Model-Based Synthetic Sampling for Imbalanced Data

Chien-Liang Liu, *Member, IEEE* and Po-Yen Hsieh

Abstract—Imbalanced data is characterized by the severe difference in observation frequency between classes and has received a lot of attention in data mining research. The prediction performances usually deteriorate as classifiers learn from imbalanced data, as most classifiers assume the class distribution is balanced or the costs for different types of classification errors are equal. Although several methods have been devised to deal with imbalance problems, it is still difficult to generalize those methods to achieve stable improvement in most cases. In this study, we propose a novel framework called model-based synthetic sampling (MBS) to cope with imbalance problems, in which we integrate modeling and sampling techniques to generate synthetic data. The key idea behind the proposed method is to use regression models to capture the relationship between features and to consider data diversity in the process of data generation. We conduct experiments on 13 datasets and compare the proposed method with 10 methods. The experimental results indicate that the proposed method is not only comparative but also stable. We also provide detailed investigations and visualizations of the proposed method to empirically demonstrate why it could generate good data samples.

Index Terms—Imbalanced data, over-sampling, synthetic sampling, model-based approach

1 Introduction

Imbalanced data is characterized by severe class distribution skews and has received considerable attention in data mining and machine learning communities [1]. The American Association for Artificial Intelligence (AAAI) and International Conference on Machine Learning (ICML) individually held workshops on this problem [2] and even some researchers indicated that the imbalance problem is among the top 10 challenges of data mining research [3]. Imbalanced data occurs in several real-world domains. In the medical domain, the predictive model that helps diagnose breast cancer through mammography [4] encounters a class imbalance problem, as most patients do not suffer from breast cancer. In manufacturing, it is fairly important to identify faults in a system and even find the fault types [5]. However, the fault detection models in manufacturing always receive large quantities of healthy data samples, but the number of fault events is limited.

The last decade has witnessed the great success of machine learning owing to the explosion of data and advancements in computing power. Machine learning has been successfully applied to many application domains, including, but not limited to, the medical domain [6], finance domain [7], and manufacturing domain [8]. As machine learning classifiers learn from imbalanced data, their prediction performances often deteriorate significantly [2]. This is because most machine learning algorithms assume that the underlying class distribution is balanced [9] or the costs for different classification errors are equal [2]. Therefore, the imbalanced problem would bias the classification decision toward the majority class. As we are always interested in the minority class (e.g., the positive case for medical diagnosis and fault event for manufacturing), the aforementioned problems really cause a substantially impact in practice.

Although many methods dealing with imbalanced data have been proposed in recent years, it is difficult to generalize them to achieve stable improvement in most cases. The sampling method is probably one of the most widely used methods in dealing with imbalanced data, as it is easy to implement. One drawback of the sampling method is that data samples are increased or decreased without considering the underlying data distribution. Over-sampling may result in an overfitting problem, while under-sampling may discard representative samples. Furthermore, many synthetic sampling approaches generate synthetic samples based on k-nearest neighbors, which may be biased by the samples in the minority class.

In this work, we propose a model-based synthetic sampling (MBS) method, which is a new framework that over-samples the minority class instances from a new aspect. The proposed model belongs to the over-sampling technique, and the goal is to generate synthetic samples that could capture the relationship between the features of training samples that are in the minority class, while keeping the variability of the data samples. Compared with previous methods, the proposed method generates synthetic samples based on several ideas. First, the proposed method uses the modeling technique to capture trends or regression lines of the features for the training samples in the minority class.

• *The authors are with the Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu 300, Taiwan. E-mail: cliu@mail.nctu.edu.tw, b10071007@hotmail.com.*

Manuscript received 23 Jan. 2018; revised 30 Jan. 2019; accepted 3 Mar. 2019. Date of publication 18 Mar. 2019; date of current version 7 July 2020. (Corresponding author: Chien-Liang Liu.) Recommended for acceptance by L. B. Holder. Digital Object Identifier no. 10.1109/TKDE.2019.2905559

1041-4347 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of South Florida. Downloaded on May 13,2023 at 03:30:20 UTC from IEEE Xplore. Restrictions apply.

Second, it generates temporary data samples by sampling available feature values. Finally, it transforms temporary data samples into synthetic data via the constructed model. In the experiments, we compare the proposed method with several alternatives on thirteen datasets, and the experimental results indicate that the proposed method is not only comparative, but also stable. We also provide detailed investigations and visualizations of the proposed method to empirically demonstrate why it could generate good data samples.

The contributions of this work are listed as follows. First, we focus on imbalanced data problems and propose a model-based synthetic sampling method. Second, we design several experiments to assess the proposed method. We conduct experiments on thirteen datasets and compare the results of the proposed method with those of ten competitive methods. The experimental results indicate that the proposed method is comparative and outperforms other alternatives in most cases. Finally, we provide detailed investigations and use a visualization technique to empirically show that the proposed method performs well when compared with the other alternatives. Moreover, the proposed method is a data-level algorithm, which is easy and flexible to extend. We combine the proposed method with the boosting technique to devise a method called MBSBoost, and compare the performance of the combination with two state-of-the-art ensemble-based methods with regards to the imbalance problem. The experimental results indicate that the MBSBoost works well and stably on the thirteen datasets.

The rest of this paper is organized as follows. Section 2 presents related surveys about the imbalanced data and the methods. Section 3 then introduces the proposed model. Next, Section 4 summarizes the experimental settings and results. Section 5 presents the discussion and analysis. Conclusions are finally drawn in Section 6.

2 Related Work

Imbalance problems are harmful to many kinds of classifiers. Sun et al. (2009) [10] have analyzed the difficulties of learning from imbalanced data for several machine learning algorithms, including decision tree, artificial neural network, and support vector machines (SVM) [11]. Consequently, numerous researchers have devoted time to designing methods for imbalanced data, and these methods could be categorized into two types, data-level and algorithm-level methods [10]. We also conducts a literature survey of ensemble-based approaches for the imbalance problem, as applying the ensemble learning technique to deal with imbalance problems has become popular in recent years.

2.1 Data-Level Approach

The data-level approach adjusts the class distribution by resampling the original data and generating synthetic data to remedy imbalance problems. As this approach is always applied to data before constructing a classification model, it could be regarded as part of the pre-processing step. The main advantage of the data-level approach is that the methods utilized are independent of classifiers.

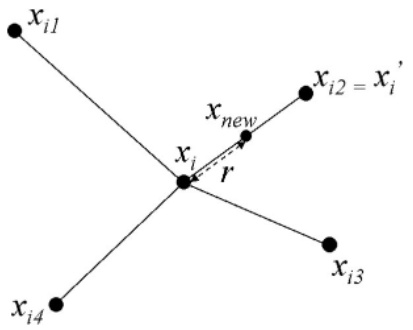


Fig. 1. Synthetic data generation based on SMOTE.

The resampling methods for imbalanced data originated from the research of Kubat and Matwin [12]. They can be roughly categorized into over-sampling and under-sampling methods. The goal of over-sampling is to increase the number of samples in the minority class by resampling or generating synthetic data, while that of under-sampling is to decrease the number of samples in the majority class by removing samples from it. The original resampling methods are random over-sampling and random under-sampling [13]. Random over-sampling selects minority data by random sampling, i.e., the probability of each sample is the same, with replacement and then adds the selected samples to the original dataset. Random under-sampling selects samples from majority set without replacement and then removes the selected data from the dataset.

The mechanics of random over-sampling adjust the class distribution by simple sampling, and it merely replicates the same minority class samples. In contrast, the synthetic sampling approach generates new synthetic data points to increase the number of the minority samples. This approach is inspired by a technique used in handwritten character recognition [14]. Owing to the limited data in a handwritten problem, researchers usually create additional training data by skewing or rotating the original data. Chawla et al. [14] developed the synthetic minority over-sampling technique (SMOTE) based on the aforementioned idea. SMOTE first finds the k-nearest neighbors for each minority class sample. Then, it randomly samples one neighbor from the k-nearest neighbors and generates a synthetic sample along the line segment between each minority class sample and its selected neighbor. The generation formula for SMOTE is as follows:

$$x_{new} = x_i + (x_i' - x_i) * r, \tag{1}$$

where x_i is the minority class sample under consideration, x_i' is the selected neighbor for x_i, and r is a random number distributed uniformly from 0 to 1. The geometric meaning is illustrated in Fig. 1, in which the x_i at the center of figure is the minority class sample, the surrounding $x_{i1} - x_{i4}$ are the 4 nearest neighbors, and x_i' is the selected neighbor. According to the formula listed in Equation (1), x_{new} depends on the random number r, which determines the location of synthetic data in linear interpolation between x_i and x_i'. Given that r close to 0, the synthetic sample will be close to x_i. In contrast, the synthetic sample will be close to x_i' as r approaches 1.

