Decision Support Systems 106 (2018) 15–29

Contents lists available at ScienceDirect

Decision Support Systems

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

A synthetic informative minority over-sampling (SIMO) algorithm leveraging support vector machine to enhance learning from imbalanced datasets

Saeed Piri [a], Dursun Delen [b,*], Tieming Liu [c]

[a] Department of Management Science and Information Systems, Spears School of Business, Oklahoma State University, Stillwater, OK 74078, USA
[b] Department of Management Science and Information Systems, Center for Health Systems Innovation, Spears School of Business, Oklahoma State University, Tulsa, OK 74106, USA
[c] Department of Industrial Engineering and Management, College of Engineering, Architecture and Technology, Oklahoma State University, Stillwater, OK 74078, USA

ARTICLE INFO

Article history:
Received 23 June 2017
Received in revised form 17 October 2017
Accepted 25 November 2017
Available online 29 November 2017

Keywords:
Predictive modeling
Machine learning
Imbalanced data
Over-sampling
Support vector machines
Performance metrics

ABSTRACT

Developing decision support systems (DSS) based on imbalanced datasets is one the critical challenges in data mining and decision-analytics. A dataset is called imbalanced when the number of examples from one class outnumbers the number of the instances from another class. Learning from imbalanced datasets is one of the major challenges in machine learning. While a standard classifier could have a very good performance on a balanced dataset, when applied to an imbalanced dataset, its performance deteriorates dramatically. This poor performance is rather troublesome, especially in detecting the minority class, which usually is the class of interest. Therefore, the poor performance of machine learning techniques, which are used to develop DSS, negatively affect the practicality of DSS in real word problems. Over-sampling the minority class is one of the most promising remedies for imbalanced data learning. In this study, we propose a new synthetic informative minority over-sampling (SIMO) algorithm leveraging support vector machine (SVM). In this algorithm, first SVM is applied to the original imbalanced dataset, then, minority examples close to the SVM decision boundary, as the informative minority examples are over-sampled. We also developed another version of SIMO and call it weighted SIMO (W-SIMO). W-SIMO is different from SIMO in the degree of over-sampling the informative minority examples. In W-SIMO, incorrectly classified informative minority examples are over-sampled with a higher degree compared to the correctly classified informative minority examples. In this way, there is more focus on incorrectly classified minority examples. The over-sampled dataset can be used to train any classifier. We applied these algorithms to the 15 publicly available benchmark imbalanced datasets and assessed their performance in comparison with existing approaches in the area of imbalanced data learning. The results showed that our algorithms had the best performance in all datasets compared to other approaches.

1. Introduction

Developing decision support systems (DSS) based on imbalanced datasets is one the critical challenges in data mining and decision making. A dataset is called imbalanced when the distribution of different classes in the data is not similar. For instance, in the case of two-class data, there are many more examples of one class (negative examples) compared to the other class (positive examples). Let us call the class with fewer examples the minority class, and the class with more examples the majority class.

Decision-making based on imbalanced datasets is very common in real-life problems, especially in decision support systems that are based on classification. For example, if a sample of people were tested for a specific disease, only a small portion of them would actually have the disease. Therefore, in building of the most of clinical decision support systems we deal with imbalanced data sets. For instance, Piri et al. [1] developed a clinical DSS for diabetic retinopathy. In their data, only 5% of the patients had retinopathy and 95% of them did not have the disease. There are several other clinical DSS that are developed based on imbalanced datasets, predicting heart transplantation outcomes through data analytics Dag et al. [2] predicted heart transplantation outcomes by analyzing imbalanced data. Finance is another area that extensively deals with decision-making based on analyzing imbalanced datasets. For instance, in credit card fraud detection, only a few numbers of transactions in the whole sample of transactions are actually fraud [3]. Another example is predicting bankruptcy in medium-sized enterprises by learning from

* Corresponding author at: Regents Professor of Management Science and Information Systems, Spears and Patterson Endowed Chairs in Business Analytics, Director of Research—Center for Health Systems Innovation, Spears School of Business, Oklahoma State University, 700 North Greenwood Ave., Tulsa, OK 74106, USA.
E-mail addresses: saeed.piri@okstate.edu (S. Piri), dursun.delen@okstate.edu (D. Delen), tieming.liu@okstate.edu (T. Liu).

https://doi.org/10.1016/j.dss.2017.11.006
0167-9236/

imbalanced data [4]. In imbalanced datasets, the prediction accuracy, especially for the minority class, is a critical challenge. When the standard machine learning techniques are applied to the imbalanced data, the result will be in favor of the majority class, i.e. a big portion of the minority class examples will be classified as the majority. In real world applications, the detection accuracy of the minority class is critically important because the minority class usually is the class of interest. Thus, misclassifying the minority class has much higher cost compared to misclassifying a majority class example. To make it clearer, compare the cost of misclassifying a cancerous patient as non-cancerous to the cost of misclassifying a non-cancerous as cancerous; in the former case, the misclassification may lead to death of a person but in the latter case, there will be some more tests and screenings. All of the aforementioned examples, indicate the extremely importance role of imbalanced data learning in the performance of the decision support systems built on imbalanced datasets.

There are various approaches to improve the performance of predictive modeling in imbalanced datasets. Sampling methods are those that either increase the number of minority examples by generating synthetic examples, or decrease the number of majority examples by removing some of them. Another popular approach is assigning different misclassification costs for various classes; this approach is called cost-sensitive. There are other methods and algorithms that we cover in the Literature review section. In this research, we propose a novel synthetic informative minority over-sampling (SIMO) algorithm, which employs support vector machine (SVM) to enhance learning from imbalanced datasets. Here we discuss why we chose over-sampling versus other methods to handle the imbalanced data learning challenge, and why we chose SVM. First, to apply a sampling method, no extra information is required other than the dataset itself [5]. However, in cost-sensitive methods, the information about the misclassification cost for each class is required, while this kind of information is unknown. The only known fact is that the misclassification cost for minority class is higher than misclassification cost for majority class [6,7]. Second, we apply over-sampling versus under-sampling. The major limitation in under-sampling is the possibility of losing important information by removing some parts of the data, while there is not such a problem in over-sampling.

There are three main reasons for choosing SVM as the classifier. First, this method has a very strong and at the same time simple theoretical background which makes it easy to explain intuitively [8]. Second, this method develops a hyperplane (decision boundary) that separates the data space for classifying the data points (examples). It is known that the data points near the decision boundary are more important and difficult to classify [9]. Therefore, identifying the near boundary data samples is rather easy in SVM. Finally, SVM has been shown to have a very good performance and high generalization power in many practical applications compared to other machine learning techniques [8,10].

The proposed algorithm, SIMO, generates synthetic minority data points that are located near the boundary between two classes in the data space. After applying SIMO in an imbalanced dataset, the number of minority class data points will be increased and the dataset will be more balanced. In this research, we developed another version of SIMO, which we call weighted SIMO (W-SIMO). In W-SIMO, after identifying the informative minority examples, they are grouped into two categories. First, those that are correctly classified by the SVM, and second, those that are incorrectly classified by the SVM. At the over-sampling stage, more data points are generated in the space of the minority data examples that are misclassified. The over-sampled dataset through SIMO and W-SIMO can be used by other machine learning techniques and it is not limited only to the SVM. We expected that our proposed algorithms would have a competitive performance in imbalanced data learning compared to other existing methods. To test this hypothesis, we performed extensive numerical experiments and provide the results of this analysis.

The remainder of this manuscript is organized as follows. In Section 2, existing methods and algorithms in the area of imbalanced data learning are reviewed. In Section 3, we provide the SVM formulation and discuss its deficiency in imbalanced datasets. In Section 4, SIMO and W-SIMO algorithms and their characteristics are described. Section 5 provides numerical analysis to assess the performance of the SIMO and W-SIMO compared to other existing algorithms. Finally, Section 6 contains the conclusion and discussion about this study.

2. Literature review

Studying the imbalanced data classification has received a considerable amount of attention in recent years. He and Garcia [7] classified the different approaches of analyzing imbalanced data into four main classes,

- Sampling methods
- Cost-sensitive methods
- Kernel-based methods and active learning methods
- Other methods such as, one-class learning, novelty detection, etc.

In this section, we briefly review the research studies that are the most related to our study.

2.1. Sampling methods

The aim of the sampling methods is to reach some degree of balanced distribution in the dataset. These methods can be categorized into two major streams, those that under-sample the majority class and those that over-sample the minority class. In under-sampling methods, some parts of the majority examples are removed. As a result, the distribution of the classes will be more balanced. The simplest method in this category is the random under-sampling. There is not any specific mechanism for under-sampling in this approach and it functions merely randomly. Other under-sampling approaches such as BalancedCascade and EasyEnsemble presented by Liu et al. [11] are called informed under-sampling. In EasyEnsemble, several samples of the majority class data are taken and combined with minority class data. Multiple models are built based on these datasets, and at the end an ensemble model makes the final decision. The main criticism of the under-sampling methods is that by removing some parts of the data, potential important information in the data can be lost.

Over-sampling on the other hand, is to re-sample or generate extra examples of the minority class. The most basic over-sampling method is random over-sampling in which minority examples in the data are randomly duplicated. The main downside of random over-sampling is over-fitting. Another major approach in over-sampling is synthetic data generation. SMOTE (Synthetic Minority Over-Sampling Technique) is one of the most well-known methods in synthetic data generation. In this method, synthetic data points are generated on the line connecting the minority samples to their k nearest minority class neighbors [12]. The major drawback in SMOTE is that it may lead to over-generalization because it blindly generates synthetic data points without considering the majority data points that might be located near the minority examples. This over-generalization might lead to overlapping between classes [13].

There are extensions to the SMOTE that tried to improve the performance of this technique. Han et al. [14] proposed a synthetic over-sampling method named Borderline-SMOTE. In this method, only a subset of minority data points is over-sampled by SMOTE technique. Those minority data points are located near the border of two classes. Borderline minority data points are identified as minority examples that most of their nearest neighbors belong to the majority class. One limitation for Borderline-SMOTE is its mechanism for identifying the borderline and noise data points. In this method, a minority data is identified as noise, only if all of its neighbors are majority. However, in cases that there only two minority data points surrounded by majority examples, Borderline-SMOTE consider them borderline, while they are obviously noises. On the other hand, Bunkhumpornpat et al. [15] introduced a method named Safe-Level SMOTE. This method calculates a parameter

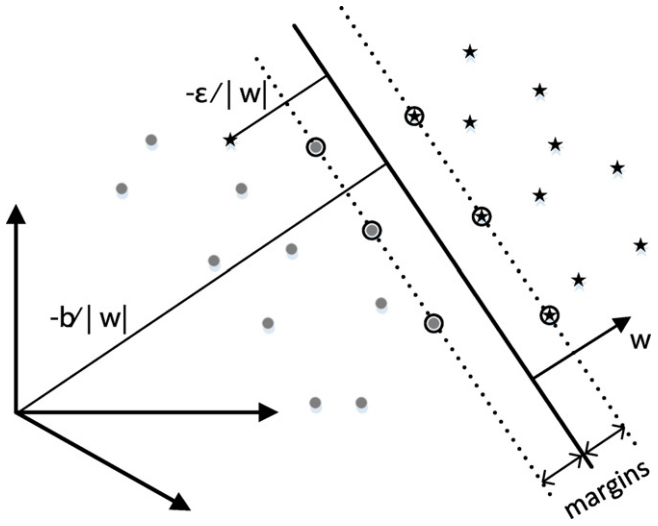


Fig. 1. Linear SVM hyperplane.

called safe-level. The greater that a safe-level is for a minority example shows that example is farther away from the borderline. After identifying the minority examples in safe regions, those data points will be over-sampled using SMOTE. Safe-level SMOTE focuses on the minority data points that are in the safe regions of the data space, while those are the data points that are easy to classify, and the main challenge is classifying the minority examples near the classes boundary. Cieslak et al. [16] introduced the cluster SMOTE method. This method first clusters the minority examples, and then over-samples data points within each cluster by applying SMOTE.

