out that all the binary datasets whose imbalance ratios are higher than 1.5 in 2015 UCR repository have been added into this group, including Hr, Sb, POC, Lt2, PPOC, E200, Eq, and Wf.

For the second group, the minority class of each dataset is constructed by merging two or three smallest original classes, and the majority class is composed of the remaining original classes. If the small original classes have very limited samples, we combine three smallest original classes into the minority class, otherwise, two smallest original classes are merged. The datasets of this group are to simulate the scenario that the minority class is indeed multimodal. We call them the multi-modal data group. Since our OHIT considers the multi-modality of minority class, OHIT is expected to perform well on this group. Table 2 summarizes the data characteristics of this group, where the feature dimension is greater than the number of minority samples on all the datasets.

Assessment metrics. In imbalanced learning area, F-value and G-mean are two widely used comprehensive metrics which can reflect the compromised performance on the majority class and minority class. The definitions of them are as follows:

$$F - value = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \quad G - mean = \sqrt{Recall \cdot Specificity},$$

where recall and precision are the measures of completeness and exactness on the minority class, respectively; specificity is the measure of prediction accuracy on the majority class. Another popular overall metric is the Area Under the receiver operating characteristics Curve (AUC) [15]. Unlike F-value and G-mean, AUC is not

Table 1: Summary of the Imbalanced Unimodal Time-series Datasets Used in the Experiments

Dataset	Minority Class	Length	Training		Test	
			Class Distribution	IR	Class Distribution	IR
Yoga(Yg)	’1’	426	137/163	1.19	1393/1607	1.15
Herring(Hr)	’2’	512	25/39	1.56	26/38	1.46
Strawberry(Sb)	’1’	235	132/238	1.8	219/394	1.8
PhalangesOutlinesCorrect(POC)	’0’	80	628/1172	1.87	332/526	1.58
Lighting2(Lt2)	’-1’	637	20/40	2	28/33	1.18
ProximalPhalanxOutlineCorrect(PPOC)	’0’	80	194/406	2.09	92/199	2.16
ECG200(E200)	’-1’	96	31/69	2.23	36/64	1.78
Earthquakes(Eq)	’0’	512	35/104	2.97	58/264	4.55
Two_Patterns(Tp)	’2’	128	237/763	3.22	1011/2989	2.96
Car	’3’	577	11/49	4.45	19/41	2.16
ProximalPhalanxOutlineAgeGroup(PPOA)	’1’	80	72/328	4.56	17/188	11.06
Wafer(Wf)	’-1’	152	97/903	9.3	665/5499	8.27

IR is the imbalance ratio (#majority class samples/#minority class samples).

Table 2: Summary of the Imbalanced Multi-modal Time-series Datasets Used in the Experiments

Dataset	Minority Class	Length	Training		Test	
			Class Distribution	IR	Class Distribution	IR
Worms(Ws)	’5’, ’2’, ’3’	900	31/46	1.48	73/108	1.48
Plane(Pl)	’3’, ’5’	144	36/69	1.92	54/51	0.944
Haptics(Ht)	’1’, ’5’	1092	51/104	2.04	127/181	1.43
FISH	’4’, ’5’	463	43/132	3.07	57/118	2.07
UWaveGestureLibraryAll(UWGLA)	’8’, ’3’	945	206/690	3.35	914/2668	2.92
InsectWingbeatSound(IWS)	’1’, ’2’	256	40/180	4.5	360/1620	4.5
Cricket_Z(CZ)	’3’, ’5’	300	52/338	6.5	78/312	4
SwedishLeaf(SL)	’10’, ’7’	128	54/446	8.26	96/529	5.51
FaceAll(FA)	’1’, ’2’	131	80/480	12	210/1480	7.05
MedicalImages(MI)	’5’, ’6’, ’8’	99	23/358	15.57	69/691	10
ShapesAll(SA)	’1’, ’2’, ’3’	512	30/570	19	30/570	19
NonInvasiveFatalECG_Thorax1(NIFT)	’1’, ’23’	750	71/1729	24.35	100/1865	18.65

affected by the decision threshold of classifiers. In this paper, we use F-value, G-mean, and AUC to assess the performance of algorithms.

Base classifier. Previous studies [3–5] have been shown that Support Vector Machines (SVM) in conjunction with the oversampling technique SPO (/INOS/MoGT) can acquire better performance in terms of F-value and G-mean than the state-of-the-art approaches 1NN [24] and 1NN-DTW [32] for classifying imbalanced time series data. Hence, we select SVM with linear kernel as base classifier for achieving the experimental comparisons.