Authorized licensed use limited to: University of South Florida. Downloaded on May 13,2023 at 03:30:20 UTC from IEEE Xplore. Restrictions apply.

The invention of SMOTE has inspired many researchers to develop different synthetic sampling methods, including borderline-SMOTE [15], safe-level-SMOTE [16], and ADASYN [17]. Borderline-SMOTE is based on SMOTE and the concept is to consider the nature of the classifier. The task of most classification algorithms is to learn the borderline between different classes in the training phase. Those data points far away from the borderline probably contribute less to the classification than those close to borderline; thus, borderline-SMOTE only focuses on minority class samples that are close to borderline to increase data size. Safe-level-SMOTE [16] is also an extension of SMOTE, but the idea behind it is quite different from that of borderline-SMOTE. Borderline-SMOTE only focuses on the samples in the borderline region, leading to an ambiguous and dangerous situation. In safe-level-SMOTE, each minority sample is assigned a safe level, and the synthetic sample will be positioned closer to the minority sample with a higher safe level.

To alleviate the problem of the small disjunct caused by within-class imbalance, Jo and Japkowicz [18] proposed cluster-based over-sampling (CBO), which is a combination of a resampling technique and an unsupervised learning method. The CBO method applies K-means to the majority and the minority classes respectively to find the sub-groups within majority and minority classes. Then, it performs over-sampling on each sub-group. The CBO method could avoid the small disjunct problem, as sub-concepts that are too small will not be present within each class. Yen and Lee [19] devised the under-sampling method based on clustering (SBC), which is also a sampling method integrated with unsupervised learning method. The SBC applies K-means to the entire data set without separating data into minority and majority classes. Then, it samples majority class samples from each cluster, and the number of selected samples for each cluster is determined by a specific formula. Finally, the selected majority class samples are combined with all minority class samples for training.

One way to generate synthetic samples with good quality is to discover the structure underlying the data, and then perform over-sampling on the discovered structure. Xie et al. [20] proposed a method called the minority over-sampling technique based on local densities in low-dimensional space (MOT2LD), which first applies t-SNE [21] to reduce the dimensionality of the training samples into a two-dimensional space, and then uses a density-peak algorithm to learn the cluster structure of the training samples in the low-dimensional space. The minority class samples are sampled according to local majority count and local minority density, and the synthetic samples are created by interpolation between two samples within cluster. Besides MOT2LD, DBSMOTE [22] uses a similar framework to explore the structure of minority data. First, DBSMOTE applies DBSCAN [23] to discover density-reachable graphs of clusters. Second, DBSMOTE calculates the mean of each cluster as pseudo-centroids. The main difference between DBSMOTE and MOT2LD is that DBSMOTE over-samples along a shortest path from each minority sample to the pseudo-centroid of its cluster.

One important characteristic of time-series data is that the adjacent data points in the time-series are usually dependent and highly correlated. However, conventional over-sampling approaches do not consider the correlation structure within the time-series data, so they are not suitable for time-series data. Thus, Cao et al. [24] focused on imbalanced time-series classification and proposed a novel integrated over-sampling (INOS) method that integrates the over-sampled population from their proposed enhanced structure preserving over-sampling (ESPO) and the interpolation-based method to generate synthetic samples. The ESPO performs over-sampling in the transformed signal space by estimating the covariance structure of the minority-class samples and regularizing the unreliable eigen spectrum. Because sequence classification has to consider the temporal structure of sequences, Gong and Chen [25] proposed the use of representation learning and kernel learning to learn a discriminative kernel feature space, on which over-sampling is performed. The representation learning is based on a recurrent neural network (RNN), since RNN could deal with multivariate and varying length sequences naturally. Once the learning process is completed, the data are projected into the kernel feature space, and SMOTE is performed on the kernel space to oversample minority samples.

2.2 Algorithm-Level Approach

The methods at the algorithm-level avoid the imbalance problem by directly modifying the classifier or using different mis-classified costs to enhance the entire performance of the model. These methods intervene in the training stage, so most of them are not independent of classifiers. In contrast, the data-level approach operates at the data level; therefore, various classifiers could benefit from it.

The fundamental purpose of cost-sensitive learning is to use a cost matrix to adjust the penalties for different errors. In most settings, we are interested in the minority class. Given that the minority class is the positive class and majority class is the negative class, the penalty or cost for false negatives (FN) is higher than that for false positives (FP), so that the learning algorithm can emphasize the importance of the minority. Wu and Chang [26] proposed a method called kernel-boundary alignment (KBA), which redesigns the kernel trick according to the imbalanced data distribution. One of the most popular kernel functions is the radial basis function (RBF), and the KBA is based on RBF as well. The kernel function for KBA involves not only the distance between all data points and the support vectors, but also the class distribution of the support vectors. Thus, the imbalanced condition is under consideration in the training process.

Based on kernel logistic regression, Ohsaki et al. [27] proposed a method called confusion-matrix-based kernel logistic regression (CM-KLOGR) for imbalanced data. The key idea is to include the weighted harmonic mean of various performance metrics from the confusion matrix in the loss function. The CM-KLOGR involves two steps. The first step is to pre-train a model with cross-entropy loss, which is the same as the original kernel logistic regression. The second step is to retrain the model, initialized by the pre-training parameters, with their proposed loss function to simultaneously consider various performance metrics for imbalanced data, so that the minority class will not be ignored.

Deep learning has become one of the most popular research topics in recent years, and the key idea behind it is

Authorized licensed use limited to: University of South Florida. Downloaded on May 13,2023 at 03:30:20 UTC from IEEE Xplore. Restrictions apply.

to learn good feature representation from data. Khan et al. [28] proposed a cost-sensitive (CoSen) deep neural network to learn feature representations for both the majority and minority classes, but their work only focused on image domains. Huang et al. [29] proposed the use of a euclidean embedding function to deal with imbalanced classification problems, such that the embedded features are discriminative without any possible local class imbalance. In their proposed learning network architecture, they formulated a quintuplet sampling method to consider an anchor, a within-cluster neighbor, a between-cluster neighbor, a within-class neighbor, and a between-class neighbor in a mini-batch, so that the learned embedding function could preserve locality across clusters and discrimination between classes. Their proposed method could only be applied to images, and computationally expensive data pre-processing, including clustering and quintuplet construction, is required for each round of model learning. Dong et al. [30] addressed the problem of imbalanced data for the multi-label classification problem in deep learning, and introduced incremental minority class discrimination learning by formulating a class rectification loss regularization, which imposes an additional batch-wise class balancing on top of the cross-entropy loss to rectify model learning bias due to the over-representation of the majority classes.

Most classification algorithms are discrimination-based, indicating that the classifiers attempt to construct a decision boundary to separate the data points into different classes. However, the one-class learning approach, which is recognition-based, is distinct from those approaches. One-class learning learns from a data set comprising only samples from one class to capture the pattern. Once the similarity between new sample and the pattern of the learned class exceeds the threshold, the sample should be identified as being in the same class, and vice versa [10]. Raskutti and Kowalczyk [31] have shown that one-class SVM [32] is stable as the data is highly imbalanced. This is because the one-class learning approach essentially focuses on one class in the training stage, thus the learning process is not susceptible to imbalance problems.

2.3 Ensemble Approach

Ensemble learning is a machine learning approach that trains a set of hypotheses and combines them to make a prediction. The most common methods in ensemble learning include boosting [33], bagging (bootstrap aggregating) [34], and stacking [35]. Several researchers have applied boosting to imbalance problems and devised novel algorithms that integrate either overs-ampling or under-sampling methods into ensemble learning framework. The boosting approach is similar to the bagging approach in terms of constructing multiple classifiers by sampling, but boosting tends to select the samples that are mis-classified by previous classifiers in general.

The SMOTEBoost algorithm [36] is an integration of SMOTE and AdaBoost [37], [38]. It not only assigns higher weights to the mis-classified samples, but also performs SMOTE to increase the number of minority data at each round. The DataBoost-IM algorithm [39] combines AdaBoost with a new synthetic sampling method based on the Gaussian distribution. The key process is to generate synthetic data for the sample that is difficult to classify no matter whether it belongs to the majority or the minority class. At each round, the DataBoost-IM algorithm identifies the samples that are most difficult to classify and then generates synthetic data for the majority and the minority classes individually with different over-sampling rates.