Barua et al. [17] proposed a majority weighted minority oversampling technique that first identifies hard to learn minority examples by considering their distance from the majority neighbors, and then it over-samples those examples using a clustering approach. Sáez et al. [18] suggested a framework called SMOTE-IPF. In their framework, the data is first over-sampled by SMOTE method and then noisy data points are filtered by applying iterative-partitioning filter (IPF) method. Through a series of numerical experiments, they showed the efficiency of their framework. There are other studies in the area of synthetic data generation [14,19,20]. Generally speaking, synthetic oversampling significantly improves the classification accuracy, especially for the minority class [21]. Another advantage is that by generating the synthetic minority data (not simply replicating existing minority data), the minority region is generalized and overfitting can be avoided [22]. For a more comprehensive review of the sampling methods, we refer readers to He and Garcia [7].

2.2. Cost-sensitive methods

Unlike sampling methods that alter the distribution of the data through either generating synthetic minority data points or removing some portion of majority data points, the idea of cost-sensitive methods is based on the different misclassification costs for different classes in the dataset. Usually the cost of misclassifying the minority class is much higher than the majority class misclassification [23]. To perform cost-sensitive methods, a matrix, called cost matrix is required. This matrix shows the misclassification cost for different classes in the dataset [24]. The main concern about cost-sensitive methods is that in most of

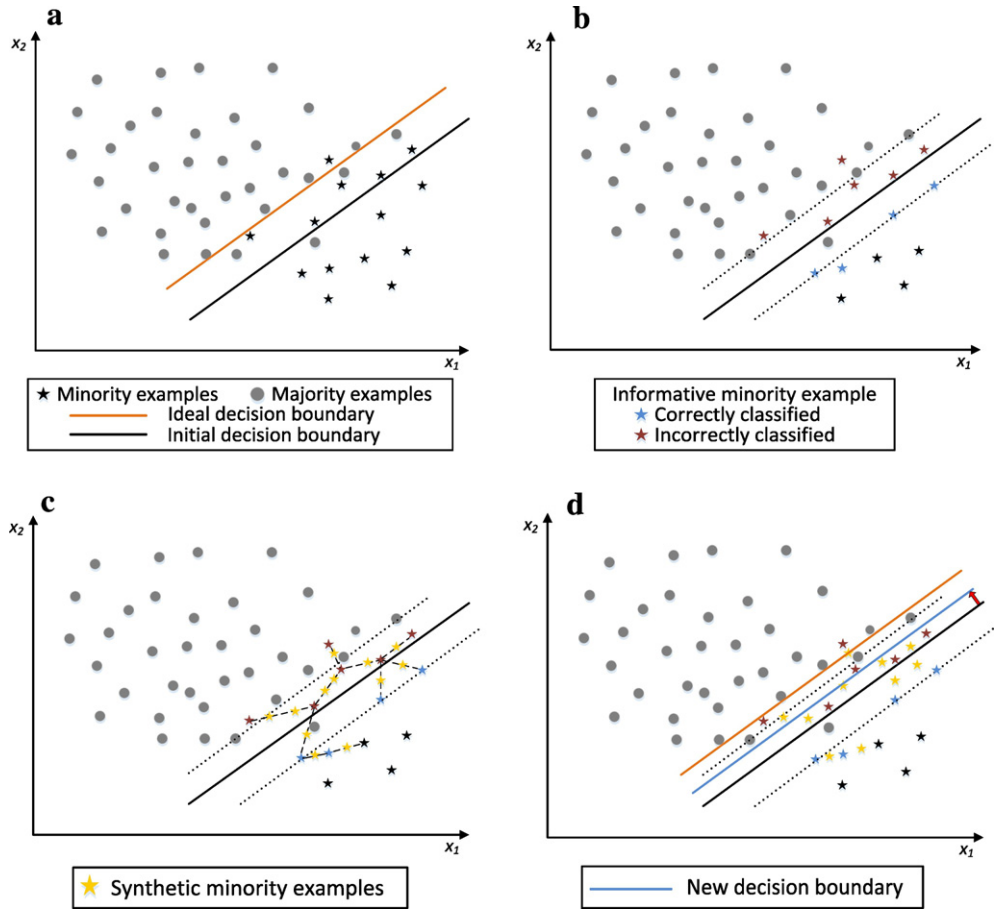
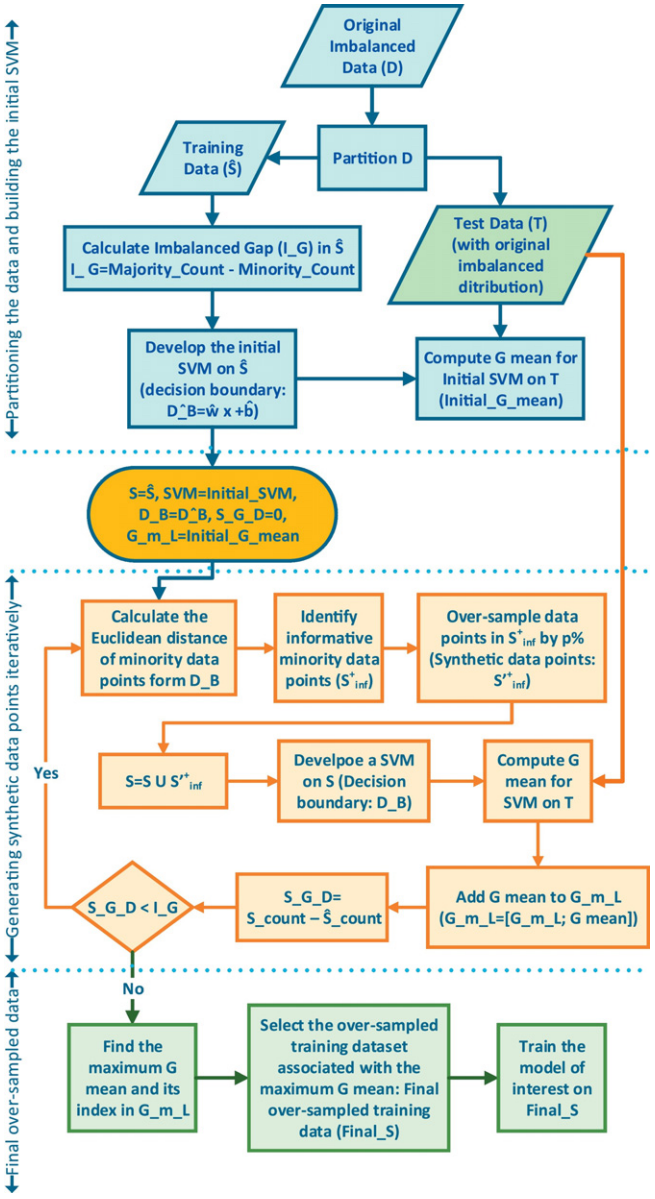
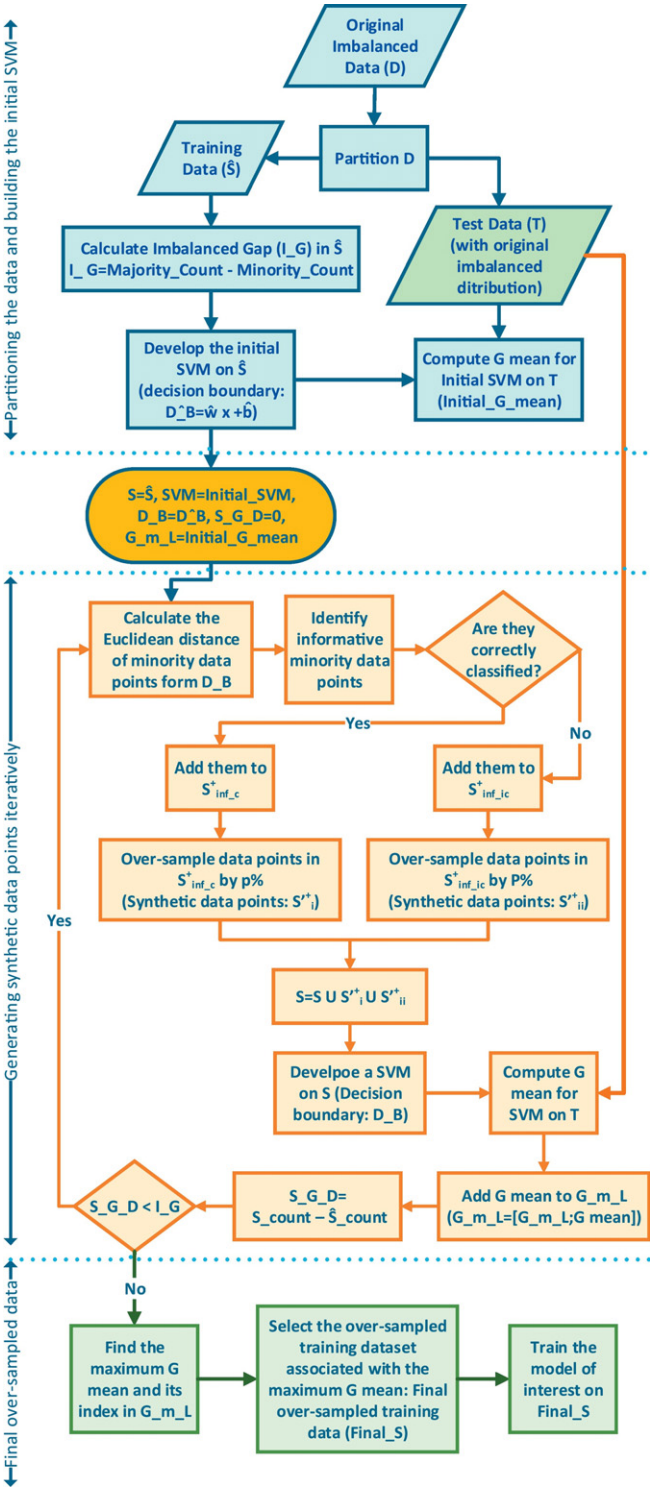


Fig. 2. W-SIMO algorithm mechanism (simplified).



Flowchart 1. SIMO algorithm.



Flowchart 2. W-SIMO algorithm.

the situations the exact misclassification cost related to various classes is unknown [6].

There are three major categories in cost sensitive approaches [7]. The first category includes techniques that assign various weights to the examples in the dataspace. Methods in this category are motivated by the AdaBoost algorithm [25]. AdaBoost is a meta-algorithm that begins with the original dataset and trains a model on this dataset. Incorrectly classified examples are identified, and in the next iteration more weight (higher error cost) will be assigned to them. In this way, more focus will be on the examples that are misclassified. This process repeats and the classifier performance improves. The second group encompasses approaches are those that use ensemble schemes integrated with cost-sensitive approaches. Many of the research studies in these two categories have combined various weighting and adaptive boosting techniques. For instance Sun et al. [26] and Fan et al. [27] proposed algorithms for updating the weights in AdaBoost in imbalanced data learning. Lee et al. [28] used SVM to adjust the weights of the examples in AdaBoost to learn from imbalanced data. In the third category, cost-sensitive methods incorporate the misclassification costs directly into the classifiers. Cost-sensitive decision tree [29], cost-sensitive neural networks [30], and cost-sensitive SVM [31] are in this category.

2.3. Kernel-based methods

Kernel-based methods are mostly integrated with SVM. Many researchers have studied imbalanced data learning through support vector machine. Wu and Chang [32] developed a boundary-alignment algorithm, which makes a change in the kernel function to move the

boundary toward the negative instances. Akbani et al. [33] proposed an algorithm by integrating the different error cost method [31] and the SMOTE over-sampling method, however they performed the SMOTE over-sampling independent from the SVM model. Wang and Japkowicz [34] applied boosting and asymmetric error cost for minority and majority classes. Mathew et al. [35] proposed a kernel-based SMOTE for SVM. In their approach, the over-sampling through the SMOTE technique happens in kernel feature space. Yu et al. [10] developed the SVM-OTHR algorithm. In this algorithm, they adjusted the decision threshold by moving the decision hyperplane toward the majority class data. Lee et al. [28] proposed an improved weighted support vector machine for imbalanced data learning. Jian et al. [36] developed a sampling framework by using support vector machine. They over-sampled minority data points and under-sampled majority data point, and classified the examples by an ensemble of support vector machines. To enhance the imbalanced data learning performance, Shao et al. [37] proposed a weighted Lagrangian twin support vector machine. They introduced a graph based under-sampling in their algorithm. However, their approach is very sensitive to its parameters and it takes a long time to find the reasonable values for the parameters, therefore it is not very easy and practical to use their approach in real word problems.

