Mathematical Biosciences and Engineering

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

MBE, 19(6): 6204–6233.
DOI: 10.3934/mbe.2022290
Received: 13 December 2021
Revised: 06 March 2022
Accepted: 28 March 2022
Published: 15 April 2022

Research article

A new approach to generating virtual samples to enhance classification accuracy with small data—a case of bladder cancer

Liang-Sian Lin[1], Susan C Hu[2,*], Yao-San Lin[3], Der-Chiang Li[4] and Liang-Ren Siao[4]

[1] Department of Information Management, National Taipei University of Nursing and Health Sciences, Ming-te Road, Taipei 112303, Taiwan
[2] Department of Public Health, College of Medicine, National Cheng Kung University, University Road, Tainan 70101, Taiwan
[3] Singapore Centre for Chinese Language, Nanyang Technological University, Ghim Moh Road Singapore 279623, Singapore
[4] Department of Industrial and Information Management, National Cheng Kung University, University Road, Tainan 70101, Taiwan

* **Correspondence:** Email: shuhu@mail.ncku.edu.tw; Tel: 8862757575x53134; Fax: 8862374252.

Abstract: In the medical field, researchers are often unable to obtain the sufficient samples in a short period of time necessary to build a stable data-driven forecasting model used to classify a new disease. To address the problem of small data learning, many studies have demonstrated that generating virtual samples intended to augment the amount of training data is an effective approach, as it helps to improve forecasting models with small datasets. One of the most popular methods used in these studies is the mega-trend-diffusion (MTD) technique, which is widely used in various fields. The effectiveness of the MTD technique depends on the degree of data diffusion. However, data diffusion is seriously affected by extreme values. In addition, the MTD method only considers data fitted using a unimodal triangular membership function. However, in fact, data may come from multiple distributions in the real world. Therefore, considering the fact that data comes from multi-distributions, in this paper, a distance-based mega-trend-diffusion (DB-MTD) technique is proposed to appropriately estimate the degree of data diffusion with less impacts from extreme values. In the proposed method, it is assumed that the data is fitted by the triangular and trapezoidal membership functions to generate virtual samples. In addition, a possibility evaluation mechanism is proposed to measure the applicability of the virtual samples. In our experiment, two bladder cancer datasets are used to verify the effectiveness of the proposed DB-MTD method. The experimental results demonstrated that the proposed method outperforms other VSG techniques in classification and regression items for small bladder cancer datasets.

Keywords: small data approach; virtual sample generation; data-driven model

1. Introduction

Bladder cancer is a common cancer in the human urinary system. Some bladder cancer studies have reported that this cancer is related to tumor suppressor genes and oncogene genes, such as multidrug resistance (MDR), topoisomerase II (Topo II), Rb, epidermal growth factor receptor (EGFR), HER-2 (Neu), c-ErbB-3, c-ErbB-4, Cyclin A, Cyclin D1, P16, Cdc 2, Bcl-2, Bax, etc. [1–7]. These genes can be used to diagnose bladder cancer [1,4,6] and analyze the effects of X-ray treatment among bladder cancer patients [5,7]. For example, higher Cyclin D1 values indicate that patients are in the early stage of bladder cancer [2]. The over-expression values of EGFR, Neu, c-ErbB-3, c-ErbB-4 may reflect the treatment outcomes of Cobalt-60 (Co-60) radiation therapy on bladder cancer [3]. These studies often suffer from a small sample problem because it is difficult to obtain a sufficient number of gene expression profiles from patients due to the need for a costly genome sequencing procedure [8,9]. However, for researchers who adopt data-driven models as forecasting tools to predict cancer, it is necessary to collect sufficient data and find representative samples from which to build data-driven models with good learning performance [10–12]. With a limited number of gene profiles, traditional machine learning methods cannot construct data-driven models that will provide accurate predictions of bladder cancer. For this reason, small data learning has become an important challenge to bladder cancer prediction with small datasets.

1.1. Small data learning

Previous research approaches to the problem of small data learning can be divided into two types: The first proposes a better learning model for different data sets. For example, Mao et al. [13] proposed a modified Mahalanobis-Taguchi System to extract important information for the purpose of improving classification accuracy for a high-dimensional small dataset. Izonin et al. [14] proposed the input-doubling method to improve prediction performance of the compressive strength of trabecular bone with only 77 observations. Due to data diversity, it is difficult to apply the same model to different datasets. Researchers must fine-tune the parameters of data-driven models to improve the learning of models using small data. The second approach is a virtual sample generation (VSG) method used to expand the quantity of data. VSG methods have been developed in various fields, such as diagnostic classification of tumors [1,4,6,15], prediction of the treatment effects of radiotherapy on tumors [5,7], and engineering applications [16–18]. These methods based on the concept of VSG are typically classified into three types: resampling-based VSG, model-based VSG, and diffusion-based VSG. The bootstrap method and the Synthetic Minority Over-sampling

Technique (SMOTE) [19] typify resampling-based VSG methods. The bootstrap method replaces the original sample with a synthetic sample by re-sampling the original samples [20]. The SMOTE method draws out a neighboring example of the original data as a new example to fill gaps between data. The neighboring example is calculated as shown in the following equation: $x_{synthetic} = x_i + random(0,1)\cdot(x_j - x_i), \forall i \neq j$, where $x_{synthetic}$ is a synthetic virtual sample different from x_i, and x_i is resampled data and x_j is the neighboring data of x_i, e.g., Figure 1.

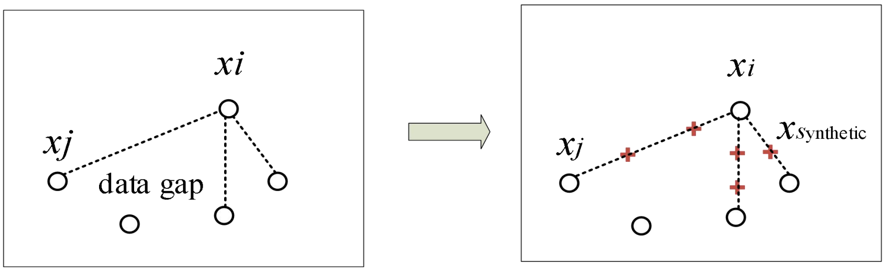


Figure 1. Synthetic sample generation of SMOTE.

Many studies have combined bootstrap methods with data-driven models to achieve good learning results [21–23]. For example, Lee et al. [21] created samples using the combined bootstrap method to construct a support vector regression (SVR), when only a small number of samples from the health care system were used to predict oscillometric measurement of blood pressure (BP). La Rocca and Perna [23] proposed a combination of extreme learning machine (ELM) models with bootstrap samples to improve the learning accuracy of small data sets. These experimental results showed that the proposed method led to a significant improvement in the performance of small data learning. The problem with bootstrapping, however, is that it can repeatedly overlearn the same sample, thus creating an overfitting problem. In this regard, Lee et al. [21] applied a bootstrap to a SVR to reduce the probability of overfitting. Unfortunately, most machine learning models are not immune to this problem. From the perspective of augmenting data diversity, some scholars suggest that by treating data that approximates the original sample as a virtual sample, the model can learn diverse data and reduce the incidence of overfitting.

Model-based VSG methods depend on machine learning models or artificial neural networks to find the relevance between input attributes and the output targets of small data in high dimensional space. The characteristics of original data can be learned by generating virtual samples in high dimensional space. Cho et al. [24] randomly generated virtual samples and selected virtual samples that were similar to the original examples using a multilayer perceptron neural network. Huang and Moraga [25] suggested the use of the diffusion-neural-network (DNN), which utilizes information diffusion and fuzzy theory to derive artificial samples in DNNs. In the field of image recognition, the generative adversarial net (GAN) model [26], which is specifically designed to generate synthetic images. The concept of the GAN is to generate a fake image that approximates a true image to increase the amount of training data. However, the GAN model is computationally intensive and may suffer from mode collapses due to low image variability when the true image data has an extremely skewed distribution [27].

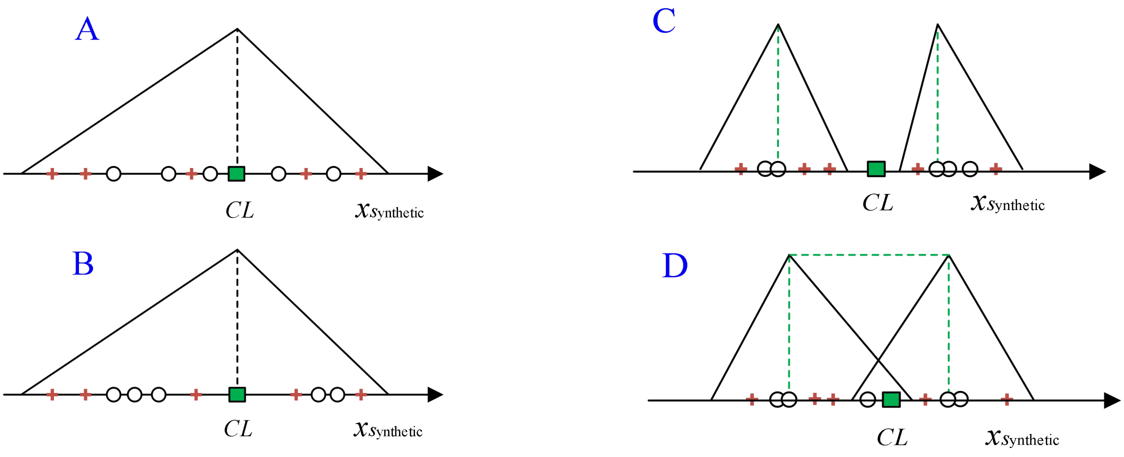
Diffusion-based VSG methods are represented based on information derived from original data.

The method usually assumes that data comes from a mathematical function or a specific probability distribution to generate virtual samples. The most representative method is mega-trend-diffusion (MTD) method [28]. Based on a triangular membership function (MF), the MTD method is to estimate domain of small data and generate virtual samples within the domain. Methods based on MTD method have been developed to improve the prediction performance of data-driven models with small datasets in practical applications [29–32]. For example, Majid et al. [30] applied the MTD technique to perform estimation of the data range in minority class data. Their experimental results showed that the generated virtual samples improved the classification accuracy of four data-driven models (k-nearest neighbor, support vector machines, Naïve Bayes, and random forest) for the prediction of human breast and colon cancers.

As an alternative to the MTD method, Yang et al. [33] proposed the VSG method based on a Gaussian distribution (GD), and virtual samples were generated within the estimated data domain. However, in the real world, data may be fitted using multiple distributions. The MTD and GD methods did not consider generating virtual samples based on multiple distributions. For this reason, based on multiple triangular MFs, Zhu et al. [31] proposed the multi-distribution MTD (MD-MTD) to generate virtual samples intended to improve the learning accuracies of models of purified terephthalic acid production. In addition, the MD-MTD method has been applied to enhance the prediction of an applicant risk assessment of an early online lending platform [32]. In addition to the assumption of multiple triangular MFs, Wang et al. [34] assumed that the data came from a mixture of normal distributions to create a large amount of data from a similar process and fuse data using a transfer learning method. The distribution of the fused data can be inferred using a nonparametric method. The estimated distribution is used to generate virtual samples to improve prediction in other processes. Although their method achieved good prediction results, it is more complex than the MD-MTD method. In addition, virtual samples generated using their method follow symmetric normal distributions. However, in fact, it is more appropriate to consider using asymmetric distributions to generate virtual samples for small datasets [31,32]. Therefore, based on the assumption of multi-distributions, in this paper, a distance-based mega-trend-diffusion (DB-MTD) technique is proposed to avoid over-diffusion of data due to extreme values and estimate the central locations (CLs) of multi-distributions. The detailed motivation for this work is explained in the following section.