The parameter C of SVM is optimized by a nested 5-fold cross-validation over the training data. The considered values are $\{2^{-3}, 2^{-2}, \ldots, 2^{10}\}$. For each experimental dataset, the oversampling algorithm is applied to handle the training data so as to balance class distribution. Given that oversampling techniques involve the use of random numbers in the process of yielding synthetic samples, we run the oversampling method 10 times on the training data, the final performance result is the average of 10 results classifying the test data.

3.2 Comparison of Oversampling Methods

To evaluate the effectiveness of OHIT, we compare OHIT with existing representative oversampling methods, including random oversampling ROS, interpolation-based synthetic oversampling SMOTE, structure-preserving oversampling MDO and INOS, and the mixture model of Gaussian trees MoGT. With respect to the setting of parameters, the parameter values of OHIT are $k1 = 1.5\sqrt{n}$, $\kappa = \sqrt{n}$, and $drT = 0.9$, where n is the number of minority samples. All the other methods use the default values recommended by the corresponding authors. Specifically, the neighbor parameter K of SMOTE is set to 5; the parameters $K1$ and $K2$ in MDO are 5 and 10, respectively; for INOS, 70% of the synthetic samples are generated from the Gaussian distribution reflected the covariance structure of minority class (i.e., $r = 0.7$); in MoGT, the Bayesian information criterion is used to determine the number of mixture components.

Tables 3 and 4 respectively present the classification performances of all the compared algorithms on the unimodal datasets and multimodal datasets, where *original* represents SVM without combining any oversampling. For two data groups, *original* shows the worst results on most of the datasets in terms of F-value and

Table 3: Performance Results of all the Compared Methods on the Imbalanced Unimodal Datasets.

Metrics	Methods	Datasets												Average
		Yg	Hr	Sb	POC	Lt2	PPOC	E200	Eq	TP	Car	PPOA	Wf	
F-measure	Original	*.5652*	NaN	**.9571**	*.3584*	*.5778*	*.7013*	*.7143*	NaN	*.5780*	*.6875*	*.4242*	*.3163*	*.5812*
	ROS	0.6073	0.4053	0.9513	**0.5642**	0.6117	0.7379	0.7431	0.1107	0.6367	0.8128	**0.5320**	0.6511	0.6137
	SMOTE	0.5949	**0.5027**	*0.9366*	0.4809	**0.6961**	0.7244	0.7678	0.1946	0.6379	**0.8399**	0.4369	0.5997	0.6177
	MDO	0.6072	0.4055	0.9478	0.5535	0.6299	0.7471	0.7336	*0.0624*	0.6282	0.6875	0.4955	0.6003	0.5915
	INOS	0.6130	0.4174	0.9485	0.5060	0.6039	0.7426	0.7537	0.0750	0.6469	0.7809	0.5051	0.6150	0.6007
	MoGT	0.5717	*0.3920*	0.9451	0.5352	0.6533	0.7431	**0.7739**	0.1870	0.5918	0.6875	0.4360	**0.6840**	0.6001
	OHIT	**0.6134**	0.4655	0.9513	0.5497	0.6543	**0.7496**	0.7650	**0.1971**	**0.6490**	0.8265	0.5096	0.5606	**0.6243**
G-mean	Original	.6180	*.0000*	.9688	*.4753*	*.6388*	*.7506*	*.7725*	*.0000*	*.6818*	*.7421*	*.6261*	*.4366*	*.5592*
	ROS	0.6386	0.5093	0.9677	**0.6349**	0.6624	0.8130	0.8002	0.2655	0.7724	0.8576	0.7691	**0.8301**	0.7101
	SMOTE	0.6232	**0.5824**	*0.9603*	0.5675	**0.7222**	0.8058	0.8219	0.3822	0.7827	**0.8823**	**0.8311**	0.8095	0.7309
	MDO	0.6399	0.5108	0.9651	0.6300	0.6734	0.8139	0.7904	0.1842	0.7586	0.7421	0.7259	0.8129	0.6873
	INOS	**0.6448**	0.5207	0.9667	0.5960	0.6555	0.8179	0.8092	0.2029	0.7785	0.8264	0.7758	0.8110	0.7005
	MoGT	*0.6071*	0.4993	0.9640	0.6160	0.6887	0.8178	**0.8261**	0.3788	0.7420	0.7421	0.7176	0.8129	0.7010
	OHIT	0.6421	0.5584	**0.9685**	0.6258	0.6952	**0.8217**	0.8181	**0.3956**	**0.7830**	0.8694	0.7928	0.8083	**0.7316**
AUC	Original	.6771	*.2490*	.9898	.6691	.7056	**.9044**	.9032	*.4681*	.8496	.9294	.8908	.8019	*.7532*
	ROS	0.6772	0.6174	0.9908	0.6687	*0.6992*	*0.8837*	0.8935	0.5320	0.8529	**0.9360**	*0.8731*	**0.8847**	0.7924
	SMOTE	0.6620	**0.6335**	0.9929	*0.6307*	0.7218	0.8860	0.9000	**0.5658**	0.8555	0.9311	0.9008	0.7595	0.7866
	MDO	0.6770	0.6029	0.9920	**0.6752**	**0.7267**	0.8987	0.8955	0.5314	0.8559	0.9259	0.9002	0.8754	0.7964
	INOS	**0.6862**	0.6205	0.9919	0.6511	0.7000	0.8880	*0.8924*	0.5317	**0.8625**	0.9309	0.9075	0.8625	0.7938
	MoGT	*0.6368*	0.6235	*0.9892*	0.6696	0.7013	0.8882	**0.9125**	0.5266	*0.8204*	*0.9067*	0.8962	*0.7578*	0.7774
	OHIT	0.6821	0.6270	**0.9931**	0.6645	0.7063	0.8989	0.9008	0.5341	0.8599	0.9340	**0.9098**	0.8714	**0.7985**