Tang and Zhang [38] proposed a granular SVM with repetitive under-sampling. They utilized SVM for under-sampling in a way that they repeatedly developed SVM models and each time discarded the negative (majority class) support vectors from the data. Even though they performed the under-sampling integrated with the SVM, the problem of losing potential important information by under-sampling still exists. As Akbani et al. [33] showed in their paper, under-sampling the majority class may decrease the total error, but it usually deteriorates the performance of the SVM on the test data, because it fails to approximate the orientation of the ideal hyperplane. Batuwita and Palade [39] suggested an over-sampling method in which they selected the majority examples near the boundary as the informative negative data points, and then they randomly over-sampled the minority examples to have relatively balanced data. This work can be critiqued in two ways. First, they focused on the informative majority examples, while the primary interest in imbalanced datasets is on the minority examples, therefore the focus on the informative majority examples may lead to even more bias toward the majority class. Second, they simply applied random over-sampling that is not as powerful as synthetic data generation methods and may lead to over-fitting. The two former studies did not compare their model's performance with other existing methods; therefore, it is not easy to comment on generalizability and efficiency of their model [40]. Proposed a preprocessing approach using SVM for imbalanced data. In their approach, they first trained SVM on the original data, and then replaced the actual target variable value by the SVM predicted value. They claimed that SVM will classify a portion of the majority examples as minority, and therefore the processed data will have a more balanced distribution. Their claim is questionable, because in imbalanced data learning most of the time there is poor accuracy on minority class and good accuracy on majority. This means that most of the minority examples are misclassified as majority not the other way around. They tested their approach only on one dataset; therefore, their results could be because of the characteristics of that special dataset.

In this research, we propose a novel over-sampling algorithm leveraging SVM. We can numerate several advantages for our proposed algorithm. First, it leverages a powerful classifier, i.e. SVM, and therefore better results are expected compared to other pre-processing approaches. Second, we conduct over-sampling rather than under-sampling that may lead to information loss due to discarding a fraction of data. Finally, we perform the over-sampling only on the informative minority examples. In this way, we generate the least amount of synthetic data points; therefore, the distribution of the training data will not change dramatically. In addition, because the amount of synthetic generated data is much less compared to other existing methods such as SMOTE, Borderline SMOTE, Safe-Level SMOTE, and Cluster-SMOTE, the computational cost of training machine learning techniques will be lower.

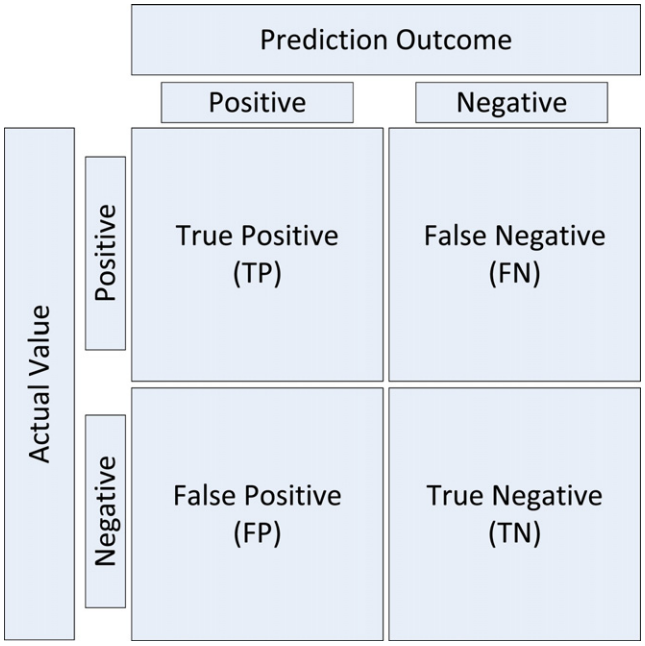


Fig. 3. Confusion matrix.

3. Support vector machines (SVM)

SVM is a machine learning technique that can be applied to both regression and pattern recognition (classification) problems. For the classification, SVM develops a decision boundary that separates two classes in the data space. To build this decision boundary, SVM maximizes the separating margin between two classes in the data space while it minimizes the classification error. Fig. 1 shows a linear SVM decision

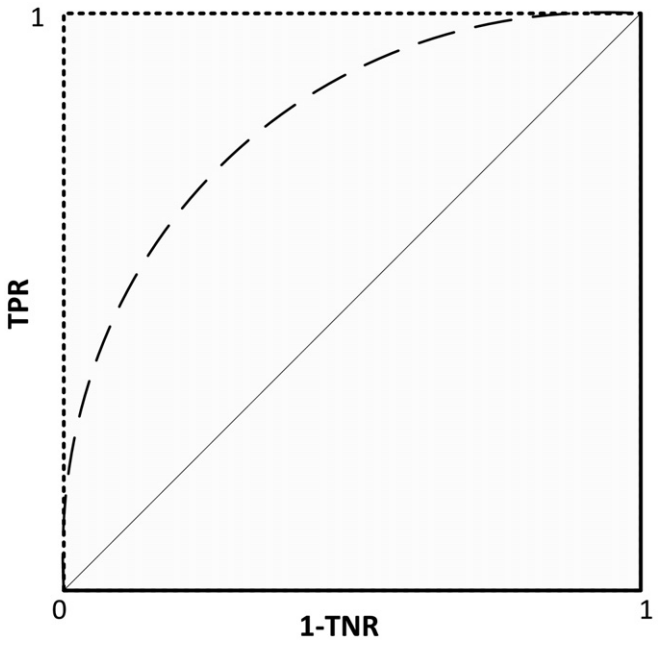


Fig. 4. ROC chart.

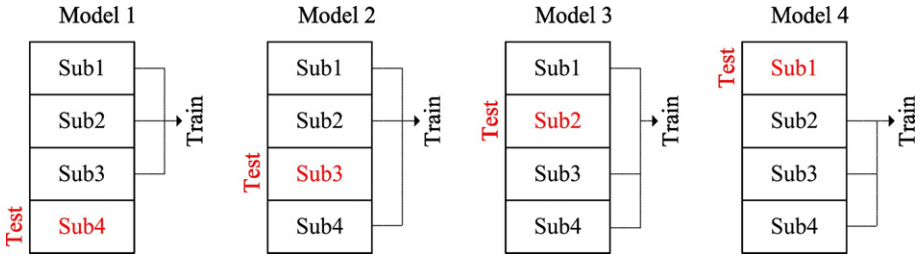


Fig. 5. 4-Fold cross validation mechanism.

boundary. Dots and stars denote the two classes in the data. The data points that lie on the margins at both sides of the decision boundary are called support vectors. These support vectors are shown in Fig. 1 with a circle around them. w is the normal to the decision boundary and $b/|w|$ is the perpendicular distance of the decision boundary from the origin [41]. When two classes are not completely separable, some of the examples will be misclassified. In Fig. 1, one star data point has misclassified as a dot, the distance of this point from the decision boundary is $-\varepsilon/|w|$.

SVM can be applied to both linear and non-linear separable problems. When two classes are not linearly separable, kernel trick can be employed and the data is mapped to a feature space (using a mapping function $\phi(.)$), which is in a higher dimension [42]. In the feature space, two classes will be linearly separable and the problem will be handled similar to the linearly separable case.

The decision boundary of the SVM has the following formulation (Formulation (1)),

$$w^T\phi(x) + b \tag{1}$$

where w is obtain from Formulation (2),

$$w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \tag{2}$$

To determine the class of a new sample, x, a sign function (sgn(.)) is used, it is obtained using,

$$y = \ sgn\{w^T\phi(x) + b\} \tag{3}$$

3.1. SVM on imbalanced datasets

Although SVM has a very good performance on balanced datasets, when applied to imbalanced datasets, its performance deteriorates dramatically, especially on the minority class. The SVM decision boundary in an imbalanced dataset is closer toward the minority class region compared to the ideal classification decision boundary. As a result, a considerable number of minority class examples will be misclassified as the majority. Wu and Chang [43] mentioned two reasons for this decision boundary skewness. The first reason is in regard to the imbalanced training data ratio, because the negative data points outnumber the positive examples, these positive examples are further away from the "ideal" decision boundary compared to the majority examples. Second, the imbalanced supports vector ratio, because the number of the negative (majority class) support vectors is much more than the positive (minority class) support vectors, a positive test data point might have more negative support vector neighbors, and as a result will be misclassified as negative (majority) class. Akbani et al. [33] pointed out another reason for the skewed decision boundary. The objective of the SVM model is to maximize the margin between two classes as well as minimizing the classification errors and there is a tradeoff between these two. When the number of negative examples is much more than the positive ones, the cumulative misclassification cost of the positive points is relatively small, therefore SVM tends to maximize the margin to its highest possible degree by classifying most (sometimes all) of the examples as negative. Thus, the decision boundary will be shifted toward the minority class region. In the next section, we describe our proposed remedy to this problem.

4. SIMO and W-SIMO algorithms

In this study, we developed a novel synthetic informative minority over-sampling (SIMO) algorithm integrated with SVM. As we mentioned earlier, when SVM is applied to an imbalanced dataset, the decision boundary will be closer to the minority class space in favor of the majority class examples. Therefore, a considerable portion of minority examples will be misclassified. In SIMO, we generate synthetic data points that belong to the minority class. In this way, the distribution of the dataset will be more balanced and a better performance will be expected from machine learning techniques. Research has shown the data points that are close to the boundary of classes are the important data points in forming the classifiers [7]. Therefore, in SIMO we focus on the minority data points near the boundary of two classes.

The first step in performing SIMO (Algorithm 1 and Flowchart 1) is to partition the dataset into training and test datasets. This partitioning is conducted in a way that the imbalance ratio in training and test datasets will be the same as the imbalance ratio in the original dataset. The reason for partitioning the data is to avoid biases and to assess the SIMO performance fairly on imbalanced data with the original imbalance ratio (test dataset). Next, we calculate the imbalanced gap in the training dataset. Imbalanced gap is the difference between the number of majority examples and minority examples in the training dataset. Imbalanced gap is the upper bound for generating the synthetic data points in our algorithm. In the next stage, we develop a SVM on the original imbalanced training dataset and evaluate this initial model by computing the G mean. G mean is an evaluation metric that is widely used in imbalanced data learning. More details about G mean and its

Table 1
Notations for Algorithms 1 and 2.

D: Initial imbalanced dataset
$\hat{S}$: Initial imbalanced training dataset
T: Imbalanced test dataset
Δ: Top Δ% of minority data points close to decision boundary
p: Oversampling degree for minority informative data points that are correctly classified at each iteration
P: Oversampling degree for minority informative data points that are incorrectly classified at each iteration
S_G_D: Synthetic generated data points count
G_m_L: ***G mean*** variation log in each iteration

Table 2
Benchmark datasets characteristics.

