Mathematical Biosciences and Engineering

MBE, 20(10): 17672–17701.
DOI: 10.3934/mbe.2023786
Received: 05 July 2023
Revised: 06 September 2023
Accepted: 11 September 2023
Published: 15 September 2023

http://www.aimspress.com/journal/MBE

Research article

Improved support vector machine classification for imbalanced medical datasets by novel hybrid sampling combining modified mega-trend-diffusion and bagging extreme learning machine model

Liang-Sian Lin[1,*], Chen-Huan Kao[1], Yi-Jie Li[1], Hao-Hsuan Chen[1] and Hung-Yu Chen[2]

[1] Department of Information Management, National Taipei University of Nursing and Health Sciences, Taipei 112303, Taiwan
[2] Department of Information Management, National Chin-Yi University of Technology, Taichung 411030, Taiwan

* **Correspondence:** Email: lianghsien@ntunhs.edu.tw.

Abstract: To handle imbalanced datasets in machine learning or deep learning models, some studies suggest sampling techniques to generate virtual examples of minority classes to improve the models' prediction accuracy. However, for kernel-based support vector machines (SVM), some sampling methods suggest generating synthetic examples in an original data space rather than in a high-dimensional feature space. This may be ineffective in improving SVM classification for imbalanced datasets. To address this problem, we propose a novel hybrid sampling technique termed modified mega-trend-diffusion-extreme learning machine (MMTD-ELM) to effectively move the SVM decision boundary toward a region of the majority class. By this movement, the prediction of SVM for minority class examples can be improved. The proposed method combines α-cut fuzzy number method for screening representative examples of majority class and MMTD method for creating new examples of the minority class. Furthermore, we construct a bagging ELM model to monitor the similarity between new examples and original data. In this paper, four datasets are used to test the efficiency of the proposed MMTD-ELM method in imbalanced data prediction. Additionally, we deployed two SVM models to compare prediction performance of the proposed MMTD-ELM method with three state-of-the-art sampling techniques in terms of geometric mean (G-mean), F-measure (F1), index of balanced accuracy (IBA) and area under curve (AUC) metrics. Furthermore, paired t-test is used to elucidate whether the suggested method has statistically significant differences from the other sampling techniques in terms of the four evaluation metrics. The experimental results demonstrated that the

proposed method achieves the best average values in terms of G-mean, F1, IBA and AUC. Overall, the suggested MMTD-ELM method outperforms these sampling methods for imbalanced datasets.

Keywords: imbalanced datasets; hybrid sampling approach; support vectors; virtual examples

1. Introduction

The imbalanced data classification problem frequently occurs in medical applications, including diabetes classification [1,2], cancer diagnosis [3−5] and biomedical data classification [6−9]. An imbalanced medical dataset indicates that the number of negative examples such as healthy individuals drastically exceed the number of positive examples or patients with diseases. Researchers thus often place much effort towards learning patterns in those minority patients with cancer or other rare diseases. Under this circumstance, traditional machine learning and deep learning models are often distorted towards the majority class on prediction results. As a result, these models often exhibit lower classification performance for the minority class. Under this scenario, these learning models fail to provide credible prediction results for doctors to make correct treatment decisions according to a patient's conditions. Consequently, we note researchers have devoted significant efforts for developing effective methods to overcome the imbalanced dataset problem in academic and real-world applications.

To deal with imbalanced datasets, some researchers proposed sampling techniques for balancing class distributions to improve overall classification accuracy of learning models. These sampling approaches can be classified into three categories: 1) under-sampling method; 2) over-sampling method; and 3) hybrid sampling method.

The 1) under-sampling method aims at reducing learning bias towards the majority class by removing some negative examples. Babar and Ade [10], for example, proposed an under-sampling technique based on the multi-layer perceptron (MLP) model to identify valuable samples and eliminate noise in the majority class. In [10], they divided majority class examples into several clusters and filtered critical examples according to stochastic measure evaluation. To improve breast cancer prediction with imbalanced data, Zhang et al. [11], for example, proposed an under-sampling method which utilizes k-means algorithm to select representative examples close to original examples in the majority class. Vuttipittayamongkol and Elyan [12] suggested an overlap-based under-sampling method that utilizes k-nearest neighbor (KNN) algorithm to find dangerous minority class examples (i.e., positive examples) that are surrounded by most of the majority class examples (i.e., negative examples). They excluded these negative examples to enhance prediction for positive examples.

The 2) over-sampling method directly raises the quantity of examples in the minority class by creating synthetic samples. The synthetic minority oversampling technique (SMOTE) proposed by Chawla et al. [13] is the most representative technique among over-sampling methods. In [13], they create new minority class examples using a linear interpolation method. In addition, they use the KNN algorithm to select new examples belonging to the minority class. To avoid synthetic examples falling into the majority class area, Bunkhumpornpat et al. [14] proposed the safe-level-SMOTE method to generate safe positive examples close to original positive examples. To reduce false positive rates, Cieslak et al. [15] proposed a clustering-based SMOTE sampling method (cluster-SMOTE), which partitions the original dataset into several subsets and generates new minority class examples using SMOTE with these subsets. Other than generating examples within the safe minority class region, de

la Calleja et al. [16] proposed the synthetic multi-minority oversampling (SMMO) method, which resamples misclassified positive examples as new instances to improve prediction accuracy of learning models for the minority class. Furthermore, Farquad and Bose [17] employed the support vector machines (SVM) model as a pre-processor (named the SVM-balance method) to resample misclassified data close to the raw minority class example as new samples for pushing the decision boundary toward the majority class.

The 3) hybrid sampling method is a combination of under-sampling and over-sampling methods. For instance, Wang [18] proposed a hybrid sampling SVM method which removes negative examples that are far from SVM's decision boundary and uses the SMOTE algorithm to create minority class examples with several training subsets. To improve SVM imbalanced classification on breast cancer diagnosis, Zhang and Chen [19] presented a hybrid of random over sampling example (ROSE), k-means and support vector machine (RK-SVM) methods, which consists of using ROSE to resample samples in the minority class and using k-means clustering method for keeping informative samples in the majority class.

As previously stated, the kernel-based over-sampling methods can effectively push SVM's decision boundary toward the majority class. However, when the margin on the hypersphere is very short, linearly interpolated examples using SMOTE may become dangerous new examples in the minority class, since they are very proximate to the area of the majority class. As a result, new examples of minority class may be regarded as noise or outliers that worsen classification accuracy of SVM for the minority class. Conversely, when the margin is wide, although safer examples of the minority class can be created, this softly shifts the SVM decision boundary toward the majority class. This may have tiny effects for improving SVM classification of skewed datasets. Overall, based on the above-mentioned problem, we note two challenging research questions (RQs), as follows:

RQ1: According to the above-mentioned studies, kernel-based SVM is prone to misclassifying minority class examples located near the decision boundary. To improve SVM classification for minority class examples, some studies aimed to generate kernel-based synthetic minority class examples to adjust SVM's decision boundary towards the region of majority class. We consulted the findings in [18] and [19], finding a hybrid sampling method can more effectively improve classification performances of SVM as compared to using a single under-sampling or over-sampling method. But which kind of hybrid sampling methods for creating synthetic examples of minority class and screening representative examples of majority class can further improve SVM imbalanced classification?

RQ2: The kernel-based oversampling methods aimed to create virtual samples of minority class nearby the SVM decision boundary. However, the generated virtual samples may be surrounded by most of the majority class examples. They are considered as danger minority class examples or noise, which distort learning of SVM. What kind of learning models can be used to monitor the similarity between synthetic examples and original data to screen acceptable minority class examples?

In order to address SVM imbalanced classification on RQ1 and RQ2, we develop a novel hybrid sampling method termed modified mega-trend-diffusion-extreme learning machines (MMTD-ELM) to adjust the SVM decision boundary to achieve improvement of SVM classification for imbalanced datasets. The major contributions of this paper are as follows:

a) To reduce bias of majority class examples for SVM models, based on a fuzzy triangular membership function (MF), we propose an under-sampling method using α-cut fuzzy number to screen representative SVs of majority class. The MF value of the example represents the possibility that the example belongs to the majority class. The higher MF value indicates that the

example has more representations for constructing the SVM decision boundary of the majority class. When the example has a higher value, it indicates a higher effect in predicting majority class.

b) To avoid generated new examples falling into the area of majority class, we proposed a modified MTD method, in which, MTD as proposed by Li et al. [20], is deployed to estimate the data range of support vectors of the minority class and generate the virtual data's inputs within the estimated data range. To predict labels corresponding to the virtual data's inputs, we construct a bagging-based extreme learning machine (ELM) model. In this paper, we feed the ELM using different datasets resampled from an original dataset. The bagging method, proposed by Breiman [21], can enable the ELM model to capture diverse patterns between inputs and output. With a bagging strategy, the prediction accuracy of the ELM model can be improved at identifying the virtual data's output.

c) Some studies about hybrid sampling methods [18,19] measured distance of majority class examples from each other and removed unrepresentative examples that were far from the SVM's decision boundary. However, the distance-based sampling method is easily impacted by noise or outliers. Differing from their papers, we developed a hybrid sampling method named MMTD-ELM, which consists of a under-sampling α-cut fuzzy number technique for screen representative examples of the majority class and a over-sampling MMTD technique for producing synthetic examples of minority class. In the proposed under-sampling method, we use MF value to measure the representation of the majority class example with low impact of noise, in which, MF value is used to measure potential information of majority class examples. By removing some examples and creating new examples near the decision boundary, we can effectively shift the SVM decision boundary towards the region of the majority class. By this shift, more minority class examples can be correctly predicted but only a few majority class examples may be misclassified. As a result, the proposed method can further improve SVM classification of the minority class.

In this paper, three medical datasets obtained from the Knowledge Extraction based on Evolutionary Learning (KEEL) dataset repository [22] and one medical dataset obtained from microarray gene expression cancer data [23] are used to test efficacy of the proposed MMTD-ELM method. Based on the four datasets, we compared the MMTD-ELM method with the IMB (using imbalanced datasets) method, which only uses an original imbalanced dataset without generating new examples, and three other sampling methods. The three sampling methods include the SMOTE method for interpolating examples of minority class, the SVM-balance method [17] for randomly generating SVs in the minority class, and the cluster-SMOTE method [15] for a distance-based hybrid sampling for imbalanced datasets. We construct two types of SVM models: SVM with polynomial kernel (SVM_poly) and SVM with radial basis function kernel (SVM_rbf) to test classification performance using these methods. Four evaluation metrics: geometric mean (G-mean) as seen in [24], F-measure (F1), index of balanced accuracy (IBA) as seen in [25] and area under curve (AUC) as seen in [26] are used to measure classification results with imbalanced datasets. Additionally, the paired t-test is used to examine whether the proposed MMTD-ELM method has statistically significant differences from the other methods in terms of four evaluation indicators. According to our experimental results, the proposed MMTD-ELM method outperforms the other four methods. For instance, when imbalance ratio between majority class and minority class is at 9:1, based on the four datasets, the proposed MMTD-ELM method achieves the best average values in terms of G-mean (0.901 and 0.914), F1 (0.877 and 0.885), IBA (0.719 and 0.742) and AUC (0.841 and 0.854) metrics for SVM_poly and SVM_rbf models, respectively.