Best (/Worst) results are highlighted in bold (/italics) Type.

Table 4: Performance results of all the Compared Methods on the Imbalanced Multi-modal Datasets.

Metrics	Methods	Datasets												Average
		Ws	PI	Ht	FISH	UWGLA	IWS	CZ	SL	FA	MI	SA	NIFT	
F-measure	Original	*.0267*	*.9623*	*.4388*	*.8627*	.7178	.6182	NaN	*.3667*	.8262	*.2000*	.6667	**.7345**	*.5837*
	ROS	0.4639	0.9670	0.5365	0.9004	*0.6533*	*0.5852*	*0.3699*	**0.8120**	**0.8338**	0.3509	0.4777	0.6947	0.6371
	SMOTE	0.5049	**0.9804**	0.6220	0.8753	0.7050	0.6601	**0.5528**	0.6385	0.8099	0.3928	*0.4748*	*0.3544*	0.6309
	MDO	0.4926	0.9718	0.6077	0.8841	0.6757	0.6749	0.4309	0.7325	0.8193	**0.6099**	0.6837	0.6458	0.6857
	INOS	0.4680	0.9679	0.5924	0.8955	0.7350	0.6853	0.4727	0.7855	0.8147	0.4363	0.4758	0.6804	0.6675
	MoGT	0.4679	0.9751	0.6153	0.8882	0.7279	**0.6962**	0.3902	0.7868	*0.7563*	0.4429	0.6195	0.5874	0.6628
	OHIT	**0.5053**	0.9670	**0.6293**	**0.9006**	**0.7405**	0.6831	0.5148	0.7866	0.8108	0.4489	**0.6872**	0.6927	**0.6972**
G-mean	Original	*.1165*	*.9623*	*.5385*	*.8749*	.7925	*.7077*	*.0000*	*.4778*	*.8768*	*.3580*	*.7290*	.8036	*.6031*
	ROS	0.5485	0.9669	0.6119	**0.9216**	*0.7670*	0.7720	0.5301	0.8778	0.8860	0.6328	0.8665	0.8072	0.7657
	SMOTE	0.5682	**0.9801**	0.6719	0.9177	0.8360	0.8443	**0.7536**	0.8752	0.8993	0.7561	0.9111	**0.8794**	0.8244
	MDO	0.5716	0.9717	0.6670	0.9003	0.7855	0.7980	0.5986	0.8479	0.8882	0.7727	0.9302	*0.7857*	0.7931
	INOS	0.5527	0.9679	0.6544	0.9157	0.8416	0.8474	0.6420	0.8902	0.9007	0.7656	0.8999	0.8367	0.8096
	MoGT	0.5533	0.9747	0.6728	0.9054	0.8288	0.8407	0.5760	0.9049	0.8871	0.7387	0.9078	0.8030	0.7994
	OHIT	**0.5768**	0.9670	**0.6843**	0.9134	**0.8448**	**0.8501**	0.6914	**0.9071**	**0.8996**	**0.7799**	**0.9305**	0.8520	**0.8247**
AUC	Original	*.4359*	.9975	.7101	.9496	.9072	.8835	.7349	.9587	.9598	.8559	.9271	.9605	.8567
	ROS	0.5549	0.9961	*0.6735*	0.9487	*0.8627*	*0.8388*	0.7214	0.9506	**0.9605**	*0.7120*	*0.9110*	0.9708	*0.8418*
	SMOTE	0.5738	0.9981	0.7125	0.9500	0.9093	0.8978	**0.8220**	0.9444	0.9531	0.7977	0.9242	*0.9534*	0.8697
	MDO	0.5591	0.9990	0.7112	0.9457	0.8770	0.8753	0.7602	*0.9214*	0.9544	0.8053	0.9331	0.9681	0.8592
	INOS	0.5646	0.9956	0.7084	0.9517	0.9074	0.9009	0.7738	0.9493	0.9525	0.8547	0.9212	0.9721	0.8710
	MoGT	0.5598	0.9967	0.7107	*0.9447*	0.9101	0.8961	*0.6951*	0.9387	*0.9389*	0.8384	0.9267	0.9560	0.8593
	OHIT	0.5709	**0.9985**	**0.7285**	**0.9519**	**0.9110**	**0.9011**	0.8141	**0.9610**	0.9541	**0.8767**	**0.9362**	**0.9734**	**0.8815**