1.2. Motivation

The traditional MTD method uses the midpoint of the data as an estimation of the CL of the data and estimates the skewness of the data by the amount of data greater or less than CL. The concept of data diffusion in MTD is that when the data is more skewed, the data domain is more diffused. However, when estimating the skewness of data in terms of data quantity, there is a problem: a small data set with about the same amount of data on both sides of the CL is estimated to be an un-skewed distribution for the reasons illustrated using Scenario A and Scenario B in Figure 2(a). Scenario A considers the amount of data used to estimate the degree of skewness of the data. As a result, the degree of skewness on the left side of the data is not significant enough to increase the spread of the data range. Scenario B considers the distance between the data and the CL to estimate the degree of skewness, and the data on the left is farther away from the CL, which means that the skewness between the left and the CL is greater, and the data distribution is close to the left skewed distribution. Therefore, the data in the left-hand area has a larger diffusion of data, meaning that more potential information is hidden in this region. Thus, more virtual samples can be explored in that domain. On the other hand, the MTD method assumes the data is fitted in a unimodal triangular MF, and it ignores the possibility that the data may come from a multi-modal triangular MF. This is illustrated in Scenarios C and D in Figure 2(b). The data in Scenario C is fitted in two triangular MFs, where the distance between the two MFs is relatively large. Therefore, it is regarded as multiple triangular MFs. The data in Scenario D indicates that when two triangular MFs have overlapping regions, this means that the similarity between the data is high, and it also means that potential information is hidden in the overlapping region. Therefore, in this paper, the two triangular MFs are fused into a trapezoidal MF.



(a) MTD method

(b) The proposed DB-MTD method

Figure 2. Two types of data diffusion.

As mentioned above, data diffusion depends on the degree of data skewness. When extreme values exist in a small dataset, the estimated skewness is excessively left or right, which results in over-diffusion of data. Most virtual samples generated in an over-extended domain are unsuitable and cannot effectively improve the learning of models. To address this problem, the MD-MTD method can be used to modify the estimation of data skewness in order to avoid excessive data diffusion. However, because data skewness in MD-MTD is still determined by the amount of data, the problem where different small datasets result in the same data diffusion shown in Scenario A and Scenario B still exists and thus needs to be solved. In addition, although the MD-MTD method estimates the CL of multi-distributions, it does not consider multiple CLs for multi-distributions. Therefore, in this paper, a distance-based mega-trend-diffusion (DB-MTD) method is proposed to extend the data range and reduce the influences of extreme values when there are small datasets. In our method, data skewness is determined by the distances between examples. When most of the examples are farther from the CL, this means that a significant amount of information is hidden between these examples and the CL, and vice versa. In addition, in this paper, a plausibility assessment mechanism (PAM) is proposed to select appropriate virtual samples generated by multiple triangular MFs or one trapezoidal MF.

This study validates the proposed DB-MTD approach using one bladder cancer classification case from Liao [6] to diagnose bladder cancer and a regression case from Tsai et al. [7] to predict the effects of treatment of bladder cancer patients with radiotherapy. In this study, a data-driven model using a back-propagation neural network (BPNN) was used to compare the learning performance between the proposed DB-MTD method and other VSG methods. The experimental results of the two bladder cancer datasets showed that the learning performance of the proposed method outperformed the other VSG methods. A paired-test was also used to verify whether there were statistically significant differences in the BPNN model for the two bladder cancer datasets.

The remainder of this study is organized as follows: Section 2 is a review of the literature on VSG methods, as well as a brief introduction to the learning tools used in the experimental validation. Section 3 explains the details of the proposed DB-MTD. Section 4 is the description of the two bladder cancer datasets and provides the experimental results. The conclusions are given in Section 5.

2. Background

Many studies have attempted to improve the learning performance of small datasets by generating virtual samples that been added into the original dataset to expand the amount of training data [24,25,28,33,31]. The generation of virtual samples has been verified as an effective strategy by which to extract meaningful information from a small dataset and increase the forecasting accuracy of models. Three related studies on virtual sample generation are reviewed in this section.

2.1. Diffusion neural network (DNN)

Huang and Moraga [25] proposed the DNN method, which applies a diffusion function to generate virtual samples on both sides of a given data point. In their approach, the data point is designated as the midpoint of a fuzzy normal function. After determining the possibility of *n* midpoints, the DNN approach then carried out symmetric diffusion on both sides of the midpoint to derive 2*n* virtual samples from Eqs (1)–(5). In Eqs (1) and (2), *u* and *v* are virtual samples of the input and output derived by using the original data; h_{set} is the diffusion coefficient obtained from Eq (3); "*r*" represents the linear correlation coefficient of the input *x* and the output *y*, and $\psi(r)$ represents the membership function value of the linear correlation coefficient *r*, as shown in Eq (5). After obtaining the corresponding values of the fuzzy normal membership function using Eqs (4), (1) and (2) are applied to produce virtual samples, *u* and *v*, after which the possibilities of the virtual samples are entered into a neural network for training, as shown in Figure 3.

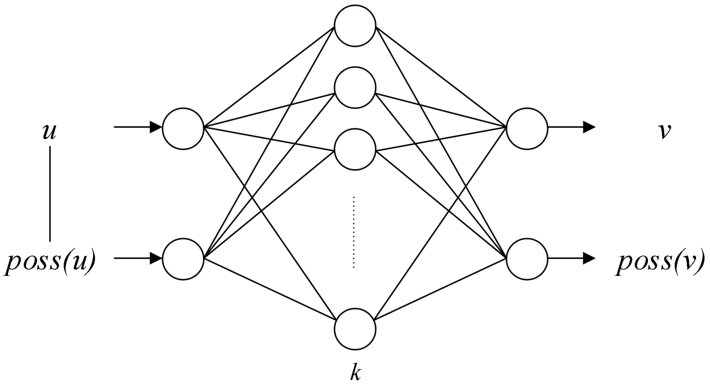


Figure 3. Small sample neural network (*2-k-2* scheme).

$$u = x \pm \sqrt{-2h_{set}^2 \ln \psi(r)} \tag{1}$$

$$h_{set} = \begin{cases} 0.6841(b-a), & \text{for } n=5 \\ 0.5404(b-a), & \text{for } n=6 \\ 0.4482(b-a), & \text{for } n=7 \\ 0.3839(b-a), & \text{for } n=8 \\ \dfrac{2.6851(b-a)}{n-1}, & \text{for } n \geq 9 \end{cases}, \text{ where } a = \min_{1\leq i\leq n}\{x_i\},\ b = \max_{1\leq i\leq n}\{x_i\} \tag{2}$$

$$v = y \pm \sqrt{-2h_{set}^2 \ln \psi(r)} \tag{3}$$

$$poss(x,u) = \frac{1}{h_{set}\sqrt{2\pi}} e^{\left[-\frac{(x-u)^2}{2h^2}\right]} \tag{4}$$

$$\psi(r) = \psi\left(0.9 + m\times 10^{-2}\right) \mapsto \underbrace{0.9...9}_{m9s}, \forall r \in \{0.91, 0.92, ..., 0.99\}, m = 1,...,9 \tag{5}$$

2.2. The mega-trend-diffusion (MTD) approach

Li et al. [28] proposed the mega-trend-diffusion (MTD) method to estimate the triangular membership function used to generate virtual samples. This method assumes that data follows the assumptions below:

1) Each variable is independent of the other.
2) Each variable is fitted into a unimodal distribution.

The virtual sample generation of data range can be estimated using Eqs (6) and (7).

$$a = u_{set} - Skew_L \times \sqrt{-2 \times \frac{\hat{s}_x^2}{N_L} \times \ln(\varphi)} \tag{6}$$

$$b = u_{set} + Skew_U \times \sqrt{-2 \times \frac{\hat{s}_x^2}{N_U} \times \ln(\varphi)} \tag{7}$$

where $u_{set} = (\max + \min)/2$ is the midpoint, and the coefficients of skewness are defined as $Skew_L = \frac{N_L}{N_L + N_U}$ and $Skew_U = \frac{N_U}{N_L + N_U}$, which refers to the degree of diffusion on two sides of u_{set}; $\sqrt{-2 \times \frac{\hat{s}_x^2}{N_L} \times \ln(\varphi)}$ and $\sqrt{-2 \times \frac{\hat{s}_x^2}{N_U} \times \ln(\varphi)}$ are defined as the proportion of the degree of diffusion, and $\varphi = 10^{-20}$ and $\hat{s}_x^2$ represent the sample variance, where the virtual sample is generated randomly between interval [*a*,*b*], as shown in Figure 4(a).

2.3. The Multi-distribution MTD (MD-MTD) approach

Zhu et al. [31] presented the multi-distributions MTD method to avoid over-diffusion of small data. They modified the MTD method to avoid extreme values to influence estimations of data skewness and CL. In their paper, the data skewness and CL are modified as:

$$Skew_L = \frac{N_L}{N_L + N_U + m} \tag{8}$$

$$Skew_U = \frac{N_U}{N_L + N_U + m} \tag{9}$$

$$u_{set} = \begin{cases} x_{\left[\frac{n}{2}+1\right]} & , \ if \ n \text{ is odd} \\ \frac{1}{2}\left(x_{\left[\frac{n}{2}\right]} + x_{\left[\frac{n}{2}+1\right]}\right), & if \ n \text{ is even} \end{cases} \tag{10}$$

where *m* is a shape parameter used to adjust the degree of data skewness, and $x_{[\cdot]}$ is an order statistic. The *m* is set to one in their method. The N_L (N_U) is the number of values smaller (greater) than u_{set}. The range estimation of MD-MTD method is defined by Eqs (6) and (7). Due to the adjusted skewness on the left and right side of CL, the data diffusion is reduced into a narrower domain [*a*,*b*], as shown in Figure 4(b).

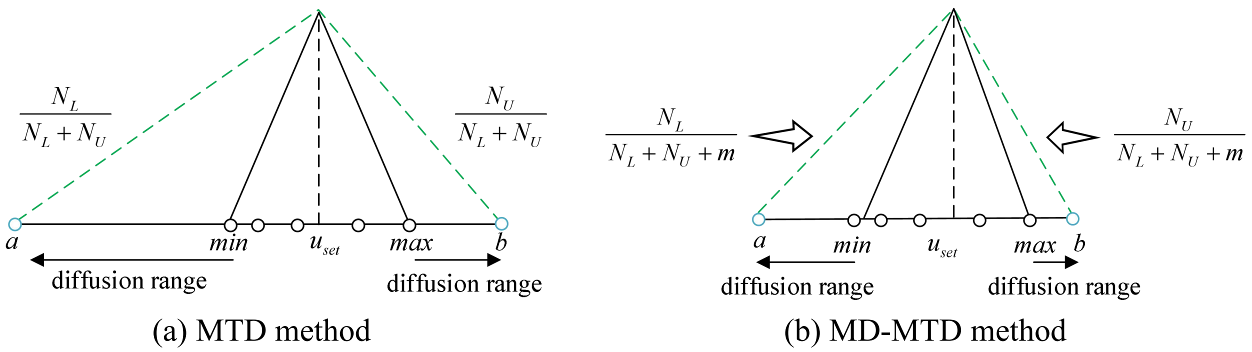


Figure 4. Data diffusion in the MTD and MD-MTD methods.