The aforementioned approaches combine over-sampling methods with ensemble learning, but over-sampling methods always generate more training samples, requiring more training time to learn model parameters. In contrast, under-sampling removes data from the training set, giving it a base to reduce the training time. Thus, Liu et al. [40] combined ensemble learning and under-sampling to propose two algorithms, EasyEnsemble and BalanceCasCade. EasyEnsemble samples a subset of the majority, and uses this subset as well as all the minority data to train a classifier. The process is repeated several times and all classifiers are trained on the balanced data. BalanceCasCade has the same structure as EasyEnsemble, but the correctly classified majority samples will be removed at each iteration. That is because, as the remaining majority samples are difficult to classify, they are believed to be important data for the model. Besides, Wang and Pineau [41] have extended several ensemble learning algorithms, such as SMOTEBOOST and RUSBOOST, to deal with imbalanced data problems presented in online learning. Galar et al. [42] have provided a review on ensembles for the class imbalance problem, in which they developed a thorough empirical comparison by considering the most significant published papers, and proposed a taxonomy for bagging, boosting, and hybrid-based approaches.

3 Model-Based Synthetic Sampling

This section introduces the proposed method, including notations, the algorithm, and our motivation.

3.1 Concept and Motivation

The over-sampling methods require replication of minority class samples, leading to many overlapping points in the feature space. Ideally, the importance of the minority data is enhanced, but the small disjunct problem is still severe, as the scope of the minority data is not enlarged. This situation forces the classifier to develop a large number of specific and narrow decision regions to classify each point correctly, but it is difficult for those decisions to be generalized to unseen data [14].

One of the purposes of SMOTE is to tackle the aforementioned problem. SMOTE achieved this by creating synthetic data to enlarge the scope of the minority data in the feature space, so that the generalization may be improved. Nevertheless, SMOTE has been criticized for having an over-generalization problem, as it blindly generalizes the region of the minority class samples without considering the majority data [16] and may produce the synthetic data situated in the clusters of the majority class. Although borderline-SMOTE was proposed to avoid this problem, it is not easy to define the borderline region. According to the original definition, the minority class sample is viewed as noise if all the nearest neighbors are majority data ($m = k$) and considered as borderline if $m = k - 1$, but actually these two cases are not

Authorized licensed use limited to: University of South Florida. Downloaded on May 13,2023 at 03:30:20 UTC from IEEE Xplore. Restrictions apply.

TABLE 1
The List of Feature Models

Model	Label	Feature
Model 1	F1	F2-F10
Model 2	F2	F1 and F3-F10
Model 3	F3	F1-F2 and F4-F10
	.	
	.	
	.	
Model 10	F10	F1-F9

significantly different [16]. Safe-level-SMOTE, which is an extension of SMOTE, uses a complicated definition for the degree of safety to modify the location of synthetic data. However, this method has to set not only the k value to perform SMOTE, but also an additional k value for calculating the safe-level for all samples. Practically, it is difficult to determine the values of these parameters.

The approaches related to SMOTE capture simple linear relationships between data points and ignore the global characteristics of the minority data. The sampling procedure in DataBoost-IM independently samples feature values from the Gaussian distribution for each synthetic sample without considering the relationship between features. In this work, we propose a new framework called MBS to tackle imbalance problems and attempts to create synthetic data by modeling and resampling techniques.

3.2 Notation

In this section, the notations that will be used in the following sections are introduced. Each data object $\mathbf{x}^{(i)}$ is represented as a feature vector of length m, i.e., $\mathbf{x}^{(i)} = [\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \ldots, \mathbf{x}_m^{(i)}]$. $\mathbf{x}_{-j}$ is introduced to represent all the features of data $\mathbf{x}$ except feature j. The data collection is represented by $\mathbf{D} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}]$, indicating that the number of data objects is n and $\mathbf{x}^{(i)} \in \mathbb{R}^m$. To simplify the explanation, we focus on binary classification, but the proposed work could be extended to multi-class problems. The focus of this study is imbalanced data; therefore, data could be separated into two classes, namely, the minority class and majority class. We use $\mathbf{D}^s$ to denote the minority class samples and use $\mathbf{D}^l$ to represent majority class samples. Thus, the entire data set can be divided into two partitions, namely, $\mathbf{D} = \mathbf{D}^s \cup \mathbf{D}^l$.

We propose to generate temporary synthetic data by sampling features, so we further introduce the value set $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m]$ for all the possible feature values of the minority class, so that $\forall \mathbf{x}^{(i)} \in \mathbf{D}^s$, $\mathbf{x}_1^{(i)} \in \mathbf{v}_1, \mathbf{x}_2^{(i)} \in \mathbf{v}_2, \ldots$, and $\mathbf{x}_m^{(i)} \in \mathbf{v}_m$.

3.3 Proposed Method

The goal of MBS is to generate synthetic data that keeps the characteristics of the original data. In this work, we assume there exists a relationship between features, and the relationship could be used to capture the characteristics of the features. Thus, we propose to build models to learn the relationship between features and generate synthetic data in terms of the learned feature models. MBS involves three steps: (1) training feature models, (2) sampling features to generate temporary sampling data, and (3) generating final synthetic data. This section introduces each step in detail and presents the proposed algorithm. The proposed method is a framework, and the feature models could be replaced with different models. In the implementation, we use linear and non-linear regression models as the feature models.

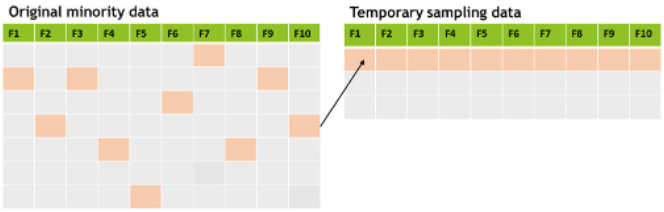


Fig. 2. Illustration of the sampling for each feature.

3.3.1 Training Feature Models

Each data sample is characterized by its features, so we propose to use a model-based approach to find the relationship between features. Given a data sample $\mathbf{x}^{(i)}$, we use one of the features, say $\mathbf{x}_j^{(i)}$, as the label, and the remaining features, $\mathbf{x}_{-j}^{(i)}$, as the elements of the new feature vector to compose a new training sample. Following the above scheme, one can have n training samples, each of which is located in a space of size $\mathbb{R}^{m-1}$.

For each feature $\mathbf{x}_j^{(i)}$, we could construct a mapping function or model. Thus, one can obtain m models following the same process. Assume the number of features is 10, and the features are F1, ..., F10. As illustrated in Table 1, the label for model 1 is F1, and the remaining features are the features of data samples during the training process. Following the same principal, the label for model 2 is F2, and the rest are features for training, and so on. The objective of this step is to capture the relationship between features in the minority class samples by modeling, and then use the learned models to adjust temporary synthetic data as described below.

3.3.2 Sampling Features to Generate Temporary Sampling Data

Given feature domains $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m]$ for all the features of the training samples in the minority class, we propose to use the sampling with replacement technique to generate the initial synthetic data. Given the value set $\mathbf{v}_1 = \{\mathbf{v}_{11}, \mathbf{v}_{12}, \ldots, \mathbf{v}_{1n}\}$, the first step is to sample a value from the set $\mathbf{v}_1$ with replacement, and the sampled one is the value of the first feature, namely $x_1^{(i)}$, for the temporary synthetic data sample $x^{(i)}$. Then, we can apply the sampling process to obtain the second feature value for the temporary synthetic data sample until the mth feature.

Fig. 2 shows an illustration of the process, in which we simplify the problem to assume that the number of possible values for all the ten features are all seven. Once the feature sampling process is completed, one can obtain a temporary data sample. Instead of replicating the original data to generate synthetic data, we use a sampling approach to generate temporary synthetic data by sampling from features, so as to increase the diversity of the synthetic data.

Authorized licensed use limited to: University of South Florida. Downloaded on May 13,2023 at 03:30:20 UTC from IEEE Xplore. Restrictions apply.

Fig. 3. Illustration of the predicting feature.

3.3.3 Generating Final Synthetic Data

Once the feature models and temporary synthetic data samples are available, the final step is to use the feature models to predict features of the final synthetic data with the temporary synthetic data samples as inputs. This final step aims to make the synthetic data emulate the feature relationships which the real observations have. Given an initial synthetic data sample $\mathbf{x}^{(i)}$ obtained from the previous step, one can predict $\mathbf{x}_j^{(i)}$, the jth feature of the final synthetic data sample, with the learned model j and the input $\mathbf{x}_{-j}^{(i)}$.

As shown in Fig. 3, F1 of the final synthetic data is obtained from the prediction of model 1, in which the inputs are the features of the temporary sampling data except F1, i.e., F2-F10. Then, one can follow the same process to obtain the remaining features of the final synthetic data. For example, F2 of the final synthetic data is obtained from the prediction of model 2, in which the inputs are the features of the temporary sampling data except F2, i.e., F1 and F3-F10. In the implementation, we repeat the synthetic data generation process five times, and we also conduct experiments to analyze the impact of the number of iterations on the performances.