The remainder of this paper is organized as follows: Section 2 introduces the SVM model; Section 3

illustrates the complete implementation procedure for the proposed MMTD-ELM method; Section 4 provides the description of four medical datasets and discusses the experimental results; and Section 5 concludes and discusses future work.

2. Related works

In this section, we present a literature review of sampling techniques for improving SVM imbalanced classification. Additionally, we introduce the SVM model for classification tasks.

2.1. Sampling approaches for improving SVM imbalanced classification

SVM proposed by Cortes and Vapnik [27], is a typical kernel-based learning algorithm to address classification work in many fields, such as air quality classification [28,29], medical diagnosis [19,30,31] and speech recognition [32,33]. The kernel-based SVM model maps original data onto a high-dimensional feature space to identify examples between different classes. The decision boundary is constructed by SVM that can effectively separate examples of different classes while minimizing training error. The examples located on SVM's decision boundary are called support vectors (SVs). The classification ability of the SVM model depends on the quantity of support vectors on the decision boundary. However, with skewed datasets, prediction results of SVM often tend towards the majority class because the learning model is trained using tiny examples of the minority class. To improve SVM classification performance for imbalanced datasets, some research has suggested employing sampling techniques to generate artificial minority class examples to balance data class distributions. However, randomly generating artificial examples cannot significantly improve SVM classification accuracy for skewed datasets since SVM's decision boundary may be slightly shifted towards the majority class. To deal with this issue, Zeng and Gao [34] addressed a kernel-based SMOTE method to generate virtual samples near the decision boundary of SVM on the minority class side to extend the margin of SVM's hyperplane. Other than over-sampling methods with SVs, Luo et al. [35] presented a hybrid sampling support vector data description (SVDD) method, which randomly deletes SVs in the majority class and generates SVs in the minority class using SMOTE to obtain balanced training datasets. However, eliminating SVs of the majority class may omit critical information for classifying majority class. At the same time, generating SVs using SMOTE might lead to new SVs surrounded by most majority class examples that become noise or outliers in the minority class.

2.2. SVM

Given a dataset of n samples: $(\vec{x}_1, y_1), (\vec{x}_2, y_2), ..., (\vec{x}_i, y_i)$, i = 1,2,...,n, where $\vec{x}_i \in \mathbb{R}^m$ is the input vector and $y_i \in \{-1, +1\}$ is the label of ith sample. According to the formula in [27], SVM classification satisfies the following condition:

$$\begin{cases} \mathbf{w}^T \varphi(\vec{x}_i) + b \geq +1, & \text{if } y_i = +1 \\ \mathbf{w}^T \varphi(\vec{x}_i) + b \leq -1, & \text{if } y_i = -1 \end{cases}, \tag{1}$$

where $\mathbf{w}$ represents weight vector, b is the bias and $\varphi(\cdot)$ is the mapping function for projecting original inputs onto a high-dimensional feature space. Based on Eq (1), SVM classification for these samples with different classes can be determined by Eq (2).

$$\text{sign}\left(\mathbf{w}^T\varphi(\vec{x}_i)+b\right). \tag{2}$$

According to the principle of structural risk minimization, the construction of SVM can be defined as a primal optimization problem, as follows:

$$\begin{aligned} &\text{Min}\ \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\xi_i) \\ &\text{s.t.}\ \ y_i(\mathbf{w}^T\cdot\varphi(\vec{x}_i)+b)\geq 1-\xi_i \\ &\xi_i\geq 0,\ i=1,2,\ldots,n, \end{aligned} \tag{3}$$

where $\|\mathbf{w}\|^2 = \mathbf{w}^T\cdot\mathbf{w}$, C is the error penalty parameter to control trade-off between acceptable classification error and maintaining decision boundary with maximum margin and ξ_i is a slack variable to allow tolerance for misclassification errors. According to Karush-Kuhn-Tucker (KKT) optimality conditions, we can reformulate Eq (3) as a quadratic optimization problem. To solve this optimization problem, we derive the problem with Lagrange multipliers α_i, as:

$$\begin{aligned} &\text{Max}\ \ \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(\vec{x}_i,\vec{x}_j) \\ &\text{s.t.}\ \ \sum_{i=1}^{n}\alpha_i y_i = 0 \\ &0\leq\alpha_i\leq C,\ i=1,2,\ldots,n, \end{aligned} \tag{4}$$

where when α_i is not equal to zero, it is called a support vector (SV) on the decision boundary and $k(\vec{x}_i,\vec{x}_j)$ is a kernel function noted as $\langle\varphi(\vec{x}_i),\varphi(\vec{x}_j)\rangle, \forall i\neq j$ for mapping non-linear $\vec{x}_i$ onto a high-dimensional space, as depicted in Figure 1.

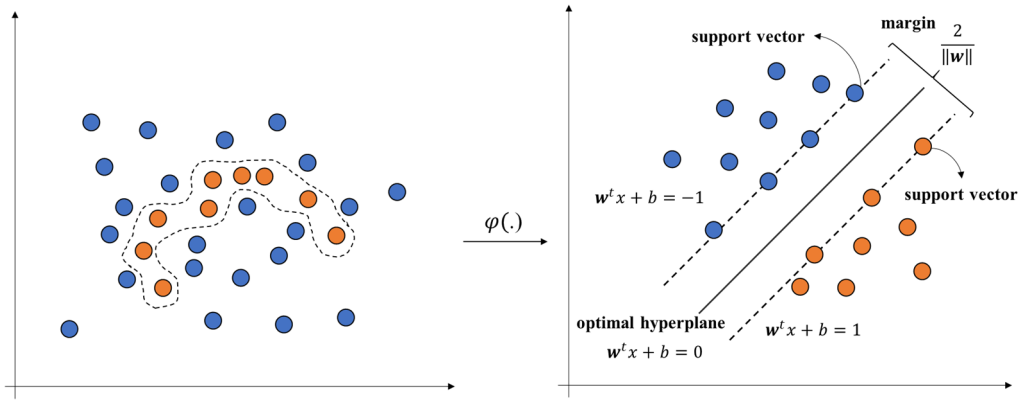


Figure 1. Mapping data by $\varphi(\cdot)$.

There are four categories of $k\left(\vec{x}_i,\vec{x}_j\right)$: linear kernel $\vec{x}_i^T\cdot\vec{x}_j$, polynomial kernel

$[\gamma \cdot (\vec{x}_i^T \cdot \vec{x}_j) + r]^d$, radial basis function kernel $\exp(-\gamma \| \vec{x}_i - \vec{x}_j \|^2)$ and sigmoid kernel $\tanh(\gamma(\vec{x}_i^T \cdot \vec{x}_j) + r)$, in which, $d \in \mathbb{N}$, $r \in \mathbb{N}$ and $\gamma \in \mathbb{R}^+$.

3. The proposed MMTD-ELM method

In this paper, we develop a unique hybrid sampling technique for improving SVM classification for skewed datasets. We explain the proposed method in depth in the following sections.

3.1. Method

Given a dataset has *n* samples with *m* input variables $X_1, X_2, ..., X_m$ and one output variable Y, which are denoted as $\{(\vec{x}_1, y_1), (\vec{x}_2, y_2), ..., (\vec{x}_i, y_i)\}$, $i = 1,2,..., n$. The $\vec{x}_i$ expresses *i*th data vector, and y_i is its label. We utilize the min-max data normalization to eliminate effects between *m* input variables $X_1, X_2, ..., X_m$ with different scales before implementing the suggested MMTD-ELM technique. The data normalization formula is expressed as:

$$\tilde{x}_{i,j} = \frac{x_{i,j} - \min(X_j)}{\max(X_j) - \min(X_j)} \in [0,1], j = 1,2,\ldots,m, \tag{5}$$

where $\tilde{x}_{i,j}$ is normalized data, $\max(X_j)$ is the maximum value of the *j*th input variable and $\min(X_j)$ is the minimum value of the *j*th input variable. To address classification problems posed by imbalanced datasets, the proposed MMTD-ELM hybrid sampling method consists of two stages: under-sampling and over-sampling stages. To balance skewed class distribution, in the under-sampling stage, we screen representative support vectors (SVs) in the majority class; in the over-sampling stage, we create new minority class examples. By implementing the proposed MMTD-ELM method, a new balanced dataset is obtained. The designed hybrid sampling procedure is illustrated in Figure 2.

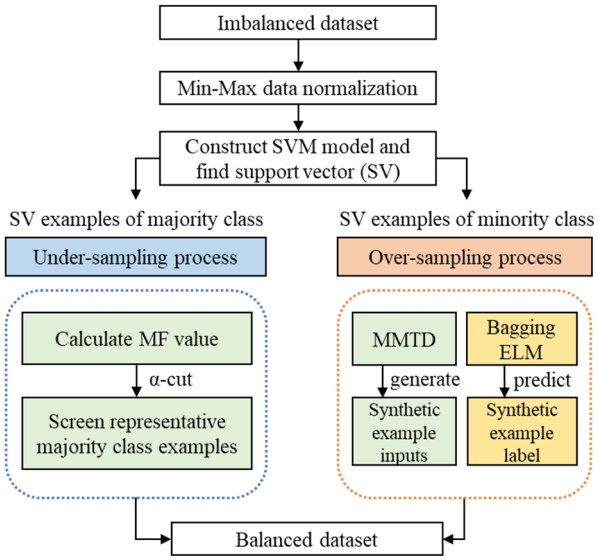


Figure 2. Proposed MMTD-ELM method procedure.

3.2. Implementing MMTD-ELM technique

The proposed method is aimed at effectively shifting the SVM model decision boundary towards the majority class to improve SVM classification for imbalanced datasets. The proposed method is developed to select significant majority class SVs and create artificial minority class SVs within the estimated data domain. First, we employ the MMTD method to estimate data range of SVs in both the majority and minority classes, respectively. Based on estimation of SVs in the majority class, we calculate MF values of SVs and find valuable majority class SVs with an α-cut. In addition, we generate new SVs close to original SVs in the minority class within the estimated data range of minority class SVs. Additionally, we depict data domain estimation of SVs using the MMTD method and proposed hybrid sampling approach in Figure 3.