Dataset	Minority class	Majority class	# of variables	# of records	Imbalance ratio
Liver disorders (liver)	"1"	"2"	7	345	1:1.38
Ionosphere	Bad	Good	34	351	1:1.79
Pima Indians Diabetes (Pima)	"1"	"0"	8	768	1:1.87
Breast Cancer Wisconsin Original (BreastCO)	Malignant	Benign	10	699	1:1.91
Iris	Versicolor	All other	5	150	1:2
Yeast	NUC	All other	8	1484	1:2.6
Statlog Vehicle Silhouettes (vehicle)	Van	All other	18	846	1:3.25
Contraceptive Method Choice (CMC)	Long-term	All other	9	1473	1:3.42
Breast Cancer Wisconsin_20% (BreastC20)	Malignant	Benign	10	699	1:3.91
Connectionist Bench_Vowel Recognition (vowel)	"0" & "1"	All other	11	990	1:4.5
Ecoli	pp	All other	8	336	1:5.46
Libras Movement_12 (Libras12)	"1" & "2"	All other	91	360	1:5.88
Libras Movement_34 (Libras34)	"3" & "4"	All other	91	360	1:6.34
Glass identification (glass)	"7"	All other	9	214	1:6.38
Breast Cancer Wisconsin_10% (BreastC10)	Malignant	Benign	10	699	1:8.9

computation is provided in Section 5.1. As it can be seen in Fig. 2a, the initial SVM decision boundary is close to the minority class data space in favor of majority class data space and the ideal decision boundary should be located farther away from the minority dataspace.

The next step in SIMO is to calculate the Euclidean distance of the minority data points from the SVM decision boundary. As we mentioned earlier, data points close to the boundary of classes are important and informative. In order to select the informative minority data points,

Table 3
Comparing the performance of various imbalance data learning approaches (using G mean).

		Original data	Under sampling	SMOTE	BorSMOTE	Safe level SMOTE	Cluster SMOTE	SMOTE-IPF	Cost sensitive	SIMO	W-SIMO
Liver	G mean	64.86%	65.37%	65.16%	64.62%	64.93%	65.89%	**66.16%**	65.23%	**68.62%**	**69.08%**
	95% HCI	0.92%	1.68%	1.65%	1.25%	2.22%	1.19%	1.40%	1.14%	1.10%	1.50%
	Rank	9	5	7	10	8	4	**3**	6	**2**	**1**
Ionosphere	G mean	82.92%	82.70%	83.36%	82.29%	82.47%	83.29%	**83.58%**	83.19%	**84.69%**	**84.99%**
	95% HCI	1.97%	1.40%	1.48%	1.42%	2.12%	1.85%	1.49%	1.04%	0.90%	1.77%
	Rank	7	8	4	10	9	5	**3**	6	**2**	**1**
Pima	G mean	70.12%	74.07%	74.33%	72.98%	**74.61%**	74.44%	74.34%	74.11%	**75.48%**	**76.26%**
	95% HCI	0.91%	1.25%	0.90%	0.67%	0.77%	0.90%	0.87%	1.14%	0.69%	0.67%
	Rank	10	8	6	9	**3**	4	5	7	**2**	**1**
BreastCO	G mean	96.71%	96.84%	97.09%	96.74%	96.96%	97.00%	**97.45%**	97.14%	**97.87%**	**97.94%**
	95% HCI	0.32%	0.63%	0.36%	0.43%	0.37%	0.43%	0.36%	0.34%	0.29%	0.26%
	Rank	10	8	5	9	7	6	**3**	4	**2**	**1**
Iris	G mean	57.72%	72.14%	74.54%	72.48%	75.17%	75.82%	**76.20%**	74.48%	**78.13%**	**78.38%**
	95% HCI	7.67%	2.64%	2.60%	3.04%	1.55%	2.39%	2.12%	2.77%	1.80%	1.41%
	Rank	10	9	6	8	5	4	**3**	7	**2**	**1**
Yeast	G mean	41.08%	70.42%	70.90%	69.19%	70.69%	**70.96%**	70.68%	70.89%	**72.25%**	**72.12%**
	95% HCI	0.55%	0.78%	0.62%	0.53%	0.49%	0.55%	0.51%	0.64%	0.39%	0.46%
	Rank	10	8	4	9	6	**3**	7	5	**1**	**2**
Vehicle	G mean	95.63%	95.83%	95.98%	95.78%	95.83%	95.76%	**96.05%**	95.90%	**96.58%**	**96.81%**
	95% HCI	0.51%	0.53%	0.59%	0.59%	0.46%	0.45%	0.47%	0.63%	0.32%	0.51%
	Rank	10	7	4	8	6	9	**3**	5	**2**	**1**
CMC	G mean	0.24%	65.45%	65.15%	64.72%	**65.60%**	65.38%	65.38%	65.55%	**66.06%**	**66.55%**
	95% HCI	0.02%	1.17%	0.70%	0.49%	0.63%	0.55%	0.62%	0.25%	1.00%	0.49%
	Rank	10	5	8	9	**3**	7	7	4	**2**	**1**
BreastC20	G mean	96.20%	96.12%	96.01%	**96.28%**	95.88%	96.12%	96.07%	96.12%	**97.53%**	**97.55%**
	95% HCI	0.48%	0.82%	0.69%	0.49%	0.76%	0.44%	0.61%	0.73%	0.53%	0.40%
	Rank	4	6	9	**3**	10	6	8	6	**2**	**1**
Vowel	G mean	86.85%	87.90%	89.03%	89.77%	88.90%	**89.98%**	89.26%	88.10%	**91.06%**	**91.10%**
	95% HCI	0.55%	0.90%	0.59%	0.51%	0.99%	0.91%	0.87%	0.75%	0.79%	0.98%
	Rank	10	9	6	4	7	**3**	5	8	**2**	**1**
Ecoli	G mean	71.28%	89.29%	89.55%	84.80%	90.05%	**90.10%**	89.90%	90.03%	**91.85%**	**91.88%**
	95% HCI	1.62%	1.10%	1.14%	1.00%	0.65%	0.79%	0.75%	0.80%	0.76%	0.41%
	Rank	10	8	7	9	4	**3**	6	5	**2**	**1**
Libras12	G mean	66.35%	33.99%	43.77%	44.69%	43.93%	48.34%	58.06%	**70.69%**	**87.07%**	**85.49%**
	95% HCI	5.33%	3.53%	0.97%	2.45%	0.90%	1.31%	2.64%	6.63%	1.67%	2.49%
	Rank	4	10	9	7	8	6	5	**3**	**1**	**2**
Libras34	G mean	84.04%	88.93%	89.36%	87.57%	**89.90%**	88.43%	89.72%	89.76%	**91.64%**	**91.87%**
	95% HCI	3.24%	2.51%	1.73%	2.01%	1.56%	1.13%	1.56%	1.86%	1.70%	1.33%
	Rank	10	7	6	9	**3**	8	5	4	**2**	**1**
Glass	G mean	91.62%	91.45%	91.52%	91.41%	91.09%	91.95%	**91.99%**	91.40%	**92.84%**	**92.86%**
	95% HCI	2.31%	1.30%	2.03%	1.67%	1.59%	1.23%	1.36%	1.10%	1.55%	1.16%
	Rank	5	7	6	8	10	4	**3**	9	**2**	**1**
BreastC10	G mean	94.06%	95.50%	94.53%	94.41%	94.72%	94.47%	**95.70%**	94.59%	**95.98%**	**96.09%**
	95% HCI	1.65%	1.35%	0.85%	1.05%	1.01%	0.41%	0.76%	0.87%	0.74%	0.83%
	Rank	10	4	7	9	5	8	**3**	6	**2**	**1**

Table 4
Comparing the performance of various imbalance data learning approaches (using AUC).

		Original data	Under sampling	SMOTE	BorSMOTE	Safe level SMOTE	Cluster SMOTE	SMOTE-IPF	Cost sensitive	SIMO	W-SIMO
Liver	AUC	66.53%	65.62%	65.39%	64.86%	65.18%	**66.64%**	66.40%	65.50%	**68.97%**	**69.45%**
	95% HCI	0.72%	1.63%	1.56%	1.30%	2.06%	1.20%	1.32%	1.09%	0.97%	1.38%
	Rank	4	6	8	10	9	**3**	5	7	**2**	**1**
Ionosphere	AUC	83.94%	83.34%	83.98%	82.73%	83.30%	**84.06%**	83.92%	83.43%	**85.22%**	**85.59%**
	95% HCI	1.73%	1.19%	1.34%	1.33%	1.88%	1.66%	1.34%	0.95%	0.84%	1.76%
	Rank	5	8	4	10	9	**3**	6	7	**2**	**1**
Pima	AUC	72.03%	74.26%	74.49%	73.21%	**74.76%**	74.59%	74.52%	74.29%	**75.71%**	**76.47%**
	95% HCI	0.74%	1.21%	0.87%	0.61%	0.74%	0.89%	0.84%	1.07%	0.65%	0.66%
	Rank	10	8	6	9	**3**	4	5	7	**2**	**1**
BreastCO	AUC	96.72%	96.85%	97.10%	96.76%	96.97%	97.01%	**97.48%**	97.15%	**97.89%**	**97.97%**
	95% HCI	0.32%	0.63%	0.36%	0.43%	0.37%	0.43%	0.36%	0.33%	0.29%	0.26%
	Rank	10	8	5	9	7	6	**3**	4	**2**	**1**
Iris	AUC	63.82%	73.60%	75.50%	74.05%	75.93%	76.77%	**77.14%**	75.46%	**79.12%**	**79.32%**
	95% HCI	3.29%	2.66%	2.66%	2.95%	1.54%	2.05%	1.87%	2.49%	1.45%	1.47%
	Rank	10	9	6	8	5	4	**3**	7	**2**	**1**
Yeast	AUC	57.56%	70.57%	71.05%	70.18%	70.78%	**71.07%**	70.85%	70.98%	**72.45%**	**72.33%**
	95% HCI	0.20%	0.71%	0.50%	0.54%	0.42%	0.52%	0.48%	0.56%	0.45%	0.46%
	Rank	10	8	4	9	7	**3**	6	5	**1**	**2**
Vehicle	AUC	95.68%	95.86%	96.02%	95.83%	95.87%	95.80%	**96.10%**	95.94%	**96.60%**	**96.84%**
	95% HCI	0.50%	0.53%	0.58%	0.59%	0.45%	0.44%	0.46%	0.61%	0.31%	0.50%
	Rank	10	7	4	8	6	9	**3**	5	**2**	**1**
CMC	AUC	50.24%	65.85%	65.47%	65.80%	65.82%	65.86%	**66.06%**	65.81%	**66.35%**	**66.76%**
	95% HCI	0.02%	1.17%	0.65%	0.44%	0.58%	0.54%	0.53%	0.23%	0.71%	0.61%
	Rank	10	5	9	8	6	4	**3**	7	**2**	**1**
BreastC20	AUC	96.23%	96.14%	96.04%	**96.29%**	95.91%	96.14%	96.11%	96.14%	**97.54%**	**97.59%**
	95% HCI	0.46%	0.81%	0.68%	0.48%	0.75%	0.44%	0.61%	0.73%	0.52%	0.39%
	Rank	4	6	9	**3**	10	6	8	6	**2**	**1**
Vowel	AUC	87.50%	87.96%	89.08%	89.88%	88.96%	**90.10%**	89.32%	88.29%	**91.16%**	**91.25%**
	95% HCI	0.50%	0.88%	0.59%	0.50%	0.98%	0.89%	0.84%	0.72%	0.75%	0.88%
	Rank	10	9	6	4	7	**3**	5	8	**2**	**1**
Ecoli	AUC	75.29%	89.47%	89.70%	85.06%	90.20%	**90.22%**	90.02%	90.19%	**91.96%**	**91.97%**
	95% HCI	1.16%	1.07%	1.17%	0.97%	0.63%	0.81%	0.74%	0.78%	0.74%	0.41%
	Rank	10	8	7	9	4	**3**	6	5	**2**	**1**
Libras12	AUC	72.81%	55.65%	55.47%	51.93%	54.95%	55.59%	70.06%	**74.03%**	**88.15%**	**86.83%**
	95% HCI	3.14%	1.73%	1.42%	2.16%	1.48%	1.65%	2.49%	5.37%	1.41%	1.93%
	Rank	4	6	8	10	9	7	5	**3**	**1**	**2**
Libras34	AUC	85.17%	89.12%	89.61%	87.88%	**90.17%**	88.76%	89.97%	89.94%	**91.82%**	**92.03%**
	95% HCI	2.61%	2.50%	1.65%	1.90%	1.42%	1.07%	1.48%	1.80%	1.63%	1.29%
	Rank	10	7	6	9	**3**	8	4	5	**2**	**1**
Glass	AUC	92.07%	91.78%	91.85%	91.86%	91.56%	92.25%	**92.31%**	91.70%	**93.13%**	**93.22%**
	95% HCI	2.08%	1.21%	1.89%	1.57%	1.47%	1.15%	1.28%	1.13%	1.38%	1.03%
	Rank	5	8	7	6	10	4	**3**	9	**2**	**1**
BreastC10	AUC	94.26%	95.57%	94.64%	94.53%	94.85%	94.58%	**95.83%**	94.68%	**96.07%**	**96.18%**
	95% HCI	1.57%	1.30%	0.78%	0.93%	0.96%	0.39%	0.71%	0.78%	0.71%	0.78%
	Rank	10	4	7	9	5	8	**3**	6	**2**	**1**

we identify those that are close to the SVM decision boundary (have the least Euclidean distance from the decision boundary). Therefore, after calculating the Euclidean distance of the minority data points from the decision boundary, the top Δ% of them that are the closest ones to the decision boundary will be selected as informative minority data points (Fig. 2b). Next, we generate synthetic data points in the space of the informative minority examples and append the generated data points to the training dataset. We use SMOTE approach for generating synthetic data points in the minority data space. In this approach, first, the K nearest neighbors of each informative minority data point is identified (the neighbors are also from informative minority examples). Then, a synthetic data point is generated on the line connecting the informative minority data point and its neighbors.

$$x_{synthetic} = x_i + (\hat{x}_i - x_i)\delta \tag{4}$$

where x_i is an informative minority data point, $\hat{x}_i$ is one of the K nearest neighbors of x_i, and δ is a random number between 0 and 1.