Best (/Worst) results are highlighted in bold (/italics) Type.

G-mean, while OHIT achieves the best average performances in all the metrics.

In order to verify whether OHIT can significantly outperform the other compared algorithms, we perform the Wilcoxon signed-ranks test on the classification results of Tables 3 and 4. The test results are summarized in Table 5, where “+” and “*” denote the corresponding p value is not greater than 0.05 and 0.1, respectively. From Table 5, one can see that the p values on most of the significant tests are not beyond 0.05, and there are more significant differences on the multimodal datasets in comparison with the unimodal datasets.

It is worth noting that the significant difference has not been found between OHIT and SMOTE over the unimodal data group. To more granularly investigate the performance differences of these two algorithms, we compute the recall, specificity, and precision values of them on the unimodal datasets. The results are summarized in Table 6. According to Table 6, SMOTE performs better in recall, but does not statistically outperform OHIT in terms of recall;

Table 5: Summary of *p*-values of Wilcoxon Significance Tests Between OHIT and each of the Other Compared Methods

OHIT vs	Unimodal data			Multi-modal data		
	F-measure	G-mean	AUC	F-measure	G-mean	AUC
Original	**9.8e-4**$_{+}$	**9.8e-4**$_{+}$	**0.0552**$_{*}$	**0.0122**$_{+}$	**4.9e-4**$_{+}$	**0.0044**$_{+}$
ROS	0.4131	**0.0342**$_{+}$	**0.0425**$_{+}$	**0.042**$_{+}$	**0.0015**$_{+}$	**0.0034**$_{+}$
SMOTE	0.6772	0.9097	0.3013	**0.0342**$_{+}$	0.6221	**0.0342**$_{+}$
MDO	**0.0342**$_{+}$	**0.0093**$_{+}$	0.377	0.1099	**0.0015**$_{+}$	**0.0024**$_{+}$
INOS	**0.0342**$_{+}$	**0.0049**$_{+}$	**0.021**$_{+}$	**0.0425**$_{+}$	**0.0068**$_{+}$	**0.0015**$_{+}$
MoGT	**0.064**$_{*}$	**0.0122**$_{+}$	**0.0269**$_{+}$	**0.0269**$_{+}$	**0.0015**$_{+}$	**4.9e-4**$_{+}$

while OHIT obtains the higher specificity and precision values on most of the datasets, and is significantly better than SMOTE in specificity (/precision) at a significant level of 0.05 (/0.1). From this result, we can find that, compared to OHIT, SMOTE boosts the performance of minority class more aggressively, but at the same time causes the misclassification of more majority samples. One main reason may be that high dimensionality can aggravate the over-generalization problem of SMOTE. Since the space between two minority samples is increased exponentially with dimensionality, the synthetic samples interpolated by SMOTE can fall in huge region in high-dimensional space. Greatly expanding the minority class regions is beneficial to predict the minority samples, but it can also increase the risk of invading majority class regions.