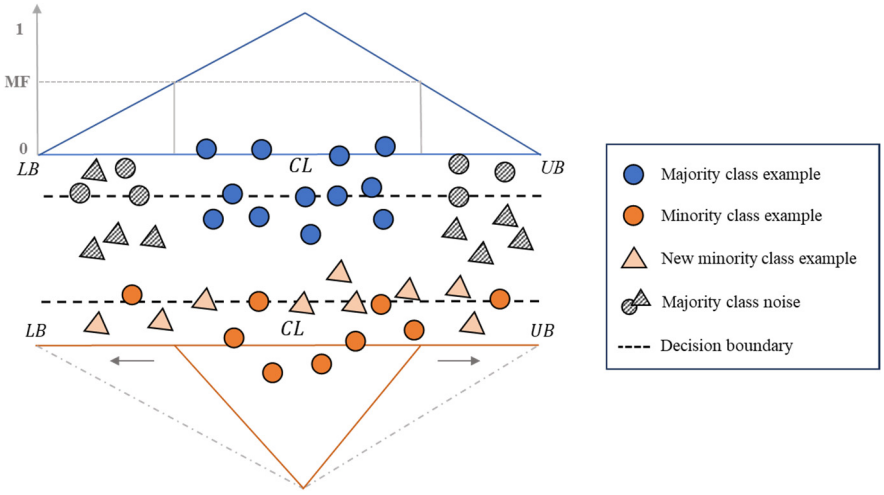


Figure 3. Proposed hybrid sampling approach.

3.3. Estimate data range using MMTD method

In this paper, we use MMTD method to estimate data range [LB,UB] of SVs on the decision boundary. They are calculated by Eqs (6) and (7), respectively.

$$\mathrm{LB}_j = \begin{cases} \mathrm{CL}(X_j^{SV}) - \mathrm{Skew}_{\mathrm{LB},j} \times \sqrt{-2 \times \dfrac{\hat{s}^2(X_j^{SV})}{N_{\mathrm{LB},j}} \times \ln(\varphi)}, & \text{if } \mathrm{LB}_j \le \min(X_j^{SV}) \\ \min(X_j^{SV}), & \text{if } \mathrm{LB}_j > \min(X_j^{SV}) \end{cases}, \tag{6}$$

$$\mathrm{UB}_j = \begin{cases} \mathrm{CL}(X_j^{SV}) + \mathrm{Skew}_{\mathrm{UB},j} \times \sqrt{-2 \times \dfrac{\hat{s}^2(X_j^{SV})}{N_{\mathrm{UB},j}} \times \ln(\varphi)}, & \text{if } \mathrm{UB}_j \ge \max(X_j^{SV}) \\ \max(X_j^{SV}), & \text{if } \mathrm{UB}_j < \max(X_j^{SV}) \end{cases}, \tag{7}$$

where LB and UB are the lower and upper bound, respectively. $N_{\mathrm{LB},j}$ ($N_{\mathrm{UB},j}$) is the number of values smaller (greater) than $\mathrm{CL}(X_j^{SV})$ for *j*th input variable. If the $N_{\mathrm{LB},j}$ ($N_{\mathrm{UB},j}$) is zero, the lower bound LB_j (the upper bound UB_j) cannot be defined, and we set them as $\min(X_j^{SV})$ and $\max(X_j^{SV})$,

respectively. The φ is set to 10^{-20}, $\hat{s}^2(\cdot)$ represents sample variance and $\mathrm{CL}(\cdot)$ is the median of samples. They are presented as follows:

$$\hat{s}(x)=\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i-\bar{x}}\,, \tag{8}$$

$$\mathrm{CL}=\begin{cases} x_{\left[\frac{n}{2}+1\right]} & ,\ if\ n \text{ is odd} \\ \frac{1}{2}(x_{\left[\frac{n}{2}\right]}+x_{\left[\frac{n}{2}+1\right]}), & if\ n \text{ is even} \end{cases}. \tag{9}$$

In addition, $\mathrm{Skew}_{\mathrm{LB},j}$ ($\mathrm{Skew}_{\mathrm{UB},j}$) is defined as coefficients of skewness on left (right) side of CL, which are calculated as follows:

$$\mathrm{Skew}_{\mathrm{LB},j}=\frac{N_{\mathrm{LB},j}}{N_{\mathrm{LB},j}+N_{\mathrm{UB},j}+\sigma}, \tag{10}$$

$$\mathrm{Skew}_{\mathrm{UB},j}=\frac{N_{\mathrm{UB},j}}{N_{\mathrm{LB},j}+N_{\mathrm{UB},j}+\sigma}, \tag{11}$$

in which, σ is a shape parameter for adjusting the degree of data skewness. In this paper we set σ to one. By the MMTD method, the data range of the SVs set can be estimated.

3.4. The under-sampling method using α-cut

To screen representative instances in the majority class, we employ the MMTD method to evaluate the data domain of majority class SVs. Based on a triangular MF, the MF value of the support vector(s) is calculated as follows:

$$\mathrm{MF}(x)=\begin{cases} 0 & ,\ x=\mathrm{LB}_j^{SV,M}\ \text{or}\ x=\mathrm{UB}_j^{SV,M} \\ \dfrac{x-LB_j^{SV,M}}{CL_j^{SV,M}-LB_j^{SV,M}} & ,\ \mathrm{LB}_j^{SV,M}\le x<\mathrm{CL}(X_j^{SV,M}) \\ \dfrac{UB_j^{SV,M}-x}{UB_j^{SV,M}-CL_j^{SV,M}} & ,\ \mathrm{CL}(X_j^{SV})<x\le\mathrm{UB}_j^{SV,M} \\ 1 & ,\ x=\mathrm{CL}(X_j^{SV,M}) \end{cases} ,j=1,2,\ldots,m, \tag{12}$$

where x is the support vector(s) of majority class and $\mathrm{MF}(x)\in[0,1]$. In this paper, we utilize α-cut $\in[0,1]$ for selecting valuable SVs according to MF value. The α-cut is a crisp set represented as follows:

$$A_\alpha=\left\{x\in x\middle|\mathrm{MF}(x)\ge\alpha\right\}, \tag{13}$$

in which, from Eq (12), α-cut can be derived as follows:

$$A_{\alpha} = [A_{\alpha,lower\ bound}, A_{\alpha,upper\ bound}], \tag{14}$$

where $A_{\alpha,lower\ bound} = LB_j^{SV,M} + \alpha \cdot (CL_j^{SV,M} - LB_j^{SV,M})$ and $A_{\alpha,upper\ bound} = UB_j^{SV,M} - \alpha \cdot (UB_j^{SV,M} - CL_j^{SV,M})$. By A_{α}, we can implement the under-sampling process to find representative SVs of the majority class. The derivation of Eq (14) is shown as follows:

$$\frac{x - LB_j^{SV,M}}{CL_j^{SV,M} - LB_j^{SV,M}} > \alpha$$
$$\alpha \cdot (CL_j^{SV,M} - LB_j^{SV,M}) > x - LB_j^{SV,M}$$
$$x > LB_j^{SV,M} + \alpha \cdot (CL_j^{SV,M} - LB_j^{SV,M})$$
$$\Rightarrow A_{\alpha,lower\ bound} = LB_j^{SV,M} + \alpha \cdot (CL_j^{SV,M} - LB_j^{SV,M})$$

and

$$\frac{UB_j^{SV,M} - x}{UB_j^{SV,M} - CL_j^{SV,M}} > \alpha$$
$$UB_j^{SV,M} - x > \alpha \cdot (UB_j^{SV,M} - CL_j^{SV,M})$$
$$x < UB_j^{SV,M} - \alpha \cdot (UB_j^{SV,M} - CL_j^{SV,M})$$
$$\Rightarrow A_{\alpha,upper\ bound} = UB_j^{SV,M} - \alpha \cdot (UB_j^{SV,M} - CL_j^{SV,M})$$

As a result, from Eq (14), valuable majority class SVs can be kept within data range $[A_{\alpha,\text{lower bound}}, A_{\alpha,\text{upper bound}}]$.

3.5. The over-sampling method using bagging ELM model

The data range [LB,UB] of SV in the minority class can be estimated by MMTD method in Section 3.3. We randomly generate virtual SVs inputs within estimated [LB,UB], as shown in Figure 4.

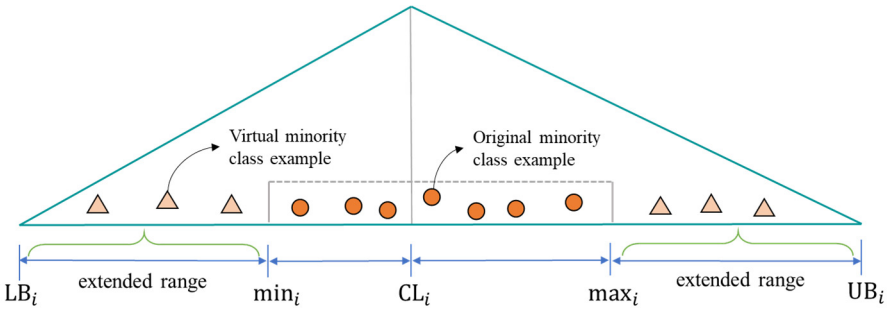


Figure 4. Estimated data domain.

As for prediction of virtual SV output, we deploy the extreme learning machines (ELM) proposed by Huang et al. [36] to monitor virtual SV output. The ELM is a feed-forward neural network, which consists of an input layer, hidden layer and output layer. The ELM model architecture is depicted in Figure 5.

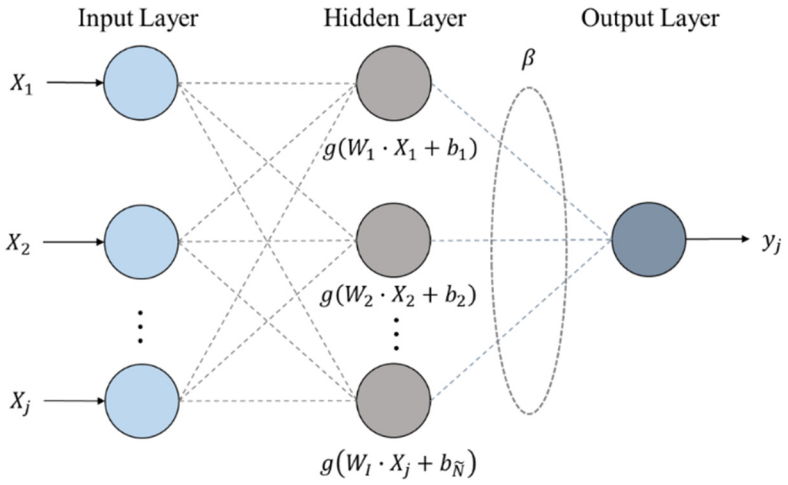


Figure 5. ELM architecture.

In Figure 5, the ELM's outcome can be expressed as follows:

$$\sum_{i=1}^{\tilde{N}} \beta_i \cdot f(W_i \cdot X_j + b_i) = y_j, j = 1,2,\ldots,m \tag{15}$$

where $\tilde{N}$ is the quantity of neurons in the hidden layer, β_i is weight between hidden layer and output layer, f is activation function, W_i is weight between input layer to hidden layer, b_i is a bias and y_j is model outcome. The detailed steps for training the ELM model are listed as follows:

Step 1. Randomly assign initial values of weight W_i and bias b_i in the hidden layer.