2.4. Back-propagation neural network (BPNN)

The back-propagation neural network (BPNN) was adopted in the present study as the data-driven model, which is a supervised learning model applicable for classification or forecasting problems. The input and output of the training data were used to build a neural network, and the weight of each node could be adjusted based on the difference between the degree of the expected output value and the actual observed value between each layer. A mapping function was built to process the weights in the different layers, as shown in Eq (11).

$$Y = f\ (\sum_{i=1}^{P} W_i X_i) + b \tag{11}$$

The output result for the input can be expressed as Eq (11), where X_i and Y are *i*th input variable and one output variable, respectively; W is the weighting value between the node and other nodes in the layer; $f(\cdot)$ is an activate function for nodes; *b* represents the product-sum of the weights, which must be greater than an error value, so the input nodes can be transferred into other nodes between linking layers, and p is the number of inputs from other nodes. In this paper, $f(\cdot)$ is set as the sigmoid activation function to update the weights of the nodes in the *j*th hidden layer as follows:

$$f_j(x) = \frac{1}{1+e^{-x}},\ 0 < f_j(x) < 1 \tag{12}$$

In a basic BPNN scheme, nodes are composed of so-called “layers,” as shown in Figure 5, where the input and output layers are, respectively, the attributes and target of data. A hidden layer is mainly used to process the data, and the number of layers is determined according to the data complexity. Generally speaking, a higher number of hidden layers and more nodes produce fewer learning errors, but the network scheme in the training process is more complex and has a longer convergence time.

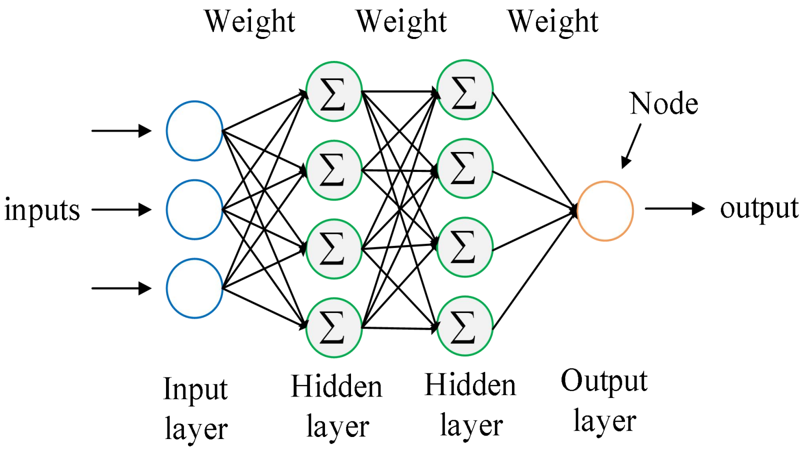


Figure 5. Basic BPNN (Input layer - *j* Hidden layers - Output layer).

3. Methodology

In this paper, we propose a unique distance-based mega-trend-diffusion method, including the skewness degree, the diffusion coefficient, the selection of u_{set}, and the number of u_{set}. This section illustrates the proposed method in detail, including the plausibility assessment mechanism (PAM) in Subsection 3.8 used to select suitable virtual samples.

3.1. Symbol definitions

Assume that a training dataset has *n* examples with *m*-1 input attributes $\{X_j \mid j=1,2,...,m-1\}$, and one output target X_m, denoted as $T=\{(\vec{x}_1,\vec{y}_1),\ (\vec{x}_2,\vec{y}_2),...,(\vec{x}_i,\vec{y}_i)\}$, $i=1,2,...,n$, where $\vec{x}_i=\{x_{i,1},x_{i,2},...,x_{i,m-1}\}$ has *m*-1 variables and $\vec{y}_i$ is the target of $\vec{x}_i$. The elements of the variables are $\{x_{i,i} \mid i=1,2,...,n;\ j=1,2,...,m\}$ in the training dataset, as shown in Table 1.

Table 1. Symbol definitions in a training dataset.

NO.	Input attributes					Output
	X_1	…	X_j	…	X_{m-1}	X_m
1	$x_{1,1}$	…	$x_{1,j}$	…	$x_{1,m-1}$	$x_{1,m}$
…	…	…	…	…	…	…
i	$x_{i,1}$	…	$x_{i,j}$	…	$x_{i,m-1}$	$x_{i,m}$
…	…	…	…	…	…	…
n	$x_{n,1}$	…	$x_{n,j}$	…	$x_{n,m-1}$	$x_{n,m}$

3.2. Data pre-processing

This subsection introduces the Min-Max data normalization method and *k*-nearest neighbor (*k*NN) algorithm used for imputation of missing data.

3.2.1. Min-Max data normalization

Because the proposed method is based on the distances between examples to measure the degree of data diffusion, before performing our proposed method, a data normalization process is necessary to avoid the effects from different variable scales that affect the data diffusion measurement. In this paper, we applied the Min-Max data normalization process to transform the domain of original data $x_{i,j}$ into [0,1]. The transforming formula is expressed as:

$$x_{i,j}^{*} = \frac{x_{i,j} - \min_j}{\max_j - \min_j} \in [0,1] \tag{13}$$

where $x_{i,i}^{*}$ is transformed value and $\max_i$ ($\min_i$) is the maximum(minimum) value of the X_i variable. Min-Max normalization is a linear transformation intended to maintain the relative distances between the original examples. As a result, the proposed method can be effectively performed with transformed examples for small datasets with different variable scales.

3.2.2. *k*NN in missing data imputation

When missing values occur in the original dataset, they increase the amount of data diffusion measurement error due to an incomplete data structure. There are two types of methods that can be used to deal with missing values. One is the ignoring missing value method in which examples with missing values are deleted, and the other is the data imputation method, where missing values are replaced with plausible values. In this paper, we adopt the data imputation method to fill in missing values instead of deleting them due to the very limited amount of data. Common data imputation methods include mean imputation, imputation with distributions, *k*NN imputation, etc. Some studies have suggested that *k*NN has better imputation performance than other methods [35–38]. For example, Jadhav et al. [38] compared *k*NN with other methods in terms of data imputation, and *k*NN significantly outperformed the other methods for numerical datasets. Because the proposed method is applicable to numerical variables, we use the *k*NN imputation method to handle data with missing values.

The *k*NN is an instance-based algorithm proposed by Cove and Hart [39]. This algorithm classifies examples based on similarity measurements that are made between examples. For data imputation, the *k*NN is used to impute missing values by selecting the nearest neighbors of examples with missing values. In this paper, we use the Euclidean distance as the similarity measurement to select the nearest neighbors of the example, as follows:

$$Euclidean\ distance = \sqrt{\sum_{i=1}^{k}(x_{i,j} - x_{i,j})^2} \tag{14}$$

where k is the number of nearest neighbors, and j represents jth data variable. We illustrate a simple example of *k*NN for data imputation in Figure 6. When we set k as three, the original data with a missing value has two neighbors, where the missing value can be imputed using the nearest neighbors.

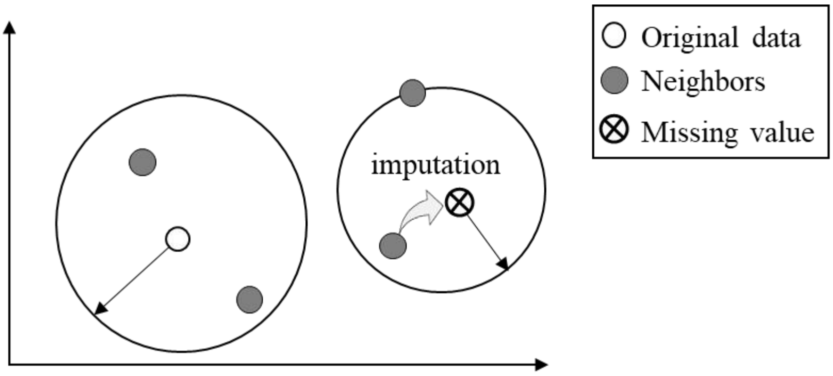


Figure 6. The *k*NN for data imputation.

3.3. Measurement of the degree of skewness

In the original MTD, the number of samples occurring on two sides of u_{set} is used to determine the degree to which the data is skewed. When the number of samples is significantly larger on one side, the MTD formula is not suitable for estimating the data distribution. In this paper, a unique MTD method is proposed based on the distance between samples. The idea of this paper is that when the sum of the distances between the samples on one side is less than that on the other side, this indicates that the sample distribution is skewed to one side.

The distance between the samples is defined by using the distance equation derived from the Minkowski Distance, as noted by Cha [40], where $\text{Distance} = \left(\sum_{i=1}^{d}|P_i - Q_i|^h\right)^{\frac{1}{h}}$. Since in this paper, it is assumed that each variable is independent, only a 1-D data distribution is considered. That is, $\text{Distance} = \sum_{i=1}^{d}|P_i - Q_i|$, where P_i represents the ith sample point x_i, and Q_i is the u_{set} of variables. The degree to which the data is skewed is considered in Eqs (18) and (19).

$$G = \sum_{i=1}^{n}|x_i - u_{set}| \tag{15}$$

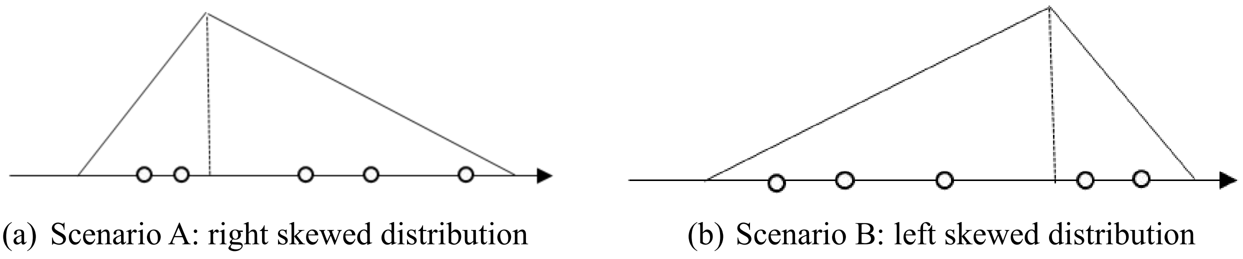
$$G_L = \sum_{i=1}^{n} \left| x_i - u_{set} \right| = \sum_{i=1}^{n} u_{set} - x_i, \text{ for } x_i < u_{set} \tag{16}$$

$$G_U = \sum_{i=1}^{n} \left| x_i - u_{set} \right| = \sum_{i=1}^{n} x_i - u_{set}, \text{ for } x_i > u_{set} \tag{17}$$

$$Skew_L = \frac{G_L}{G_L + G_U} \tag{18}$$

$$Skew_U = \frac{G_U}{G_L + G_U} \tag{19}$$

where G_L (or G_U) represents the sum of the distance between the smaller (or greater) sample, as compared to u_{set} and u_{set}. Figure 7(a) (or (b)) shows that if $Skew_L$ is smaller (or greater) than $Skew_U$, the data distribution is inferred as being a right (or left) skewed distribution.


(a) Scenario A: right skewed distribution

(b) Scenario B: left skewed distribution

Figure 7. Skewed distribution.