At this stage, we have a new training dataset that includes more minority examples compared to the previous training dataset (Fig. 2c). The number of synthetically generated data points and their indices will be recorded at each iteration. Next, a new SVM will be developed on the updated training dataset. The decision boundary of this new SVM will be shifted toward the majority class data space closer to the ideal decision boundary (Fig. 2d). The reason is that by generating synthetic minority examples, the imbalance ratio of the training dataset will be reduced and following that, the imbalance ratio of the support vectors will be alleviated. Therefore, the decision boundary will be shifted toward the majority class dataspace (As we discussed in detail in Section 3.1, the position of the SVM decision boundary only depends on the support vectors). The new SVM will be assessed by computing the G mean, and the G mean will be logged into a vector for further evaluations. Again, in the updated training dataset, the Euclidean distance of the minority data points from the new SVM decision boundary is calculated, informative ones will be selected, and new synthetic minority data points will be generated. Another SVM will be developed on the updated dataset, the SVM will be assessed, and the results will be recorded. These steps will be repeated until

the number of synthetically generated examples reaches the imbalanced gap.

The performance of machine learning techniques highly depends on the structure and complexity of datasets. In our algorithm, in each iteration, we create a new updated dataset by generating more synthetic minority examples. Even though the performance of the SVM improves on the updated training datasets compared to the original imbalanced dataset, the improvement in the performance of the SVM in each iteration compared to the previous iteration is not guaranteed in all datasets. In the other words, the G mean might not always be increasing through the iterations. Therefore, we keep track of the SVM performances and their corresponding training dataset in each iteration. At the end of the loop, the best performing model is identified by comparing the G mean values, and the training dataset associated with that model/iteration will be selected as the final over-sampled training dataset.

In this study, we proposed another version of SIMO that we call weighted synthetic informative minority over-sampling (W-SIMO). Steps 1 to 7 in W-SIMO (Algorithm 2 and Flowchart 2) are the same as SIMO, i.e. an initial SVM is developed on the original imbalanced training dataset, and top Δ% minority data points close to the SVM decision boundary are identified as informative minority examples. In the next step, informative minority examples will be classified into two groups: first, those that are correctly classified ($S^+_{\text{inf\_c}}$) through SVM, and second, those that are incorrectly classified ($S^+_{\text{inf\_ic}}$). The data points in $S^+_{\text{inf\_ic}}$ will be over-sampled to a higher degree compared to the data points in $S^+_{\text{inf\_c}}$. We adopt this idea from AdaBoost, which pays more attention to the incorrectly classified examples [25]. In W-SIMO, by over-sampling the examples in $S^+_{\text{inf\_ic}}$ with a higher degree, we consider them even more informative compared to the examples in $S^+_{\text{inf\_c}}$. It means that more synthetic minority data points will be generated in the space of $S^+_{\text{inf\_ic}}$ examples. After over-sampling (synthetically generating) the informative minority data points in $S^+_{\text{inf\_ic}}$ and $S^+_{\text{inf\_c}}$, the reminder of the W-SIMO is similar to SIMO. Applying the SIMO and W-SIMO is not limited to the SVM. We use SVM in our algorithms to identify informative data points to over sample them, however the final over-sampled data can be used in any other machine learning technique, such as decision tree, logistic regression, and random forest. The notations of Algorithms 1 and 2 are shown in Table 1.

Algorithm 1- SIMO

Given D, Δ, p

1. Partition D into Training $\hat{S}$, and Test T datasets
2. Calculate the *Imbalanced_Gap* in $\hat{S}$

$$Imbalance\_Gap = Majority\_Count - Minority\_Count$$

3. Develop the *Initial SVM* model on $\hat{S}$, *Initial SVM* decision boundary: $\widehat{D\_B} = \hat{w}^T x + \hat{b}$

$$\{\hat{w} = \sum_{j=1}^{N_s} \hat{a}_i y_i \phi(x_i)\}$$

4. Compute *G mean* for *Initial SVM* on T: *Initial_G Mean*
5. $S = \hat{S}$, $SVM = Initial\ SVM$, $D\_B = \widehat{D\_B}$, $G\_m\_L = Initial\_G\ Mean$, $S\_G\_D = 0$

While $S\_G\_D < Imbalance\_Gap$

6. Calculate the Euclidean distance of minority data points form D_B

$$Euc\_D(x^{k+}) = \frac{|\sum_{t=1}^{m} w_t x_t^{k+} + b|}{\sqrt{\sum_{t=1}^{m} w_t{}^2}}$$

7. Identify informative minority data points: S^+_{inf}

Top Δ% of minority data points close to D_B based on the Euclidean distance

8. Over-sample data points in S^+_{inf} by p%, name the synthetic generated data points $\hat{S}^+_{inf}$
9. $S = S \cup \hat{S}^+_{inf}$
10. Calculate the number of synthetic generated data points

$$S\_G\_D = S\_count - \hat{S}\_count$$

11. Develop a support vector machine on S, SVM
12. Compute *G mean* for SVM on T
13. Add the *G mean* to the G_m_L, $(G\_m\_L = [G\_m\_L; G\ mean])$

End

14. Find the maximum *G mean* and its index in G_m_L
15. Select the over-sampled training dataset associated with the maximum *G mean*
16. Train the model of interest on the final over-sampled training dataset
17. Evaluate the model on the test dataset by computing the *G mean* and *AUC*

Algorithm 2- W-SIMO

Given $D, \Delta, p, P, (p < P)$

1. Partition D into Training $\hat{S}$, and Test T datasets
2. Calculate the *Imbalanced_Gap* in $\hat{S}$

$$Imbalance\_Gap = Majority\_Count - Minority\_Count$$

3. Develop the *Initial SVM* model on $\hat{S}$, *Initial SVM* decision boundary: $\widehat{D\_B} = \hat{w}^T x + \hat{b}$

$$\{\hat{w} = \sum_{j=1}^{N_s} \hat{a}_i y_i \phi(x_i)\}$$

4. Compute *G mean* for *Initial SVM* on T: *Initial_G Mean*
5. $S = \hat{S}$, $SVM = Initial\ SVM$, $D\_B = \widehat{D\_B}$, $G\_m\_L = Initial\_G\ Mean$, $S\_G\_D = 0$

While $S\_G\_D < Imbalance\_Gap$

6. Calculate the Euclidean distance of minority data points form D_B

$$Euc\_D(x^{k+}) = \frac{|\sum_{t=1}^{m} w_t x_t^{k+} + b|}{\sqrt{\sum_{t=1}^{m} w_t{}^2}}$$

7. Identify informative minority data points: S^+_{inf}

Top Δ% of minority data points close to D_B based on the Euclidean distance

8. Classify informative minority data points using the *SVM* model, form:
   i. $S^+_{\text{inf\_c}}$, informative minority data points that are *correctly* classified
   ii. $S^+_{\text{inf\_ic}}$, informative minority data points that are *incorrectly* classified
9. Over-sample data points in $S^+_{\text{inf\_c}}$ by p%, name the synthetic generated data points $\hat{S}^+_i$
10. Over-sample data points in $S^+_{\text{inf\_ic}}$ by P%, name the synthetic generated data points $\hat{S}^+_{ii}$
11. $S = S \cup \hat{S}^+_i \cup \hat{S}^+_{ii}$
12. Calculate the number of synthetic generated data points

$$S\_G\_D = S\_count - \hat{S}\_count$$

13. Develop a support vector machine on S, SVM
14. Compute *G mean* for SVM on T
15. Add the *G mean* to the G_m_L, $(G\_m\_L = [G\_m\_L; G\ mean])$

End

16. Find the maximum *G mean* and its index in G_m_L
17. Select the over-sampled training dataset associated with the maximum *G mean*
18. Train the model of interest on the final over-sampled training dataset
19. Evaluate the model on the test dataset by computing the *G mean* and *AUC*

5. Numerical experiments

In this section, we provide the results of our numerical experiments to assess the performance of SIMO and W-SIMO compared to other existing algorithms in imbalanced data learning. First, we describe the evaluation metrics that we used for the assessments. Second, we provide the characteristics of the benchmark imbalanced datasets that we used. Third, we present the results of the numerical experiments. Finally, we provide the results of sensitivity analysis on SIMO parameters.

5.1. Evaluation metrics

In classification or pattern recognition problems confusion matrix plays an important role to assess the predictive models. Fig. 3 shows a confusion matrix. As was pointed out earlier, in this study, we consider the minority class as positive, and the majority class as negative class. Accuracy of prediction (Formulation (5)) is a common evaluation metric in the balanced datasets; however, it is misleading in assessing the predictive models when applied in imbalanced datasets. Consider an imbalanced dataset with the 10% rate of the positive examples. Because negative examples outnumber the positive ones, simply classifying all of the examples as negative will result in a 90% accuracy. Therefore, in imbalanced datasets other appropriate evaluation metrics such as sensitivity, specificity, G mean, and AUC should be applied [44].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \tag{5}$$

Sensitivity or true positive rate (TPR) (it is also called hit rate or recall) is a metric that evaluates the accuracy of predicting the positive examples. On the other hand, specificity or true negative rate (TNR) assesses the accuracy of detecting the negative examples. Formulations (6) and (7) show the calculation of TPR and TNR.

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{TNR} = \frac{\text{TN}}{\text{FP} + \text{TN}} \tag{7}$$

TPR and TNR assess the detection accuracy in positive and negative examples separately. Therefore, considering one of them without the other one would not be helpful, therefore, we need a metric such as G mean that incorporates these two metrics at the same time. G mean is the geometric mean of TPR and TNR (Formulation (8)). Thus, any model with poor performance on either positive or negative examples will have a low G mean.

$$\text{G mean} = \sqrt{\text{TPR} \times \text{TNR}} \tag{8}$$

Another assessment tool that is independent of the data distribution is Receiving Operator Characteristic (ROC) chart. ROC shows the tradeoff between TPR and TNR by manipulating the decision cut-off. Decision cut-off is the threshold value for decision making based on the output of a predictive model. When the decision cut-off for a model is 0, all of the examples will be classified as positive, therefore TPR = 100% but TNR = 0%. On the other hand, if decision cut-off is 1, TPR = 0% and TNR = 100%. Thus, by changing the decision cut-off from 1 to 0, we can increase the TPR, and TNR will decrease at the same time. In ROC chart, the x-axis shows the 1-TNR and y-axis denotes the TPR, in this way the graph will be increasing. Each point on the ROC chart shows the value of TPR and 1-TNR for a specific decision cut-off value. The closer the ROC chart to the top left point, the better the performance of the classifier. Fig. 4 shows a ROC chart, the 45-degree line is the base line model (random), the dash line corresponds to a good performing model, and dotted line is for the perfect model.