Step 2. Calculate hidden layer output matrix H, as follows:

$$H = \begin{bmatrix} f(W_1 \cdot X_1 + b_1) & f(W_2 \cdot X_1 + b_2) & \ldots & f(W_{\tilde{N}} \cdot X_1 + b_{\tilde{N}}) \\ f(W_1 \cdot X_2 + b_1) & f(W_2 \cdot X_1 + b_2) & \ldots & f(W_{\tilde{N}} \cdot X_2 + b_{\tilde{N}}) \\ \ldots & \ldots & \ldots & \ldots \\ f(W_1 \cdot X_n + b_1) & f(W_2 \cdot X_1 + b_2) & \ldots & f(W_{\tilde{N}} \cdot X_n + b_{\tilde{N}}) \end{bmatrix}_{n \times \tilde{N}}. \tag{16}$$

Step 3. Solve the following formula to find the weight β_i, as:

$$\beta = H^{-1}T, \tag{17}$$

where T is the target value in the output layer.

In this paper, we employ the sigmoid function as the activation function, as follows:

$$f(x) = \frac{1}{1+e^{-x}}, \quad 0 < f(x) < 1. \tag{18}$$

In addition, we use classification error rate to measure ELM model prediction accuracy. If model prediction is greater than 0.5, it is considered positive class. Conversely, if predicted value is less than or equal to 0.5, it is considered negative class. To optimize overall ELM model weights, we employ the bagging method [21] to resample original datasets to create multiple training datasets for retraining the ELM model. The bagging method is beneficial for training datasets with skewed class, since it creates several datasets by resampling from original datasets that allow the ELM model to learn different

patterns between inputs and output. In the fine-tuning process, we update these weights for 10 epochs each iteration at a learning rate of 1×10^{-3} until a total of 100 epochs, as illustrated in Figure 6.

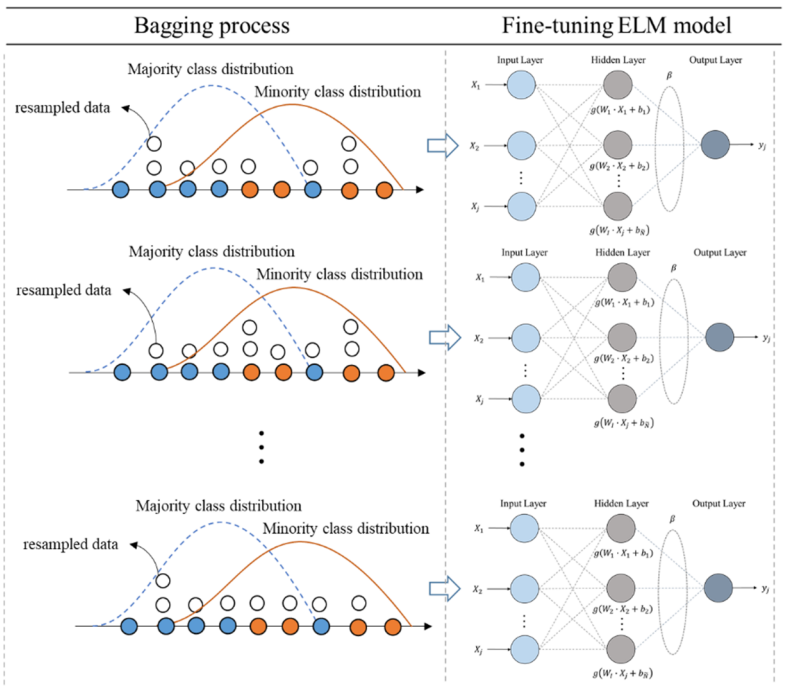


Figure 6. Bagging-based ELM model.

3.6. Proposed MMTD-ELM procedure

In this section, the hybrid sampling scheme for the proposed MMTD-ELM method is depicted as Figure 7. After completing the proposed implementation procedure for balancing the imbalanced training dataset, we constructed two SVM models using the balanced training dataset. Finally, we measure prediction accuracy of SVM model for testing dataset in terms of G-mean, F1, IBA and AUC metrics.

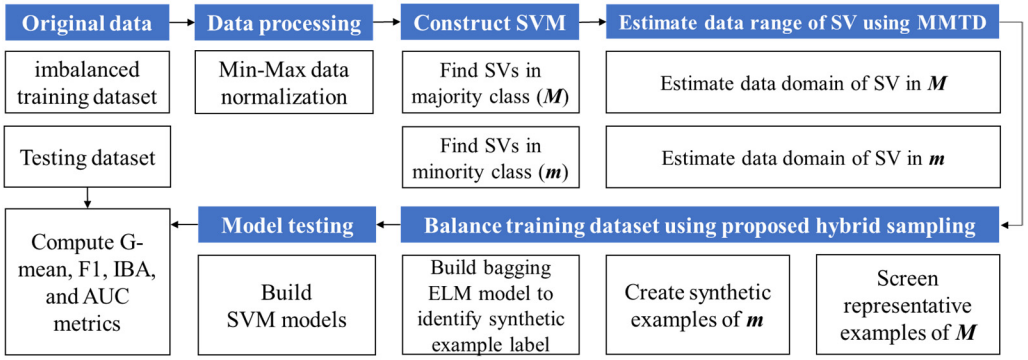


Figure 7. Proposed MMTD-ELM method implementation.

In the following, we summarize the implementation procedure explaining the MMTD-ELM

method in steps as described in Table 1.

Table 1. The MMTD-ELM algorithm.

Input:

An imbalanced training dataset.

Output:
Balanced training dataset.

Begin:

Definition:
m: minority class
M: majority class
#: the quantity of examples
#***M***/#***m***: imbalance ratio (IR)

Step 1. Split original dataset into imbalanced training dataset according to IR value and testing dataset.

Step 2. Normalize the imbalanced training dataset by min-max data normalization in Eq (5).

Step 3. Construct SVM model and find SVs in ***m*** class as SV_m and ***M*** class as SV_M, respectively.

Step 4. Estimate data domain $[LB_M, UB_M]$ for SV_M.

Step 5. Estimate data domain $[LB_m, UB_m]$ for SV_m.

Step 6. Calculate $A_\alpha = [A_{\alpha,lower\ bound}, A_{\alpha,upper\ bound}]$ for SV_M as in Eq (14) according to α-cut value.

Step 7. Remove unrepresentative SV_M outside data domain A_α.

Step 8. Train bagging ELM model for 10 epochs each time until epochs accumulate to 100 with an initial learning rate of 1×10^{-3} for support vector SV_m and SV_M.

Step 9. While the quantity of ***m*** class < the quantity of ***M*** class:
$SV^*_{m,input}$ ←Generate synthetic input variables in ***m*** class within data domain $[LB_m, UB_m]$.
$SV^*_{m,output}$ ←Feed $SV^*_{m,input}$ into trained bagging ELM model to predict its label.

If the prediction belongs to ***m*** class **then**:

Add $\{SV^*_{m,input}, SV^*_{m,output}\}$ into the imbalanced training dataset.

Else

continue.

Endif

return Balanced training dataset.

End

4. Experiment

In this section, we will describe four benchmark datasets used in our experiments as well as their experimental results. The experiment was executed with a computer equipped with Intel(R) Core(TM) i7-13700KF and 64 GB memory. Under the Ubuntu 22.04.1 LTS operating system, the experiment is implemented using Python 3.10.6 programming language for data processing and

constructing SVM predictive models. In this paper, we configure SVM models with the scikit-learn package (version 1.3.0) [37].

4.1. Dataset description

We used the four datasets to test prediction performance using the proposed MMTD-ELM method. The four datasets consist of new-thyroid1, Ecoli2, and Wisconsin (Diagnostic) obtained from the KEEL dataset repository, and one high-dimensional lung cancer microarray dataset downloaded from microarray gene expression cancer data [23]. We summarized the number of input features, the amount of data and other information for the four datasets in Table 2, in which, #instances represents the amount of data, #features represents the quantity of input features and #class indicates the number of categories. In addition, #M and #m indicate the quantity of majority and minority class examples, respectively. The imbalanced ratio (IR) is defined as #M/#m.

Table 2. Dataset description.

No.	Dataset	#instances	#features	#M	#m	#class	IR
1	new-thyroid1	215	5	180	35	2	5.14
2	Ecoli2	336	7	284	52	2	5.46
3	Wisconsin (Diagnostic)	569	30	357	212	2	1.68
4	Lung cancer	181	1626	150	31	2	4.84

4.2. Evaluation metrics

When a training dataset has imbalanced class distributions, the accuracy rate metric is not suitable to fully evaluate classification performance. As a result, we use the confusion matrix to evaluate classification performance of predictive models for imbalanced datasets. The confusion matrix consists of the predictive model's outcome and the actual output as presented in Table 3.

Table 3. Confusion matrix.

Actual \ Predicted	Positive class	Negative class
Positive class	True positive (TP)	False negative (FN)
Negative class	False positive (FP)	True negative (TN)

In this paper, we define positive class (minority class) as 1 and negative class (majority class) as 0. Considering classification accuracy for both negative class and positive class, four evaluation metrics, G-mean, F1, IBA and AUC, are used to measure classification performance for imbalanced datasets. The G-mean is defined as the geometric mean of Recall and Specificity as in Eq (19), in which, Recall (Specificity) represents the proportion of correctly predicted positive (negative) class examples to actual positive (negative) class examples. They are calculated as TP / (TP + FN) and TN / (TN + FP), respectively. F1 is the harmonic mean of Precision and Recall as calculated in Eq (20), where

Precision = TP / (TP + FP). In addition, IBA and AUC have comprehensive evaluations for overall classification results for imbalanced datasets as presented in Eqs (21) and (22). In Eq (22), $Rank_i$ represents the ranking of the ith instance in the TP set. In addition, |TP| and |TN| represent the amount of TP and TN, respectively.

$$\text{G-mean} = \sqrt{\text{Recall} \cdot \text{Specificity}}, \tag{19}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{20}$$

$$\text{IBA} = 1 + (\text{Recall} - \text{Specificity}) \cdot \text{Recall} \cdot \text{Specificity}, \tag{21}$$

$$\text{AUC} = \frac{\sum_{i=1}^{|\text{TP}|} (Rank_i - i)}{|\text{TP}| \cdot |\text{TN}|}. \tag{22}$$

4.3. Experiment design

In order to test effects of the proposed method for imbalanced datasets, we create imbalanced dataset scenarios for the four datasets. We randomly draw 100 data from an original dataset as training datasets according to IR values of 4 and 9, respectively. The remaining data is set as a testing dataset. Based on the four datasets, we compare prediction performance between the proposed MMTD-ELM method for hybrid sampling examples near the SVM's decision boundary, IMB method for only using imbalanced datasets and three sampling methods: SMOTE for generating new minority class examples by interpolating between original minority class examples, SVM-balance for generating minority class examples nearby the SVM's decision boundary and Cluster-SMOTE for generating new minority class examples and excluding unrepresentative majority class examples. In addition, we construct two types of SVM models with polynomial kernel (SVM_poly) and radial basis kernel (SVM_rbf) as predictive models to compare prediction performance across these five methods. The two SVM models are constructed with scikit-learn tool (version 1.3.0) [37]. The SVM_poly model is configured with {kernel: poly; cost penalty C:10; degree:2} and the SVM_rbf model is configured with {kernel: rbf; cost penalty C:10; gamma: "auto"}, where "auto" is defined as 1/the number of input features.

