Decision Support Systems 105 (2018) 66–76

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Rebuilding sample distributions for small dataset learning

Der-Chiang Li *, Wu-Kuo Lin, Chien-Chih Chen, Hung-Yu Chen, Liang-Sian Lin

Department of Industrial and Information Management, National Cheng Kung University, University Road, Tainan 70101, Taiwan, ROC

ARTICLE INFO

Article history:
Received 31 March 2017
Received in revised form 28 October 2017
Accepted 29 October 2017
Available online 3 November 2017

Keywords:
Small data
Virtual sample
Data preprocessing

ABSTRACT

Over the past few decades, a few learning algorithms have been proposed to extract knowledge from data. The majority of these algorithms have been developed with the assumption that training sets can denote populations. When the training sets contain only a few properties of their populations, the algorithms may extract minimal and/or biased knowledge for decision makers. This study develops a systematic procedure based on fuzzy theories to create new training sets by rebuilding the possible sample distributions, where the procedure contains new functions that estimate domains and a sample generating method. In this study, two real cases of a leading company in the thin film transistor liquid crystal display (TFT-LCD) industry are examined. Two learning algorithms—a back-propagation neural network and support vector regression—are employed for modeling, and two sample generation approaches—bootstrap aggregating (bagging) and the synthetic minority oversampling technique (SMOTE)—are employed to compare the accuracy of the models. The results indicate that the proposed method outperforms bagging and the SMOTE with the greatest amount of statistical support.

1. Introduction

Over the past few decades, numerous machine learning algorithms have been developed to extract knowledge from data [1]. However, the majority of these algorithms were developed based on the assumption that training sets can represent the properties of populations. Conversely, if the training data contain insufficient information about the populations, the learning algorithms may output less precise results for future events.

1.1. Background

Although issues related to big-data learning have only attracted attention in recent years, issues related to small-data learning were revealed by Student's t-distribution [2] in 1908. The collection of additional samples to enlarge a sample size and ensure that algorithms can perform sufficient learning is sometimes difficult and/or expensive in certain situations, such as the diagnoses of rare diseases [3,4], examination of deoxyribonucleic acid (DNA) microarrays [5], pattern recognition with limited pixels [6,7], development of new products [8], and systems in their initial stages [9]. Methods for effectively learning robust and accurate information from small data is an issue that is worthy of additional research.

To demonstrate how small data affect the learning results of most algorithms, Fig. 1 displays two possible distributions of two small datasets with regard to their populations. In Fig. 1(a), the instances are evenly distributed in a population. Although most learning approaches can extract exact knowledge from a population, only a small amount of information will be obtained. Conversely, in Fig. 1(b), the instances are concentrated in a part of the population. The majority of learning approaches will produce biased outcomes regardless of the data size.

In addition to the sample distribution, another issue that can cause insufficient information to be obtained is the gaps between two observations in small data. As shown in Fig. 2, although the observations are evenly distributed in the population, gaps exist between two observations in a small dataset. These gaps (referred to as information gaps) should be filled with observations in a complete dataset; however, these observations are not available. Most learning algorithms fail to train their patterns with the unavailable instances in the information gaps in small datasets, and therefore, the obtained information is inadequate. For example, most tree-based algorithms, such as the C4.5 decision tree [10], need to partition continuous data into discrete intervals before evaluating the classification purity. However, the expected size of an interval is usually unavailable in small datasets since some intervals that contain no observations are integrated with their nearest intervals. If an insufficient number of candidate positions exist for the purity evaluation, then the trees that are built and the resulting hierarchy of the classification rules will be small.

1.2. Related studies

Virtual sample generation (VSG) methods can be employed to address the learning problem of small data. These methods are a type of data-preprocessing method that is applied in the process of knowledge discovery in databases (KDD) [11]; research has demonstrated their

* Corresponding author.
E-mail address: lidc@mail.ncku.edu.tw (D.-C. Li).

https://doi.org/10.1016/j.dss.2017.10.013
0167-9236/

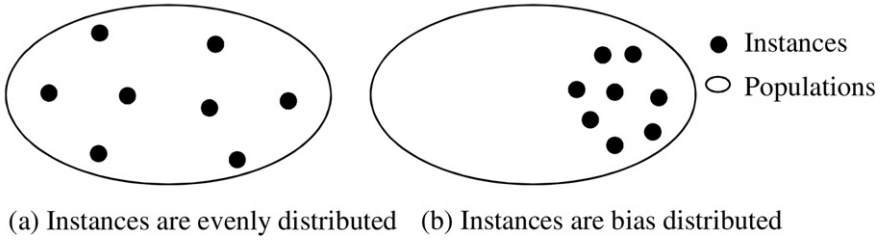

Fig. 1. Two situations in which small data may be distributed relative to the populations.

effectiveness [12]. One of the most extensively applied VSG methods is the bootstrapping procedure (BP) [13], which creates new training sets (referred to as bootstrapping sets) by resampling instances from the original data with a certain probability. The benefit of this approach is that most learning algorithms train a sample at least twice to gradually revise the identified patterns, which enables them to represent the behaviors of the actual data. To overcome the over-fitting issue in training sets, numerous ensemble learning methods were developed, such as bagging [14] and random forests [15], which employ BP to create bootstrapping sets for algorithms to build classifiers and determine classes by voting. Currently, bagging and random forests are extensively applied to extract knowledge from big data since each bootstrapping set can denote one evenly distributed part of a population.

When applying bagging or random forests to learn with small data, the use of bootstrapping sets may create two issues: an unstable data structure and overfitting, as shown in Fig. 3(a) and Fig. 3(b), respectively. A comparison of Fig. 2 with Fig. 3(a) reveals that certain observations in Fig. 2 are missing in Fig. 3(a) because they were not selected with a certain probability when forming the bootstrapping sets. The number of observations is very small, and thus, the difference between the features of the two bootstrapping sets in Fig. 3(a) is large. Since the amount of information provided by small data is minimal, any missing observations in the bootstrapping sets can increase the loss of information. Although we can double the observations to form the bootstrapping sets, as shown in Fig. 3(b), this step usually causes the patterns identified by the algorithms to represent the behaviors of a few observations, which causes overfitting. The amount of information provided by the set in Fig. 3(b) does not increase because the increased information is the same information provided by the same observations.

The synthetic minority over-sampling technique (SMOTE) [16] was proposed to generate artificial samples that differ from the original samples in the minority class. Based on the *k*-nearest neighbors, the SMOTE generates synthetic data along continuous vectors between the minority class's instances and their nearest neighbors, as shown in Fig. 4. Although the information gaps in the minority class are filled with synthetic instances, they are distributed within the domain of the real instances in the minority class. The method employed by the SMOTE to generate samples is simple and does not consider the possible distributions of the entire minority class.