3.4. Diffusion coefficient

In the MTD, the diffusion coefficient is set as $\frac{\hat{s}_x^2}{N_L}$ or $\frac{\hat{s}_x^2}{N_U}$ to fine-tune the degree of diffusion on two sides of u_{set}. When extreme values exist in the dataset, then the variations among the samples are significant, so the sample variance $\hat{s}_x^2$ leads to excessive effects on the estimate of the diffusion coefficient h_{set}. In order to reduce this effect, h_{set} is set as $\frac{|\hat{s}_x|}{G_L}$ or $\frac{|\hat{s}_x|}{G_U}$. Therefore, in order to reduce the influence of outliers by using the distance function G on two sides of u_{set}, we adjust the diffusion coefficient as follows: $\sqrt{-2\times\frac{|\hat{s}_x|}{G_L}\times\ln(\varphi)}$ or $\sqrt{-2\times\frac{|\hat{s}_x|}{G_U}\times\ln(\varphi)}$. If G_L is greater than G_U, then it will be skewed to the left as a small h_{set}, and if there are potential outliers, then the *G* value in this direction will be increased. Thus, the proposed h_{set} can be used to mitigate the increasingly excessive diffusion.

3.5. Selection of u_{set}

This selection of u_{set} is often seriously affected when there are many extreme values in a dataset, and thus, the midpoint u_{set} that is obtained may not represent the tendency of the dataset although u_{set} can be set as the mean of the samples, and the mean can then be used as the estimation of the data tendency when the number of samples is sufficient. However, setting the mean of u_{set} is unstable in terms of estimating the data tendency in small samples. Thus, when considering the estimation of the tendency of the dataset for a small dataset, the median and mode are considered to be slightly affected by extreme values. *k* modes can be used to discover the frequency of repeated values in the dataset to predict the tendency of the dataset. Therefore, the proposed setting of u_{set} can be defined as follows:

$$u_{set,k}=\begin{cases}Me, \text{ where } k=1\\ Mo_k, \text{ where } 2\le k\le\left\lfloor\frac{n}{2}\right\rfloor\end{cases}\tag{20}$$

$$Me=\begin{cases}\dfrac{x_{\left\lceil\frac{n}{2}\right\rceil}+x_{\left\lceil\frac{n}{2}+1\right\rceil}}{2}, \text{ for } n=2r, \text{ where } \forall r\in N\\ x_{\left\lceil\frac{n+1}{2}\right\rceil}, \text{ for } n=2r+1, \text{ where } \forall r\in N\end{cases}\tag{21}$$

$$Mo_k=x_i, \text{ for } i=1,2,...,n, \text{ where } 2\le k\le\left\lfloor\frac{n}{2}\right\rfloor\tag{22}$$

Given a dataset $X=\{x_1,x_2,...,x_n\}$, *n* represents the number of samples, and *k* is the number of modes, where the mode is defined as identical data that occurs more than twice in a dataset. If no mode exists in a dataset, then the u_{set} is the median.

3.6. Estimation of the data range

Assuming a dataset $X=\{x_1,x_2,...,x_n\}$, the forms of Eqs (6),(7) and Eqs (16),(17) can be modified with *k* u_{set} as follows:

$$a_k = \begin{cases} u_{set,k} - Skew_{L,k} \times \sqrt{-2 \times \frac{|\hat{s}_x|}{G_{L,k}} \times \ln(\varphi)} \\ u_{set,k}, \text{ where } G_{L,k} = 0 \end{cases} \tag{23}$$

$$b_k = \begin{cases} u_{set,k} + Skew_{U,k} \times \sqrt{-2 \times \frac{|\hat{s}_x|}{G_{U,k}} \times \ln(\varphi)} \\ u_{set,k}, \text{where } G_{U,k} = 0 \end{cases} \tag{24}$$

$$G_{L,k} = \sum_{i=1}^{n} \left| x_i - u_{set,k} \right| = \sum_{i=1}^{n} u_{set,k} - x_i, \text{ for } x_i < u_{set,k} \tag{25}$$

$$G_{U,k} = \sum_{i=1}^{n} \left| x_i - u_{set,k} \right| = \sum_{i=1}^{n} x_i - u_{set,k}, \text{ for } x_i > u_{set,k} \tag{26}$$

Eqs (25) and (26) represent the sum of the distances between x_i and $u_{set,k}$. $G_{L,k}$ and $G_{U,k}$ represent the sum of the distances between the sample points fewer and greater than *k*th u_{set}, respectively. Therefore, in this case, we use the distance function on two sides of *k*th u_{set} as an expression of skewness, as shown in Eqs (27) and (28).

$$Skew_{L,k} = \frac{G_{L,k}}{G_{L,k} + G_{U,k}} \tag{27}$$

$$Skew_{U,k} = \frac{G_{U,k}}{G_{L,k} + G_{U,k}} \tag{28}$$

In Eqs (23) and (24), considering the diffusion coefficient presented in Section 3.4, two specific cases are discussed: If the *k*th mode is one of the extreme values, this will cause $G_{L,k}$ or $G_{U,k}$ to be close to 0 because $x_i \approx u_{set} \rightarrow |x_i - u_{set}| \rightarrow 0$; then, the diffusion in the direction of the extreme value cannot be defined, and the lower bound a_k or the upper bound b_k of the *k*th mode is set as $u_{set,k}$. In addition, when the total number of samples is one, both $G_{L,k}$ and $G_{U,k}$ are 0.

3.7. Virtual sample generation with the proposed method

Since the previous MTD method set u_{set} as the midpoint, the method is susceptible to extreme values. In addition, the method assumes that the data is fitted into a unimodal distribution, but the data may come from a multi-modal distribution. For these two reasons, the new DB-MTD method proposed in this paper generates virtual samples from a unimodal distribution or a multi-modal distribution, where the median or mode is set as u_{set} to reduce the influence of outliers. The process proposed in the DB-MTD used to generate virtual samples is shown in Figure 8.

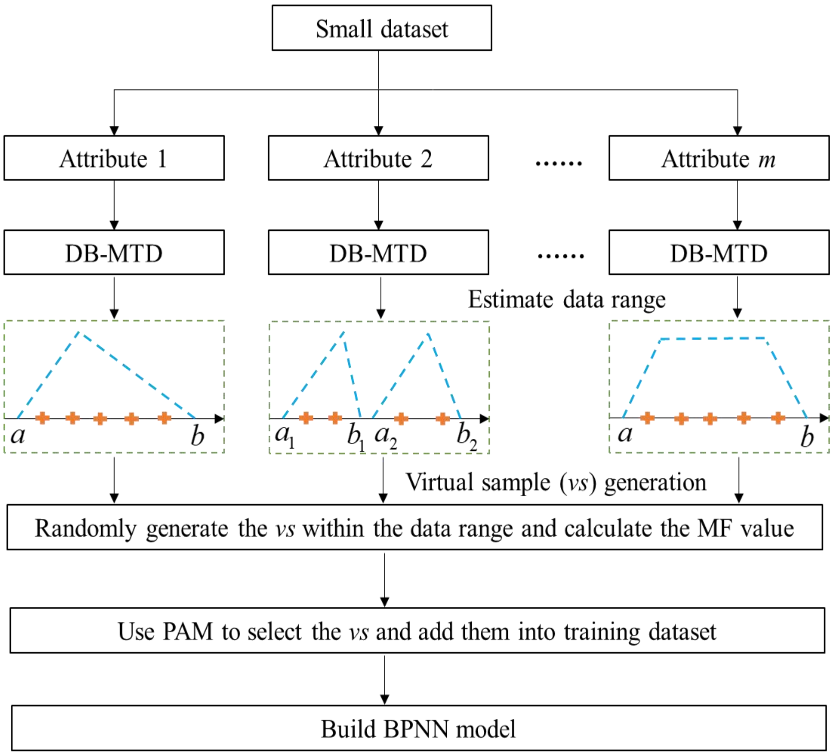

Small dataset
Attribute 1
Attribute 2
......
Attribute *m*
DB-MTD
DB-MTD
......
DB-MTD
Estimate data range
a
b
a_1
b_1
a_2
b_2
a
b
Virtual sample (*vs*) generation
Randomly generate the *vs* within the data range and calculate the MF value
Use PAM to select the *vs* and add them into training dataset
Build BPNN model

Figure 8. Diagram of the proposed process.

When using the proposed method to estimate the data range, some possible scenarios are illustrated as follows:

1) When a dataset does not have a mode, the median is taken as the u_{set}, as shown in Figure 9, Scenario A. Or, when the mode exists in the dataset, then u_{set} is the mode, as shown in Figure 9, Scenario B.
2) When there is only one mode, the u_{set} is the extreme minimum, as shown in Figure 9, Scenario C. If u_{set} is the extreme maximum, the estimation of membership function is as shown in Figure 9, Scenario D.
3) When there is more than one mode, then it is necessary to generate more than one triangular fuzzy membership function, and we thus need to consider whether multiple triangular membership functions will overlap or not.

a. When there is no overlapping area between any two modes, there are two triangular membership functions, as shown in Figure 9, Scenario E.

b. When these two triangular fuzzy membership functions overlap, there is one trapezoidal membership function, as shown in Figure 9, Scenario F.

Since the overlapping of two triangular fuzzy membership functions in Scenarios E and F mean that important data may be located in the overlapping area, the value of the membership function of the sample point is set as 1. In other words, we thus set the possibility of the virtual sample being located in the interval as 1.

After estimating the data range given by Eqs (23) and (24), the virtual sample is randomly generated within the specified range with a uniform distribution.

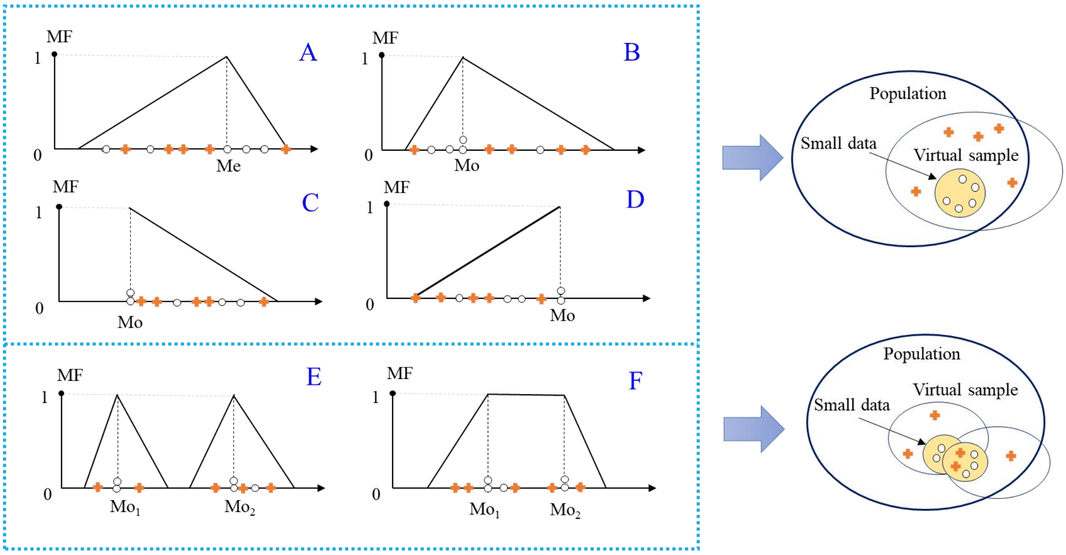


Figure 9. The mixture MFs.