3.4 Proposed Algorithm

Algorithm 1 shows that the inputs include the minority data $\mathbf{D}^s$, over-sampling rate p, the feature value set $\mathbf{V}$, and the number of iterations for repeating the generation process T. First, the temporary sampling data set $\mathbf{R}$ and the synthetic data set $\mathbf{S}$ are initialized as empty matrices of size $m \times n^{syn}$ as shown in Line 4 and Line 5 of Algorithm 1. Next, we train m feature models for the m features, in which model_j $(1 \leq j \leq m)$ is trained with $\mathbf{x}_j$ as the label and $\mathbf{x}_{-j}$ as the features. Lines 6-8 show the steps for training feature models. The second step of the proposed algorithm is to generate temporary data samples by applying the random with replacement technique on the features as presented in Lines 9-13. The final step is to generate synthetic data samples as listed in Lines 14-19, in which $\mathbf{S}_j$ denotes the jth feature of the synthetic data sample and is obtained from the prediction of model_j with $\mathbf{R}$ as the input.

The proposed algorithm is a data-level algorithm; therefore the outputs are the synthetic data samples. Once the generation process is completed, we use the final synthetic data as well as the original data to train the classifier.

Algorithm 1. MBS Algorithm

Input: $\mathbf{D}^s$: minority data, p: over-sampling rate, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m]$: possible feature values of minority class, T: total iterations for repeating generation process
Output: $\mathbf{S}$: synthetic data

1 n^s = number of minority class samples
2 m = number of features
3 $n^{syn} = n^s * p$ (synthetic data size)
4 $\mathbf{R} = \phi$ (temporary data with m columns and n^{syn} rows)
5 $\mathbf{S} = \phi$ (synthetic data with m columns and n^{syn} rows)
6 **for** j = 1 to m **do**
7 train model_j with $\mathbf{x}_j$ as label and $\mathbf{x}_{-j}$ as features
8 **end**
9 **for** i = 1 to n^{syn} **do**
10 **for** j = 1 to m **do**
11 $\mathbf{R_{ij}}$ = randomly sample a value from $\mathbf{v}_j$ with replacement
12 **end**
13 **end**
14 **for** t = 1 to T **do**
15 **for** j = 1 to m **do**
16 $\mathbf{S}_j$ = predict feature j by model_j and $\mathbf{R}$
17 **end**
18 $\mathbf{R} = \mathbf{S}$ (update the temporary dataset for predicting by predicted dataset $\mathbf{S}$)
19 **end**
20 return $\mathbf{S}$

3.5 Discussion

The proposed method is a framework that involves three steps. The first and final steps are related to regression models, while the goal of the second step is to randomly generate temporary synthetic samples. We briefly provide the motivation for using these two components to discuss their effectiveness in the proposed method.

In this work, we propose a data-level algorithm that utilizes the over-sampling technique to generate synthetic samples. Methods that generate synthetic samples to balance class distribution usually make specific assumptions during the generating process. For example, SMOTE assumes that the synthetic samples are located in the line between two minority samples, so it is expected that SMOTE considers local information to generate synthetic samples. Moreover, some researchers assume that the minority samples should be generated on a new feature space that could preserve structure information [24], [25].

To generate synthetic samples with good quality, we assume there exists a relationship between features that could be characterized by regression models. Thus, we would like to use simple models to capture the relationship, as the number of minority samples is always limited in an imbalance problem. We attempt to learn the trends or regression lines of the features from minority samples, so that the regression lines can facilitate the generation of final synthetic samples when given temporary synthetic samples.

It is worth mentioning that randomness is an important ingredient in sampling-based methods. In the second step, the temporary sampling data are generated randomly, so as to increase the diversity of the synthetic data. According to

Authorized licensed use limited to: University of South Florida. Downloaded on May 13,2023 at 03:30:20 UTC from IEEE Xplore. Restrictions apply.

TABLE 2
Summary Description of Datasets

Name	Data size	# of Features	Imbalance ratio
Pima Indian Diabetes	768	8	268:500(34.9%)
Haberman's Survival	306	3	81:225(26.5%)
Satimage	6435	36	626:5809(9.7%)
E.coli	336	7	35:301(10.4%)
Shuttle	43500	9	37:43463(0.09%)
Ionosphere	351	18	126:225(35.9%)
Vehicle	846	34	199:647(23.5%)
Give Me Some Credit	150000	10	10026:139974(6.7%)
Diabetes	101766	9	11357:90409(11.16%)
Hmeq	5960	10	1189:4771(20%)
Promotion	25000	4	1716:23284(6.9%)
Bank	45211	11	5289:39922(11.7%)
Spambase	4601	57	1813:2788(39.4%)

our survey, Guo and Viktor have used similar approaches to generate synthetic samples to balance class distributions as described in Section 2.2 of [39]. The key idea behind data generation in our work and [39] is similar, namely, to generate diverse synthetic samples.

The data generation step in the proposed method is similar to the rule of nominal attribute proposed by Guo and Viktor [39]. We do not make an assumption about the distribution underlying the original training attributes; instead, we use models to adjust the attribute values in order to emulate the real feature relationships. Guo and Viktor applied this technique to deal with an imbalanced data problem [39], and improve classification performance by using boosting with data generation [43]. We conduct experiments to compare the proposed framework with several alternatives, and investigate the effectiveness of the steps involved in the proposed framework.

4 Experiment and Discussion

4.1 Datasets

We conducted experiments on thirteen datasets to evaluate the performance of the proposed method and other competitive methods. The summaries of the datasets are listed in Table 2, including the data size, number of features, and imbalance ratio. All of the data used in the experiments are publicly available datasets, in which "Give Me Some Credit","Diabetes","Hmeq", as well as "Promotion" are from Kaggle,[1] while the others are from UCI machine learning repository.[2] Detailed introduction for each dataset is available on github[3] due to the space limit.

4.2 Evaluation Metric

Accuracy is probably one of the most commonly used performance metrics for classification tasks. However, the limitation of accuracy as the performance measure on an imbalanced dataset was quickly established, and receiver operating characteristic (ROC) curves soon emerged as a popular choice [44], in which the x-axis is the false positive rate (FPR) and the y-axis is the true positive rate (TPR). To compare classifiers, one may want to reduce ROC performance to a single scalar value representing expected performance. The area under the curve (AUC) is an alternative method for evaluating classifier performance, explaining why this work uses AUC as the evaluation metric.

1. https://www.kaggle.com/
2. https://archive.ics.uci.edu/ml
3. https://github.com/b10071007/Model-Based-Synthetic-Sampling

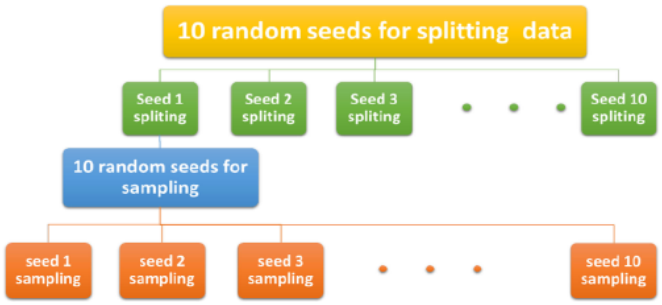


Fig. 4. Structure for splitting and sampling.

4.3 Experimental Settings

In the experiments, we compared the proposed method with several competitive methods, including random over-sampling, random under-sampling, SMOTE, borderline-Smote1, borderline-Smote2, safe-level-SMOTE, ADASYN, cluster-based over-sampling, and the under-sampling method based on clustering. These methods are classical and commonly used in dealing with imbalanced data, explaining why they were selected in the experiments.

Sampling methods belong to the stochastic process, so two confounding factors may lead to unstable experimental results: (1) different random seeds for splitting training and testing data, and (2) different random seeds for performing sampling method. Therefore, we proposed a more rigorous design in our experiments as illustrated in Fig. 4, in which we used 10 different seeds to split data into training and testing data, and performed the sampling methods for each splitting data with another 10 different seeds. In each data splitting, we could collect 10 performance results for each method, and we used the average performance as the result for this splitting. Then, we averaged the performance from 10 splittings to get the final performance result for each method; in other words, the final performance result was based on 100 experimental results.

The proposed method and the competitive methods are data-level algorithms, and the experiments focused on a classification task, so the experiments required a classifier to perform the classification task. We used logistic regression as the classifier, as it is a commonly used classification algorithm.

Moreover, the proposed method is a framework, and we can use different modeling methods as our feature models. The purpose of the feature model is to discover the relationship between features, so any model could be used in the proposed method. This work focuses on numeric features, so the feature models are regression models. Note that most over-sampling methods also focus on numeric features to generate data samples. Following the setting of the classification algorithm, we selected a linear algorithm and two non-linear algorithms as the feature models to evaluate the performance of the proposed method, in which the linear

Authorized licensed use limited to: University of South Florida. Downloaded on May 13,2023 at 03:30:20 UTC from IEEE Xplore. Restrictions apply.