4.4. An example using the proposed MMTD-ELM method

In this section, to explain the proposed MMTD-ELM method in depth, based on the Ecoli2 dataset, we create a training dataset with IR = 9 as an example. The training dataset has 10 minority class data and 90 majority class data as listed in Table 4. The minority (m) class is labeled as "Positive" and the majority (M) class is labeled as "Negative". Seven variables: Mcg, Gvh, Lip, Chg, Aac, Alm1 and Alm2 are set as input features. The implementing steps for the MMTD-ELM method are explained, as follows:

Table 4. The training dataset.

	input features							output
No.	Mcg	Gvh	Lip	Chg	Aac	Alm1	Alm2	Class
1	0.69	0.80	0.48	0.50	0.46	0.57	0.26	Positive
2	0.63	0.86	0.48	0.50	0.39	0.47	0.34	Positive
3	0.64	0.81	0.48	0.50	0.37	0.39	0.44	Positive
4	0.62	0.83	0.48	0.50	0.46	0.36	0.40	Positive
5	0.76	0.73	0.48	0.50	0.44	0.39	0.39	Positive
6	0.69	0.65	0.48	0.50	0.63	0.48	0.41	Positive
7	0.69	0.66	0.48	0.50	0.41	0.50	0.25	Positive
8	0.63	1.00	0.48	0.50	0.35	0.51	0.49	Positive
9	0.62	0.78	0.48	0.50	0.47	0.49	0.54	Positive
10	0.74	0.82	0.48	0.50	0.49	0.49	0.41	Positive
11	0.43	0.32	0.48	0.50	0.33	0.45	0.52	Negative
12	0.52	0.81	0.48	0.50	0.72	0.38	0.38	Negative
…	…	…	…	…	…	…	…	…
100	0.44	0.49	0.48	0.50	0.39	0.38	0.40	Negative

Step 1. Convert training data into domain [0,1] by min-max data normalization given by Eq (5).
Step 2. Find support vectors using the SVM_poly model. We listed the support vectors for negative class and positive class in Tables 5 and 6, respectively.

Table 5. Support vectors of majority class.

	input features							output
No.	Mcg	Gvh	Lip	Chg	Aac	Alm1	Alm2	Class
1	0.57	0.74	0.00	0.00	0.90	0.29	0.41	Negative
2	0.86	0.61	0.00	0.00	0.60	0.72	0.82	Negative
3	0.76	0.50	0.00	0.00	0.92	0.32	0.33	Negative
4	0.60	0.28	0.00	0.00	0.30	0.22	0.46	Negative
5	0.43	0.26	0.00	0.00	0.44	0.08	0.00	Negative
6	0.85	0.39	1.00	1.00	0.47	0.41	0.33	Negative
7	0.63	0.38	0.00	0.00	0.43	0.16	0.36	Negative
8	0.96	0.37	0.00	0.00	0.64	0.47	0.38	Negative
9	0.57	0.74	0.00	0.00	0.90	0.29	0.41	Negative

Table 6. Support vectors of minority class.

	input features							output
No.	Mcg	Gvh	Lip	Chg	Aac	Alm1	Alm2	Class
1	0.77	0.73	0.00	0.00	0.55	0.54	0.28	Positive
2	0.69	0.77	0.00	0.00	0.55	0.26	0.44	Positive
3	0.86	0.64	0.00	0.00	0.52	0.30	0.42	Positive
4	0.77	0.53	0.00	0.00	0.78	0.42	0.45	Positive
5	0.77	0.54	0.00	0.00	0.48	0.45	0.27	Positive
6	0.69	0.70	0.00	0.00	0.56	0.43	0.59	Positive

Step 3. Estimate data range of each input feature of majority class support vectors using MMTD method, as listed in Table 7.

Table 7. Estimates of the data range of majority class support vectors.

	input features						
Estimates	Mcg	Gvh	Lip	Chg	Aac	Alm1	Alm2
LB_M	0.34	0.05	−0.93	−0.93	0.08	−0.10	−0.07
CL_M	0.70	0.39	0.00	0.00	0.53	0.30	0.37
UB_M	1.05	0.72	1.00	1.00	0.99	0.71	0.81

Step 4. Calculate $A_\alpha = [A_{\alpha=0.25,lower\ bound}, A_{\alpha=0.25,upper\ bound}]$ as seen in Eq (14), where α is set at 0.25. The $A_{\alpha=0.25,lower\ bound}$ is calculated as $LB_M + 0.25 \cdot (CL_M - LB_M)$ and $A_{\alpha=0.25,upper\ bound}$ is calculated as $UB_M - 0.25 \cdot (UB_M - CL_M)$. The calculations are shown in Table 8.

Table 8. Calculation of A_α at α = 0.25.

	input features						
Estimates	Mcg	Gvh	Lip	Chg	Aac	Alm1	Alm2
$A_{\alpha=0.25,lower\ bound}$	0.427	0.136	−0.700	−0.700	0.196	0.000	0.037
$A_{\alpha=0.25,upper\ bound}$	0.965	0.634	0.750	0.750	0.827	0.606	0.703

Step 5. Remove majority class examples outside the range A_α. The deleted examples are listed in Table 9.

Table 9. Deleted majority class examples.

	input features							output
No.	Mcg	Gvh	Lip	Chg	Aac	Alm1	Alm2	Class
1	0.57	0.74	0.00	0.00	0.90	0.29	0.41	Negative
2	0.86	0.61	0.00	0.00	0.60	0.72	0.82	Negative
3	0.76	0.50	0.00	0.00	0.92	0.32	0.33	Negative
4	0.43	0.26	0.00	0.00	0.44	0.08	0.00	Negative
5	0.85	0.39	1.00	1.00	0.47	0.41	0.33	Negative

Step 6. Estimate data range of each input feature of minority class support vectors using MMTD method, as listed in Table 10.

Table 10. Estimates of data range of minority class support vectors.

	input features						
Estimates	Mcg	Gvh	Lip	Chg	Aac	Alm1	Alm2
LB_m	0.60	0.45	0.00	0.00	0.28	0.21	0.18
CL_m	0.77	0.67	0.00	0.00	0.55	0.43	0.43
UB_m	0.86	0.89	0.00	0.00	0.74	0.65	0.68

Step 7. Create synthetic minority class example within the estimated range $[LB_m, UB_m]$ and input them into the trained bagging ELM model to determine if it belongs to the minority class.

Step 8. Repeat Step 7 until 75 (= 90 – 5 − 10) synthetic examples of the minority class are created. These generated examples are listed in Table 11.

Table 11. Synthetic minority class dataset.

	input features							output
No.	Mcg	Gvh	Lip	Chg	Aac	Alm1	Alm2	Class
1	0.55	0.85	0.48	0.50	0.47	0.38	0.33	Positive
2	0.58	0.83	0.48	0.50	0.34	0.36	0.29	Positive
…	…	…	…	…	…	…	…	…
75	0.56	0.80	0.48	0.50	0.42	0.40	0.47	Positive

Step 9. Add synthetic minority class dataset into original training dataset to build up a balanced training dataset. We depict them in Figure 8.

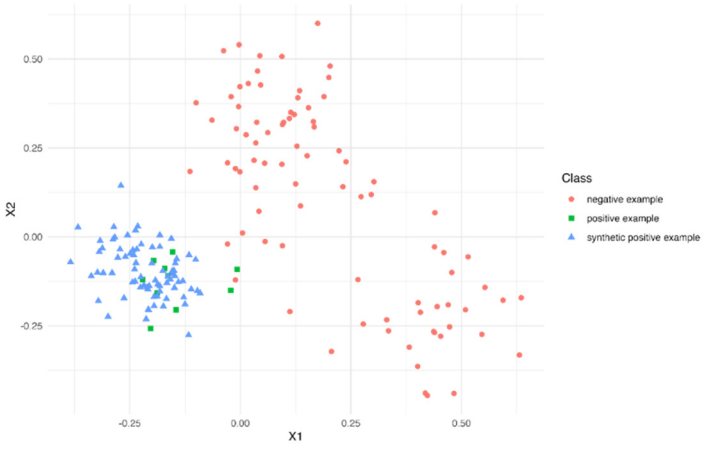


Figure 8. Balanced training dataset.

4.5. Statistical tests with experimental results

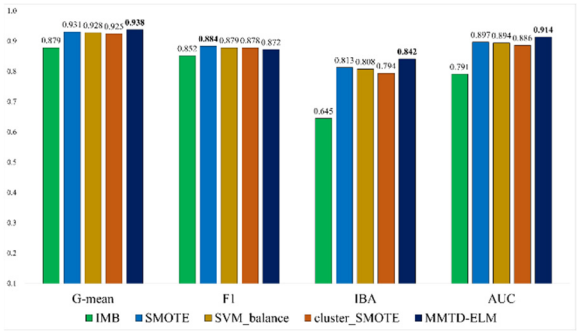
In this section, we use the paired *t*-test to assess whether there are significant differences between the proposed MMTD-ELM method and the *i*th method in IMB, SMOTE, SVM-balance and Cluster-SMOTE. In the paired t-test procedure, we set the null hypothesis H_0 and the alternative hypothesis H_1 as:

$$\begin{cases} H_0 : \mu_{MMTD-ELM} - \mu_{ith} = 0 \\ H_1 : \mu_{MMTD-ELM} - \mu_{ith} \neq 0 \end{cases}, \tag{23}$$

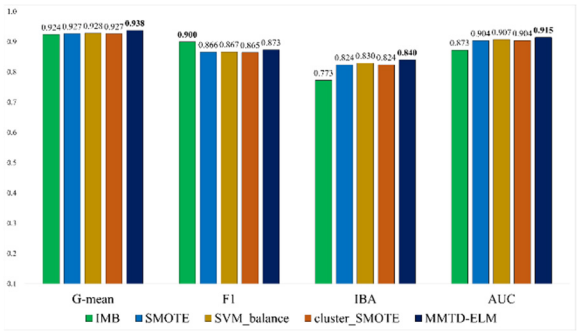
where $\mu_{MMTD-ELM} - \mu_{ith}$ indicates the average of differences of classification results between the MMTD-ELM method and *i*th method for G-mean, F1, IBA or AUC metric with 50 experiments. In addition, we set the significance level α at 0.05. When p-value is less than the significance level α, the hypothesis H_0 is rejected, indicating there is a significant difference for G-mean, F1, IBA or AUC metric. We used the symbol "*" to indicate that the classification capability of the proposed MMTD-ELM method has statistically significant effects over the other methods.