Since the fuzzy theory was proposed by Zadeh [17] in 1965, it has been employed to handle uncertain events. For example, to expand

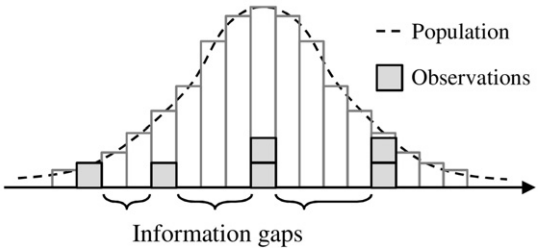

Fig. 2. Distribution of a small dataset and its unknown population.

crisp observations to fill the information gaps caused by a lack of data, Huang [18] proposed the principle of information diffusion, in which a normal diffusion function that is developed based on fuzzy theories is defined as

$$\tilde{f}_n(x) = \frac{1}{nh\sqrt{2\pi}} \sum_{i=1}^{n} \exp\left[-\frac{(x-x_i)^2}{2h^2}\right], \tag{1}$$

where h is the diffusion coefficient and n is the attribute size. In 2004, Huang and Moraga [19] proposed the diffusion neural network (DNN), which derives a pair of artificial samples for each observation based on Eq. (1) to fill the information gaps. Since the parameter n in Eq. (1) are determined to be a constant when a dataset is given, Huang and Moraga [19] applied the part $\exp[-(x-x_i)^2/2h^2]$ to derive the virtual values of an observation (x, y) as $x' = x \pm \sqrt{-2h_x{}^2 \ln\psi(r)}$ and $y' = y \pm \sqrt{-2h_y{}^2 \ln\psi(r)}$ for a two-dimensional dataset, where h are the diffusion coefficients, which are the inductions from a large amount of simulation results in DNN, r is the correlation coefficient of input X and output Y, and $\psi(r)$ is the transforming function, which is defined as

$$\psi(r) = \psi\left(0.9 + m \times 10^{-2}\right) \mapsto 0.9\ldots99_{\ m\ \ 9s} \quad \forall r \in \{0.91,\ 0.92,\ \ldots,\ 0.99\}, \tag{2}$$

mapping r into a possibility value. For example, if r is 0.93 (or 0.96), then m is 3 (or 6) and the possibility $\psi(r)$ is 0.999 (or 0.999999).

Since DNN needs r to be >0.9, the applicability of the DNN method is limited with regard to most practical cases. In addition, the distributions constructed by DNN have information gaps because DNN only considers the behavior of an individual observation rather than the behavior of an entire dataset.

1.3. Motivation

To improve the robustness and/or accuracy of the forecasting models produced by data preprocessing when learning with small data, this study proposes a systematical procedure to create new training sets by rebuilding possible sample distributions. The procedure contains a set of new functions that estimate the possible ranges of observations and a sample generation method that considers the relations among attributes, where the functions and the method are developed based on the fuzzy normal function [18] and fuzzy concepts, respectively.

In the experiments, two learning algorithms—a back-propagation neural network (BPN) and support vector regression (SVR)—are adopted to build models. Two VSG approaches—bagging (using BP) and the SMOTE—are employed to compare the effectiveness of the models. Two real cases of a leading company in the thin film transistor liquid crystal display (TFT-LCD) industry in Taiwan are examined. The results indicate that the proposed method is more effective than bagging and the SMOTE for learning from the two cases with the greatest amount of statistical support.

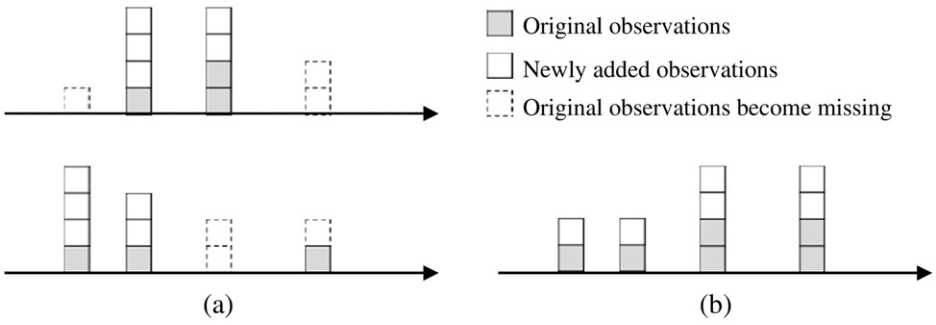

Fig. 3. Bootstrapping sets of small data, where (a) are two possible sets, and (b) is the set for which the observations are doubled.

The remainder of this study is organized as follows. Section 2 introduces the proposed procedure, and Section 3 describes the experimental environment and the background of the two real cases. Section 4 discusses the experimental results, and Section 5 presents the conclusions of this paper.

2. Proposed methodology

Two steps can be employed to enhance the data structures of small data by rebuilding the possible sample distributions: estimating the sample distributions and creating new samples. Before introducing the proposed method, the notations in this work are defined.

2.1. Definitions of notations

Assume that a small dataset with $m-1$ input attributes $\{ X_j | j=1,2,\ldots,m-1 \}$, one output attribute X_m, and n instances, as listed in Table 1. The elements of the attributes are $\{x_{i,j} | i=1,2,\ldots,n \ ; \ j=1,2,\ldots,m \}$, and LB_j, CL_j, UB_j, $\min_j$, and $\max_j$ are the estimated lower bound, center location, upper bound, minimum, and maximum values of $\{ X_j | j=1,2,\ldots,m \}$, respectively. In this section, these symbols will be used in the equations, figures, and tables.

2.2. Estimating sample distributions

In fuzzy theories, three major types of membership functions (MFs) are commonly applied to denote sample distributions, including Gaussian, trapezoidal and triangular MFs. However, the Gaussian MF restricts the distributions to symmetrical distributions, whereas they may also be asymmetrical in reality. In previous studies [20,21] the comparison of triangular MFs and trapezoidal MFs indicated that the triangular MF is preferred and the trapezoidal MF will increase the computational complexity. Therefore, this study adopts the triangular MFs to denote the possible sample distributions of small data. Three steps are taken to complete this task: determining the location centers, deriving the domain bounds, and building the sample distributions with triangular MFs.

2.2.1. Determining location centers

In contrast with DNN, in which virtual values are created on both sides of an individual observation, this study estimates the possible distribution of observations. We need to determine the center location (CL) of a distribution to denote the position of the height in a triangular MF. Three possible candidates are the mode (Mo), the mean, and the median (Me). Mo is not applicable in this study because it does not usually exist in small data, with the exception of designed experiments. The mean is not considered as a suitable CL for small data since it is more likely to be affected by an extreme outlier than Mo and Me when sample sizes are very small, as shown by the example in Fig. 5, in which a box plot is drawn to scale the observations with only one observation on the right side of the mean. Although taking the mean or Me as CL does not make a difference, since their deviation is small when no observations are identified as an outlier, taking the mean as CL would probably increase the risk of making one observation represent a half distribution when an outlier exists. Therefore, Me is more suitable than Mo as the mean to be applied to small datasets as the CL in this study and is calculated by

$$CL_j = \begin{cases} \dfrac{x'_{\frac{n}{2},j} + x'_{\frac{n}{2}+1,j}}{2}, & \text{if } n = 2z, \\ x'_{\frac{n+1}{2},j}, & \text{if } n = 2z+1, \end{cases} \tag{3}$$

where $\forall z \in \mathrm{N}$, $\{ x'_{i,j} | \ i=1,2,\ldots,n; \ j=1,2,\ldots,m \}$ are the sorted values of X_j, and n is the sample size.

2.2.2. Deriving domain bounds

In contrast to a DNN, in which the part $\exp[-(x-x_i)^2/2h^2] \in (0, \ 1]$ in Eq. (1) is used to derive the virtual values of an observation, this research applies this expression to estimate the possible domain of an attribute. Huang and Moraga [19] considered that $|r| = \exp[-(x-x_i)^2/2h^2]$, where r is the correlation coefficient between input X and output Y was reasonable. To satisfy the network structures to make the learning procedure converge, they decided to set $\psi(|r|) = \exp[-(x-x_i)^2/2h^2]$, where $\psi(|r|)$ is a transforming function that maps $|r|$ into a possibility value, as formulated in Eq. (2). We redefine this part as

$$\varphi(|r_{p,q}|) = \exp\left[-\frac{(CL_p - B_p)^2}{2\hat{s}_p^{\,2}}\right] \tag{4}$$

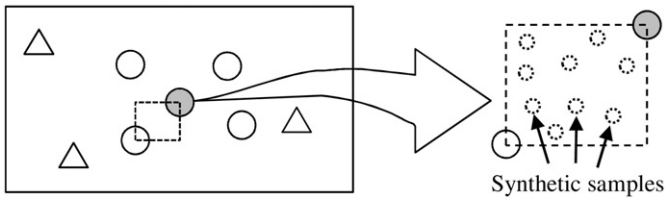

Fig. 4. Sample generating mechanism of SMOTE.