TABLE 3
Average AUC Results

Dataset	Baseline	MBS_linear	MBS_CART	MBS_SVR	Over	Under	CBO	SBC	Smote	B_Smote1	B_Smote2	Safe_Smote	ADASYN
Pima	0.8246	**0.8254**	0.8194	0.8207	0.8242	0.8225	0.7994	0.8209	0.8238	0.8226	0.8232	0.8248	0.8224
Haberman	0.6795	**0.6873**	0.6762	0.675	0.6805	0.68	0.5612	0.6593	0.6825	0.6854	0.6854	0.6839	0.6804
Satimage	0.7618	0.7645	0.7525	**0.7649**	0.7612	0.7611	0.7514	0.7492	0.7613	0.7577	0.7574	0.7625	0.7543
Ecoli	0.9235	**0.9296**	0.9265	0.927	0.9197	0.9192	0.9052	0.9035	0.9233	0.9259	0.9248	0.927	0.9177
Shuttle	0.8678	**0.8902**	0.8835	0.8878	0.8696	0.8698	0.837	0.8837	0.8684	0.8727	0.8769	0.8689	0.8861
Ionosphere	0.8387	**0.8661**	0.857	0.8539	0.8425	0.8112	0.817	0.7952	0.8373	0.853	0.8494	0.8511	0.8541
Vehicle	0.9733	0.9882	0.992	0.9804	0.9737	0.9703	0.9862	0.9716	0.9836	0.9898	**0.9930**	0.9809	0.9905
Credit	0.6995	**0.7178**	0.6601	0.6503	0.7004	0.7005	0.6961	0.7062	0.7003	0.6996	0.6996	0.6993	0.7011
Diabetes	0.6333	**0.6348**	0.6136	0.5602	0.6334	0.6335	0.5470	0.6048	0.6322	0.6321	0.6329	0.6328	0.6308
Hmeq	0.7837	0.7837	0.7798	0.7672	0.7838	**0.7839**	0.6653	0.7749	0.7837	0.7818	0.7823	0.7832	0.7830
Promotion	0.6499	0.6499	0.6490	0.6431	0.6498	0.6499	0.6418	0.6426	0.6499	0.6492	0.6493	0.6494	**0.6500**
Bank	0.8567	**0.8600**	0.8553	0.8273	0.8584	0.8583	0.7537	0.8433	0.8581	0.8590	0.8591	0.8588	0.8595
Spambase	0.9713	0.9712	0.9658	0.9602	**0.9714**	0.9659	0.9375	0.8647	0.9661	0.9535	0.9693	0.9658	0.9693

TABLE 4
Ranking Results

Dataset	Baseline	MBS_linear	MBS_CART	MBS_SVR	Over	Under	CBO	SBC	Smote	B_Smote1	B_Smote2	Safe_Smote	ADASYN
Pima	3	**1**	12	11	4	8	13	10	5	7	6	2	9
Haberman	9	**1**	10	11	6	8	13	12	5	2	2	4	7
Satimage	4	2	11	**1**	6	7	12	13	5	8	9	3	10
Ecoli	7	**1**	4	2	9	10	12	13	8	5	6	2	11
Shuttle	12	**1**	5	2	9	8	13	4	11	7	6	10	3
Ionosphere	9	**1**	2	4	8	12	11	13	10	5	7	6	3
Vehicle	11	5	2	9	10	13	6	12	7	4	**1**	8	3
Credit	9	**1**	12	13	5	4	11	2	6	7	7	10	3
Diabetes	4	**1**	10	12	3	2	13	11	7	8	5	6	9
Hmeq	4	3	10	12	2	**1**	13	11	5	9	8	6	7
Promotion	5	4	10	11	6	2	13	12	3	9	8	7	**1**
Bank	9	**1**	10	12	6	7	13	11	8	4	3	5	2
Spambase	2	3	8	10	**1**	7	12	13	6	11	5	9	4

algorithm was linear regression and the non-linear algorithms were CART and SVR with the radial basis function.

All the methods in the experiments required specification of the hyper-parameters, including the over-sampling rate, under-sampling rate, and k nearest neighbors. Seven of the competitive methods use the over-sampling approach, while the remaining two methods use the under-sampling approach. We set the over-sampling rate as 100 percent, which doubled the proportion of minority data, and under-sampling rate as 50 percent, which removed half of the majority data. Furthermore, we set k as five for the methods involving SMOTE, and set the degree of safety as five for borderline-SMOTE, safe-level-SMOTE and ADASYN according to the original papers. As for the proposed method, the number of iterations was five.

4.4 Experimental Results

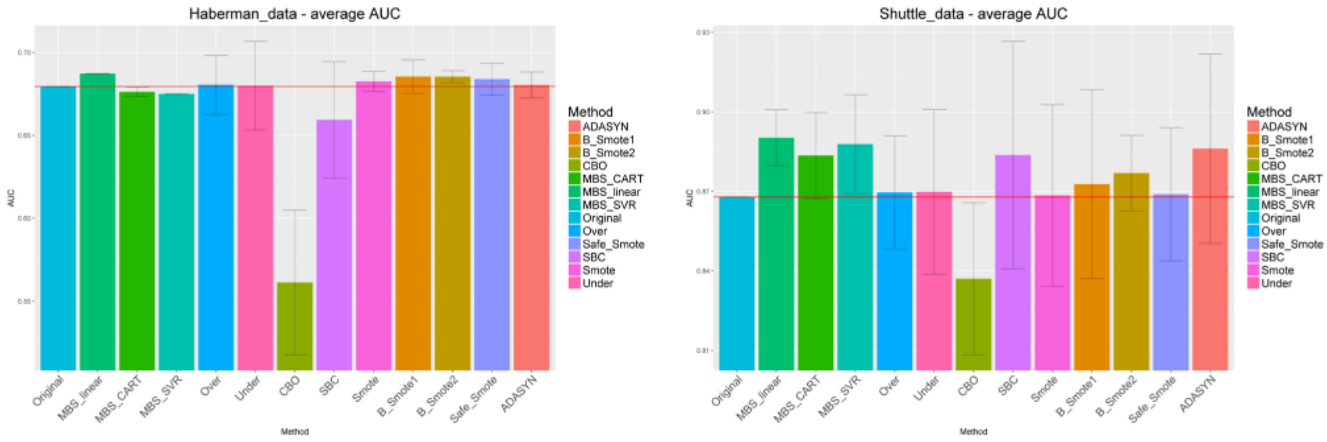
We applied all the methods to the thirteen datasets and used AUC as the performance metric. The code is available on github.[4] In the experiments, we had a baseline method called "baseline", which trained the model using original data without a sampling method. Additionally, the proposed method allows linear and non-linear regression methods to discover the relationship between features. We used "MBS_linear" to represent MBS with a linear feature model, and used "MBS_CART" and "MBS_SVR" to represent two kinds of non-linear feature models. Finally, the experiments used "Over" and "Under" to denote random over-sampling and random under-sampling, respectively; "CBO" and "SBC" to represent the cluster-based over-sampling method and under-sampling method, respectively; "B_Smote1" and "B_Smote2" to denote two forms of borderline-SMOTE; and "Safe_Smote" to represent safe-level-SMOTE.

The experimental results, including the average AUCs for all methods, are presented in Table 3. The experimental results indicate that the proposed method with a linear feature model, namely MBS_linear, generally outperforms the other alternatives. We summarize the performance rankings for all the methods on the thirteen datasets in Table 4. The ranking results indicate that MBS_linear indeed outperforms other state-of-the-art sampling methods. Among the 13 datasets, MBS_linear achieves the best performances on 8 datasets, and gets the second place on one dataset. In contrast, the proposed method with CART and SVR as the feature model, namely MBS_CART and MBS_SVR, fail to perform well on most datasets.

We further selected two datasets to present the mean and the 95 percent confidence interval of AUC for different sampling methods by bar plot. Fig. 5 shows the results. The experimental results indicate that MBS_linear is effective and

4. https://github.com/b10071007/Model-Based-Synthetic-Sampling

Authorized licensed use limited to: University of South Florida. Downloaded on May 13,2023 at 03:30:20 UTC from IEEE Xplore. Restrictions apply.



(a) Haberman Dataset (b) Shuttle Dataset

Fig. 5. AUC result of Haberman and Shuttle datasets for linear classifier.

stable, as it achieves a higher AUC mean and narrower AUC confidence interval than the competitive methods. We also carried out a Wilcoxon signed-rank test to verify whether MBS_linear outperforms other methods significantly across different datasets. The Wilcoxon signed-rank test, which is a kind of non-parametric statistical hypothesis test, was applied to assess whether the population mean rank of two paired samples differ. The results shown in Table 5 indicate that MBS_linear is significantly better than other methods, as we set the confidence level 0.95, namely, $\alpha = 0.05$.