An easier way to assess the models and compare different classifiers is to measure the area under the curve (AUC) in ROC chart. AUC takes values between 0 and 100%. AUC for the base line model is 50%, and therefore, classifiers with AUC below 50% are even worse than random guess. The closer the AUC of classifier to 100%, the better the performance of the classifier.

5.2. Datasets

In this study, we used 15 benchmark imbalanced datasets that are publicly available in UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/). We tried to use datasets with various imbalance ratios from 1:1.38 to 1:8.9, i.e. the percentage of minority class in the benchmark datasets ranges from 42% to 10%. To test SIMO and W-SIMO on datasets with more severe imbalance ratio, we randomly removed some portions of minority class examples from Breast Cancer dataset and generated datasets with 1:3.91 (BreastC20 dataset) and 1:8.9 (BreastC10) imbalance ratio. Table 2 shows the name and characteristics of these datasets.

5.3. Results

In this study, we compared the performance of our algorithms, SIMO and W-SIMO to seven other existing approaches in imbalanced data learning. We also provided the modeling results on the original imbalanced data for reference. For all of the algorithms, we used the parameters suggested by their developers. We assessed SIMO and W-SIMO in comparison with these algorithms: under-sampling, SMOTE, borderline SMOTE, safe-level SMOTE, cluster SMOTE, SMOTE-IPF, and cost sensitive SVM. In cost sensitive SVM, we assigned the error cost of the two classes based on the imbalance ratio in the dataset. For instance, if the imbalance ratio in a data is 1:4, the error cost for the minority class is 4 times greater that the error cost for majority class.

To avoid over-fitting and fairly assess the generalizability and performance of various approaches, we applied 4-fold cross validation in our numerical experiments [45]. In a 4-fold cross validation, the original dataset is partitioned into four mutually exclusive and exhaustive subsets with equal sizes (Sub_1, Sub_2, Sub_3, and Sub_4). Then, the models are developed four times, each time the model is trained on three of the subsets, and is tested on the fourth one. The final performance will be the average of the models 1, 2, 3, and 4. Fig. 5 shows the mechanism of 4-fold cross validation.

In order to further reduce the effect of randomness, we ran each 4-fold cross validation on all approaches 10 times. Therefore, each approach has been applied to each dataset 40 times. As a result, for each evaluation metric we have both average value and 95% confidence interval. Tables 3 and 4 show the performance of all of the 9 imbalanced data learning approaches as well as the learning from original imbalanced dataset in a linear SVM classifier. The first row for each dataset in this table shows the evaluation metric average (G mean in Table 3 and AUC in Table 4), the second row shows the half of the 95% confidence interval width (HCI) for the evaluation metric, and the third row shows the performance ranking of each approach compared to other approaches.

As it can be seen in Tables 3 and 4, in all of the 15 imbalanced datasets, our proposed algorithms, SIMO and W-SIMO had the best performance compared to other approaches (approaches with ranks 1, 2, and 3 are **bolded** in Tables 3 and 4). In addition, the difference between the G mean and AUC value for SIMO and W-SIMO and other approaches is significant. To show this difference and the achieved improvement thorough applying our algorithm, we calculated the difference between the G mean and AUC of our algorithm and the G mean and AUC of the best algorithm among other approaches (the approach with rank 3 in Tables 3 and 4) in all datasets. We also calculated the difference between the G mean and AUC of the best and second best algorithms among approaches other than our algorithm (the approaches with rank 3 and 4 in Tables 3 and 4) in all 15 benchmark datasets. Table 5 shows the average of these differences in all datasets. We ran a *t*-test to compare the improvement from the approach with rank 3 to our algorithm to the achieved improvement from the approach with rank 4 to the approach with rank 3 (best and second best approaches not including SIMO and W-SIMO). The *p*-values for G mean and AUC were 0.0106 and 0.0136 respectively. Therefore, the *t*-test showed that the difference between our algorithm and best algorithm among other existing approaches was significantly greater than the difference between the approaches with rank 3 and 4 at the confidence level of 95%. Table 6 demonstrates the overall ranking of SIMO and W-SIMO compared to other imbalanced data learning approaches when applied to linear SVM. The overall

Table 5
Average difference between the performance of our algorithm and other approaches.

	Average difference between our algorithm and the best one among other approaches	Average difference between the best and second best ones among other approaches
G mean	2.36%	0.44%
AUC	2.2%	0.23%

Table 6
Overall ranking on linear SVM.

Approach	G mean	AUC
W-SIMO	1.1	1.1
SIMO	1.9	1.9
SMOTE-IPF	4.6	4.5
Cluster SMOTE	5.3	5.0
Cost sensitive	5.7	6.1
SMOTE	6.3	6.4
Safe level SMOTE	6.3	6.7
Under sampling	7.3	7.1
BorSMOTE	8.1	8.1
Original data	8.6	8.1

ranking is calculated based on the average of various approaches' ranking in 15 benchmark datasets. Since W-SIMO and SIMO had the first and second places in all datasets, their overall ranking is 1.1 and 1.9 respectively.

As we mentioned earlier, the oversampled training data by SIMO and W-SIMO can be used in any other machine learning technique. Therefore, SIMO and W-SIMO can be considered pre-processing oversampling algorithms. To evaluate the performance of SIMO and W-SIMO in other data mining techniques, we applied them in SVM with RBF kernel function, logistic regression, and decision tree. Tables 7, 8, and 9 present the overall rankings of our algorithms as well as their counterparts when applied to SVM with RBF kernel, logistic regression, and decision tree in all benchmark datasets. As it can be seen in Tables 7, 8, and 9, the overall ranking of W-SIMO and SIMO is not about 1 and 2, unlike what we observed in Table 6. This means that our algorithms were not always the best when applied in machine learning techniques other than linear SVM. In fact, these results were expected since SIMO and W-SIMO are imbedded into linear SVM, therefore, we expected them to have a better performance in linear SVM. Even though our algorithms were not always the best ones in other machine learning techniques, their overall performance was better compared to other approaches. As Tables 7, 8, and 9 show, either SIMO or W-SIMO was the best overall algorithm in SVM with RBF kernel, logistic regression, and decision tree.

As we noted in the Introduction section, one of the reasons that we used SVM in our algorithm was its great performance and accuracy compared to other machine learning techniques. The results of the numerical experiments in logistic regression and decision tree showed that our algorithm was not always the best in all datasets in these data mining techniques. However, when we compared the best performing algorithms (imbalanced data learning algorithms, such as SIMO, SMOTE, and under-sampling) in each machine learning technique in each dataset, it turned out that SVM always outperformed other data mining techniques. Therefore, our algorithm might not always have the best performance when applied to logistic regression and decision tree, but its performance in SVM is better and has higher G mean and AUC. Table 10 demonstrates these results. For each dataset, we provide the G mean and AUC of the best imbalanced data learning approach in each of the four machine learning techniques, linear SVM, SVM with RBF kernel, logistic regression, and decision tree. The **bold** numbers show the best performing machine learning technique in each dataset and the underlined numbers are the results of our algorithms. Only in three datasets, the best performing model was not incorporated with our algorithm; those cases are shown in ***italic bold***. The output of our algorithm in those cases is shown in parenthesis and they are not much lower than the best performing approaches. Overall, no one can claim that their algorithm is the best performing algorithm in all datasets, because the performance of a technique or algorithm highly depends on the distribution, size, and complexity of datasets, however, the overall performance of algorithms on multiple datasets from various domains can be a fair comparison measure.

Table 8
Overall ranking on logistic regression.

Approach	G mean	AUC
W-SIMO	2.6	2.4
Cluster SMOTE	3.3	3.6
SMOTE-IPF	3.9	4.2
SMOTE	4.6	4.5
SIMO	4.8	4.9
Safe level SMOTE	5.4	5.4
Under sampling	6	5.8
BorSMOTE	6.8	6.7
Original data	7.6	7.5

Another advantage of our proposed algorithm is that it makes a minimal alteration to the original distribution of the dataset. While other over-sampling approaches generate enough data points to completely fill the imbalanced gap in the data, SIMO and W-SIMO only focus on the informative data points close to the decision boundary between two classes in the data, and therefore, they do not generate as many synthetic data points as other over-sampling methods. Table 11 demonstrates the imbalanced gap between majority and minority class in various datasets. It also shows the average number of data points generated by our algorithms as well as other over-sampling approaches. The number in parenthesis shows the amount of the synthetically generated data points as a percentage of the total imbalanced gap in the training datasets. As it can be seen, SIMO and W-SIMO usually generate less number of data points compared to other over-sampling methods. This result shows two advantages of our proposed algorithms. First, our algorithms do not dramatically change the distribution of the data from its original shape. Second, with less amount of data generated, the further computational cost in training the machine learning techniques will be lower.

5.4. Sensitivity analysis

For applying SIMO and W-SIMO, their parameters, i.e. Δ, p, and P need to be specified. To evaluate the performance of SIMO in different parameters values, we performed a sensitivity analysis. In the sensitivity analysis, we considered values 10% to 50% for Δ, and 5% to 50% for p. Table 12 depicts the results of the sensitivity analysis for $\Delta = 10$, 20, 30, and 40% and $p = 10$ and 40%. As it can be seen in Table 12, different values of parameters do not make a considerable difference in the performance of SIMO. Therefore, SIMO is not very sensitive to the value of its parameters. Moreover, except in 4 cases, in all of the other cases, with even the worst parameters value, SIMO had a better performance compared to the 3rd best approach. Based on this analysis, we suggest the following policy for choosing the parameters values. When the imbalance ratio of the data is high (the minority class rate below 20%), it is better to select

Table 7
Overall ranking- SVM-RBF kernel.

Approach	G mean	AUC
W-SIMO	2.9	2.8
Cluster SMOTE	3.8	3.9
SIMO	4.2	4.2
SMOTE-IPF	4.9	4.8
Cost sensitive	5.0	4.9
Under sampling	5.6	5.6
Safe level SMOTE	6.3	6.3
SMOTE	6.8	6.9
BorSMOTE	7.1	7.2
Original data	8.4	8.4

Table 9
Overall ranking on decision tree.

Approach	G mean	AUC
SIMO	2.8	2.7
SMOTE-IPF	3.0	3.1
Under sampling	3.9	3.8
W-SIMO	4.2	4.3
SMOTE	5.2	5.1
Cluster SMOTE	5.5	5.2
Original data	5.5	5.9
Safe level SMOTE	6.8	6.5
BorSMOTE	8.1	8.4

Table 10
The performance of best approach in each machine learning technique.