Table 1
Small dataset.

No. of instances	Inputs					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	$x_{n,1}$	…	$x_{n,j}$	…	$x_{n,m-1}$	$x_{n,m}$

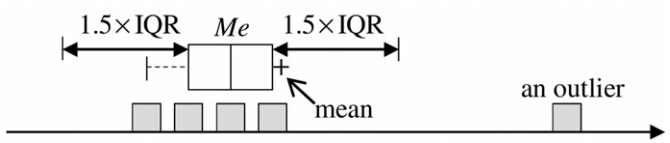

Fig. 5. Box plot to scale a small dataset that has an outlier.

Fig. 6. The situation that $[B_p^L, B_p^U]$ fails to cover $[\min_p, \max_p]$.

to derive the sample boundaries B_p of X_p, where CL_p is the center location of X_p, $r_{p,q}$ is the relation between X_p and X_q, $p,q \in \{1,2,\ldots,m\}$ and $p \neq q$, $\hat{s}_p$ is the sample standard deviation of X_q, and $\varphi(\cdot)$ is the transforming function defined as Eq. (5). Since $\psi(\cdot)$ only considers the situation in which $|r|$ is >0.9, we define Eq. (5)—a modified sigmoid function—to make $\varphi(r)$ between (0, 1) when the relation $r \in [-1, 1]$ is given. Similar to how $\psi(\cdot)$ acts in a DNN, $\varphi(\cdot)$ makes the diffusion width narrower when r is larger; conversely, this width is wider when r is smaller.

$$\varphi(r) = \frac{1}{1 + \exp(-10(|r| - 0.5))} \tag{5}$$

However, before introducing how to derive sample boundaries, we need to identify suitable indicators to be employed in Eq. (4) when working with small datasets. Eq. (4) contains two indicators: the correlation coefficient $r_{p,q}$ and the sample standard deviation $\hat{s}_p$.

The correlation coefficient $r_{p,q}$ is not suitable for measuring the relations in small datasets since it is computed by taking two means ($\overline{X}_p$and $\overline{X}_q$) as the bases, which increases the risk that the results may be affected by outliers. Therefore, this study adopts the indicator trend similarity between attributes (TSA) [22] to replace r. The TSA was developed based on a non-parametric method to measure the relation (occurring trend) between two attributes in small data without considering the distances between observations and their *CL*s to avoid the effect of any potential outlier on the results. The similar trend $g(i)_{p,q}$ of the i^{th} instance between X_p and X_q is computed as

$$g(i)_{p,q} = \begin{cases} 1, & \text{if } (x_{i,p} - CL_p)(x_{i,q} - CL_q) > 0 \\ 0, & \text{if } (x_{i,p} - CL_p)(x_{i,q} - CL_q) = 0 \\ -1, & \text{if } (x_{i,p} - CL_p)(x_{i,q} - CL_q) < 0 \end{cases}, \tag{6}$$

where $i = 1,2,\ldots n$, $p,q \in \{1,2,\ldots,m\}$ and $p \neq q$; and the similar trend $S_{p,q} \in [-1, 1]$ between X_p and X_q is obtained by

$$S_{p,q} = \frac{1}{n}\sum_{i=1}^{n} g(i)_{p,q}, \quad p,q \in \{1,2,\ldots,m; p \neq q\}. \tag{7}$$

The sample standard deviation $\hat{s}_p = \sqrt{\sum_{i=1}^{n}(x_{i,p} - \overline{x}_p)^2/n-1}$ is the indicator that measures the average of the Euclidean distances from the mean $\overline{x}_p$ to the observations $x_{\cdot,p}$. $\hat{s}_p$ is also considered to be unsuitable for assessing the degree of dispersion in small datasets because it takes $\overline{x}_p$ as the data center, which is very likely to be affected by outliers. This research employs the Euclidean distances to evaluate the degree of dispersion from the observations to *Me*. The new indicator d_p for X_p is then calculated by

$$d_p = \sqrt[u]{\sum_{i=1}^{n}(x_{i,j} - CL_j)^u / n}, \tag{8}$$

where $p = 1,2,\ldots,m$, $u = 1,2,\ldots$, and u is set to two in this study. Therefore, the final equation for deriving the boundaries of an attribute is formulated as

$$\varphi(S_{p,q}) = \exp\left[-\frac{(CL_p - B_p)^2}{2d_p{}^2}\right], \tag{9}$$

where $S_{p,q} \in [-1, 1]$ is the similar trend between X_p and X_q; CL_p and B_p are the center location (the median) and the bound values of X_p, respectively; $p,q \in \{1,2,\ldots,m\}$; and $p \neq q$. We can derive the bound values as

$$B_p = CL_p \pm d_p\sqrt{-2\ln(\varphi(S_{p,q}))}. \tag{10}$$

When considering the effect of the locations of observations relative to CL_p on the distribution skewness, B_p is redefined as

$$B_p^L = CL_p - \min\left\{d_p^L\sqrt{-2\ln(\varphi(S_{p,q}))} \mid q = 1,2,\ldots,m, \ q \neq p\right\} \tag{11}$$

$$B_p^U = CL_p + \min\left\{d_p^U\sqrt{-2\ln(\varphi(S_{p,q}))} \mid q = 1,2,\ldots,m, \ q \neq p\right\}, \tag{12}$$

where d_p^L and d_p^U are the averaged Euclidean distances computed by the observations, whose values are smaller than CL_p and larger than CL_p, respectively. The reason for taking the minimum diffusion width in Eq. (11) and Eq. (12) is to obtain the intersection when working with multiple attributes. When considering the situation in which the boundary values do not cover the current value domain $[\min_p, \max_p]$ in X_p, as shown in Fig. 6, the lower bound LB_p and the upper bound UB_p are determined as

$$LB_p = \begin{cases} B_p^L, & \text{if } B_p^L \leq \min_p \\ \min_p, & \text{if } B_p^L > \min_p \end{cases} \tag{13}$$

$$UB_p = \begin{cases} B_p^U, & \text{if } B_p^U \geq \max_p \\ \max_p, & \text{if } B_p^U < \max_p \end{cases}, \tag{14}$$

respectively.

2.2.3. Building sample distributions

When the three parameters LB_j, CL_j, and UB_j of X_j are obtained, the process continues to construct a fuzzy triangular MF to denote the possible sample distribution, as shown in Fig. 7, where $j = 1,2,\ldots,m$. The MF function $\mathrm{MF}_j(x)$ is defined as

$$\mathrm{MF}_j(x) = \begin{cases} (x - LB_j)/(CL_j - LB_j), & \text{if } x \leq CL_j, \\ (UB_j - x)/(UB_j - CL_j), & \text{if } x > CL_j, \\ 0, & \text{if } x < LB_j \text{ or } x > UB_j \end{cases} \tag{15}$$

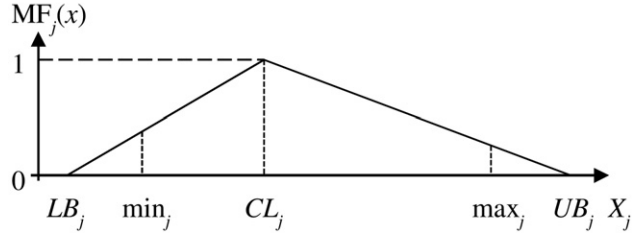

Fig. 7. Sample distribution with a triangular MF.