4.6. Experimental results

In this section, we implemented a total of 50 experiments to compare classification results among the five methods on the four datasets. In Figure 9(a) and (b), for example, when IR value was set at 4, classification results using the proposed MMTD-ELM method (deep blue line) are better than those of IMB (green line), SMOTE (blue line), SVM-balance (earthy yellow line) and Cluster-SMOTE (orange line) methods on SVM_poly and SVM_rbf models, respectively. When IR value is increased from 4 to 9, the MMTD-ELM method still outperforms the other four methods in terms of the four evaluation metrics, as displayed in Figure 10.

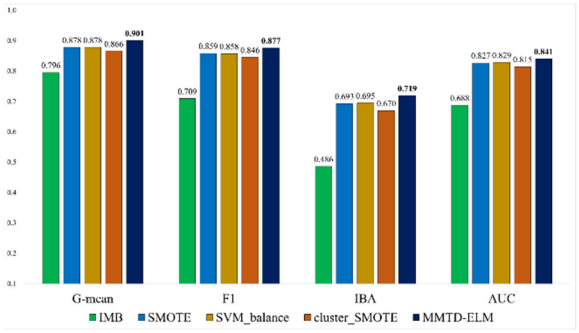


(a) for the SVM_poly model

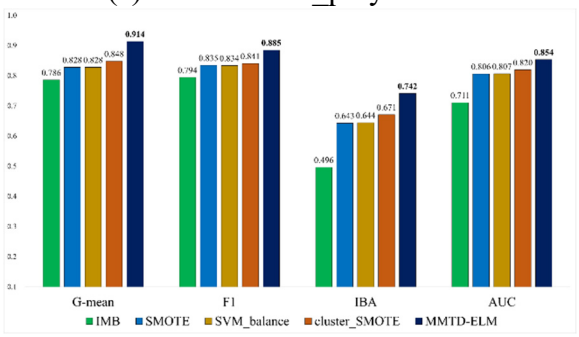


(b) for the SVM_rbf model

Figure 9. Compared methods' classification results at IR = 4.



(a) for the SVM_poly model



(b) for the SVM_rbf model

Figure 10. Compared methods' classification results at IR = 9.

Additionally, in Tables 12 and 13, we list other experimental results for the four datasets with IR values of 4 and 9. In these tables, the values in **bold** represent the best classification results among five methods in items of G-mean, F1, IBA and AUC metrics. In Table 12, for example, the proposed MMTD-ELM method in terms of AUC metric can achieve the best average scores of 0.958 and 0.973 on the SVM_poly and SVM_rbf models for new-thyroid1 dataset, respectively. Note that, on the SVM_poly and SVM_rbf models, the prediction accuracy using the suggested MMTD-ELM method can achieve improvement of $0.919 - 0.543 = 0.376$ and $0.947 - 0.852 = 0.095$ in terms of IBA metric, respectively.

Table 12. Average of results for IR = 4.

Dataset	new-thyroid1							
Classifier	SVM_poly				SVM_rbf			
Method	G-mean	F1	IBA	AUC	G-mean	F1	IBA	AUC
IMB	0.857	0.846	0.543	0.735	0.957	**0.946**	0.852	0.920
SMOTE	**0.965**	**0.939**	0.887	0.941	0.965	0.879	0.942	0.971
SVM-balance	0.961	0.927	0.88	0.937	0.964	0.875	0.943	0.972
Cluster-SMOTE	0.953	0.929	0.843	0.917	0.968	0.897	0.941	0.970
MMTD-ELM	0.964	0.897	**0.919**	**0.958**	**0.974**	0.920	**0.947**	**0.973**
Dataset	Ecoli2							
Classifier	SVM_poly				SVM_rbf			
Method	G-mean	F1	IBA	AUC	G-mean	F1	IBA	AUC
IMB	0.798	0.696	0.435	0.651	0.871	**0.779**	0.624	0.785
SMOTE	**0.888**	**0.722**	0.745	0.860	0.885	0.714	0.748	0.863
SVM-balance	0.886	0.717	0.744	0.860	**0.890**	0.718	0.763	0.872
Cluster-SMOTE	0.877	0.711	0.718	0.844	0.871	0.684	0.733	0.856
MMTD-ELM	0.885	0.695	**0.764**	**0.873**	0.881	0.687	**0.768**	**0.877**
Dataset	Wisconsin (Diagnostic)							
Classifier	SVM_poly				SVM_rbf			
Method	G-mean	F1	IBA	AUC	G-mean	F1	IBA	AUC
IMB	0.882	0.874	0.634	0.793	0.894	0.887	0.653	0.803
SMOTE	0.891	0.883	0.648	0.800	0.895	0.888	0.656	0.805
SVM-balance	0.888	0.880	0.638	0.794	0.897	0.890	0.664	0.810
Cluster-SMOTE	0.890	0.882	0.645	0.798	0.896	0.890	0.661	0.808
MMTD-ELM	**0.905**	**0.898**	**0.690**	**0.827**	**0.919**	**0.911**	**0.738**	**0.856**
Dataset	Lung cancer							
Classifier	SVM_poly				SVM_rbf			
Method	G-mean	F1	IBA	AUC	G-mean	F1	IBA	AUC
IMB	0.977	0.99	0.968	0.985	0.973	**0.988**	**0.962**	**0.982**
SMOTE	0.979	0.991	0.972	0.986	0.962	0.983	0.948	0.975
SVM-balance	0.978	0.99	0.970	0.986	0.962	0.983	0.948	0.975
Cluster-SMOTE	0.978	0.990	0.970	0.986	0.971	0.987	0.961	0.981
MMTD-ELM	**0.998**	**0.998**	**0.993**	**0.996**	**0.976**	0.975	0.907	0.952

Table 13. Average of results for IR = 9.

Dataset	new-thyroid1							
Classifier	SVM_poly				SVM_rbf			
Method	G-mean	F1	IBA	AUC	G-mean	F1	IBA	AUC
IMB	0.803	0.784	0.417	0.646	0.871	0.862	0.577	0.758
SMOTE	0.923	0.917	0.736	0.855	0.957	0.943	**0.863**	**0.927**
SVM-balance	**0.923**	0.917	**0.738**	**0.856**	0.955	0.940	0.860	0.925
Cluster-SMOTE	0.908	0.900	0.692	0.829	**0.957**	**0.945**	0.858	0.924
MMTD-ELM	0.922	**0.916**	0.732	0.853	0.933	0.929	0.765	0.872
Dataset	Ecoli2							
Classifier	SVM_poly				SVM_rbf			
Method	G-mean	F1	IBA	AUC	G-mean	F1	IBA	AUC
IMB	0.725	0.334	0.283	0.529	0.788	0.69	0.405	0.629
SMOTE	**0.887**	0.775	0.726	0.849	**0.884**	0.772	0.716	0.843
SVM-balance	0.886	0.770	**0.734**	**0.854**	0.883	0.770	**0.718**	**0.844**
Cluster-SMOTE	0.870	0.748	0.704	0.836	0.875	0.748	0.710	0.840
MMTD-ELM	0.883	**0.786**	0.688	0.825	0.881	**0.782**	0.686	0.824
Dataset	Wisconsin (Diagnostic)							
Classifier	SVM_poly				SVM_rbf			
Method	G-mean	F1	IBA	AUC	G-mean	F1	IBA	AUC
IMB	0.829	0.813	0.484	0.690	0.817	0.799	0.457	0.670
SMOTE	0.827	0.811	0.480	0.687	0.824	0.807	0.474	0.682
SVM-balance	0.827	0.811	0.478	0.686	0.824	0.807	0.475	0.682
Cluster-SMOTE	0.827	0.811	0.478	0.686	0.825	0.809	0.478	0.685
MMTD-ELM	**0.844**	**0.830**	**0.518**	**0.715**	**0.853**	**0.839**	**0.546**	**0.734**
Dataset	Lung cancer							
Classifier	SVM_poly				SVM_rbf			
Method	G-mean	F1	IBA	AUC	G-mean	F1	IBA	AUC
IMB	0.825	0.905	0.760	0.885	0.668	0.826	0.546	0.786
SMOTE	0.875	0.932	0.83	0.918	0.648	0.816	0.519	0.773
SVM-balance	0.876	0.932	0.831	0.919	0.651	0.818	0.524	0.775
Cluster-SMOTE	0.858	0.923	0.806	0.907	0.736	0.860	0.638	0.829
MMTD-ELM	**0.954**	**0.974**	**0.938**	**0.969**	**0.989**	**0.991**	**0.969**	**0.984**

4.7. Analysis of the experimental results

In this section, based on the four datasets, we calculate the average (Avg) and standard deviation (SD) of classification accuracy in terms of G-mean, F1, IBA and AUC metrics as seen in Tables 14 and 15. In Table 14, for example, the values of "0.938" and "0.023" indicate Avg and SD of prediction results using the proposed MMTD-ELM method for G-mean metric on the SVM_poly model, respectively. Additionally, we rank the five methods, to select the best methods in terms of G-mean, F1, IBA and AUC metrics. In Tables 14 and 15, we can see that the proposed method has the best ranking averages on the four evaluation metrics. In Table 15, for example, on the SVM_poly and SVM_rbf models, the proposed MMTD-ELM method achieves better ranking value among the five

methods in terms of G-mean (1.830 and 2.005), F1 (1.755 and 1.885), IBA (1.930 and 2.125) and AUC (1.935 and 2.115), respectively.

Table 14. Compared results between MMTD-ELM and other methods at IR = 4.