Although the major advantage of OHIT is capable of dealing with the multi-modality of minority class, OHIT also exhibits the performance superiority on the unimodal datasets in comparison with MDO and INOS (Table 5). A congenital deficiency of MDO is that the covariance structure of minority class adopts the sample covariance matrix. INOS uses a regularization procedure to fix the unreliable eigenspectrum of sample covariance matrix, but does not consider the negative influence of outliers for the estimation of covariance matrix. Compared to INOS, OHIT can utilize DRSNN clustering to eliminate the outliers of minority class.

3.3 Evaluation of Separate OHIT Procedures

This experiment aims to evaluate the impacts of DRSNN clustering and shrinkage estimation on the performance of OHIT. To this end, we compare OHIT with the following OHIT variants: 1) OHIT without DRSNN clustering (denoted by OHIT/DRSNN); 2) OHIT without using shrinkage technique to improve covariance matrix estimator (OHIT/shrinkage); 3) OHIT replacing the shrinkage estimate of covariance matrix with the eigenspectrum regularization (OHIT with ER, the considered regularization is employed in INOS to alleviate the overadapted problem of sample covariance matrix).

Table 7 summarizes the average performance values of OHIT and its three variants (due to the limitation of space, the detailed experimental results are provided in Tables S1 and S2 in the supplementary material). The corresponding Wilcoxon test results between OHIT and each of its variants are presented in Table 8. We can find that OHIT is significantly better than all the variants in most of the cases, and more obvious advantages have been shown on the multimodal data group in comparison with the unimodal data group. It indicates that both DRSNN clustering and the shrinkage estimation of covariance matrix have positive effects on making OHIT to achieve better performance for high-dimensional imbalanced time series classification.

3.4 Comparison of Oversampling Mechanisms on a Toy Dataset

We visually compare OHIT and the other compared algorithms based on a two-dimensional toy dataset. Fig. 2a illustrates a balanced distribution, where each class has three modes and each mode contains 500 samples. In Fig. 2b, the minority class represented by blue pluses randomly retains 50 samples for each mode, so as to form imbalanced class distribution. Figs. 2c, 2d, 2e, 2f, 2g, and 2h show the augmented data after conducting ROS, SMOTE, MDO, INOS, MoGT, and OHIT on the minority class in sequence, where the introduced synthetic samples are denoted by red asterisks.

Based on these figures, the following observations can be obtained. 1) ROS does not effectively expand the regions of minority class, as the generated synthetic samples come from the replications of original minority samples (Fig. 2c). 2) SMOTE interpolates the synthetic samples between pairs of neighboring minority samples, which only considers the local characteristic of minority samples. Hence, the generated synthetic samples does not reflect the whole structures contained in the modes (Fig. 2d). 3) In MDO and INOS, the assumption, the minority class is unimodal, can lead to erroneous covariance matrix. The introduced synthetic samples can totally distort the original structure of each mode (Figs. 2e and 2f). 4) Although MoGT takes the multi-modality into account by building multiple Gaussian tree models for the minority class, the modes of minority class are not captured correctly on this toy dataset (Fig. 2g). In fact, the authors of MoGT assign the number of Gaussian tree models in manual way when modelling the minority class. However, the number of modes is unknown in practice. The developed algorithm should have the capability of detecting the modes of minority class automatically. 5) Different from MoGT, OHIT has identified all the modes correctly. Among all the compared algorithms, the augmented data by OHIT is the most similar to the original balanced data (Fig. 2a vs Fig. 2h).

4 CONCLUSION

The learning from imbalanced time-series data is challenging, since time series data tends to be high-dimensional and highly correlated in variables. In this study, we have proposed a structure preserving oversampling OHIT for the classification of imbalanced time-series data. To acquire the covariance structure of minority class correctly, OHIT leverages a DRSNN clustering algorithm to capture the multi-modality of minority class in high-dimensional space, and uses the shrinkage technique of covariance matrix to alleviate the problem of limited samples. We evaluated the effectiveness of OHIT on both the unimodal datasets and multi-modal datasets. The experimental results showed that OHIT can significantly outperform existing typical oversampling solutions in most of cases, and each of DRSNN clustering and shrinkage technique is important for enabling OHIT to gain better performance for classifying imbalanced time-series data.

Table 6: Recall, Specificity, and Precision of SMOTE and OHIT on the Imbalanced Unimodal Datasets.