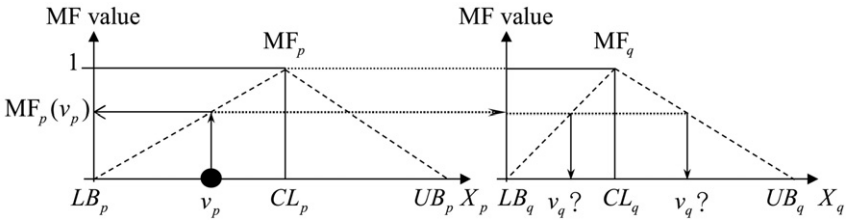

Fig. 8. Obtaining two possible v_q by projecting $MF_p(v_p)$ onto MF_q.

2.3. Generating samples

This subsection introduces how to generate the corresponding synthetic values of an attribute when the other attribute values are given, how to create the given attribute values, and how to handle data with three or more dimensions.

2.3.1. Generating corresponding attribute values

To create samples with consideration of the relations among attributes, a procedure is developed based on the MFs and TSA. To explain the procedure, an example is shown in Fig. 7, in which MF_p and MF_q denote the two sample distributions of X_p and X_q, respectively; $p,q \in \{1,2,\ldots,m\}$; and $p \neq q$. By projecting the $MF_p(v_p)$ of a virtual value v_p onto MF_q, we can infer two possible virtual values v_q in X_q. The sign of $S_{p,q}$ indicates the corresponding v_q when v_p is given. Taking Fig. 8 as an example, if $S_{p,q}$ is positive, we choose the left v_q; otherwise, we select the right v_q. The equation to obtain v_q is formulated as

$$v_q = \begin{cases} LB_q + \mathrm{MF}_p(v_p)(CL_q - LB_q), & LB_q \le v_q \le CL_q \\ UB_q - \mathrm{MF}_p(v_p)(U_q - CL_q) & , CL_q < v_q \le UB_q \\ 0 & , \text{otherwise} \end{cases} \tag{16}$$

However, slight deviations between two real world values in different instances always exist, where the deviations are probably not measured due to the preciseness of the scales. Therefore, the strategy employed by this study is to project one value interval instead of one exact value, as in the two examples shown in Fig. 9 and Fig. 10. The interval bounds $[v_p^-, v_p^+]$ of v_p are obtained by

$$v_p^- = \begin{cases} v_p - \theta_{p,q}(UB_p - LB_p), & \text{if } v_p - \theta_{p,q}(UB_p - LB_p) \ge LB_p \\ LB_p, & \text{if } v_p - \theta_{p,q}(UB_p - LB_p) < LB_p \end{cases} \tag{17}$$

$$v_p^+ = \begin{cases} v_p + \theta_{p,q}(UB_p - LB_p), & \text{if } v_p + \theta_{p,q}(UB_p - LB_p) \le UB_p \\ UB_p, & \text{if } v_p + \theta_{p,q}(UB_p - LB_p) > UB_p \end{cases}, \tag{18}$$

where $UB_p - LB_p$ is the sample range, and $\theta_{p,q} \in (0,1)$ is the diffusion coefficient, which determines the diffused widths.

In this study, we consider a simple linear relation as an example to transform $|S_{p,q}|$ into $\theta_{p,q}$ as

$$\theta_{p,q} = a \times |S_{p,q}| + b, \tag{19}$$

where a and b are the coefficients to be determined. Apart from Eq. (19), other functions, such as a modified bipolar sigmoid, is applicable here. However, the transformation functions should adhere to the following principles:

1. When $|S_{p,q}|$ is closer to 0, which implies that extracting the occurring trend between X_p and X_q from the observations is difficult, then the range $[v_q^-, v_q^+]$ within which v_q may be located can be very wide.
2. When $|S_{p,q}|$ is closer to 1, the interval width $v_p^+ - v_p^-$ and $v_q^+ - v_q^-$ are narrow, which decreases the probability that v_q may be located on the opposite side relative to CL_q to retain their high relation degree.

If we expect the minimum diffusion coefficient and maximum diffusion coefficient to be 10% (or 1%) and 90% (or 99%), respectively, when $|S_{p,q}|$ are 1 and 0, respectively, then we can derive $a = -0.8$ (or -0.98) and $b = 0.9$ (or 0.99) as the coefficients in Eq. (19).

Note that three situations need to be considered when computing the corresponding interval bounds $[v_q^-, v_q^+]$: the side of the location of v_p that is relative to CL_p, the sign of $S_{p,q}$, and whether v_p^- or v_p^+ are on the opposite side of CL_p (as in the example in Fig. 10). A total of eight conditions can be applied to determine how to compute v_q^- and v_q^+. Regardless of which of the eight conditions occurs, we can obtain two v_q^- and two v_q^+ by solving the equations as $MF_q(v_q^-) = MF_p(v_p^-)$ and $MF_q(v_q^+) = MF_p(v_p^+)$, respectively, and determine the proper v_q^- and v_q^+ according to the three previously mentioned situations. When v_q^- and v_q^+ are obtained, v_q is created by drawing one value from a uniform distribution $[v_q^-, v_q^+]$.

2.3.2. Generating given attribute values

The previous subsection indicated two types of virtual values: the corresponding value v_q of X_q and the given value v_p of X_p. In this study, v_p is generated by the possibility assessment mechanism (PAM). When sample sizes are small, observations located near distribution edges are more likely to be identified as noise (outliers) by learning algorithms than observations located around distribution centers, where the noise would probably reduce the accuracy of any identified patterns. This paper employs PAM to decrease and increase the sizes of the synthetic values near the edges (LB_p and UB_p) and CL_p, respectively, and make the distribution of v_p obey the shape of MF_p. In computer programming, the distribution of v_p will present a uniform distribution $[LB_p, \quad UB_p]$ if we directly take random seeds from a uniform distribution [0, 1] to generate v_p. PAM is a computer programming technique

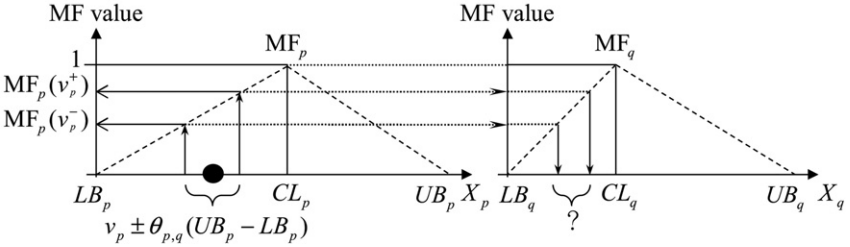

Fig. 9. Projecting $[MF_p(v_p^-), MF_p(v_p^+)]$ onto MF_q to estimate possible ranges of X_q when $S_{p,q} > 0$ and $S_{p,q}$ is large.