	SVM-Linear		SVM-RBF		Logistic regression		Decision tree	
	G mean	AUC	G mean	AUC	G mean	AUC	G mean	AUC
Liver	**69.08%**	**69.45%**	62.31%	64.09%	65.19%	66.77%	62.51%	63.18%
Ionosphere	84.99%	85.59%	**94.11%** (94.00%)	**94.19%** (94.04%)	81.54%	82.62%	87.68%	87.79%
Pima	**76.26%**	**76.47%**	70.51%	70.62%	74.60%	74.66%	69.25%	69.38%
BreastCO	**97.94%**	**97.97%**	97.04%	97.16%	96.52%	96.54%	94.76%	94.84%
Iris	78.38%	79.32%	**97.06%**	**97.14%**	75.12%	75.50%	94.31%	94.45%
Yeast	**72.25%**	**72.45%**	70.31%	70.39%	70.88%	71.17%	66.28%	66.39%
Vehicle	**96.81%**	**96.84%**	95.90%	96.04%	96.25%	96.30%	91.56%	91.64%
CMC	**66.55%**	**66.76%**	66.17%	66.35%	65.74%	65.92%	61.82%	61.98%
BreastC20	**97.55%**	**97.59%**	97.34%	97.35%	96.37%	96.43%	93.61%	93.57%
Vowel	91.10%	91.25%	**99.61%**	**99.66%**	90.35%	90.41%	95.67%	95.62%
Ecoli	91.88%	91.97%	**93.56%** (91.93%)	**93.67%** (92.72%)	90.69%	90.79%	86.20%	86.88%
Libras12	87.07%	88.15%	**97.68%**	**97.79%**	39.37%	42.83%	84.34%	85.86%
Libras34	91.87%	92.03%	**93.01%**	**93.15%**	82.84%	83.06%	82.75%	83.13%
Glass	**92.86%**	**93.22%**	89.43%	89.98%	91.62%	91.93%	92.40%	92.63%
BreastC10	96.09%	96.18%	**96.76%**	**96.81%**	94.76%	94.85%	92.23%	92.36%

higher values for Δ and p, i.e. values between 30 and 40% for Δ, and values between 25 and 50% for p. The reason is that, because the number of the minority data points in highly imbalanced datasets is very low, by selecting relatively higher values for Δ, we consider greater number of minority data points for over-sampling. Therefore, we avoid the potential overfitting that might happen. On the other hand, for datasets with lower imbalanced ratio (the minority class rate between 20 and 40%), choosing lower values for Δ and p will generate better results. Selecting the parameters for W-SIMO follows the same policy with one difference, and that is selecting a higher value for P compared to p. Our suggestion based on the sensitivity analysis is to choose 20 to 30% greater values for P. For example, if $p = 20\%$, values between 40 and 50% are appropriate for P.

6. Discussion and conclusion

Imbalanced datasets are widespread in various domains such as healthcare, finance, and information system security. Nowadays, a large portion of decision support systems is built by analyzing data. Therefore, decision support systems in the above mentioned domains are affected by the imbalanced data learning challenges. In an imbalanced dataset, the number of examples belonging to one class outnumbers the number of examples from the other class. Therefore, in an imbalanced dataset, there are majority and minority classes of examples. Training machine-learning techniques using imbalanced datasets is a critical challenge in data analytics. The prediction accuracy of a data mining technique, especially prediction accuracy of detecting the minority class in an imbalanced dataset, is inferior to the performance of the same technique when applied to a balanced dataset. There has been an enormous effort to address the problem of imbalanced data learning in recent years. Sampling methods along with cost sensitive approaches are among the most efficient remedies to the imbalanced data learning problem. Improving the prediction accuracy of machine learning techniques when applied to imbalanced datasets, leads to better decision making in real world problems. This improved decision making is critically important when we are dealing with imbalanced datasets, because in most the cases, the minority class is the class of interest for decision makers. As a result, any effort toward enhancing imbalanced data learning, strengthen the efficiency of the decision support systems.

In this study, we proposed a synthetic informative minority oversampling (SIMO) algorithm integrated with SVM to enhance the performance of machine learning techniques when applied to imbalanced datasets. In this algorithm, first SVM is applied to the original imbalanced dataset. In the next step, minority examples close to the SVM

Table 11
Imbalanced gap and average # of synthetically generated data points (% of the imbalance gap).

	Imbalanced gap in training data	Other approaches	SIMO	W-SIMO
Liver	41	41 (100%)	20 (48.8%)	22 (53.7%)
Ionosphere	75	75 (100%)	18 (24%)	22 (29.3%)
Pima	174	174 (100%)	104 (59.8%)	96 (55.1%)
BreastCO	154	154 (100%)	25 (16.2%)	30 (19.5%)
Iris	38	38 (100%)	25 (65.8%)	24 (63.1%)
Yeast	467	467 (100%)	392 (83.9%)	373 (79.9%)
Vehicle	336	336 (100%)	50 (14.9%)	46 (13.7%)
CMC	606	606 (100%)	543 (89.6%)	569 (93.9%)
BreastC20	229	229 (100%)	13 (5.7%)	18 (7.9%)
Vowel	473	473 (100%)	173 (36.6%)	152 (32.1%)
Ecoli	174	174 (100%)	60 (34.5%)	58 (33.3%)
Libras12	175	175 (100%)	29 (16.6%)	17 (9.7%)
Libras34	180	180 (100%)	20 (11.1%)	17 (9.4%)
Glass	117	117 (100%)	5 (4.2%)	6 (5.1%)
BreastC10	282	282 (100%)	35 (12.4%)	19 (6.7%)

Table 12
Sensitivity analysis on SIMO parameters.

		SIMO									3rd best approach
		$\Delta = 10\%$		$\Delta = 20\%$		$\Delta = 30\%$		$\Delta = 40\%$		Best Δ & p	
		$p = 10\%$	$p = 40\%$	$p = 10\%$	$p = 40\%$	$p = 10\%$	$p = 40\%$	$p = 10\%$	$p = 40\%$		
Liver	G mean	**68.99%**	68.70%	68.91%	68.73%	68.57%	68.58%	68.32%	68.42%	$\Delta = 10\%$	66.16%
	AUC	**69.34%**	68.91%	69.26%	68.94%	68.74%	68.67%	68.60%	68.60%	$p = 10\%$	66.64%
Ionosphere	G mean	**85.14%**	84.94%	85.10%	84.49%	84.74%	84.46%	84.49%	84.18%	$\Delta = 20\%$	83.58%
	AUC	**85.71%**	85.55%	85.63%	84.64%	85.43%	85.01%	85.16%	84.87%	$p = 05\%$	84.06%
Pima	G mean	75.88%	75.59%	**75.99%**	75.90%	75.65%	75.75%	75.64%	75.52%	$\Delta = 20\%$	74.61%
	AUC	76.08%	75.80%	**76.22%**	76.12%	75.98%	75.97%	75.86%	75.64%	$p = 10\%$	74.76%
BreastCO	G mean	98.14%	98.07%	98.13%	98.12%	**98.15%**	97.88%	97.90%	97.76%	$\Delta = 15\%$	97.45%
	AUC	98.15%	98.09%	98.15%	98.13%	**98.16%**	97.90%	97.91%	97.77%	$p = 15\%$	97.48%
Iris	G mean	78.45%	78.45%	**78.91%**	78.24%	78.76%	78.03%	78.44%	78.08%	$\Delta = 20\%$	76.20%
	AUC	79.15%	79.18%	**79.68%**	78.97%	79.46%	78.80%	79.20%	78.61%	$p = 25\%$	77.14%
Yeast	G mean	71.86%	**72.33%**	72.21%	72.03%	72.31%	71.80%	71.80%	71.58%	$\Delta = 10\%$	70.96%
	AUC	72.10%	**72.51%**	72.40%	72.26%	72.46%	72.05%	72.06%	71.79%	$p = 15\%$	71.07%
Vehicle	G mean	96.74%	96.94%	96.84%	96.77%	**97.03%**	96.73%	96.92%	96.75%	$\Delta = 30\%$	96.05%
	AUC	96.76%	96.95%	96.87%	96.79%	**97.06%**	96.76%	96.94%	96.77%	$p = 30\%$	96.1%
CMC	G mean	66.03%	66.15%	66.32%	**66.48%**	66.46%	66.34%	66.14%	65.96%	$\Delta = 20\%$	65.60%
	AUC	66.33%	66.44%	66.55%	**66.77%**	66.74%	66.62%	66.44%	66.28%	$p = 40\%$	66.06%
BreastC20	G mean	97.72%	97.59%	**97.85%**	97.67%	97.47%	97.41%	97.33%	97.25%	$\Delta = 12\%$	96.28%
	AUC	97.73%	97.60%	**97.86%**	97.68%	97.49%	97.42%	97.34%	97.26%	$p = 05\%$	96.29%
Vowel	G mean	90.94%	**91.50%**	90.95%	91.22%	91.13%	90.92%	91.19%	91.00%	$\Delta = 10\%$	89.98%
	AUC	91.06%	**91.60%**	91.08%	91.35%	91.24%	91.01%	91.31%	91.12%	$p = 35\%$	90.10%
Ecoli	G mean	91.30%	91.51%	92.06%	92.02%	**92.15%**	91.74%	91.99%	91.94%	$\Delta = 12\%$	90.10%
	AUC	91.51%	91.72%	92.17%	92.13%	**92.25%**	91.84%	92.07%	92.05%	$p = 20\%$	90.22%
Libras12	G mean	86.98%	87.12%	**87.47%**	86.85%	86.67%	86.44%	86.50%	86.50%	$\Delta = 20\%$	70.69%
	AUC	88.16%	88.26%	**88.55%**	88.08%	87.77%	87.53%	87.75%	87.81%	$p = 10\%$	74.03%
Libras34	G mean	91.12%	91.31%	91.55%	91.51%	91.75%	**91.76%**	91.63%	91.71%	$\Delta = 30\%$	89.90%
	AUC	91.30%	91.50%	91.84%	91.72%	91.95%	**91.96%**	91.80%	91.87%	$p = 25\%$	90.17%
Glass	G mean	92.98%	92.67%	92.63%	92.89%	92.77%	**93.02%**	92.95%	92.51%	$\Delta = 30\%$	91.99%
	AUC	93.28%	92.98%	92.96%	93.19%	93.07%	**93.31%**	93.28%	92.86%	$p = 40\%$	92.31%
BreastC10	G mean	95.39%	95.73%	95.72%	95.90%	95.91%	95.71%	95.96%	**96.21%**	$\Delta = 40\%$	95.70%
	AUC	95.54%	95.80%	95.79%	95.99%	96.02%	95.85%	96.07%	**96.30%**	$p = 35\%$	95.83%

decision boundary are selected as the informative minority examples. Next, these examples are over-sampled to a pre-specified degree. Finally, a new SVM model is developed on the updated dataset. This process iterates until we reach a pre-specified balance level. In each iteration, we have an updated training dataset, which is formed by adding the newly generated data points to the previous dataset. Each of these training datasets is used to develop a SVM model, and the SVM model is assessed on the test dataset. At the end, the best model and its associated training dataset is selected as the final over-sampled training dataset. In this research, we also developed another version of SIMO called W-SIMO. W-SIMO is different from SIMO in the degree of over-sampling the informative minority examples. In W-SIMO, informative minority examples that are incorrectly classified are over-sampled with a higher degree compared to the informative minority examples that are correctly classified. In this way, there is more focus on incorrectly classified minority examples.

SIMO and W-SIMO have several advantages compared to other imbalanced data learning methods. First, they leverage SVM, which is a powerful machine learning technique in pattern recognition problems. Second, in SIMO and W-SIMO, we over-sample the minority examples rather than under-sampling the majority examples, therefore we avoid losing potentially useful information by discarding some portion of the data. Third, our focus in SIMO and W-SIMO is only on the data points (examples) near the decision boundary as the informative minority data points. This focus is even more important in W-SIMO where we over-sample the incorrectly classified examples with a higher degree. Therefore, SIMO and W-SIMO concentrate on the informative minority examples that usually are misclassified by standard machine learning techniques. Fourth, compared to other oversampling methods, SIMO generates fewer synthetic data points. Therefore, the changes to the original distribution of the data and further computational costs will be lower compared to other oversampling approaches. Fifth, the oversampled data through SIMO can be used to train any other machine learning technique, thus its application is not limited only to SVM. Finally, SIMO and W-SIMO are not very sensitive to their parameters, even though we suggest to select higher values for Δ and p in highly imbalanced datasets and lower values in moderately imbalanced datasets.