As mentioned above, we discuss possible scenarios of the membership functions, as shown in Figure 9, where the peak of the membership function is set at 1, and the membership function value of the upper bound and lower bound is set at 0. When k u_{set} exists, and sample x lies in the interval [a_k , b_k], the value of the membership function of x can be calculated when multiple triangular membership functions do not overlap each other, using Eq (29). When multiple triangular membership functions overlap each other, trapezoidal membership functions are used, as shown in Eq (30).

$$Triangular\ MF(x)=\begin{cases}\dfrac{x-a_k}{u_{set,k}-a_k}, \text{ as } x<u_{set,k}\\ \dfrac{b_k-x}{b_k-u_{set,k}}, \text{ as } x>u_{set,k}\\ 1, \text{ as } x=u_{set,k}\\ 0, \text{ as } x=a_k \text{ or } b_k\end{cases} \tag{29}$$

$$Trapezoidal\ MF(x)=\begin{cases}\dfrac{x-min(a_k)}{min(u_{set,k})-min(a_k)}, \text{ as } x\in\left[min(a_k),min(u_{set,k})\right]\\ \dfrac{max(b_k)-x}{max(b_k)-max(u_{set,k})}, \text{ as } x\in\left[max(u_{set,k}),max(b_k)\right]\\ 1, \text{ as } x\in\left[min(u_{set,k}),max(u_{set,k})\right]\\ 0, \text{ as } x=min(a_k) \text{ or } max(b_k)\end{cases} \tag{30}$$

3.8. Plausibility assessment mechanism (PAM)

According to the virtual sample generation method, the values of the fuzzy membership functions are used to randomly generate virtual samples in order to fill the gaps in the data in the case of small sample learning. In this paper, a procedure was developed for a PAM that could be used to select the virtual samples based on an inferred data distribution so as to increase learning robustness. This mechanism can be used to examine whether the randomly generated virtual values are qualified or not, and it is described as follows in detail: Firstly, a *tv* (temporary value) is randomly generated from the estimated interval [*a*,*b*], after which the corresponding membership function values, MF (*tv*), are discovered, as shown in Figure 10.

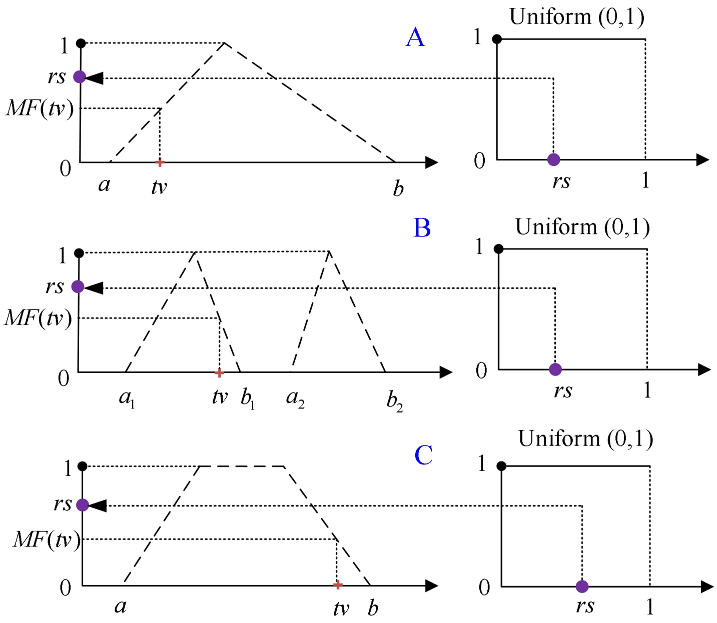


Figure 10. The proposed PAM.

Three possible scenarios are illustrated, as follows: In Figure 10, Scenario A only has one u_{set}; Scenario B is two triangular membership functions, and Scenario C is a trapezoid fuzzy membership function with two u_{set}. When determining whether *tv* can be kept as a virtual sample or not, we additionally generate a *rs* (random seed) ranging between [0,1], as shown in Figure 10. This random seed is defined based on a uniform distribution. In the testing process, if the random seed is less than the membership function value of *tv*, then this *tv* will be kept as a suitable virtual sample *v*; if not, this *tv* will be abandoned, and a new virtual value will be generated. When a *tv* is close to the u_{set}, then the membership function value of *tv* is close to 1, and thus, it will have a higher likelihood of being be kept as a qualified virtual sample, as shown in Eq (31).

$$v = tv,\ as\ rs \leq MF(tv) \tag{31}$$

4. Experiments

This section illustrates the experimental procedures designed for the purposes of this study. We compare the proposed method with other VSG methods with two bladder cancer cases. The experiments were implemented with Python 3.8.10. The experimental results are discussed in the following section.

4.1. Case description

Two bladder cancer cases are examined in our experiments. One case provided by Liao [6] is used to classify whether a patient suffers from bladder cancer, and the other case was obtained in [7] to predict a bladder cancer patient's resistance to radiotherapy.

4.1.1. Bladder cancer (BC) case

The BC dataset comprises nine bladder cancer patients and nine healthy persons, as listed in Table 2. The first nine examples are bladder cancer patients defined as class 1, and the last nine samples are healthy persons defined as class 0. Thirteen gene proteins MDR, Topo II, Rb, EGFR, Neu, c-ErbB-3, c-ErbB-4, Cyclin A, Cyclin D1, P16, Cdc 2, Bcl-2, and Bax extracted in bladder cell lines are set as the input attributes.

Table 2. BC dataset.

Bladder cell lines		Input attributes													Output
NO.	ID	MDR	Topo II	Rb	EGFR	Neu	c-ErbB-3	c-ErbB-4	Cyclin A	Cyclin D1	P16	Cdc 2	Bcl-2	Bax	class
1	BFTC-905	1.8	0.1	2.5	7	0.8	1	1	0.3	2.5	0.1	2	0.1	1.5	1
2	HT 1197	1	1.3	1	8.5	0.3	0.1	3.5	0.3	0.8	2.8	2.5	1	1.3	1
3	T24	1.5	3.5	0.7	2	1	1.5	2.5	2	2.5	2.3	2	2	0.1	1
4	TCC-SUP	0.1	1	1	1.5	0.1	0.1	3	0.1	0.8	2.3	2.8	0.1	0.2	1
5	J82	0.3	1.3	2.7	2.2	1	0.1	2.5	0.5	2	0.1	2.5	1.3	0.2	1
6	TSGH-8301	2	0.3	0.3	9	3.5	4.5	6.5	0.1	0.1	0.1	2.5	1.3	1.5	1
7	HT 1376	1	2	1.3	3	0.3	3	2.5	0.1	1.3	1.3	2.5	0.1	0.8	1
8	Sca-BER	1	1	0.5	5	0.5	6.5	6.5	0.7	0.1	0.5	2.8	1.8	0.3	1
9	5637	1.2	0.8	1.5	10	0.5	0.1	5	0.1	2.7	0.1	2.5	0.1	1.3	1
10	A	0.3	0.1	6	0.5	0.1	0.2	0.5	0.1	0.5	5	0.5	5	8	0
11	B	0.1	0.3	4	1	0.2	0.1	2	0.3	0.1	4	0.5	9	5	0
12	C	0.2	0.1	5	0.5	0.1	0.1	1.5	0.2	0.1	6	1	7	5	0
13	D	0.1	0.8	9	1.5	0.1	0.5	0.5	0.1	1	6	0.8	6	6	0
14	E	0.5	0.1	6	2	0.5	0.1	0.5	0.3	0.1	5	0.5	4	4	0
15	F	0.5	0.5	6	2	0.1	0.8	0.8	0.1	0.2	6	1	5	5	0
16	G	0.1	1	4	1	0.3	0.2	0.5	0.5	0.2	5	1.5	5	7	0
17	H	0.5	0.1	5	0.5	0.2	0.1	1	0.1	0.1	8	0.5	7	7	0
18	I	0.1	0.1	7	0.8	0.1	0.1	2	0.2	0.5	5	0.5	6	4	0

Table 3. RBC dataset.

Bladder cell lines		Input attributes														Output
NO.	ID	MDR	Topo II	Rb	EGFR	Neu	c-ErbB-3	c-ErbB-4	Cyclin A	Cyclin D1	P16	Cdc 2	Bcl-2	Bax	Co-60	Resistance to radiotherapy
1	HT 1376	1	1	-0.5	5	0.5	6.5	6.5	0.7	0.1	-0.5	2.8	1.8	-0.3	5	97
2	HT 1376	1	1	-0.5	5	0.5	6.5	6.5	0.7	0.1	-0.5	2.8	1.8	-0.3	10	90
3	HT 1376	1	1	-0.5	5	0.5	6.5	6.5	0.7	0.1	-0.5	2.8	1.8	-0.3	20	84
4	HT 1376	1	1	-0.5	5	0.5	6.5	6.5	0.7	0.1	-0.5	2.8	1.8	-0.3	30	82
5	HT 1197	1	2	-1.3	3	0.3	3	2.5	0.1	1.3	-1.3	2.5	0.1	-0.8	5	92
6	HT 1197	1	2	-1.3	3	0.3	3	2.5	0.1	1.3	-1.3	2.5	0.1	-0.8	10	78
7	HT 1197	1	2	-1.3	3	0.3	3	2.5	0.1	1.3	-1.3	2.5	0.1	-0.8	20	72
8	HT 1197	1	2	-1.3	3	0.3	3	2.5	0.1	1.3	-1.3	2.5	0.1	-0.8	30	73
9	TCC-SUP	0.1	1	-1	1.5	0.1	0.1	3	0.1	0.8	-2.3	2.8	0.1	-0.2	5	94
10	TCC-SUP	0.1	1	-1	1.5	0.1	0.1	3	0.1	0.8	-2.3	2.8	0.1	-0.2	10	70
11	TCC-SUP	0.1	1	-1	1.5	0.1	0.1	3	0.1	0.8	-2.3	2.8	0.1	-0.2	20	50
12	TCC-SUP	0.1	1	-1	1.5	0.1	0.1	3	0.1	0.8	-2.3	2.8	0.1	-0.2	30	41
13	J82	1.5	3.5	-0.7	2	1	1.5	2.5	2	2.5	-2.3	2	2	-0.1	5	84
14	J82	1.5	3.5	-0.7	2	1	1.5	2.5	2	2.5	-2.3	2	2	-0.1	10	70
15	J82	1.5	3.5	-0.7	2	1	1.5	2.5	2	2.5	-2.3	2	2	-0.1	20	48
16	J82	1.5	3.5	-0.7	2	1	1.5	2.5	2	2.5	-2.3	2	2	-0.1	30	40
17	Sca-BER	1.2	0.8	-1.5	10	0.5	0.1	5	0.1	2.7	-0.1	2.5	0.1	-1.3	5	82
18	Sca-BER	1.2	0.8	-1.5	10	0.5	0.1	5	0.1	2.7	-0.1	2.5	0.1	-1.3	10	57
19	Sca-BER	1.2	0.8	-1.5	10	0.5	0.1	5	0.1	2.7	-0.1	2.5	0.1	-1.3	20	39
20	Sca-BER	1.2	0.8	-1.5	10	0.5	0.1	5	0.1	2.7	-0.1	2.5	0.1	-1.3	30	36
21	T24	0.3	1.3	-2.7	2.2	1	0.1	2.5	0.5	2	-0.1	2.5	1.3	-0.2	5	90
22	T24	0.3	1.3	-2.7	2.2	1	0.1	2.5	0.5	2	-0.1	2.5	1.3	-0.2	10	58
23	T24	0.3	1.3	-2.7	2.2	1	0.1	2.5	0.5	2	-0.1	2.5	1.3	-0.2	20	39
24	T24	0.3	1.3	-2.7	2.2	1	0.1	2.5	0.5	2	-0.1	2.5	1.3	-0.2	30	34
25	5637	1	1.3	-1	8.5	0.3	0.1	3.5	0.3	0.8	-2.8	2.5	1	-1.3	5	83
26	5637	1	1.3	-1	8.5	0.3	0.1	3.5	0.3	0.8	-2.8	2.5	1	-1.3	10	50
27	5637	1	1.3	-1	8.5	0.3	0.1	3.5	0.3	0.8	-2.8	2.5	1	-1.3	20	32
28	5637	1	1.3	-1	8.5	0.3	0.1	3.5	0.3	0.8	-2.8	2.5	1	-1.3	30	28
29	TSGH-8301	2	0.3	-0.3	9	3.5	4.5	6.5	0.1	0.1	-0.1	2.5	1.3	-1.5	5	60
30	TSGH-8301	2	0.3	-0.3	9	3.5	4.5	6.5	0.1	0.1	-0.1	2.5	1.3	-1.5	10	30
31	TSGH-8301	2	0.3	-0.3	9	3.5	4.5	6.5	0.1	0.1	-0.1	2.5	1.3	-1.5	20	29
32	TSGH-8301	2	0.3	-0.3	9	3.5	4.5	6.5	0.1	0.1	-0.1	2.5	1.3	-1.5	30	30
33	BFTC-905	1.8	0.1	-2.5	7	0.8	1	1	0.3	2.5	-0.1	2	0.1	-1.5	5	63
34	BFTC-905	1.8	0.1	-2.5	7	0.8	1	1	0.3	2.5	-0.1	2	0.1	-1.5	10	33
35	BFTC-905	1.8	0.1	-2.5	7	0.8	1	1	0.3	2.5	-0.1	2	0.1	-1.5	20	21
36	BFTC-905	1.8	0.1	-2.5	7	0.8	1	1	0.3	2.5	-0.1	2	0.1	-1.5	30	18