Metrics	Methods	Datasets												Average
		Yg	Hr	Sb	POC	Lt2	PPOC	E200	Eq	TP	Car	PPOA	Wf	
Recall	SMOTE	0.5904	**0.4769**	**0.9913**	0.4855	0.6786	**0.8076**	**0.8639**	0.1672	**0.8675**	**0.8421**	**0.8471**	0.7129	**0.6943**
	OHIT	**0.6050**	0.4038	0.9849	**0.5587**	**0.5893**	0.7957	0.8056	**0.1845**	0.7705	0.8158	0.6941	**0.7316**	0.6616
Specificity	SMOTE	0.6580	0.7132	0.9302	0.6639	0.7697	0.8045	0.7828	**0.8792**	0.7118	0.9244	0.8160	**0.9193**	0.7978
	OHIT	**0.6817**	**0.7737**	**0.9523**	**0.7025**	**0.8212**	**0.8487**	**0.8313**	0.8489	**0.7958**	**0.9268**	**0.9069**	0.8933	**0.8319**
Precision	SMOTE	0.5996	0.5332	0.8877	0.4766	0.7160	0.6576	0.6921	**0.2346**	0.5045	**0.8379**	0.2945	**0.5183**	0.5794
	OHIT	**0.6223**	**0.5514**	**0.9200**	**0.5422**	**0.7368**	**0.7088**	**0.7287**	0.2119	**0.5607**	0.8377	**0.4032**	0.4554	**0.6066**

In terms of recall, specificity and precision, p-values of Wilcoxon test between OHIT and SMOTE are 0.1763, 0.0161, *and* 0.0674, *respectively.*

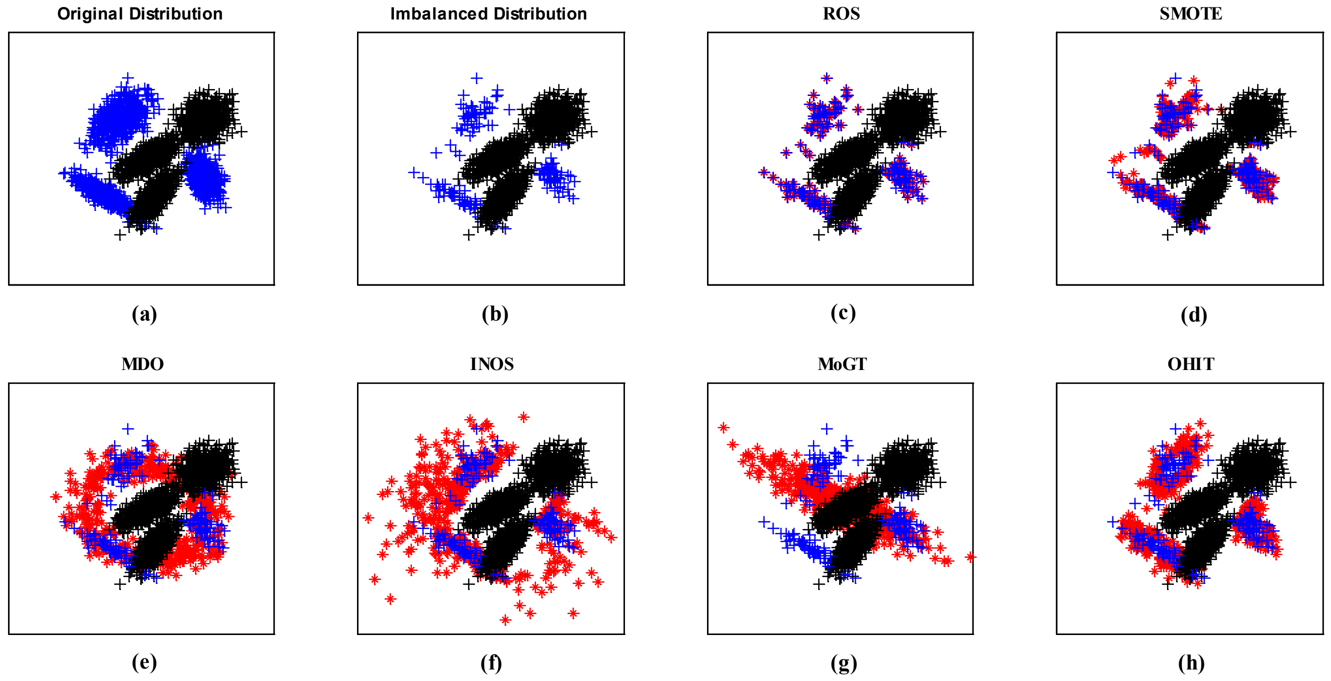

Figure 2: Visual Comparison: (a) Original Data; (b) Imbalanced Data; (c), (d), (e), (f), (g), and (h) are the Augmented Data by Performing ROS, SMOTE, MDO, INOS, MoGT, and OHIT, respectively.