5 Discussion and Analysis

We conducted experiments on thirteen datasets and compared the proposed framework with ten methods. Furthermore, in this paper, we provide a detailed investigation of the proposed method. The previous experiments indicated that the proposed method with a linear regression model outperformed that with a non-linear regression model; therefore, the following sections focus on the proposed method with a linear regression model.

5.1 Investigation of the Proposed Method

The first step of the proposed method is to construct a feature model to identify the relationship between features. We designed experiments to check whether the subsequent classifier could benefit from the feature modeling process. We created a baseline method called "No modeling", which only samples features to increase the number of data samples without modeling and predicting steps.

TABLE 5
Wilcoxon Signed-Rank Test

Comparison	Hypothesis(α=0.05)	p-value
MBS_linear vs. Baseline	Rejected	0.0033
MBS_linear vs. Over-sampling	Rejected	0.0024
MBS_linear vs. Under-sampling	Rejected	0.0012
MBS_linear vs. CBO	Rejected	0.0002
MBS_linear vs. SBC	Rejected	0.0002
MBS_linear vs. SMOTE	Rejected	0.0005
MBS_linear vs. Borderline-Smote1	Rejected	0.0012
MBS_linear vs. Borderline-Smote2	Rejected	0.0081
MBS_linear vs. Safe-level Smote	Rejected	0.0004
MBS_linear vs. ADASYN	Rejected	0.0034

The second step is to perform sampling for each feature to increase diversity of synthetic data. To analyze this step, we created three different sampling schemes that were compared with the proposed sampling approach.

- MBS with random generator:
  Randomly samples feature values from the Gaussian distribution to construct temporary sampling data.
- MBS with sampling on each sample:
  Instead of using feature-level sampling, this approach performs data-level sampling to increase the number of data samples.
- MBS with no sampling:
  Directly uses the original minority data without sampling to create temporary sampling data.

The final step is to predict features iteratively, and we conducted experiments to observe the change in performance with different numbers of iterations.

5.1.1 Experimental Settings

The evaluation settings were the same as those of the previous experiments, and we used logistic regression as the classifier and linear regression as the feature model of MBS. All variations in MBS belong to the over-sampling category and we set the generating rate as 100 percent to double the minority data. Moreover, the number of iterations was five for all variants, and then we observed the change in the performance of the original MBS at iteration one, three, and five.

5.1.2 Experimental Results

The experimental results are presented in Table 6 and the rankings of the variants are listed in Table 7. The experimental results indicate that MBS generally outperforms "No modeling". The difference between these two methods is the modeling process, and the results indicate that MBS could benefit from the modeling step.

Next, the evaluation for the second step requires a comparison of MBS and the variants of MBS. It is apparent that MBS outperforms these three alternatives, indicating that

Authorized licensed use limited to: University of South Florida. Downloaded on May 13,2023 at 03:30:20 UTC from IEEE Xplore. Restrictions apply.

TABLE 6
Average AUC for Different Variations of MBS

Dataset	MBS	No modeling	MBS (Data Level Sampling)	MBS(Random Sampling)	MBS (No sampling)
Pima	**0.8254**	0.8248	0.8246	0.825	0.8247
Haberman	0.6873	0.6818	0.6873	**0.6874**	**0.6874**
Satimage	**0.7645**	0.7628	0.7632	0.7273	0.7635
Ecoli	**0.9296**	0.9262	0.9269	0.9214	0.9278
Shuttle	0.8902	**0.8938**	0.8687	0.8625	0.8767
Ionosphere	0.8661	**0.8986**	0.8682	0.8687	0.8696
Vehicle	**0.9882**	0.9863	0.9861	0.9872	0.9851
Credit	**0.7178**	0.6897	0.6976	0.7001	0.6976

TABLE 7
Ranking for Different Variations of MBS

Dataset	MBS	No modeling	MBS (Data Level Sampling)	MBS(Random Sampling)	MBS (No sampling)
Pima	**1**	3	5	2	4
Haberman	3	5	3	**1**	**1**
Satimage	**1**	4	3	5	2
Ecoli	**1**	4	3	5	2
Shuttle	2	**1**	4	5	3
Ionosphere	5	**1**	4	3	2
Vehicle	**1**	3	4	2	5
Credit	**1**	5	3	2	3

sampling for each feature is an effective step to improve performance.

Finally, we examined the performance differences of MBS at different iteration numbers to verify the effect of the iterative step used in the final step of MBS. Fig. 6 presents the average AUC on four datasets with different numbers of iterations. The experimental results indicate that the proposed method performs stably as the number of iterations increases, and could achieve better performance on the Vehicle dataset.

5.2 Visualization and Discussion

In this section, we attempt to use visualization results to explain the reason why MBS performs well on imbalanced datasets. In the following figures, we perform principal component analysis (PCA) to project the original data points onto two-dimensional space to present visualization results. Fig. 8a shows the scatter plot of the Pima dataset, in which the majority class is labeled as 0 and minority class is labeled as 1.

The SMOTE is a classical method for dealing with imbalanced data, and the key idea behind SMOTE is to find k nearest neighbors and generate synthetic samples along the line segment between each minority and its selected neighbor. The SMOTE considers the pairwise relationship between the minority data, which belongs to local information. In contrast, MBS considers global relationships by modeling all the minority data rather than only local information.

For example, if we want to generate a new data point between point A and point B as illustrated in Fig. 7, the line between A and B could be characterized by a function $x = 2y - 1$. Therefore, the nature of linear interpolation is to estimate the relationship between features by a straight line function connecting two points only. In contrast, MBS captures the relationship with multiple specialized models for each feature, and the models are based on the regression lines of overall minority data.

Besides, the SMOTE suffers from the over-generalization problem as presented in Fig. 8b, in which the synthetic data are deep green points on feature space. As compared with

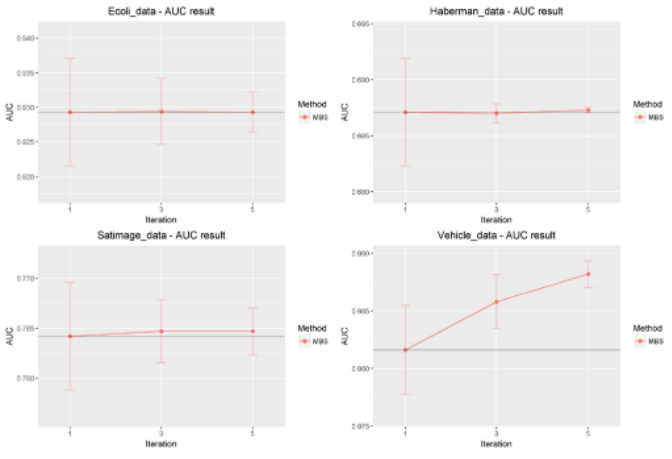


Fig. 6. Average AUC for different iterations.

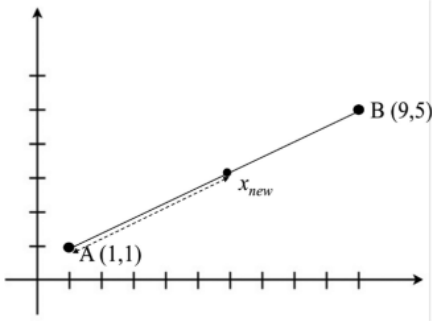


Fig. 7. Concept of interpolation.

Authorized licensed use limited to: University of South Florida. Downloaded on May 13,2023 at 03:30:20 UTC from IEEE Xplore. Restrictions apply.

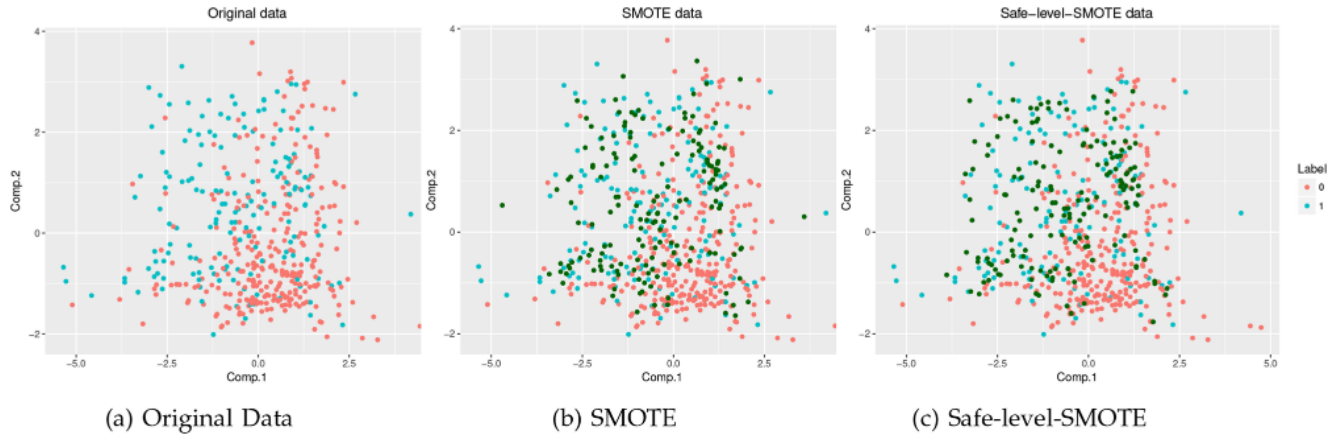


Fig. 8. Visualization of sampling data points on the Pima dataset.