4.1.2. Radiotherapy of bladder cancer (RBC) case

The RBC dataset has 36 examples used to predict the resistance to Co-60 radiotherapy for bladder cancer cells. Each example has thirteen gene proteins, as mentioned above, and one additional attribute, the energy of gamma radiation from the Co-60 isotope, as listed in Table 3. The example output represents the patients' resistance to radiotherapy treatment.

4.2. An example using the proposed DB-MTD method

We randomly drew 10 samples in the RBC dataset as an example to explain the different scenarios in the proposed DB-MTD method. The drawn samples were set as a training dataset, as listed in Table 4**.** The implementation procedure for the DB-MTD method is explained in the following discussion.

Table 4. The training dataset.

Bladder cell lines		Input attributes													
NO.	ID	MDR	Topo II	Rb	EGFR	Neu	c-ErbB-3	c-ErbB-4	Cyclin A	Cyclin D1	P16	Cdc 2	Bcl-2	Bax	Co-60
1	HT 1197	1	2	-1.3	3	0.3	3	2.5	0.1	1.3	-1.3	2.5	0.1	-0.8	5
2	HT 1376	1	1	-0.5	5	0.5	6.5	6.5	0.7	0.1	-0.5	2.8	1.8	-0.3	5
3	5637	1	1.3	-1	8.5	0.3	0.1	3.5	0.3	0.8	-2.8	2.5	1	-1.3	5
4	TCC-SUP	0.1	1	-1	1.5	0.1	0.1	3	0.1	0.8	-2.3	2.8	0.1	-0.2	10
5	5637	1	1.3	-1	8.5	0.3	0.1	3.5	0.3	0.8	-2.8	2.5	1	-1.3	30
6	J82	1.5	3.5	-0.7	2	1	1.5	2.5	2	2.5	-2.3	2	2	-0.1	30
7	Sca-BER	1.2	0.8	-1.5	10	0.5	0.1	5	0.1	2.7	-0.1	2.5	0.1	-1.3	20
8	J82	1.5	3.5	-0.7	2	1	1.5	2.5	2	2.5	-2.3	2	2	-0.1	5
9	HT 1197	1	2	-1.3	3	0.3	3	2.5	0.1	1.3	-1.3	2.5	0.1	-0.8	30
10	J82	1.5	3.5	-0.7	2	1	1.5	2.5	2	2.5	-2.3	2	2	-0.1	20

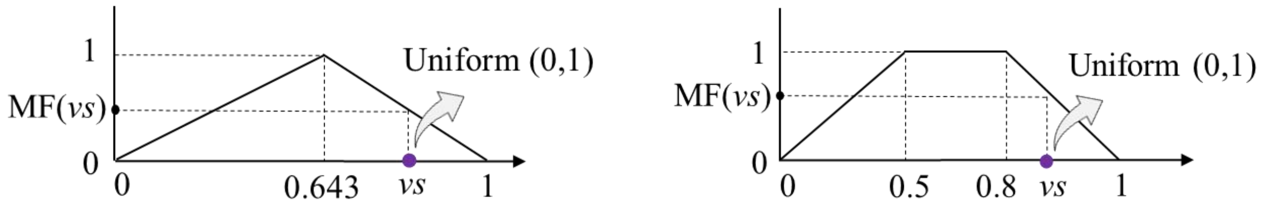
Table 5. Estimations of attributes using the DB-MTD method.

	Input attributes													
MF type	*Tri*	*Tri*	*Trap*	*Tri*	*Tri*	*Tri*	*Tri*	*Tri*	*Trap*	*Tri*	*Tri*	*Tri*	*Trap*	*Tri*
estimations	MDR	Topo II	Rb	EGFR	Neu	c-ErbB-3	c-ErbB-4	Cyclin A	Cyclin D1	P16	Cdc 2	Bcl-2	Bax	Co-60
a_1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$u_{set,1}$	0.643	0.315	0.5	0.176	0.333	0.219	0.062	0.105	0.3	0.185	0.625	0.474	0.0	0.4
b_1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
a_2	-	-	0.0	-	-	-	-	-	0.0	-	-	-	0.0	-
$u_{set,2}$	-	-	0.8	-	-	-	-	-	0.9	-	-	-	1.0	-
b_2	-	-	1.0	-	-	-	-	-	1.0	-	-	-	1.0	-

Note: "Tri" and "Trap" represents that the attribute is fitted by the triangular and trapezoidal MF, respectively.

We use the Min-Max data normalization process from Eq (13) to transform the training data domain into [0,1]. The lower bound a and upper bound b of the input attributes can be obtained from Eqs (23) and (24), as shown in Table 5. The Rb, Cyclin D1, Bax attributes are determined to be the trapezoidal MF based on our method, where we could obtain two CLs. The remaining attributes were defined as the triangular MF.

To explain the PAM used to select the virtual samples, we use the MDR and Rb attributes as examples. The virtual sample (*vs*) could be created based on different MFs in the two attributes, as shown in Figure 11. Then, the *vs* was randomly generated within [*a*,*b*] to calculate the possibility MF(*vs*). When the MF(*vs*) was greater than *rs*~Uniform(0,1), then *vs* could be regarded as an appropriate virtual sample.



(a) MF in MDR attribute

(b) MF in Rb attribute

Figure 11. The selection of the *vs*.

After performing the PAM, the virtual samples could be generated, as shown in Table 6. The virtual sample output was derived using the built BPNN model with small datasets. Then, we added the generated virtual examples into the original training dataset to build a new training dataset.

Table 6. The virtual samples using DB-MTD method.

NO.	MDR	Topo II	Rb	EGFR	Neu	c-ErbB-3	c-ErbB-4	Cyclin A	Cyclin D1	P16	Cdc 2	Bcl-2	Bax	Co-60
1	0.336	0.631	0.032	0.104	0.600	0.201	0.186	0.889	0.127	0.05	0.223	0.696	0.407	0.425
2	0.862	0.075	0.806	0.278	0.329	0.109	0.837	0.581	0.353	0.113	0.145	0.529	0.305	0.052
3	0.966	0.141	0.231	0.381	0.534	0.338	0.169	0.377	0.57	0.171	0.774	0.171	0.444	0.165
4	0.937	0.656	0.256	0.715	0.047	0.561	0.314	0.727	0.108	0.700	0.483	0.488	0.024	0.477
5	0.515	0.514	0.818	0.754	0.882	0.912	0.647	0.178	0.037	0.39	0.904	0.554	0.224	0.012
6	0.129	0.147	0.139	0.685	0.486	0.576	0.485	0.646	0.148	0.782	0.04	0.871	0.491	0.724
7	0.468	0.595	0.631	0.289	0.631	0.877	0.581	0.059	0.185	0.721	0.513	0.873	0.035	0.347
8	0.882	0.648	0.877	0.774	0.722	0.734	0.124	0.863	0.260	0.459	0.865	0.646	0.175	0.173
9	0.795	0.467	0.528	0.189	0.775	0.433	0.295	0.246	0.778	0.442	0.031	0.076	0.615	0.466
10	0.208	0.173	0.741	0.615	0.927	0.338	0.136	0.762	0.717	0.629	0.439	0.899	0.32	0.292

4.3. Experimental procedure

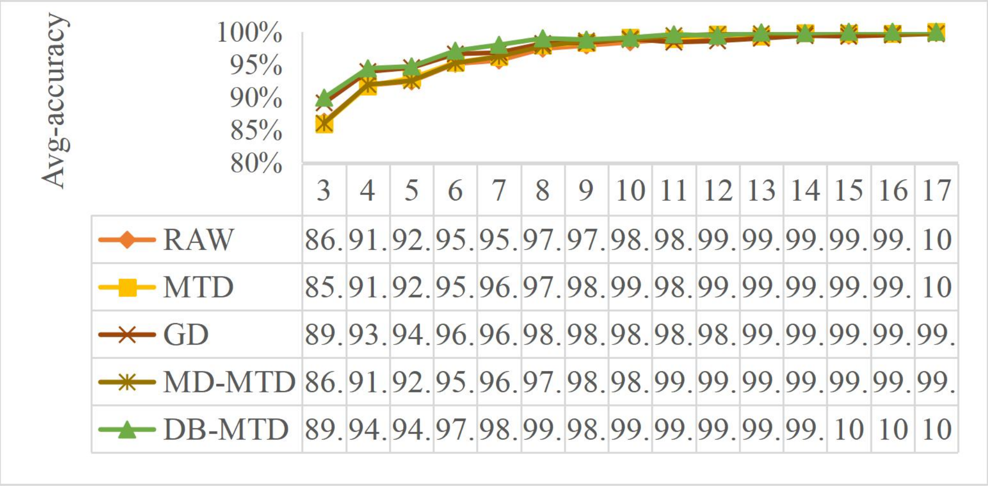
In our experiment, we used the random sampling method to create different scenarios for small sample sizes. The different training datasets were created by sampling from the original datasets with the BC dataset ranging in size from 3 to 17 and training data sizes {5,10,15,20,25,30,35} for the RBC dataset, and the remaining data was set as a testing dataset. Then, we used Eqs (23) and (24) to calculate the lower and upper bounds of the attributes of the training dataset. The virtual sample could be randomly generated from the evaluated range. In addition, we calculated the MF values of the virtual samples to select virtual samples using the PAM. The virtual data size was set as 100. The *f*-*k*-*l* BPNN network was set as a learning model in the experiment, where *f* represents *f* attributes of the training dataset as the number of nodes in the input layer; *k* is the number of nodes in the hidden layer, and *l* is the number of nodes in the output layer. The testing dataset was inputted into the learned BPNN model with 100 nodes in the hidden layer and 50 epochs in the training process. In the BC dataset, we used classification accuracy as the evaluation metric for the BPNN model. In the RBC dataset, the prediction performance of the BPNN model was evaluated using the root mean squared error (RMSE), which was defined as:

$$RMSE = \sqrt{\frac{1}{test_N}\sum_{i=1}^{test_N}(y_i - \hat{y}_i)^2} \tag{32}$$

where $test_N$ is the number of testing samples; y_i is the actual value, and $\hat{y}_i$ is the predicted values of the *i*th testing sample. A total of 100 iterations were used in this experiment. After performing the experiments, we calculated the average and standard deviation of the classification accuracy for the BC dataset and the average and standard deviation of the RMSE for the RBC dataset.