Table 7: Average Performance of OHIT and its Variants Across all the Datasets within each Group.

OHIT vs	Unimodal data			Multi-modal data		
	F-measure	G-mean	AUC	F-measure	G-mean	AUC
OHIT / DRSNN	0.6229	0.7290	0.7935	0.6641	0.8131	0.8751
OHIT/ shrinkage	0.5988	0.6977	0.7938	0.6523	0.7769	0.8475
OHIT with ER	0.5938	0.6974	0.7939	0.6619	0.7900	0.8519
OHIT	**0.6243**	**0.7316**	**0.7985**	**0.6972**	**0.8247**	**0.8815**

Best results are highlighted in bold type.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (Project No. 61872131). We thank the authors of MDO, INOS, and MoGT for sharing their algorithm codes with us.

Table 8: Summary of p-values of Wilcoxon Significance Tests Between OHIT and each of its Variants

OHIT vs	Unimodal data			Multi-modal data		
	F-measure	G-mean	AUC	F-measure	G-mean	AUC
OHIT / DRSNN	0.5771	0.1973	$\mathbf{0.0039}_{+}$	$\mathbf{0.021}_{+}$	$\mathbf{0.0054}_{+}$	$\mathbf{0.0244}_{+}$
OHIT/ shrinkage	0.1763	$\mathbf{0.0923}_{*}$	$\mathbf{0.0923}_{*}$	$\mathbf{0.0034}_{+}$	$\mathbf{0.0068}_{+}$	$\mathbf{0.0049}_{+}$
OHIT with ER	$\mathbf{0.021}_{+}$	$\mathbf{9.8e\text{-}4}_{+}$	$\mathbf{0.0356}_{+}$	$\mathbf{0.0269}_{+}$	$\mathbf{0.0313}_{+}$	$\mathbf{0.0063}_{+}$

REFERENCES

[1] Lida Abdi and Sattar Hashemi. 2016. To combat multi-class imbalanced problems by means of over-sampling techniques. *IEEE Transactions on Knowledge and Data Engineering* 28, 1 (2016), 238–251.

[2] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. 2004. Applying support vector machines to imbalanced datasets. In *European conference on machine learning*. Springer, 39–50.

[3] Hong Cao, Xiao-Li Li, David Yew-Kwong Woon, and See-Kiong Ng. 2013. Integrated oversampling for imbalanced time series classification. *IEEE Transactions*

on Knowledge and Data Engineering 25, 12 (2013), 2809–2822.

[4] Hong Cao, Xiao-Li Li, Yew-Kwong Woon, and See-Kiong Ng. 2011. SPO: Structure preserving oversampling for imbalanced time series classification. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 1008–1013.

[5] H. Cao, V. Y. Tan, and J. Z. Pang. 2014. A parsimonious mixture of Gaussian trees model for oversampling in imbalanced and multimodal time-series classification. *IEEE Transactions on Neural Networks & Learning Systems* 25, 12 (2014), 2226–2239.

[6] Lu Cao and Yi-Kui Zhai. 2016. An over-sampling method based on probability density estimation for imbalanced datasets classification. In *Proceedings of the 2016 International Conference on Intelligent Information Processing*. ACM, 44.

[7] Cristiano L Castro and Antônio P Braga. 2013. Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. *IEEE transactions on neural networks and learning systems* 24, 6 (2013), 888–899.

[8] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* (2002), 321–357.

[9] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. 2015. The ucr time series classification archive.

[10] Yu-An Chung, Hsuan-Tien Lin, and Shao-Wen Yang. 2015. Cost-aware pre-training for multiclass cost-sensitive deep learning. *arXiv preprint arXiv:1511.09337* (2015).

[11] Barnan Das, Narayanan C Krishnan, and Diane J Cook. 2015. RACOG and wRACOG: Two probabilistic oversampling techniques. *IEEE transactions on knowledge and data engineering* 27, 1 (2015), 222–234.

[12] Levent Ertoz, Michael Steinbach, and Vipin Kumar. 2002. A new shared nearest neighbor clustering algorithm and its applications. In *Workshop on clustering high dimensional data and its applications at 2nd SIAM international conference on data mining*. 105–115.