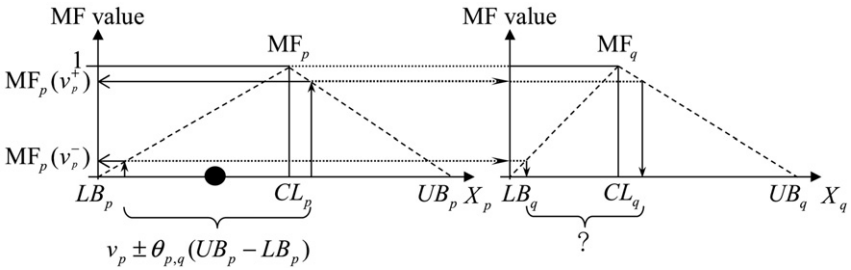

Fig. 10. Projecting $[MF_p(v_p^-), MF_p(v_p^+)]$ onto MF_q to estimate possible ranges of X_q when $S_{p,q} > 0$ and $S_{p,q}$ is small.

that was developed based on the concept of genetic algorithms [23], i.e., the 'survival of the fittest', to assess whether a random value satisfies the criterion for a suitable virtual value. The steps in PAM to create v_p are summarized as follows:

Step 1. Randomly take the value v_p from $[LB_p, \quad UB_p]$, and compute its MF value (i.e., the possibility) as $MF_p(v_p)$ using Eq. (15).
Step 2. Draw a random seed (rs) from a uniform distribution [0, 1] as the threshold value to assess whether the v_p can be reserved. Once $MF_p(v_p) > rs$, then v_p can be treated as a suitable virtual value of X_p; otherwise, v_p should be discarded.
Step 3. Repeat Steps 1 to 2 until a suitable virtual value v_p is available.

The principle of PAM, as shown in Fig. 11, is the probability that an rs is lower than $MF_p(v_p)$ and is actually $MF_p(v_p)$. Therefore, once $MF_p(v_p)$ is a larger value (i.e., v_p is closer to CL_p), v_p will have a higher probability of being a suitable virtual value. When additional v_p are generated using PAM, their distributions will obey the shapes of the estimated sample distributions rather than exhibiting a uniform distribution. Note that PAM does not intend to compare the possibility value in fuzzy theories with the probability value in statistics; however, it examines the probability of a given value (although it is an MF value) in a uniform distribution.

2.3.3. Generating high-dimensional data

When learning two-dimensional data, a virtual sample is created by employing PAM to generate a v_1 of X_1 and taking the v_1 to create its corresponding v_2 of X_2. When working with multiple dimensional data, this process can be continued by taking v_2 to create its corresponding v_3 of X_3 until all virtual values of the m attributes are created to complete one virtual sample. Although the previously mentioned process is simple, it has the potential risk of losing the relations between attributes. We consider the first three attributes X_1, X_2, and X_3 listed in Table 2 as examples. If the previously mentioned process is employed to generate samples by directly taking the sequence $\{X_1, X_2, X_3\}$, the relations between (X_1, X_2) and (X_2, X_3) may be kept in a certain degree but the relation between (X_1, X_3) may be lost, i.e., most v_1 and v_3 become located on the opposite side relative to their CLs since the process does not consider the relation between (X_1, X_3). To reduce the risk, two processes are needed: determining the generating sequence of attributes and applying a learning procedure to retain the relations.

The principles for determining the generating sequence are summarized, with an example given in Table 2.

1. Arrange the attributes whose absolute TSA ($|S|$) values are larger in the anterior parts of the sequences as much as possible, since their diffusion widths are narrower. This step can prevent the loss of additional relations in the beginning stage. In Table 2, since the maximum $|S|$ is |-0.9|, the first attribute can be decided as X_2 or X_3. When we begin searching from X_2, the second attribute is X_3 and the maximum $|S|$ in $\{|S_{3,j}|, j = 1,4,5\}$ are $|S_{3,1}| = 0.7$ and $|S_{3,4}| = 0.7$. From here, the sequence makes a branch. When the searching terminates, two possible sequences can be obtained as $\{X_2, X_3, X_1, X_4, X_5\}$ (marked as $Seq1$) and $\{X_2, X_3, X_4, X_1, X_5\}$ (marked as $Seq2$). When we begin searching from X_3, the sequence is determined as $\{X_3, X_2, X_4, X_1, X_5\}$ (marked as $Seq3$).
2. Select the sequence whose sum of $|S|$ is the largest to prevent the loss of additional total relations. In this example, the sums of $|S|$ in $Seq1$, $Seq2$, and $Seq3$ are (|-0.9| + |0.7| + |0.8| + |0.5|) = 2.9, (|-0.9| + |0.7| + |0.8| + |0.5|) = 2.9, and (|-0.9| + |-0.8| + |0.8| + |0.5|) = 3, respectively. Therefore, $Seq3$ is adopted.
3. Consider the sequence whose $|S|$ values are larger in the anterior parts of sequence once two or more sequences whose sums of $|S|$ are equivalent. Assume that a fake sequence (marked as $Seq4$), whose $|S|$ are $\{-0.9, -0.8, 0.5, 0.8\}$ and its sum of $|S|$ happens to be the same as $Seq3$, is given. $Seq3$ is preferred since its third $|S|$ (0.8) is larger than its third $|S|$ (0.5) in $Seq4$.

However, when attribute sizes are large, the relations of the created samples may present the bullwhip effect [24], i.e., the relations among attributes become biased in the posterior parts in the sequences since their $|S|$ values are smaller (even close to zero) after sequences are arranged. To control this outcome, a learning mechanism is suggested by taking the intersections of the estimated ranges of the prior attributes in the sequence. Taking $Seq3$, for instance, the final diffused range $[v_4^-, v_4^+]$ for the third attribute (X_4) is the intersection of the attributes obtained by treating (X_3, X_4) and (X_2, X_4), the final diffused range for the fourth attribute (X_1) is the final diffused range of the attributes obtained by treating (X_3, X_1), (X_2, X_1), and (X_4, X_1). This mechanism increases the time complexity to $O((m-1)!)$.

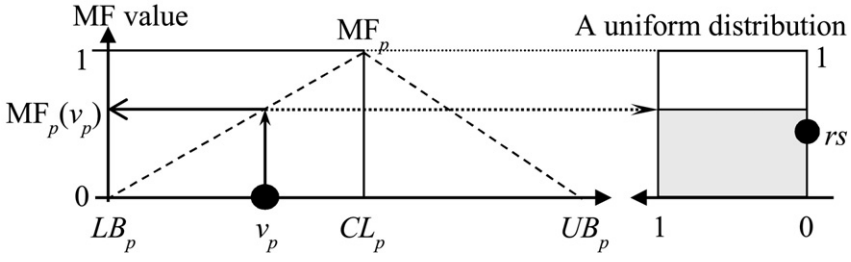

Fig. 11. Principle of the possibility assessment mechanism.

Table 2
Example of a TSA matrix in a high-dimensional dataset.

	X_1	X_2	X_3	X_4	X_5
X_1	–	0	0.7	0.8	0.5
X_2		–	−0.9	−0.8	0.6
X_3			–	0.7	0.4
X_4				–	0.5
X_5					–

3. Experimental environment

This section presents the designs of experiments and a description of two real cases of a TFT-LCD maker in Taiwan.

3.1. Experimental designs

In this study, we adopt a *k*-fold like cross-validation procedure to implement the experiment *k* times to obtain even results, where each cross-validation involves preparing, training, and testing stages, as shown in Fig. 12.