Metric	G-mean							
Classifier	SVM_poly				SVM_rbf			
Method	Avg	SD	Rank	P-value	Avg	SD	Rank	P-value
IMB	0.879	0.041	3.505	0.000*	0.924	0.762	3.045	0.000*
SMOTE	0.931	0.031	**1.955**	0.000*	0.927	0.853	2.535	0.000*
SVM-balance	0.928	0.032	2.195	0.000*	0.928	0.861	**2.390**	0.000*
Cluster-SMOTE	0.925	0.032	2.355	0.000*	0.927	0.853	2.475	0.000*
MMTD-ELM	**0.938**	**0.023**	2.235	−	**0.938**	**0.865**	2.430	−
Metric	F1							
Classifier	SVM_poly				SVM_rbf			
Method	Avg	SD	Rank	P-value	Avg	SD	Rank	P-value
IMB	0.852	0.063	2.915	0.003*	0.900	0.983	**2.180**	0.000*
SMOTE	**0.884**	0.038	**2.015**	0.001*	0.866	0.954	2.670	0.036*
SVM-balance	0.879	0.043	2.275	0.054	0.867	0.954	2.590	0.070
Cluster-SMOTE	0.878	**0.041**	2.260	0.112	0.865	0.954	2.610	0.015*
MMTD-ELM	0.872	0.045	2.775	−	**0.873**	**0.963**	2.815	−
Metric	IBA							
Classifier	SVM_poly				SVM_rbf			
Method	Avg	SD	Rank	P-value	Avg	SD	Rank	P-value
IMB	0.645	0.093	3.550	0.000*	0.773	0.039	3.360	0.000*
SMOTE	0.813	0.093	**2.000**	0.000*	0.824	0.040	2.480	0.003*
SVM-balance	0.808	0.094	2.180	0.000*	0.830	0.039	**2.280**	0.061
Cluster-SMOTE	0.794	0.095	2.475	0.000*	0.824	0.039	2.385	0.005*
MMTD-ELM	**0.842**	**0.069**	2.090	−	**0.840**	**0.026**	2.380	−
Metric	AUC							
Classifier	SVM_poly				SVM_rbf			
Method	Avg	SD	Rank	P-value	Avg	SD	Rank	P-value
IMB	0.791	0.061	3.555	0.000*	0.873	0.039	3.370	0.000*
SMOTE	0.897	0.054	**2.005**	0.000*	0.904	0.061	2.480	0.001*
SVM-balance	0.894	0.055	2.185	0.000*	0.907	0.060	**2.280**	0.024*
Cluster-SMOTE	0.886	0.056	2.465	0.000*	0.904	0.063	2.385	0.001*
MMTD-ELM	**0.914**	**0.040**	2.085	−	**0.915**	**0.049**	2.375	−

In order to further analyze classification results using these methods, we used paired t-test to demonstrate if these experimental results exhibit statistically significant differences between the proposed MMTD-ELM method and the other methods on G-mean, F1, IBA and AUC metrics. In Tables 14 and 15, the symbol "*" indicates that the MMTD-ELM method enjoys statistically significant differences (p-value < 0.05) from IMB, SMOTE, SVM-balance and cluster-SMOTE methods. In Table 14, for example, on the SMV_rbf model, the classification results using the proposed MMTD-ELM method have significant improvements (p-value = 0.003* < 0.05) as compared to the

IMB method for terms of F1 metric.

Table 15. Compared results between MMTD-ELM and other methods at IR = 9.

Metric	G-mean							
Classifier	SVM_poly				SVM_rbf			
Method	Avg	SD	Rank	P-value	Avg	SD	Rank	P-value
IMB	0.796	0.047	4.155	0.000*	0.786	0.531	4.220	0.000*
SMOTE	0.878	0.049	2.075	0.000*	0.828	0.747	2.475	0.000*
SVM-balance	0.878	0.051	2.120	0.000*	0.828	**0.748**	2.450	0.000*
Cluster-SMOTE	0.866	0.058	2.695	0.000*	0.848	0.748	2.250	0.000*
MMTD-ELM	**0.901**	**0.039**	**1.830**	−	**0.914**	0.724	**2.005**	−
Metric	F1							
Classifier	SVM_poly				SVM_rbf			
Method	Avg	SD	Rank	P-value	Avg	SD	Rank	P-value
IMB	0.709	0.119	4.080	0.000*	0.794	0.891	3.865	0.000*
SMOTE	0.859	0.047	2.120	0.000*	0.835	0.866	2.585	0.000*
SVM-balance	0.858	0.050	2.145	0.000*	0.834	0.866	2.610	0.000*
Cluster-SMOTE	0.846	0.056	2.770	0.000*	0.841	0.891	2.455	0.000*
MMTD-ELM	**0.877**	**0.041**	**1.755**	−	**0.885**	**0.983**	**1.885**	−
Metric	IBA							
Classifier	SVM_poly				SVM_rbf			
Method	Avg	SD	Rank	P-value	Avg	SD	Rank	P-value
IMB	0.486	**0.085**	4.160	0.000*	0.496	0.113	4.270	0.000*
SMOTE	0.693	0.111	2.050	0.000*	0.643	0.120	2.450	0.000*
SVM-balance	0.695	0.112	2.080	0.000*	0.644	0.124	2.475	0.000*
Cluster-SMOTE	0.670	0.116	2.675	0.000*	0.671	0.125	**2.100**	0.000*
MMTD-ELM	**0.719**	0.100	**1.930**	−	**0.742**	**0.089**	2.125	−
Metric	AUC							
Classifier	SVM_poly				SVM_rbf			
Method	Avg	SD	Rank	P-value	Avg	SD	Rank	P-value
IMB	0.688	**0.056**	4.160	0.000*	0.711	0.072	4.270	0.000*
SMOTE	0.827	0.067	2.050	0.000*	0.806	0.070	2.455	0.000*
SVM-balance	0.829	0.067	2.080	0.002*	0.807	0.072	2.480	0.000*
Cluster-SMOTE	0.815	0.069	2.670	0.000*	0.820	0.073	**2.100**	0.000*
MMTD-ELM	**0.841**	0.061	**1.935**	−	**0.854**	**0.055**	2.115	−

4.8. Summary

According to the experimental results using all five methods: IMB, SMOTE, SVM-balance, Cluster-SMOTE, and our proposed MMTD-ELM methods, listed in Tables 12–16, the findings can be summarized as follows:

a) Based on the four datasets, when IR values are set at 4 and 9, our suggested MMTD-ELM method can achieve the best classification accuracy among these methods on two types of SVM models in terms of G-mean, F1, IBA and AUC metrics, as seen in Tables 12 and 13. From these results, we can see that with increasing IR values, the proposed MMTD-ELM method consistently achieves the best classification performance in terms of G-mean, F1, IBA and AUC metrics.

Table 16. Average of results for Recall and Specificity metrics.

Dataset	new-thyroid1							
Classifier	SVM_poly				SVM_rbf			
IR	4		9		4		9	
Method	Recall	Specificity	Recall	Specificity	Recall	Specificity	Recall	Specificity
IMB	0.469	**1.000**	0.291	**1.000**	0.844	**0.997**	0.517	**1.000**
SMOTE	0.891	0.991	0.712	0.998	0.983	0.959	**0.865**	0.989
SVM-balance	0.887	0.986	**0.714**	0.998	**0.987**	0.957	0.863	0.988
Cluster-SMOTE	0.841	0.992	0.660	0.998	0.973	0.967	0.857	0.991
MMTD-ELM	**0.945**	0.971	0.707	0.998	0.969	0.976	0.745	0.999
Dataset	Ecoli2							
Classifier	SVM_poly				SVM_rbf			
IR	4		9		4		9	
Method	Recall	Specificity	Recall	Specificity	Recall	Specificity	Recall	Specificity
IMB	0.319	**0.984**	0.060	**0.998**	0.599	**0.970**	0.265	**0.993**
SMOTE	0.802	0.918	0.769	0.929	0.817	0.909	0.758	0.929
SVM-balance	0.804	0.916	**0.785**	0.922	0.834	0.910	0.763	0.926
Cluster-SMOTE	0.773	0.916	0.764	0.909	0.822	0.890	**0.765**	0.914
MMTD-ELM	**0.848**	0.898	0.704	0.947	**0.866**	0.888	0.704	0.944
Dataset	Wisconsin (Diagnostic)							
Classifier	SVM_poly				SVM_rbf			
IR	4		9		4		9	
Method	Recall	Specificity	Recall	Specificity	Recall	Specificity	Recall	Specificity
IMB	0.601	0.985	0.380	**1.000**	0.607	**0.999**	0.340	**1.000**
SMOTE	0.605	0.995	0.375	**1.000**	0.612	0.998	0.364	**1.000**
SVM-balance	0.593	**0.996**	0.373	**1.000**	0.621	0.998	0.364	**1.000**
Cluster-SMOTE	0.602	0.995	0.372	**1.000**	0.617	0.998	0.369	**1.000**
MMTD-ELM	**0.661**	0.993	**0.430**	**1.000**	**0.722**	0.989	**0.472**	0.996
Dataset	Lung cancer							
Classifier	SVM_poly				SVM_rbf			
IR	4		9		4		9	
Method	Recall	Specificity	Recall	Specificity	Recall	Specificity	Recall	Specificity
IMB	**1.000**	0.969	**1.000**	0.771	0.999	0.964	**1.000**	0.572
SMOTE	0.999	0.973	**1.000**	0.836	**1.000**	0.949	**1.000**	0.546
SVM-balance	**1.000**	0.971	**1.000**	0.837	**1.000**	0.949	**1.000**	0.551
Cluster-SMOTE	**1.000**	0.971	**1.000**	0.813	**1.000**	0.962	**1.000**	0.657
MMTD-ELM	0.993	**1.000**	**1.000**	**0.939**	0.904	**1.000**	0.974	**0.994**

b) From these experimental results listed in Tables 14 and 15, we can see that most Avg and SDs using the MMTD-ELM method obtain the best performance in terms of G-mean, F1, IBA and AUC metrics. Additionally, the proposed MMTD-ELM method has the best ranking score in terms of G-mean, F1, IBA and AUC metrics. Furthermore, most p-values are less than 0.05 at IR values of 4 and 9.
c) Although a few experimental results indicate the MMTD-ELM method does not have statistically significant prediction accuracy compared to the IMB method, the proposed MMTD-ELM method still outperforms the other methods in terms of G-mean, F1, IBA and AUC metrics.
d) In Table 16, in terms of the Recall (i.e., true positive rate) metric, we can see that the MMTD-ELM method outperforms the other methods for four experimental datasets indicating that our proposed method has better prediction accuracy for minority class (which is defined as positive class). Additionally, in terms of the Specificity (i.e., true negative rate) metric, there are only slight differences among the five methods indicating that the five methods have similar prediction performance to each other for majority class (which is defined as negative class).

In sum, the suggested MMTD-ELM method has more improvement effects and is shown to be superior to the other methods for four imbalanced datasets.