[13] Levent Ertóz, Michael Steinbach, and Vipin Kumar. 2003. Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data. In *Siam International Conference on Data Mining, San Francisco, Ca, Usa, May*.

[14] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Vol. 96. 226–231.

[15] Tom Fawcett. 2004. ROC graphs: Notes and practical considerations for researchers. *Machine learning* 31, 1 (2004), 1–38.

[16] Jerome H Friedman. 1989. Regularized discriminant analysis. *Journal of the American statistical association* 84, 405 (1989), 165–175.

[17] Keinosuke Fukunaga. 2013. *Introduction to statistical pattern recognition*. Elsevier.

[18] Michael E. Houle, Hans Peter Kriegel, Peer Kroger, Erich Schubert, and Arthur Zimek. 2010. Can Shared-Neighbor Distances Defeat the Curse of Dimensionality?. In *International Conference on Scientific & Statistical Database Management*.

[19] Andrzej Janusz, Marek Grzegorowski, Marcin Michalak, Łukasz Wróbel, and Dominik Sikora. 2017. Predicting seismic events in coal mines based on underground sensor measurements. *Engineering Applications of Artificial Intelligence* 64 (2017), 83–94.

[20] R. A. Jarvis and E. A. Patrick. 1973. *Clustering Using a Similarity Measure Based on Shared Near Neighbors*.

[21] Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. 2018. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems* 29, 8 (2018), 3573–3587.

[22] Olivier Ledoit and Michael Wolf. 2003. Honey, I Shrunk the Sample Covariance Matrix. *Social Science Electronic Publishing* 30, 4 (2003), pÃągs. 110–119.

[23] Olivier Ledoit and Michael Wolf. 2003. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance* 10, 5 (2003), 603–621.

[24] Minh Nhut Nguyen, Xiao-Li Li, and See-Kiong Ng. 2011. Positive unlabeled learning for time series classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

[25] Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. 1998. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining Knowl. Discovery* 2, 2 (1998), 169–194.

[26] J Schäfer and K Strimmer. 2009. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Stat Appl Genet Mol Biol* 4, 1 (2009), Article32.

[27] William F. Sharpe. 1963. A Simplified Model for Portfolio Analysis. *Management Science* 9, 2 (1963), 277–293.

[28] Bo Tang and Haibo He. 2017. GIR-based ensemble sampling approaches for imbalanced learning. *Pattern Recognition* 71 (2017), 306–319.

[29] Elif Derya Übeyli. 2007. ECG beats classification using multiclass support vector machines with error correcting output codes. *Digital Signal Processing* 17, 3 (2007), 675–684.

[30] Jose R Villar, Paula Vergara, Manuel Menéndez, Enrique de la Cal, Víctor M González, and Javier Sedano. 2016. Generalized models for the classification of abnormal movements in daily life and its applicability to epilepsy convulsion recognition. *International journal of neural systems* 26, 06 (2016), 1650037.

[31] Ginny Y Wong, Frank HF Leung, and Sai-Ho Ling. 2014. An under-sampling method based on fuzzy logic for large imbalanced dataset. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*. IEEE, 1248–1252.

[32] Xiaopeng Xi, Eamonn J. Keogh, Christian R. Shelton, Li Wei, and Chotirat Ann Ratanamahatana. 2006. Fast Time Series Classification Using Numerosity Reduction. In *International Conference*.

[33] Xi Zhang, Di Ma, Lin Gan, Shanshan Jiang, and Gady Agam. 2016. Cgmos: Certainty guided minority oversampling. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 1623–1631.

[34] Zhi-Hua Zhou and Xu-Ying Liu. 2006. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge & Data Engineering* 1 (2006), 63–77.

[35] Tuanfei Zhu, Yaping Lin, and Yonghe Liu. 2017. Synthetic minority oversampling technique for multiclass imbalance problems. *Pattern Recognition* 72 (2017), 327–340.

[36] Tuanfei Zhu, Yaping Lin, Yonghe Liu, Wei Zhang, and Jianming Zhang. 2019. Minority oversampling for imbalanced ordinal regression. *Knowledge-Based Systems* 166 (2019), 140–155.

[37] Ye Zhu, Ming Ting Kai, and Mark J. Carman. 2016. Density-ratio based clustering for discovering clusters with varying densities. *Pattern Recognition* 60 (2016), 983–997.