At the preparing stage, *n* instances are randomly drawn from the dataset as the training set, and the remaining *N*-*n* are regarded as the testing set. At the training stage, a control experiment is implemented with one control and three experiment groups. In the control group, the training set is regarded as a given small data set (SDS). In experiment groups 1, 2, and 3, the training set is replaced by a new training set created by BP (in bagging), the SMOTE, and the proposed method (PM), respectively, where the new training sets contain *n* real and *M* artificial instances. The algorithms that were adopted to learn the training sets are BPN and SVR. Consequently, a total of eight models are constructed. At the testing stage, the testing set is used to evaluate the forecasting errors of the eight models, where the errors are indicated by the mean absolute percentage error (MAPE) and are computed by

$$\text{MAPE} = \frac{1}{N-n}\sum_{i=1}^{N-n}\left|\frac{y_i-\hat{y}_i}{y_i}\right|, \tag{20}$$

where y_i is the output value of the i^{th} instance in the testing set, and $\hat{y}_i$ is the prediction of y_i. When the *k* cross-validations with a given training size *n* are completed, the final result is represented by the average of the *k* MAPEs, where the averaged MAPE (AvgMAPE) is computed by

$$\text{AvgMAPE} = \frac{1}{k}\sum_{j=1}^{k}\text{MAPE}_j. \tag{21}$$

To identify whether significant differences exist between group 3 and the other three groups with statistical support, paired *t*-tests with a two-tailed test are performed. The null hypothesis (H_0) and alternative hypothesis (H_a) are formulated as

$$\begin{cases} H_0 : \mu_d = 0 \\ H_a : \mu_d \neq 0 \end{cases}, \tag{22}$$

where μ_d is the average of $\{ d_j | j = 1,2,\ldots,k \}$, and d_j is the deviation between the MAPE of control group 3 and the MAPE of the other three groups in the j^{th} cross-validation run.

In addition, the experiments also contain a sensitivity analysis, in which two parameters, the training sizes *n* and the artificial sample sizes *M*, are designed. The objective of the analysis is to examine the effect of *n* and *M* on the experimental results.

The parameters in the experiments are described as follows: The number of times each experiment is repeated (k) in each *n* is 30, the sensitivity analysis of the training sizes *n* are set depending on the cases, the artificial instance sizes *M* are 100%, 200%, …, and 1000% relative to *n*, and the test level α for the paired *t*-test is 0.05.

The software that was employed to help build the models is Weka 3.80, in which BPN, SVR, and bagging are employed; these models are denoted as "MultilayerPerceptron," "SMOreg," and "bagging," respectively. Weka can also implement the SMOTE after downloading the package from the option "Package manager". The parameters for BPN and SVR in Weka are set as the defaults, where the default kernel of SVR is "PolyKernel". The parameters for bagging and the SMOTE also take the defaults, with the exception of the artificial instance sizes *M*. Although Weka's SMOTE cannot be directly applied to numerical forecasting cases, we can add fake class values to the data to create a minority class that is accepted by Weka's SMOTE.

3.2. Case description

Three major processes are required to make a TFT-LCD panel: the TFT process (or array), the CF process (color filter), and the LCD process (or cell). In the TFT and CF processes, transistors are fabricated on a glass substrate. In the LCD process, the arrayed back substrate and the CF front substrate are assembled, and then the space between these substrates is filled with liquid crystal.

Two real learning tasks are examined in the experiments, as summarized in Table 3. The first task (Case I) in the LCD process is to estimate the biased distances between the TFT substrates and the CF substrates when they are assembled using only a few samples that contain the six measured deviations on the CF substrates. The second task (Case II) in the

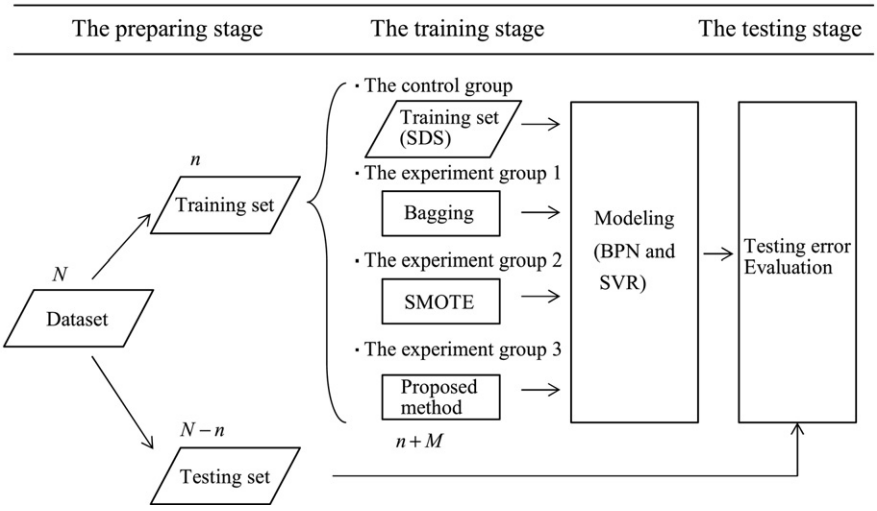

Fig. 12. Experimental design in this study.

Table 3
Two datasets examined in the empirical evaluation.

Case No.	Instance sizes	Attribute sizes	Learning targets
I	19	7	Predicting the shift between two assembled glass substrates in the LCD process using the six scales measured in the CF process.
II	30	4	Estimating the measured photo-spacer height (PSH) on a CF glass substrate when the CF process is completed.

CF process is to infer the height of the photo spacer (PSH), which creates the space between the TFT substrates and the CF substrates to be filled by the liquid crystal when the CF process is completed.

4. Experimental results and discoveries

In the sensitivity analysis, the training size n in Case I is 5, 10, and 15; in Case II, the training size n is 5, 10, 15, and 20. The experimental results are summarized in Fig. 13 and Fig. 14, where "SDS," "bagging," "SMOTE," and "PM" denote the control, experiment 1 group, experiment 2 group and experiment 3 group, respectively; the symbol "-" in "PM" indicates no significant difference between "PM" and "SDS;" the rectangles with solid lines and dotted lines indicate no significant differences between "PM" and "bagging" and "PM" and "SMOTE", respectively; and the values depicted in gray are the minimum AvgMAPEs of "bagging," "SMOTE," and "PM" for the ten artificial sample sizes M.

The findings from Fig. 13 and Fig. 14 are as follows:

1. The averaged errors (AvgMAPEs) of the two models of "SDS" monotonically decrease as the original training size n increases, which reveals that the training size affects the accuracy of the forecasting models when the sample sizes are very small