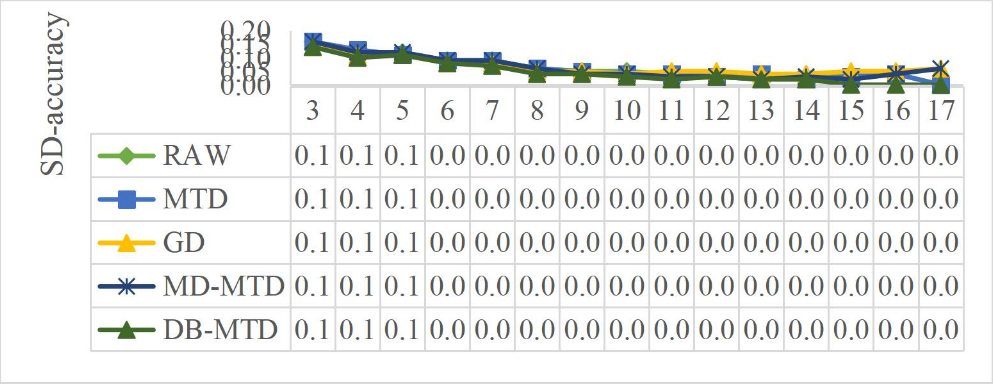
4.4. Experimental results

To verify the effectiveness of the proposed DB-MTD method, we compared it using four methods: RAW (using raw training samples), MTD (generating virtual samples based on the triangular MF), GD (generating virtual samples base on a normal distribution), and MD-MTD (generating virtual samples based on multiple distributions). For the BC dataset, the average (Avg-accuracy) and the standard deviation (SD-accuracy) of the classification accuracies are respectively shown in Figures 12 and 13. The improvements (%) in classification accuracy using the proposed DB-MTD method are listed in Table 7. In Figure 12, it can be seen that the results for most of the training subsets indicated superior classification accuracies as compared to those for the other methods under consideration. The SD-accuracy in these methods had smaller variances that were close to each other with different training data sizes. For example, when the training sample size was 3, the Avg-accuracy using the DB-MTD method was improved from 86.2 to 89.9%, where the improvement was approximately 3.7%, as shown in Table 7. For the RBC dataset, the average (Avg-RMSE) and the standard deviation (SD-RMSE) of RMSE are respectively shown in Figures 14 and 15. In Figure 14, the Avg-RMSE using DB-MTD method was improved from 24.79 to 24.47 at a training sample size of 5, where the improvement was approximately 0.32. The other improvements are shown in Table 8.



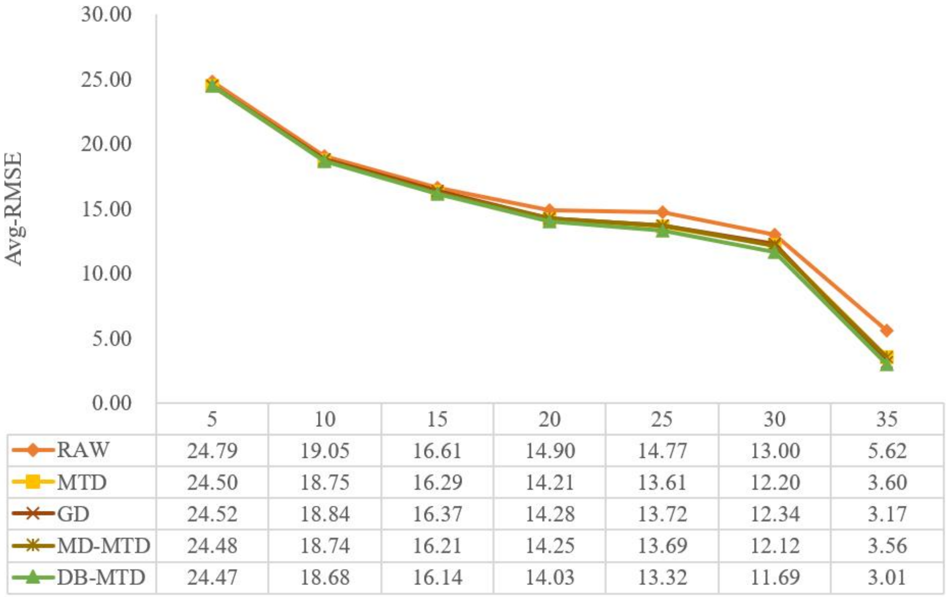
	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
RAW	86.	91.	92.	95.	95.	97.	97.	98.	98.	99.	99.	99.	99.	99.	10
MTD	85.	91.	92.	95.	96.	97.	98.	99.	98.	99.	99.	99.	99.	99.	10
GD	89.	93.	94.	96.	96.	98.	98.	98.	98.	98.	99.	99.	99.	99.	99.
MD-MTD	86.	91.	92.	95.	96.	97.	98.	98.	99.	99.	99.	99.	99.	99.	99.
DB-MTD	89.	94.	94.	97.	98.	99.	98.	99.	99.	99.	99.	99.	10	10	10

Figure 12. The Avg-accuracy for the BC dataset.



	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
RAW	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MTD	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
GD	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MD-MTD	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DB-MTD	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 13. The SD-accuracy for the BC dataset.

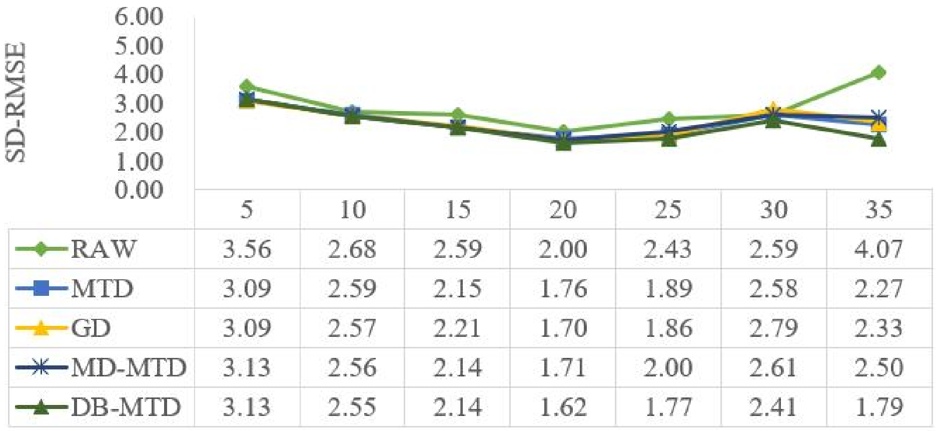


	5	10	15	20	25	30	35
RAW	24.79	19.05	16.61	14.90	14.77	13.00	5.62
MTD	24.50	18.75	16.29	14.21	13.61	12.20	3.60
GD	24.52	18.84	16.37	14.28	13.72	12.34	3.17
MD-MTD	24.48	18.74	16.21	14.25	13.69	12.12	3.56
DB-MTD	24.47	18.68	16.14	14.03	13.32	11.69	3.01

Figure 14. The Avg-RMSE for the RBC dataset.

Table 7. The improvement (%) in the classification accuracy using the proposed method.

Training data size	RAW vs DB-MTD	MTD vs DB-MTD	GD vs DB-MTD	MD-MTD vs DB-MTD
3	3.70	4.00	0.80	3.90
4	2.60	2.70	0.50	2.50
5	2.30	1.80	0.20	2.20
6	2.00	1.80	0.50	1.90
7	2.40	1.80	1.20	1.80
8	1.60	1.10	0.80	1.20
9	0.90	0.50	0.50	0.20
10	0.70	0.00	0.20	0.30
11	0.70	0.70	1.20	0.20
12	0.30	-0.20	0.80	-0.20
13	0.60	0.60	0.90	0.20
14	0.00	0.00	0.40	0.10
15	0.60	0.30	0.70	0.10
16	0.30	0.30	0.50	0.30
17	0.00	0.00	0.30	0.30



	5	10	15	20	25	30	35
RAW	3.56	2.68	2.59	2.00	2.43	2.59	4.07
MTD	3.09	2.59	2.15	1.76	1.89	2.58	2.27
GD	3.09	2.57	2.21	1.70	1.86	2.79	2.33
MD-MTD	3.13	2.56	2.14	1.71	2.00	2.61	2.50
DB-MTD	3.13	2.55	2.14	1.62	1.77	2.41	1.79

Figure 15. The SD-RMSE for the RBC dataset.

Table 8. The improvement in the RMSE using the proposed method.

Training data size	RAW vs DB-MTD	MTD vs DB-MTD	GD vs DB-MTD	MD-MTD vs DB-MTD
5	0.32	0.03	0.05	0.01
10	0.37	0.07	0.16	0.06
15	0.47	0.15	0.23	0.07
20	0.87	0.18	0.25	0.22
25	1.45	0.29	0.40	0.37
30	1.31	0.51	0.65	0.43
35	2.61	0.59	0.16	0.55

4.5. The statistical test confirming the experimental results

A paired t-test was used to verify the significance of the learning improvements using the proposed DB-MTD method in BC and RBC datasets. The *p*-values of the paired t-test for the four methods and the DB-MTD method are listed in Tables 9 and 10, where the symbol "*" indicates that the prediction results demonstrated a statistically significant difference (*p*-value < 0.05). In these tables, the results show statistical significance in terms of the difference in the Avg-accuracy for the BC dataset and the Avg-RMSE for the RBC dataset in some training data sizes. For example, when the training data size was set at 10 in the BC dataset, "0.02*" represents that the prediction results based on the DB-MTD demonstrated a significant difference in learning performance for the RAW method. Although the DB-MTD method was only statistically superior to the other methods in some training sizes, the Avg-accuracy of the DB-MTD method was greater than that when using the methods for BC dataset. It is the same results based on the Avg-RMSE were obtained for the RBC dataset, as shown in Table 10.

Table 9. The *p*-values of the paired t-test for the BC dataset on classification accuracy.

Training data size	RAW vs DB-MTD	MTD vs DB-MTD	GD vs DB-MTD	MD-MTD vs DB-MTD
3	0.06	0.87	0.69	0.66
4	0.00*	0.00*	0.01*	0.00*
5	0.91	0.73	0.31	0.79
6	0.00*	0.00*	0.01*	0.00*
7	0.00*	0.00*	0.00*	0.00*
8	0.00*	0.00*	0.00*	0.00*
9	0.01*	0.10	0.06	0.56
10	0.02*	1.00	0.44	0.32
11	0.00*	0.00*	0.00*	0.23
12	0.16	0.47	0.00*	0.62
13	0.03*	0.02*	0.00*	0.42
14	0.66	1.00	0.10	0.42
15	0.03*	0.08	0.01*	0.32
16	0.16	0.16	0.08	0.16
17	Not applicable	Not applicable	0.32	0.32

Table 10. The *p*-values of the paired t-test for the RBC dataset on RMSE.