We applied our algorithms to 15 publicly available benchmark imbalanced datasets and assessed their performance in comparison with existing approaches in the area of imbalanced data learning. These approaches were cost sensitive SVM, under sampling, SMOTE, cluster SMOTE, safe level SMOTE and borderline SMOTE as well as the original imbalanced dataset. Our algorithm had the best performance in all datasets compared to the other seven approaches in the linear SVM. In fact, the difference between our algorithm and second best algorithm was significantly greater than the difference between other algorithms (for instance, the difference between second and third best approaches). Besides linear SVM that SIMO and W-SIMO were integrated with, we also assessed SIMO and W-SIMO in other machine learning techniques such as SVM with RBF kernel, logistic regression, and decision tree. Our algorithms were not always the best in these machine learning techniques in all bench mark datasets, however their overall performances were better than all other imbalanced data learning approaches. Moreover, the results showed that the best performing machine learning technique in all datasets was either linear SVM or SVM with RBF kernel function, and except for in three datasets, our algorithms were the best ones. From the practical implication point of view, our proposed algorithm can enhance the performance of the predictive models and decision support systems in various domains such as diagnosing diseases, detecting re-admissions, and predicting the loan defaults in financial institutions among other application domains.

Our proposed algorithms may have a limitation that all of the over-sampling approaches face. This limitation is the computational time when the algorithms are applied to very large size datasets. Even though considering the recent advances in computational power of the computers, the computational time is not as critical as it used to be, we still need to enhance the speed of our algorithms in large size datasets. Therefore, we consider speeding up our algorithms in big data usage as one of the most important directions for future research. One way to achieve higher speed could be decreasing the size of the data thorough approaches such as variable selection before using the data in over-sampling algorithms. Another way could be improving the SVM training algorithms. We are considering another direction for future research, and that is applying our developed algorithms to develop a clinical decision support system for predicting kidney disease among diabetic patients. The dataset that we are going to use for that research contains the lab, demographic, clinical events, and comorbidity data of a large number of diabetic patients. We believe that this future research will reveal the performance and efficiency of our algorithm in a larger imbalanced dataset.

References

[1] S. Piri, D. Delen, T. Liu, H.M. Zolbanin, A data analytics approach to building a clinical decision support system for diabetic retinopathy: Developing and deploying a model ensemble, Decis. Support. Syst. vol. 101 (2017) 12–27 (2017/09/01).

[2] A. Dag, A. Oztekin, A. Yucel, S. Bulur, F.M. Megahed, Predicting heart transplantation outcomes through data analytics, Decis. Support. Syst. vol. 94 (Supplement C) (2017) 42–52 (2017/02/01/).

[3] P.K. Chan, F. Wei, A.L. Prodromidis, S.J. Stolfo, Distributed data mining in credit card fraud detection, Intell. Syst. Their Appl. IEEE 14 (6) (1999) 67–74.

[4] E. Tobback, T. Bellotti, J. Moeyersoms, M. Stankova, D. Martens, Bankruptcy prediction for SMEs using relational data, Decis. Support. Syst. vol. 102 (Supplement C) (2017) 69–81.

[5] P. Liu, Y. Wang, L. Cai, L. Zhang, Classifying skewed data streams based on reusing data, 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), vol. 4, IEEE 2010, pp. V490–V493.

[6] M.A. Maloof, Learning when data sets are imbalanced and when costs are unequal and unknown, ICML-2003 Workshop on Learning From Imbalanced Data Sets II, vol. 2, 2003.

[7] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. vol. 21 (9) (2009) 1263–1284.

[8] M.A. Hearst, S.T. Dumais, E. Osman, J. Platt, B. Scholkopf, Support vector machines, IEEE Intell. Syst. Appl. 13 (4) (1998) 18–28.

[9] A. Anand, G. Pugalenthi, G.B. Fogel, P. Suganthan, An approach for classification of highly imbalanced data using weighting and undersampling, Amino Acids 39 (5) (2010) 1385–1391.

[10] H. Yu, C. Mu, C. Sun, W. Yang, X. Yang, X. Zuo, Support vector machine-based optimized decision threshold adjustment strategy for classifying imbalanced data, Knowl.-Based Syst. 76 (2015) 67–78.

[11] X.-Y. Liu, J. Wu, Z.-H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Trans. Syst. Man Cybern. B Cybern. 39 (2) (2009) 539–550.

[12] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. (2002) 321–357.

[13] N.J. Benjamin, X. Wang, Imbalanced data set learning with synthetic samples, IRIS Machine Learning Workshop, 2004.

[14] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning, Advances in Intelligent Computing, Springer 2005, pp. 878–887.

[15] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-level-SMOTE: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, in: T. Theeramunkong, B. Kijsirikul, N. Cercone, T.-B. Ho (Eds.), Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27–30, 2009 Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg 2009, pp. 475–482.

[16] D.A. Cieslak, N.V. Chawla, A. Striegel, Combating imbalance in network intrusion datasets, IEEE International Conference on Granular Computing 2006, pp. 732–737.

[17] S. Barua, M.M. Islam, X. Yao, K. Murase, MWMOTE—majority weighted minority oversampling technique for imbalanced data set learning, IEEE Trans. Knowl. Data Eng. 26 (2) (2014) 405–425.

[18] J.A. Sáez, J. Luengo, J. Stefanowski, F. Herrera, SMOTE–IPF: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering, Inf. Sci. 291 (2015) 184–203.

[19] H. He, Y. Bai, E. Garcia, S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, IEEE 2008, pp. 1322–1328.

[20] A. Pourhabib, B.K. Mallick, Y. Ding, Absent data generating classifier for imbalanced class sizes, J. Mach. Learn. Res. vol. 16 (1) (2015) 2695–2724.

[21] S. Piri, D. Delen, T. Liu, H.M. Zolbanin, A data analytics approach to building a clinical decision support system for diabetic retinopathy: developing and deploying a model ensemble, Decis. Support. Syst. (2017, 101, 12–27).

[22] B. Wang, N. Japkowicz, Imbalanced data set learning with synthetic samples, Proc. IRIS Machine Learning Workshop 2004, p. 19.

[23] C. Elkan, The foundations of cost-sensitive learning, International Joint Conference on Artificial Intelligence, 17, no. 1, 2001, pp. 973–978.

[24] R. Longadge, S. Dongre, Class Imbalance Problem in Data Mining ReviewarXiv preprint arXiv:1305.1707 2013.

[25] Y. Freund, R.E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, European Conference on Computational Learning Theory, Springer, Berlin Heidelberg 1995, pp. 23–37.

[26] Y. Sun, M.S. Kamel, A.K. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recogn. vol. 40 (12) (2007) 3358–3378.

[27] W. Fan, S.J. Stolfo, J. Zhang, P.K. Chan, AdaCost: misclassification cost-sensitive boosting, ICML 1999, pp. 97–105.

[28] W. Lee, C.-H. Jun, J.-S. Lee, Instance categorization by support vector machines to adjust weights in AdaBoost for imbalanced data classification, Inf. Sci. 381 (2017) 92–103.

[29] C. Drummond, R.C. Holte, Exploiting the cost (in) sensitivity of decision tree splitting criteria, ICML, vol. 1, 2000 (no. 1).

[30] M. Kukar, I. Kononenko, Cost-sensitive learning with neural networks, ECAI 1998, pp. 445–449.

[31] K. Veropoulos, C. Campbell, N. Cristianini, Controlling the sensitivity of support vector machines, Proceedings of the International Joint Conference on AI 1999, pp. 55–60.

[32] G. Wu, E.Y. Chang, Class-boundary alignment for imbalanced dataset learning, ICML 2003 Workshop on Learning From Imbalanced Data Sets II, Washington, DC 2003, pp. 49–56.

[33] R. Akbani, S. Kwek, N. Japkowicz, Applying support vector machines to imbalanced datasets, European Conference on Machine Learning, Springer 2004, pp. 39–50.

[34] B. Wang, N. Japkowicz, Boosting support vector machines for imbalanced data sets, Knowl. Inf. Syst. 25 (1) (2010) 1–20.

[35] J. Mathew, M. Luo, C.K. Pang, H.L. Chan, Kernel-based SMOTE for SVM classification of imbalanced datasets, Industrial Electronics Society, IECON 2015-41st Annual Conference of the IEEE, IEEE 2015, pp. 001127–001132.

[36] C. Jian, J. Gao, Y. Ao, A new sampling method for classifying imbalanced data based on support vector machine ensemble, Neurocomputing vol. 193 (2016) 115–122.

[37] Y.-H. Shao, W.-J. Chen, J.-J. Zhang, Z. Wang, N.-Y. Deng, An efficient weighted Lagrangian twin support vector machine for imbalanced data classification, Pattern Recogn. vol. 47 (9) (2014) 3158–3167.

[38] Y. Tang, Y.-Q. Zhang, Granular SVM with repetitive undersampling for highly imbalanced protein homology prediction, 2006 IEEE International Conference on Granular Computing, IEEE 2006, pp. 457–460.

[39] R. Batuwita, V. Palade, Efficient resampling methods for training support vector machines with imbalanced datasets, The 2010 International Joint Conference on Neural Networks (IJCNN), IEEE 2010, pp. 1–8.

[40] M.A.H. Farquad, I. Bose, Preprocessing unbalanced data using support vector machine, Decision Support Systems, vol. 53, 1, 2012, pp. 226–233 (4//).

[41] C.J. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Disc. 2 (2) (1998) 121–167.

[42] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, Proceedings of the Fifth Annual Workshop on Computational Learning Theory, ACM 1992, pp. 144–152.

[43] G. Wu, E.Y. Chang, Adaptive feature-space conformal transformation for imbalanced-data learning, ICML 2003, pp. 816–823.

[44] N.V. Chawla, Data mining for imbalanced datasets: an overview, Data Mining and Knowledge Discovery Handbook, Springer 2005, pp. 853–867.

[45] G. Seni, J.F. Elder, Ensemble methods in data mining: improving accuracy through combining predictions, Synthesis Lectures on Data Mining and Knowledge Discovery, 2, 1, 2010, pp. 1–126.

Saeed Piri is a visiting assistant professor of Management Science and Information Systems in the Spears School of Business at Oklahoma State University (OSU). Heearned his Ph.D. in Industrial Engineering and Management from OSU in 2017, and his Master's degree in Industrial Engineering from Sharif University of Technology in 2008. His research has been published in major journals, including Decision Support systems and Expert Systems with Applications. He has presented his research in several national and international conferences, including INFORMS, DSI, and IISE. His research interests include developing decision support systems, machine learning, imbalanced data learning, ensemble models, and application of business analytics in healthcare domain.

Dursun Delen is the holder of William S. Spears and Neal Patterson Endowed Chairs in Business Analytics, Director of Research for the Center for Health Systems Innovation, and Professor of Management Science and Information Systems in the Spears School of Business at Oklahoma State University (OSU). He received his Ph.D. in Industrial Engineering and Management from OSU in 1997. Prior to his appointment as an Assistant Professor at OSU in 2001, he worked for a privately-owned research and consultancy company, Knowledge Based Systems Inc., in College Station, Texas, as a research scientist for five years, during which he led a number of decision support, information systems and advanced analytics related research projects funded by federal agencies, including DoD, NASA, NIST and DOE. His research has appeared in major journals including Decision Support Systems, Communications of the ACM, Computers and Operations Research, Computers in Industry, Journal of Production Operations Management, Artificial Intelligence in Medicine, Expert Systems with Applications, among others. He recently published seven books/textbooks in the broader are of Business Analytics. He is often invited to national and international conferences for keynote addresses on topics related to Healthcare Analytics, Data/Text Mining, Business Intelligence, Decision Support Systems, Business Analytics and Knowledge Management. He regularly serves and chairs tracks and mini-tracks at various information systems and analytics conferences, and serves on several academic journals as editor-in-chief, senior editor, associate editor and editorial board member. His research and teaching interests are in data and text mining, decision support systems, knowledge management, business intelligence and enterprise modeling.

Tieming Liu is an associate professor at the School of - Industrial Engineering and Management, Oklahoma State University. He received his doctoral degree in Transportation and Logistics from the Massachusetts Institute of Technology in 2005, his master's degree in Industrial Engineering and Management Science from Northwestern University in 2001, and his master's and bachelor's degrees in Control Theory and Control Engineering from Tsinghua University in 2000 and 1997, respectively. His research interests include supply chain management, logistics planning, and healthcare analytics. His research has been published in major journals, including IIE Transactions, Interfaces, Production and Operations Management, Naval Research Logistics, Operations Research Letters, European Journal of Operational Research, among others.