5. Conclusions

The sampling approach has been proposed as an effective technique to improve prediction accuracy in traditional machine learning and deep learning models for imbalanced datasets. This technique directly creates new examples of minority class to balance skewed data distribution. For SVM imbalanced classification, some researchers suggested generating synthetic minority class examples adjusting the SVM's decision boundary to correctly predict minority class examples as seen in [15,17,38]. Farquad and Bose [17], for example, proposed the SVM-balance method, which randomly over-samples misclassified examples near the decision boundary as new examples, to improve prediction accuracy of SVM for minority class examples. However, generated examples may be surrounded by most of the majority class examples that are thus regarded as danger examples of minority class or noise. These may lead to distortion of SVM learning. To effectively adjust SVM's decision boundary, Cieslak et al. [15] proposed the distance-based cluster-SMOTE hybrid sampling method, which creates new minority class examples and eliminates unrepresentative majority class examples. However, the cluster-SMOTE method based on distance between examples is easily impacted by noise or outliers. Differing from their papers, based on a fuzzy triangular MF, we developed a new hybrid sampling method named MMTD-ELM to screen representative majority class examples and generate synthetic minority class examples. In order to screen informative support vectors of the majority class, we developed an α-cut technique to measure representation of the majority class example. Furthermore, to create better synthetic minority class examples, we deploy a bagging ELM model to monitor the similarity between synthetic examples and original data of the minority class. As a result, when compared to the oversampling SVM-balance and distance-based hybrid sampling cluster-SMOTE methods, the proposed MMTD-ELM method achieves better prediction accuracy of SVM for skewed datasets.

In this paper, four biomedical datasets were used to elucidate effectiveness of the suggested MMTD-ELM method for SVM classification with imbalanced datasets. These experimental results demonstrate the suggested MMTD-ELM method successfully outperforms other sampling methods in

imbalanced datasets. As for research limitations, the proposed MMTD-ELM approach can be utilized to estimate the data range of numerical datasets, but it is not appropriate for datasets with discrete variables. In the future, we will further consider three directions: 1. using the proposed method for addressing other high-dimensional imbalanced microarray cancer data; 2. developing a sampling method for handling imbalanced datasets with discrete features; 3. developing a sampling method or deep learning model for imbalanced but small-sample-size datasets.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Data availability

Data available in a publicly accessible repository. The experimental datasets presented in this paper are openly available at the KEEL dataset repository [22] and the Microarray Gene Expression Cancer Data [23].

Conflict of interest

The authors declare that there are no conflicts of interest.

Acknowledgments

The National Taipei University of Nursing and Health Sciences and the National Science and Technology Council both provided funding for this research. The National Science and Technology Council, Taiwan financed this study pursuant to contract number NSTC 112-2221-E-227-003.

References

1. Q. Wang, W. Cao, J. Guo, J. Ren, Y. Cheng, D. N. Davis, DMP_MI: An effective diabetes mellitus classification algorithm on imbalanced data with missing values, *IEEE Access*, **7** (2019), 102232–102238. https://doi.org/10.1109/ACCESS.2019.2929866
2. L. Yousefi, S. Swift, M. Arzoky, L. Saachi, L. Chiovato, A. Tucker, Opening the black box: Personalizing type 2 diabetes patients based on their latent phenotype and temporal associated complication rules, *Comput. Intell.*, **37** (2021), 1460–1498. https://doi.org/10.1111/coin.12313
3. B. Krawczyk, M. Galar, Ł. Jeleń, F. Herrera, Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy, *Appl. Soft. Comput.*, **38** (2016), 714–726. https://doi.org/10.1016/j.asoc.2015.08.060
4. A. K. Mishra, P. Roy, S. Bandyopadhyay, S. K. Das, Breast ultrasound tumour classification: A machine learning-radiomics based approach, *Expert Syst.*, **38** (2021), 12713. https://doi.org/10.1111/exsy.12713

5. J. Zhou, X. Li, Y. Ma, Z. Wu, Z. Xie, Y. Zhang, et al., Optimal modeling of anti-breast cancer candidate drugs screening based on multi-model ensemble learning with imbalanced data, *Math. Biosci. Eng.*, **20** (2023), 5117–5134. https://doi.org/10.3934/mbe.2023237
6. L. Zhang, H. Yang, Z. Jiang, Imbalanced biomedical data classification using self-adaptive multilayer ELM combined with dynamic GAN, *Biomed. Eng. Online*, **17** (2018), 1–21. https://doi.org/10.1186/s12938-018-0604-3
7. H. S. Basavegowda, G. Dagnew, Deep learning approach for microarray cancer data classification, *CAAI Trans. Intell. Technol.*, **5** (2020), 22–33. https://doi.org/10.1049/trit.2019.0028
8. B. Pes, Learning from high-dimensional biomedical datasets: the issue of class Imbalance, *IEEE Access*, **8** (2020), 13527–13540. https://doi.org/10.1109/ACCESS.2020.2966296
9. J. Wang, Prediction of postoperative recovery in patients with acoustic neuroma using machine learning and SMOTE-ENN techniques, *Math. Biosci. Eng.*, **19** (2022), 10407–10423. https://doi.org/10.3934/mbe.2022487
10. V. Babar, R. Ade, A novel approach for handling imbalanced data in medical diagnosis using undersampling technique, *Commun. Appl. Electron.*, **5** (2016), 36–42. https://doi.org/10.5120/cae2016652323
11. J. Zhang, L. Chen, F. Abid, Prediction of breast cancer from imbalance respect using cluster-based undersampling method, *J. Healthcare Eng.*, **2019** (2019), 7294582. https://doi.org/10.1155/2019/7294582
12. P. Vuttipittayamongkol, E. Elyan, Overlap-based undersampling method for classification of imbalanced medical datasets, in *IFIP International Conference on Artificial Intelligence Applications and Innovations*, (2020), 358–369. https://doi.org/10.1007/978-3-030-49186-4_30
13. N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, *J. Artif. Intell. Res.*, **16** (2002), 321–357. https://doi.org/10.1613/jair.953
14. C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-Level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, **5476** (2009), 475–482. https://doi.org/10.1007/978-3-642-01307-2_43
15. D. A. Cieslak, N. V. Chawla, A. Striegel, Combating imbalance in network intrusion datasets, in *2006 IEEE International Conference on Granular Computing*, (2006), 732–737. https://doi.org/10.1109/GRC.2006.1635905
16. J. de la Calleja, O. Fuentes, J. González, Selecting minority examples from misclassified data for over-sampling, in *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference*, (2008), 276–281.
17. M. A. H. Farquad, I. Bose, Preprocessing unbalanced data using support vector machine, *Decis. Support Syst.*, **53** (2012), 226–233. https://doi.org/10.1016/j.dss.2012.01.016
18. Q. Wang, A hybrid sampling SVM approach to imbalanced data classification, *Abstr. Appl. Anal.*, **2014** (2014), 972786. https://doi.org/10.1155/2014/972786
19. J. Zhang, L. Chen, Clustering-based undersampling with random over sampling examples and support vector machine for imbalanced classification of breast cancer diagnosis, *Comput. Assisted Surg.*, **24** (2019), 62–72. https://doi.org/10.1080/24699322.2019.1649074
20. D. Li, C. Wu, T. Tsai, Y. Lina, Using mega-trend-diffusion and artificial samples in small data set learning for early flexible manufacturing system scheduling knowledge, *Comput. Oper. Res.*, **34** (2007), 966–982. https://doi.org/10.1016/j.cor.2005.05.019

21. L. Breiman, Bagging predictors, *Mach. Learn.*, **24** (1996), 123–140. https://doi.org/10.1007/BF00058655
22. J. Alcalá-Fdez, L. Sánchez, S. García, M. J. del Jesus, S. Ventura, J. M. Garrell, et al., KEEL: a software tool to assess evolutionary algorithms for data mining problems, *Soft Comput.*, **13** (2009), 307–318. https://doi.org/10.1007/s00500-008-0323-y
23. B. Haznedar, M.T. Arslan, A. Kalinli, Microarray Gene Expression Cancer Data, 2017. Available from: https://doi.org/10.17632/YNM2tst2hh.4.
24. M. Kubat, R. C. Holte, S. Matwin, Machine learning for the detection of oil spills in satellite radar images, *Mach. Learn.*, **30** (1998), 195–215. https://doi.org/10.1023/A:1007452223027
25. V. García, R. A. Mollineda, J. S. Sánchez, Theoretical analysis of a performance measure for imbalanced data, in *2010 20th International Conference on Pattern Recognition*, (2010), 617–620. https://doi.org/10.1109/ICPR.2010.156
26. J. A. Hanley, B. J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, *Radiology*, **143** (1982), 29–36. https://doi.org/10.1148/radiology.143.1.7063747
27. C. Cortes, V. Vapnik, Support-vector networks, *Mach. Learn.*, **20** (1995), 273–297. https://doi.org/10.1007/BF00994018
28. C. Liu, T. Lin, K. Yuan, P. Chiueh, Spatio-temporal prediction and factor identification of urban air quality using support vector machine, *Urban Clim.*, **41** (2022), 101055. https://doi.org/10.1016/j.uclim.2021.101055
29. Z. Wang, Y. Yang, S. Yue, Air quality classification and measurement based on double output vision transformer, *IEEE Internet Things J.*, **9** (2022), 20975–20984. https://doi.org/10.1109/JIOT.2022.3176126
30. L. Wei, Q. Gan, T. Ji, Cervical cancer histology image identification method based on texture and lesion area features, *Comput. Assisted Surg.*, **22** (2017), 186–199. https://doi.org/10.1080/24699322.2017.1389397
31. I. Izonin, R. Tkachenko, O. Gurbych, M. Kovac, L. Rutkowski, R. Holoven, A non-linear SVR-based cascade model for improving prediction accuracy of biomedical data analysis, *Math. Biosci. Eng.*, **20** (2023), 13398–13414. https://doi.org/10.3934/mbe.2023597
32. G. C. Batista, D. L. Oliveira, O. Saotome, W. L. S. Silva, A low-power asynchronous hardware implementation of a novel SVM classifier, with an application in a speech recognition system, *Microelectron. J.*, **105** (2020), 104907. https://doi.org/10.1016/j.mejo.2020.104907
33. A. A. Viji, J. Jasper, T. Latha, Efficient emotion based automatic speech recognition using optimal deep learning approach, *Optik*, (2022), 170375. https://doi.org/10.1016/j.ijleo.2022.170375
34. Z. Zeng, J. Gao, Improving SVM classification with imbalance data set, in *International Conference on Neural Information Processing*, **5863** (2009), 389–398. https://doi.org/10.1007/978-3-642-10677-4_44
35. Z. Luo, H. Parvïn, H. Garg, S. N. Qasem, K. Pho, Z. Mansor, Dealing with imbalanced dataset leveraging boundary samples discovered by support vector data description, *Comput. Mater. Continua*, **66** (2021), 2691–2708. https://doi.org/10.32604/cmc.2021.012547
36. G. Huang, Q. Zhu, C. Siew, Extreme learning machine: Theory and applications, *Neurocomputing*, **70** (2006), 489–501. https://doi.org/10.1016/j.neucom.2005.12.126
37. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al., Scikit-learn: Machine learning in Python, *J. Mach. Learn. Res.*, **12** (2011), 2825–2830

38. C. Wu, L. Chen, A model with deep analysis on a large drug network for drug classification, *Math. Biosci. Eng.*, **20** (2023), 383–401. https://doi.org/10.3934/mbe.2023018

AIMS Press