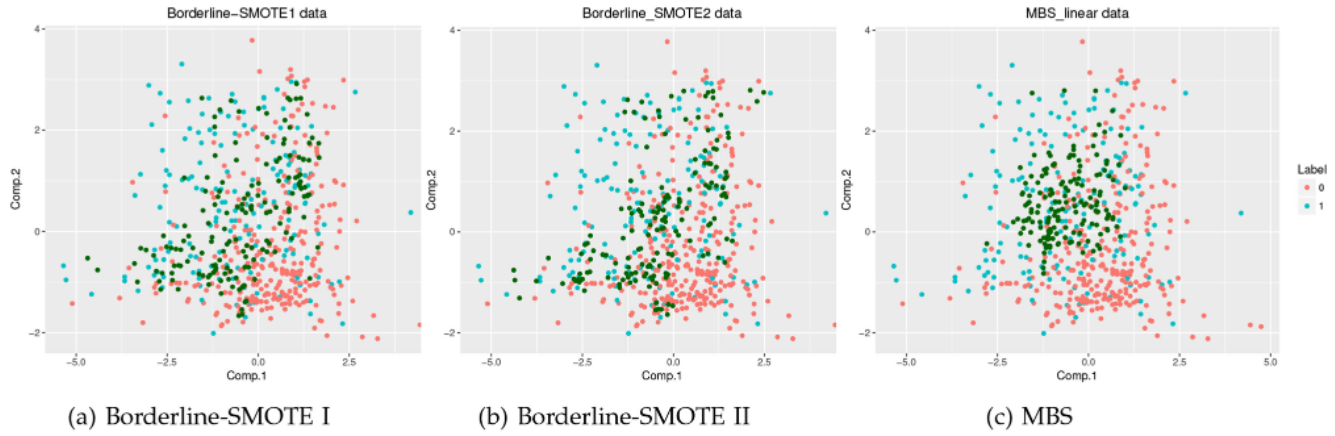


Fig. 9. Scatter plot after performing Borderline-SMOTE and MBS on the Pima dataset.

the original data points presented in Fig. 8a, a small number of synthetic samples are located in the area of the majority data.

Previous researchers have developed several extended methods of SMOTE, including safe-level-SMOTE and borderline-SMOTE. We applied these methods to the Pima dataset and present the visualization results in Figs. 8c, 9a, and 9b.

In Fig. 8c, safe-level-SMOTE generates most synthetic samples situated in the area of the majority data. As for borderline-SMOTE1 and borderline-SMOTE2 presented in Figs. 9a and 9b, they indeed generate the synthetic data close to the borderline, but also generate a small number of points that are located in the area of the majority data. In contrast, MBS performs better than those methods mentioned above as presented in Fig. 9c, which indicates that MBS generates synthetic data close to the borderline and also toward most minority data slightly. More importantly, MBS does not position any synthetic sample into the area of the majority data. As compared with other alternatives, the proposed method builds models to generate synthetic data, providing a base to learn trends of the features from data and enhance the diversity of data to avoid the drawback of resampling.

5.3 Integration of MBS and Ensemble Learning

The proposed method is a data-level algorithm, so it is easy and flexible to extend. Using the ensemble learning approach to deal with imbalance problems is popular, so we further combined the proposed MBS with the boosting technique to devise a method called MBSBoost, which is an integration of AdaBoost.M2 [45] and MBS. At each iteration of model training, MBS was performed to increase minority samples so that each weak classifier could be trained on a relatively balanced subset. We compared MBSBoost with state-of-the-art ensemble-based methods, including RUSBoost and UnderBagging. The experimental results are presented in Table 8, indicating that MBSBoost works well and stably on the thirteen datasets. Thus, the experimental

TABLE 8
Average AUC Results for Different Ensemble-Based Methods

Dataset	MBSBoost	RUSBoost	UnderBagging
Pima	0.8015	0.8079	**0.8125**
Haberman	0.6394	0.6682	**0.6816**
Satimage	**0.9498**	0.9486	0.9400
Ecoli	**0.9354**	0.9346	0.9302
Shuttle	0.9998	0.9997	**0.9999**
Ionosphere	**0.9705**	0.9661	0.9641
Vehicle	**0.9877**	0.9854	0.9761
Credit	0.6702	0.7073	**0.7631**
Diabetes	0.5567	0.5691	**0.5963**
Hmeq	**0.9233**	0.9216	0.9045
Promotion	0.5940	**0.6118**	0.5000
Bank	0.8622	**0.8691**	0.8567
Spambase	**0.9756**	0.9750	0.9665

Authorized licensed use limited to: University of South Florida. Downloaded on May 13,2023 at 03:30:20 UTC from IEEE Xplore. Restrictions apply.

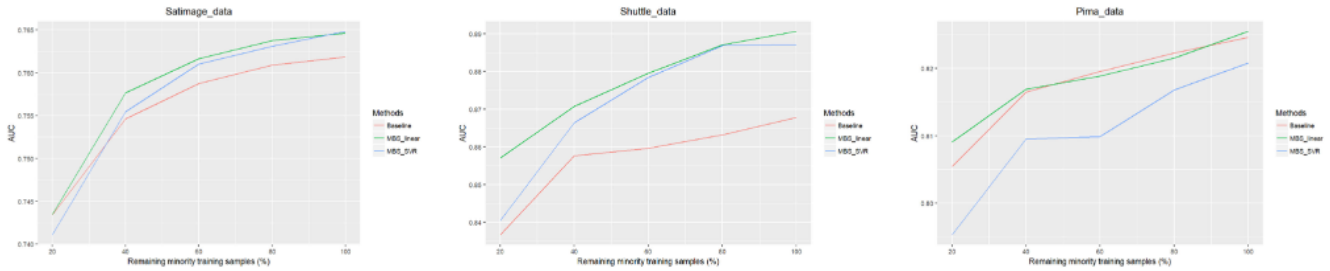


Fig. 10. Results of decreasing minority training samples experiment.

results for MBS and MBSBoost both show that the proposed MBS could generate synthetic samples with good quality.

5.4 Impact of Number of Minority Samples on Non-Linear Feature Model

The experimental results as presented in Table 3 indicate that MBS_CART and MBS_SVR, both of which using non-linear feature models in the MBS, perform worse than MBS_linear. The non-linear models generally require more training samples to learn model parameters, and always require a tuning of hyper-parameters to obtain accurate models. However, the number of training samples for the minority class is always limited in practical settings, so we conducted experiments to verify whether the number of training samples is an important factor when using non-linear regression models in the proposed framework.

The experimental settings were the same as those used in the previous experiments. The only one difference was that we decreased the minority training samples to explore the influence of different training sizes on the model performances. We set different ratios for the minority samples in the experiments, which were 100, 80, 60, 40, and 20 percent. For example, a ratio of 100 percent represents preserving all minority training samples, and a ratio of 80 percent means dropping out 20 percent of the minority training samples from the original minority samples. We selected SVR as our non-linear feature model and compared it with the baseline and linear feature model, in which the baseline is the model without over-sampling.

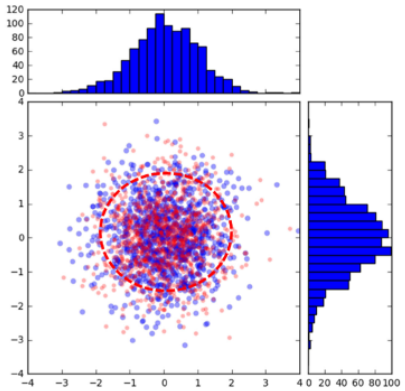
The results, which are shown in Fig. 10, indicate that the number of minority training samples has an effect on the performances of non-linear and linear feature models, but the influence on the linear model is much less than on non-linear model. Furthermore, the performance gap between the linear model and non-linear model reduces as more minority training samples become available, and increases as more samples are removed from the training set. We conclude that the linear regression model is a simple model that is more appropriate in situations where only a few training samples are available.