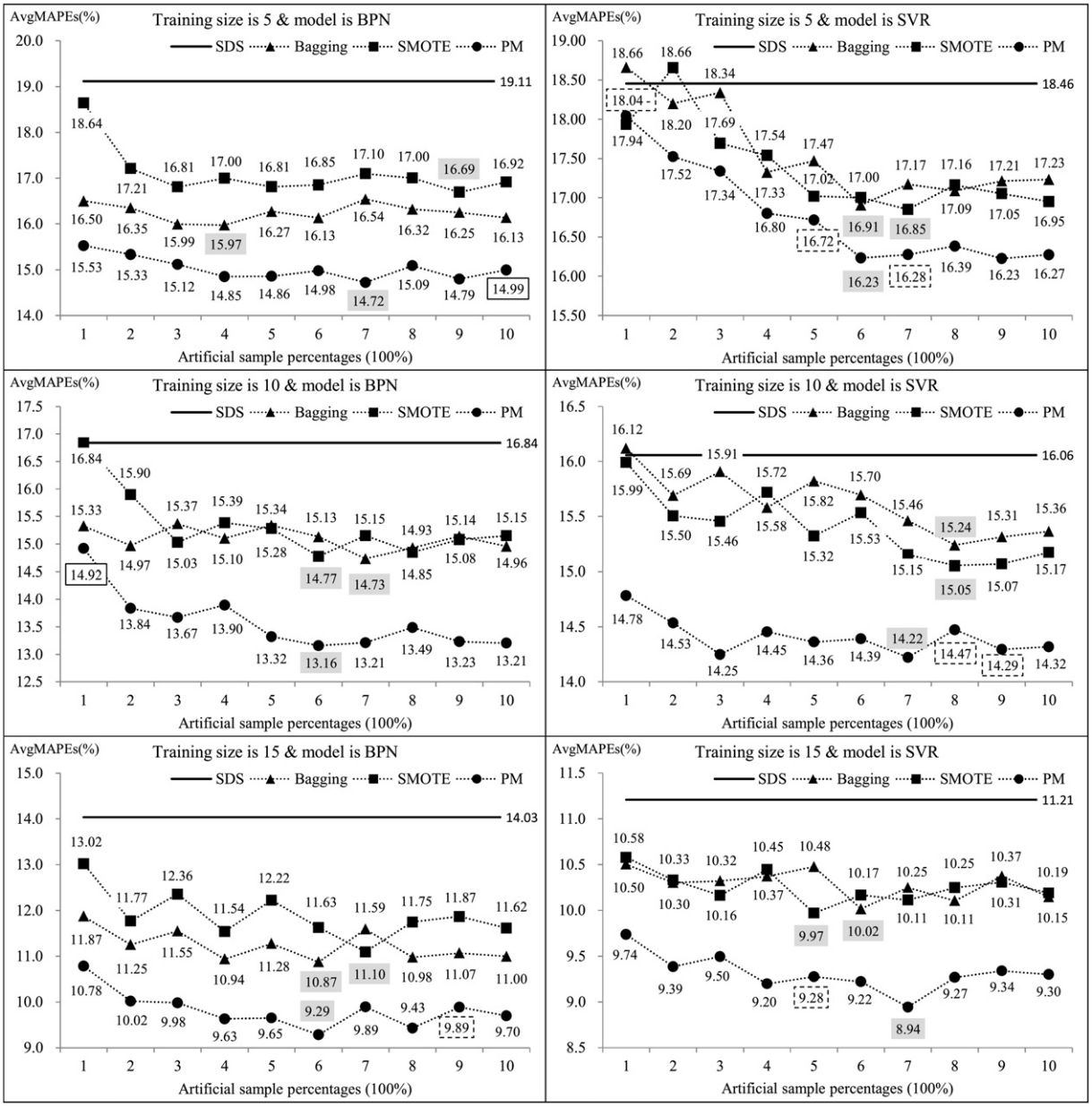

Fig. 13. Experimental results for Case I.

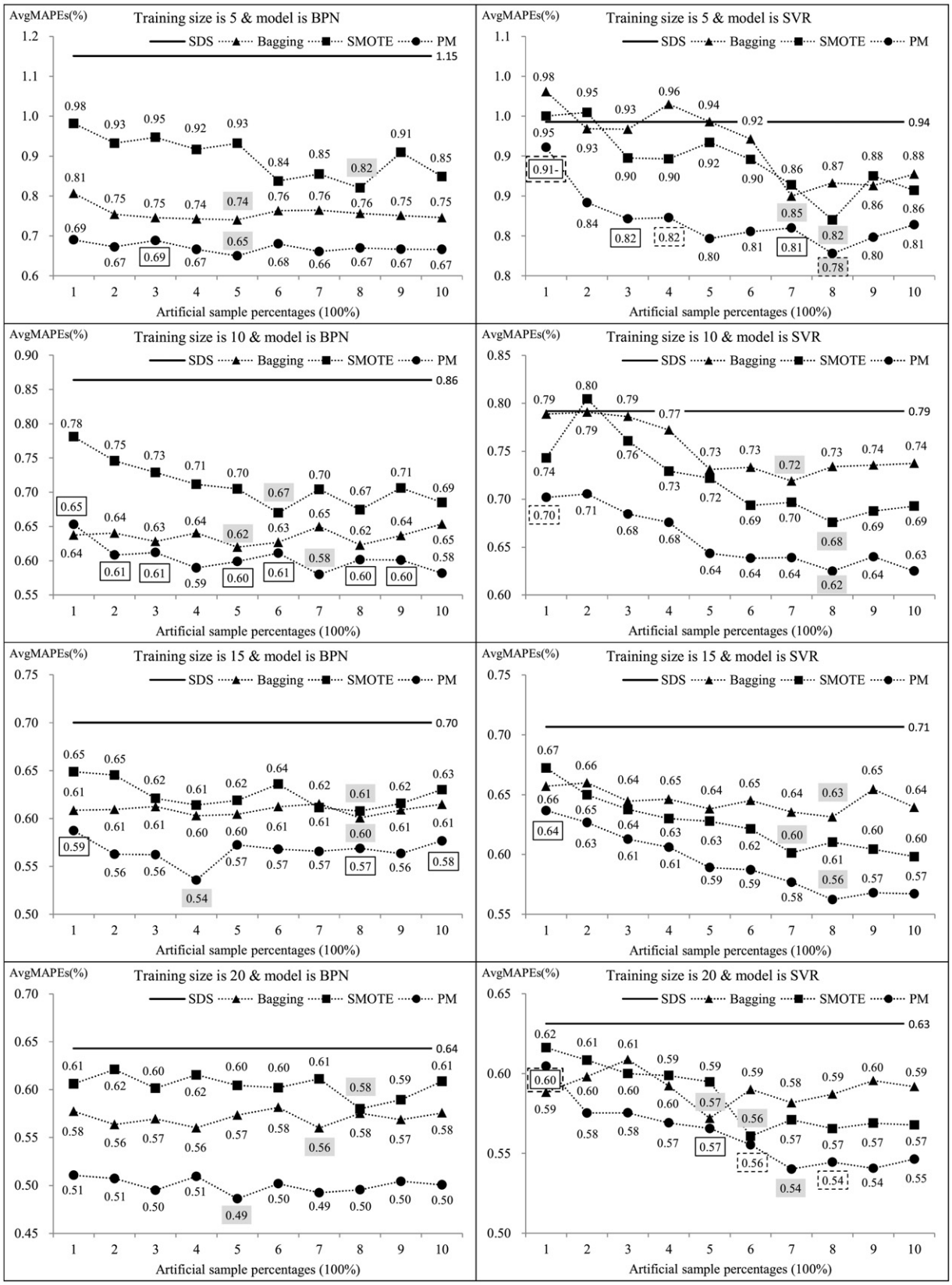

Fig. 14. Experimental results for Case II

2. When n is small (such as five or ten) and the artificial sample size M is also small (ranges from 100% to 400%), the AvgMAPEs of the three VSG approaches are very unstable. For some M, the AvgMAPEs are larger than the AvgMAPEs of "SDS"; this situation is especially distinct in "bagging" and "SMOTE." This finding implies that when n is quite small, the VSG approaches cannot improve the accuracy of the forecasting models by generating only a few samples based on the average.