Training data size	RAW vs DB-MTD	MTD vs DB-MTD	GD vs DB-MTD	MD-MTD vs DB-MTD
5	0.11	0.68	0.40	0.98
10	0.00*	0.07	0.00*	0.04*
15	0.00*	0.00*	0.00*	0.11
20	0.00*	0.01*	0.00*	0.00*
25	0.00*	0.00*	0.00*	0.00*
30	0.00*	0.00*	0.00*	0.00*
35	0.00*	0.01*	0.44	0.02*

5. Conclusions

In many fields, the virtual sample generation (VSG) approach has been regarded as an effective method to improve the learning performance of machine learning models with small sample sizes. The popular MTD method has been widely applied in many VSG studies to generate virtual samples within the estimated data range to extend the amount of the original training data. The problem with the MTD method is that extreme values have serious effects, including making it difficult to estimate the central location and data skewness. To deal with this problem, a new distance-based MTD (DB-MTD) method based on a defined distance function between data was proposed in the present work to improve the degree of data diffusion. The distance became the basis of the coefficient of skewness, which made the inferred distribution close to the pattern in the existing samples. The proposed DB-MTD coefficient could more effectively reduce the excessive diffusion problem that existed in the original MTD method based on a premise where only the triangular membership function was considered.

As to the limitations of this method, the proposed DB-MTD method can be used for small data with continuous variables, but it is not suitable for categorical or discrete variables. In our future research, two research directions can be considered: One is finding discrete density functions to create virtual samples for categorical or discrete attributes. The other is validating the proposed method on practical medical datasets.

Acknowledgments

This study is supported by the Ministry of Science and Technology and National Taipei University of Nursing and Health Sciences, Taiwan. This research was funded by the Ministry of Science and Technology Taiwan under MOST 110-2222-E-227-001-MY2.

Conflict of interest

The authors declare that there are no conflicts of interest.

References

1. P. Gontero, A. Tizzani, G. H. Muir, E. Caldarera, M. Pavone Macaluso, The genetic alterations in the oncogenic pathway of transitional cell carcinoma of the bladder and its prognostic value, *Urol. Res.*, **29** (2001), 377–387. https://doi.org/10.1007/s002400100216
2. V. Tut, K. Braithwaite, B. Angus, D. Neal, J. Lunec, J. Mellon, Cyclin D1 expression in transitional cell carcinoma of the bladder: correlation with p53, waf1, pRb and Ki67, *Br. J. Cancer*, **84** (2001), 270–275. https://doi.org/10.1054/bjoc.2000.1557
3. A. Colquhoun, S. Sundar, P. Rajjayabun, T. Griffiths, R. Symonds, J. Mellon, Epidermal growth factor receptor status predicts local response to radical radiotherapy in muscle-invasive bladder cancer, *Clin. Oncol.*, **18** (2006), 702–709. https://doi.org/10.1016/j.clon.2006.08.003

4. P. Luukka, Similarity classifier in diagnosis of bladder cancer, *Comput. Methods Programs Biomed.*, **89** (2008), 43–49. https://doi.org/10.1016/j.cmpb.2007.10.001
5. G. Y. Chao, T. I. Tsai, T. J. Lu, H. C. Hsu, B. Y. Bao, W. Y. Wu, et al, A new approach to prediction of radiotherapy of bladder cancer cells in small dataset analysis, *Expert Syst. Appl.*, **38** (2011), 7963–7969. https://doi.org/10.1016/j.eswa.2010.12.035
6. T. W. Liao, Diagnosis of bladder cancers with small sample size via feature selection, *Expert Syst. Appl.*, **38** (2011), 4649–4654. https://doi.org/10.1016/j.eswa.2010.09.135
7. T. I. Tsai, Y. Zhang, Z. Zhang, G. Y. Chao, C. C. Tsai, Considering relationship of proteins for radiotherapy prognosis of bladder cancer cells in small data set, *Methods Inf. Med.*, **57** (2018), 220–229. https://doi.org/10.3414/ME17-02-0003
8. M. D. Robinson, G. K. Smyth, Small-sample estimation of negative binomial dispersion, with applications to SAGE data, *Biostatistics*, **9** (2008), 321–332. https://doi.org/10.1093/biostatistics/kxm030
9. S. Lee, M. J. Emond, M. J. Bamshad, K. C. Barnes, M. J. Rieder, D. A. Nickerson, et al., Optimal unified approach for rare-variant association testing with application to small-sample case-control whole-exome sequencing studies, *Am. J. Hum. Genet.*, **91** (2012), 224–237. https://doi.org/10.1016/j.ajhg.2012.06.007
10. Y. Zhao, N. J. Fesharaki, H. Liu, J. Luo, Using data-driven sublanguage pattern mining to induce knowledge models: application in medical image reports knowledge representation, *BMC Med. Inf. Decis. Making*, **18** (2018), 1–13. https://doi.org/10.1186/s12911-018-0645-3
11. L. Stainier, A. Leygue, M. Ortiz, Model-free data-driven methods in mechanics: material data identification and solvers, *Comput. Mech.*, **64** (2019), 381–393. https://doi.org/10.1007/s00466-019-01731-1
12. E. Ntoutsi, P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl, M. E. Vidal, et al., Bias in data-driven artificial intelligence systems—An introductory survey, *Wiley Interdiscip. Rev.: Data Min. Knowl. Discovery*, **10** (2020), e1356. https://doi.org/10.1002/widm.1356
13. T. Mao, L. Yu, Y. Zhang, L. Zhou, Modified Mahalanobis-Taguchi System based on proper orthogonal decomposition for high-dimensional-small-sample-size data classification, *Math. Biosci. Eng.*, **18** (2020), 426–444. https://doi.org/10.3934/mbe.2021023
14. I. Izonin, R. Tkachenko, I. Dronyuk, P. Tkachenko, M. Gregus, M. Rashkevych, Predictive modeling based on small data in clinical medicine: RBF-based additive input-doubling method, *Math. Biosci. Eng.*, **18** (2021), 2599–2613. https://doi.org/ 10.3934/mbe.2020392
15. Y. Liu, Y. Zhou, X. Liu, F. Dong, C. Wang, Z. Wang, Wasserstein GAN-based small-sample augmentation for new-generation artificial intelligence: a case study of cancer-staging data in biology, *Engineering*, **5** (2019), 156–163. https://doi.org/10.1016/ j.eng.2018.11.018
16. H. Han, M. Zhou, Y. Zhang, Can virtual samples solve small sample size problem of KISSME in pedestrian re-identification of smart transportation?, *IEEE Trans. Intell. Transp. Syst.*, **21** (2020), 3766–3776. https://doi.org/10.1109/TITS.2019.2933509
17. Z. Liu, Y. Li, Small data-driven modeling of forming force in single point incremental forming using neural networks, *Eng. Comput.*, **36** (2020), 1589–1597. https://doi.org/10.1007/s00366-019-00781-6

18. Q. X. Zhu, Z. S. Chen, X. H. Zhang, A. Rajabifard, Y. Xu, Y. Q. Chen, Dealing with small sample size problems in process industry using virtual sample generation: a Kriging-based approach, *Soft Comput.*, **24** (2020), 6889–6902. https://doi.org/10.1007/s00500-019-04326-3
19. 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
20. B. Efron, R. LePage, *Introduction to Bootstrap*, Wiley & Sons, New York, 1992.
21. S. Lee, A. Ahmad, G. Jeon, Combining bootstrap aggregation with support vector regression for small blood pressure measurement, *J. Med. Syst.*, **42** (2018), 1–7. https://doi.org/10.1007/s10916-018-0913-x
22. M. F. Ijaz, M. Attique, Y. Son, Data-driven cervical cancer prediction model with outlier detection and over-sampling methods, *Sensors*, **20** (2020), 2809. https://doi.org/10.3390/s20102809
23. M. La Rocca, C. Perna, Nonlinear autoregressive sieve bootstrap based on extreme learning machines, *Math. Biosci. Eng.*, **17** (2020), 636–653. https://doi.org/10.3934/ mbe.202003
24. S. Cho, M. Jang, S. Chang, Virtual sample generation using a population of networks, *Neural Process Lett.*, **5** (1997), 21–27. https://doi.org/10.1023/A:1009653706403
25. C. Huang, C. Moraga, A diffusion-neural-network for learning from small samples, *Int. J. Approx. Reasoning*, **35** (2004), 137–161. https://doi.org/10.1016/j.ijar.2003.06.001
26. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., Generative adversarial nets, in *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*, (2014), 2672–2680.
27. X. H. Zhang, Y. Xu, Y. L. He, Q. X. Zhu, Novel manifold learning based virtual sample generation for optimizing soft sensor with small data, *ISA Trans.*, **109** (2021), 229–241. https://doi.org/10.1016/j.isatra.2020.10.006
28. D. C. Li, C. S. Wu, T. I. Tsai, Y. S. 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
29. M. R. Rahimi, H. Karimi, F. Yousefi, Prediction of carbon dioxide diffusivity in biodegradable polymers using diffusion neural network, *Heat Mass Transfer*, **48** (2012), 1357–1365. https://doi.org/10.1007/s00231-012-0982-1
30. A. Majid, S. Ali, M. Iqbal, N. Kausar, Prediction of human breast and colon cancers from imbalanced data using nearest neighbor and support vector machines, *Comput. Methods Programs Biomed.*, **113** (2014), 792–808. https://doi.org/10.1016/j.cmpb.2014.01.001
31. B. Zhu, Z. Chen, L. Yu, A novel mega-trend-diffusion for small sample, *CIESC J.*, **67** (2016), 820–826. https://doi.org/10.11949/j.issn.0438-1157.20151921
32. L. Yu, X. Zhang, Can small sample dataset be used for efficient internet loan credit risk assessment? Evidence from online peer to peer lending, *Finance Res. Lett.*, **38** (2021), 101521. https://doi.org/10.1016/j.frl.2020.101521
33. J. Yang, X. Yu, Z. Q. Xie, J. P. Zhang, A novel virtual sample generation method based on Gaussian distribution, *Knowl. Based. Syst.*, **24** (2011), 740–748. https://doi.org/10.1016/j.knosys.2010.12.010

34. K. Wang, J. Li, F. Tsung, Distribution inference from early-stage stationary data streams by transfer learning, *IISE Trans.*, (2021), 1–25. https://doi.org/10.1080/ 24725854.2021.1875520
35. O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, et al., Missing value estimation methods for DNA microarrays, *Bioinformatics*, **17** (2001), 520–525. https://doi.org/10.1093/bioinformatics/17.6.520
36. G. E. Batista, M. C. Monard, An analysis of four missing data treatment methods for supervised learning, *Appl. Artif. Intell.*, **17** (2003), 519–533. https://doi.org/10.1080/713827181
37. D. V. Nguyen, N. Wang, R. J. Carroll, Evaluation of missing value estimation for microarray data, *Data Sci. J.*, **2** (2004), 347–370. https://doi.org/10.6339/JDS.2004.02(4).170
38. A. Jadhav, D. Pramod, K. Ramanathan, Comparison of performance of data imputation methods for numeric dataset, *Appl. Artif. Intell.*, **33** (2019), 913–933. https://doi.org/10.1080/ 08839514.2019.1637138
39. T. Cover, P. Hart, Nearest neighbor pattern classification, *IEEE Trans. Inf. Theory*, **13** (1967), 21–27. https://doi.org/10.1109/TIT.1967.1053964
40. G. H. Cha, Non-metric similarity ranking for image retrieval, in *International Conference on Database and Expert Systems Applications: Springer*, (2006), 853–862. https://doi.org/ 10.1007/11827405_83