5.5 The Impact of Feature Relationship on Synthetic Samples

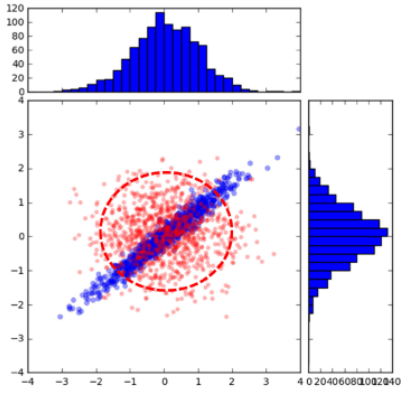
We design an experiment to discuss the impact of feature relationship on synthetic samples. To present the visualization results, we assume each data sample comprises two features. We sample features for simulated samples from two Gaussian distributions as shown in Fig. 11, which comprises two cases. In these two figures, the blue points represent the original samples and the red points are synthetic data samples that are generated by independently sampling each feature as used by Databoost-IM.

The first case assumes the two features are independent as presented in Fig. 11a, which shows the joint distribution of randomly generated data samples and the histograms in the figures represent the frequency distributions of the features. The second case assumes two features have positive correlation. Fig. 11b presents the results.

In Fig. 11a, scatter plot of the sampled points is consistent with that of the original data samples, since the features of the original data samples are independent, and the sampling follows the same assumption. In contrast, when the features are dependent but we ignore the relationship, the generating synthetic samples in Fig. 11b are unreasonable. The synthetic points tend to be present in the center of



(a) The joint distribution of the simulated samples with independent features



(b) The joint distribution of the simulated samples with dependent features

Fig. 11. Illustration of simulated samples.

Authorized licensed use limited to: University of South Florida. Downloaded on May 13,2023 at 03:30:20 UTC from IEEE Xplore. Restrictions apply.

the two Gaussian distributions, and many synthetic points are located in the area of low density.

Consequently, considering the feature relationship in generating synthetic examples is important, and this work proposes to use linear regression to capture the trend of the features. Notably, the motivation of our proposed method is to generate diverse synthesis examples and retain the original characteristics by using regression model to capture the relationship of features rather than complex patterns embedding in the data samples.

6 Conclusion

Imbalance problems have occurred in various application domains and received a considerable amount of attention recently. This work proposes a new framework called MBS to cope with imbalance problems. The proposed work integrates sampling and modeling techniques to generate synthetic data, and the generating process involves three steps. We conducted experiments on thirteen datasets and compare the proposed method with ten competitive methods. The experimental results indicate that the proposed method outperforms other alternatives in most cases in terms of effectiveness and robustness.

The proposed method is a framework, and many future research directions are possible, such as applying other regression models and other classifiers to different application domains for further analysis. Moreover, the effect of different over-sampling rates needs to be explored to find the most appropriate setting according to various conditions. Finally, although we conducted experiments on several datasets in this work, it is worthwhile to focus on a certain application domain to deeply explore the relationship between features to refine the feature model structure.

Acknowledgments

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant no. MOST 107-2221-E-009-109-MY2 and MOST 107-2218-E-009-005.

References

[1] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Progress Artif. Intell.*, vol. 5, no. 4, pp. 221–232, 2016.

[2] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[3] Q. Yang and X. Wu, "10 challenging problems in data mining research," *Int. J. Inf. Technol. Decision Making*, vol. 5, no. 04, pp. 597–604, 2006.

[4] K. S. Woods, C. C. Doss, K. W. Bowyer, J. L. Solka, C. E. Priebe, and W. P. Kegelmeyer Jr, "Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 7, no. 06, pp. 1417–1436, 1993.

[5] S. Mahadevan and S. L. Shah, "Fault detection and diagnosis in process data using one-class support vector machines," *J. Process Control*, vol. 19, no. 10, pp. 1627–1639, 2009.

[6] I. Kononenko, "Machine learning for medical diagnosis: History, state of the art and perspective," *Artif. Intell. Med.*, vol. 23, no. 1, pp. 89–109, 2001.

[7] L.-J. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1506–1518, Nov. 2003.

[8] P. K. Kankar, S. C. Sharma, and S. P. Harsha, "Fault diagnosis of ball bearings using machine learning methods," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 1876–1886, 2011.

[9] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 20–29, 2004.

[10] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 04, pp. 687–719, 2009.

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[12] M. Kubat, S. Matwin, et al., "Addressing the curse of imbalanced training sets: One-sided selection," in *Proc. Int. Conf. Mach. Learn.*, 1997, pp. 179–186.

[13] N. Japkowicz, "The class imbalance problem: Significance and strategies," in *Proc. Int. Conf. Artif. Intell.*, 2000, pp. 111–117.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[15] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Advances Intell. Comput.*, 2005, pp. 878–887.

[16] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2009, pp. 475–482.

[17] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2008, pp. 1322–1328.

[18] T. Jo and N. Japkowicz, "Class imbalances versus small disjuncts," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 40–49, 2004.

[19] S.-J. Yen and Y.-S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5718–5727, 2009.

[20] Z. Xie, L. Jiang, T. Ye, and X. Li, "A synthetic minority oversampling method based on local densities in low-dimensional space for imbalanced learning," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2015, pp. 3–18.

[21] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. Nov., pp. 2579–2605, 2008.

[22] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "DBSMOTE: Density-based synthetic minority over-sampling technique," *Appl. Intell.*, vol. 36, no. 3, pp. 664–684, 2012.

[23] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.

[24] H. Cao, X.-L. Li, Y.-K. Woon, and S.-K. Ng, "Integrated oversampling for imbalanced time series classification," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 12, pp. 2809–2822, Dec. 2013. [Online]. Available: https://doi.org/10.1109/TKDE.2013.37

[25] Z. Gong and H. Chen, "Model-based oversampling for imbalanced sequence classification," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 1009–1018.

[26] G. Wu and E. Y. Chang, "KBA: Kernel boundary alignment considering imbalanced data distribution," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 786–795, Jun. 2005.

[27] M. Ohsaki, P. Wang, K. Matsuda, S. Katagiri, H. Watanabe, and A. Ralescu, "Confusion-matrix-based kernel logistic regression for imbalanced data classification," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 9, pp. 1806–1819, Sep. 2017.

[28] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3573–3587, Aug. 2018.

[29] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5375–5384.

[30] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *CoRR*, vol. abs/1804.10851, 2018. [Online]. Available: http://arxiv.org/abs/1804.10851

[31] B. Raskutti and A. Kowalczyk, "Extreme re-balancing for SVMs: A case study," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 60–69, 2004.

[32] L. M. Manevitz and M. Yousef, "One-class SVMs for document classification," *J. Mach. Learn. Res.*, vol. 2, pp. 139–154, Mar. 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=944790.944808

Authorized licensed use limited to: University of South Florida. Downloaded on May 13,2023 at 03:30:20 UTC from IEEE Xplore. Restrictions apply.

[33] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *J.-Japanese Soc. Artif. Intell.*, vol. 14, no. 771–780, 1999, Art. no. 1612.
[34] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[35] R. Polikar, "Ensemble learning," in *Ensemble Machine Learning*. Berlin, Germany: Springer, 2012, pp. 1–34.
[36] N. Chawla, A. Lazarevic, L. Hall, and K. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Proc. Eur. Conf. Principles Data Mining Knowl. Discovery*, 2003, pp. 107–119.
[37] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Proc. Eur. Conf. Comput. Learn. Theory*, 1995, pp. 23–37.
[38] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.
[39] H. Guo and H. L. Viktor, "Learning from imbalanced data sets with boosting and data generation: The DataBoost-IM approach," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 30–39, 2004.
[40] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)*, vol. 39, no. 2, pp. 539–550, Apr. 2009.
[41] B. Wang and J. Pineau, "Online bagging and boosting for imbalanced data streams," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3353–3366, Dec. 2016.
[42] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)*, vol. 42, no. 4, pp. 463–484, Jul. 2012.
[43] H. Guo and H. L. Viktor, "Boosting with data generation: Improving the classification of hard to learn examples," in *Proc. Int. Conf. Ind. Eng. Other Appl. Appl. Intell. Syst.*, 2004, pp. 1082–1091.
[44] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *Data Mining and Knowledge Discovery Handbook*. Berlin, Germany: Springer, 2005, pp. 853–867.
[45] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Mach. Learn.*, vol. 37, no. 3, pp. 297–336, 1999.

Chien-Liang Liu received the MS and PhD degrees from the Department of Computer Science, National Chiao Tung University, Taiwan, in 2000 and 2005, respectively. He is currently an associate professor with the Department of Industrial Engineering and Management, National Chiao Tung University, Taiwan. His research interests include machine learning, data mining, and big data analytics. He is a member of the IEEE.

Po-Yen Hsieh received the BS degree from the Department of Psychology, National Chung Cheng University, Taiwan, in 2015, and the MS degree from the Department of Industrial Engineering and Management, National Chiao Tung University, Taiwan, in 2017. His research interests include machine learning and data mining.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.

Authorized licensed use limited to: University of South Florida. Downloaded on May 13,2023 at 03:30:20 UTC from IEEE Xplore. Restrictions apply.