3. When n is large (such as fifteen or twenty), the AvgMAPEs of the three VSG approaches are not only smaller than "SDS" but also more stable than "SDS", even though M is small. This finding suggests that the information about the populations in the training sets is sufficient and the artificial samples can contain correct population properties when n is large.
4. As M increases, the AvgMAPEs of the three VSG methods are stable and small and most of their minimum AvgMAPEs appear to range from 400% to 800%. In some cases, their AvgMAPEs do not continue to decrease; instead, they increase in the range of 800% to 1000%. This finding implies that too many artificial samples will cause the information contained in the original samples to be dominated by the information contained in the artificial samples, which yields biased results. However, the most suitable M to obtain the best results depends on the behaviors of the data, the operation of the kernels of the learning models, the mechanisms that underlie the VSG approaches, and the size of the data.
5. Most of the AvgMAPEs in "PM" are significantly smaller than the AvgMAPEs in "SDS", "bagging," and "SMOTE" with statistical support. This finding indicates that the proposed method outperforms bagging and the SMOTE in the two cases.
6. The difference between "bagging" and "SMOTE" is difficult to discern in the two cases. Their p-values exceed 0.05, and in some cases in Fig. 13 and Fig. 14, the AvgMAPEs are very similar. By taking their minimum AvgMAPEs and averages of AvgMAPEs as indicators, the SMOTE achieves better results than bagging when the model is SVR, whereas the reverse is true when the model is BPN.

5. Conclusions

While issues surrounding big data learning have attracted a substantial amount of attention in recent years, learning from small data is an older problem. In certain situations, data analyzers have to learn with small amounts of data, such as the two cases described in this paper. Numerous virtual sample generation approaches, which are data preprocessing methods in KDD, have been developed to extract knowledge from these data. Although BP is extensively applied to create new training sets, the data structures of the resulting sets encounter the issue of incompleteness when handling with small data. Although the SMOTE does not consider the entire distribution when generating samples, it does fill the information gaps with synthetic samples. This research proposed a systematic procedure that contains bound estimating functions and a sample generating approach to enhance the structures of small data to ensure that algorithms have sufficient training to improve the robustness and/or accuracy of the patterns that they identify.

This study examined two real cases from a leading manufacturer in the TFT-LCD industry in Taiwan, performed experiments using two algorithms—BPN and SVR—and implemented two VSG approaches—bagging (using BP) and the SMOTE—to compare the accuracies of the approaches. The results of the two cases indicated that the forecasting errors of the proposed method are smaller than the forecasting errors of bagging and the SMOTE with the majority of statistical support. The proposed method has considerable practical value for engineers in the two cases because it can help them build more robust and precise models when they try to infer the possible manufacturing results in TFT-LCD processes. Future studies should consider applying this method to small dataset learning problems in other manufacturing fields to validate the effectiveness of the approach.

References

[1] H.J. Gómez-Vallejo, B. Uriel-Latorre, M. Sande-Meijide, B. Villamarín-Bello, R. Pavón, F. Fdez-Riverola, D. Glez-Peña, A case-based reasoning system for aiding detection and classification of nosocomial infections, Decis. Support. Syst. 84 (2016) 104–116.
[2] W.S. Gosset, The probable error of a mean, Biometrika 6 (1908) 1–25.
[3] G.Y. Chao, T.I. Tsai, T.J. Lu, H.C. Hsu, B.Y. Bao, W.Y. Wu, M.T. Lin, T.L. Lu, A new approach to prediction of radiotherapy of bladder cancer cells in small dataset analysis, Expert Syst. Appl. 38 (2011) 7963–7969.
[4] C.J. Huang, H.F. Wang, H.J. Chiu, T.H. Lan, T.M. Hu, E.W. Loh, Prediction of the period of psychotic episode in individual schizophrenics by simulation-data construction approach, J. Med. Syst. 34 (2010) 799–808.
[5] Z. Huang, J. Li, H. Su, G.S. Watts, H. Chen, Large-scale regulatory network analysis from microarray data: modified Bayesian network learning and association rule mining, Decis. Support. Syst. 43 (2007) 1207–1225.
[6] P. Niyogi, F. Girosi, T. Poggio, Incorporating prior information in machine learning by creating virtual examples, Proc. IEEE 86 (1998) 2196–2209.
[7] G. Guo, C.R. Dyer, Learning from examples in the small sample case: face expression recognition, IEEE Trans. Syst. Man Cybern. B Cybern. 35 (2005) 477–488.
[8] D.C. Li, W.T. Huang, C.C. Chen, C.J. Chang, Employing virtual samples to build early high-dimensional manufacturing models, Int. J. Prod. Res. 51 (2013) 3206–3224.
[9] D.C. Li, L.S. Lin, Generating information for small data sets with a multi-modal distribution, Decis. Support. Syst. 66 (2014) 71–81.
[10] J.R. Quinlan, Learning with continuous classes, 5th Australian joint conference on artificial intelligence, Dermatol. Sin. (1992) 343–348.
[11] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery in databases, AI Mag. 17 (1996) 37.
[12] A. Dag, A. Oztekin, A. Yucel, S. Bulur, F.M. Megahed, Predicting heart transplantation outcomes through data analytics, Decis. Support. Syst. 94 (2017) 42–52.
[13] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, Chapmen & Hall, New York, 1993.
[14] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140.
[15] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.
[16] 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.
[17] L.A. Zadeh, Fuzzy sets, Inf. Control. 8 (1965) 338–353.
[18] C.F. Huang, Principle of information, Fuzzy Sets Syst. 91 (1997) 69–90.
[19] C.F. Huang, C. Moraga, A diffusion-neural-network for learning from small samples, Int. J. Approx. Reason. 35 (2004) 137–161.
[20] O.A.M. Ali, A.Y.A. Ali, B.S. Sumait, Comparison between the effects of different types of membership functions on fuzzy logic controller performance, Int. J. Emerg. Eng. Res. Technol. 3 (2015) 76–83.
[21] Z. Jin, B.K. Bose, Evaluation of membership functions for fuzzy logic controlled induction motor drive, IEEE 2002 28th Annual Conference of the Industrial Electronics Society, Vol. 02, IECON 2002, pp. 229–234.
[22] D.C. Li, W.K. Lin, L.S. Lin, C.C. Chen, W.T. Huang, The attribute-trend-similarity method to improve learning performance for small datasets, Int. J. Prod. Res. 55 (2017) 1898–1913.
[23] J. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan Press, 1975.
[24] J.W. Forrester, Industrial Dynamics, MIT Press, Cambridge, Massachusetts, 1961.

Der-Chiang Li is a Distinguished Professor at the Department of Industrial and Information Management, the National Cheng Kung University, Taiwan. He received his PhD degree at the Department of Industrial Engineering at Lamar University Beaumont, Texas, USA, in 1985. As a research professor, his current interest concentrates on machine learning with small data sets. His articles have appeared in *Decision Support Systems*, *Omega*, *Information Sciences*, European Journal of Operational Research, Computers & Operations Research, *International Journal of Production Research*, and other publications

Wu-Kuo Lin is a Ph.D. candidate at the Department of Industrial and Information Management, the National Cheng Kung University, Taiwan. His current research interests are in the area of forecasting and data mining with small data sets. His articles have appeared in *Expert Systems with Applications* and *The journal of Grey System*.

Chien-Chih Chen is a postdoctoral fellow at the Department of Industrial and Information Management, the National Cheng Kung University, Taiwan. His article has appeared in *Omega*, *Expert Systems with Applications*, *International Journal of Production Research*, *Neurocomputing*, *Computers & Industrial Engineering*, *Journal of Intelligent Manufacturing*, and other publications.

Hung-Yu Chen is a Ph.D. candidate at the Institute of Information Management, the National Cheng Kung University, Taiwan. His current research interests focus on the learning issue of small datasets.

Liang-Sian Lin is a PH.D researcher at the Department of Industrial and Information Management, the National Cheng Kung University, Taiwan. He is also working at the laboratory for small sample learning. As a research professor, his current interests concentrate on small data sets. His article has appeared in *European Journal of Operational Research*, *Decision Support Systems*, and *International Journal of Production Research*.