Proceedings of International Joint Conference on Neural Networks, San Jose, California, USA, July 31 – August 5, 2011

On Combination of SMOTE and Particle Swarm Optimization Based Radial Basis Function Classifier for Imbalanced Problems

Ming Gao, Xia Hong, Sheng Chen and Chris J. Harris

***Abstract*—The combination of the synthetic minority over-sampling technique (SMOTE) and the radial basis function (RBF) classifier is proposed to deal with classification for imbalanced two-class data. In order to enhance the significance of the small and specific region belonging to the positive class in the decision region, the SMOTE is applied to generate synthetic instances for the positive class to balance the training data set. Based on the over-sampled training data, the RBF classifier is constructed by applying the orthogonal forward selection procedure, in which the classifier structure and the parameters of RBF kernels are determined using a particle swarm optimization algorithm based on the criterion of minimizing the leave-one-out misclassification rate. The experimental results on both simulated and real imbalanced data sets are presented to demonstrate the effectiveness of our proposed algorithm.**

I. Introduction

Generally speaking, an imbalanced problem occurs when the instances in one or several classes (the majority classes) outnumber the instances of the other classes (the minority classes), which usually are the more important classes. Such an imbalance in the data represents the so-called between-class imbalance [1], in contrast to the related issue of within-class imbalance [2][3]. Imbalanced problems widely exist in the field of medical diagnosis, such as surveillance of nosocomial infection [4], cardiac care [5] and elucidating protein-protein interactions [6]) as well as in many other fields, such as fraud detection [7][8], network intrusion detection [9], telecommunication management [10], and so on. In the study of two-class imbalanced problem, the instances in the majority class are referred to as negative, while in its counterpart, the minority class, the instances are referred to as positive. Since in practice the minority class is more important, one should be more concerned with the positive instances. Imbalanced data learning has been widely researched [11]-[16]. Typically, the approaches to solving the imbalanced problem can be divided into two categories: re-sampling methods and imbalanced learning algorithms.

The re-sampling approach is actually a re-balancing process to balance the given imbalanced data set. The studies [17][18] on class distribution have shown that balanced data sets provide better learning performance than imbalanced ones, though some other studies [1][19] argue that imbalanced data sets are not necessarily responsible for the poor performance of some classifiers. Re-sampling techniques are attractive under most imbalanced circumstances. This is because re-sampling adjusts only the original training data set, instead of modifying the learning algorithm. Thus, this approach is external and transportable [18][20], and it provides a convenient and effective way to deal with imbalanced learning problems using standard classifiers. Specifically, the re-sampling methods include the random over-sampling, which randomly appends replicated instances to the positive class, and the random under-sampling, which randomly removes instances from the majority class. Alternatively, there exist the guided over-sampling and under-sampling, respectively, of which the choices to replace or to eliminate are informed rather than random. In addition, the synthetic minority over-sampling technique (SMOTE) [21] is a well acknowledged over-sampling method. In the SMOTE, instead of mere data oriented duplicating, the positive class is over-sampled by creating synthetic instances in the feature space formed by the instance and its K-nearest neighbors.

The second category, the imbalanced learning algorithms, can be regarded as a process to modify or improve the existing learning algorithms so that they can deal with imbalanced problems effectively. The imbalanced learning algorithms include the cost-sensitive method [22]-[25], the discrimination-based and recognition-based approaches [3]. Noticeably, kernel-based learning, such as support vector machine (SVM) and radial basis function (RBF), is the state-of-the-arts approach for solving imbalanced learning problems. The study [1] shows that kernel-based methods provide a relatively robust classification to imbalanced problems. Nevertheless, the detrimental effects of an imbalanced data set can be sufficiently serious to prevent kernel-based classifiers from achieving the optimal classifier's performance.

In order to achieve better classification performance, an effective approach is to integrate kernel-based classifiers with re-sampling methods. The previous studies [26]-[28] mainly focused on SVMs. Specifically, the method [26] combined the SMOTE with different costs to bias SVMs by assigning different classes with different costs so as to shift the decision boundary away from the positive instances and to define a better boundary. The work [27] proposed ensemble systems by re-sampling data sets to form the input to the standard SVM classifier, while the method [28] introduced asymmetric misclassification costs in SVMs so as to improve classification performance. Another integration of SVM with under-sampling method used the combination of the granular support vector machine (GSVM) [29] and repetitive under-sampling (RU) to form the GSVM-RU algorithm [30]. An alternative approach is to adapt kernel-based classifiers to imbalanced data sets by modifying the kernel construction

M. Gao and X. Hong are with the School of Systems Engineering, University of Reading, Reading RG6 6AY, UK (E-mails: ming.gao@pgr.reading.ac.uk; x.hong@reading.ac.uk).

S. Chen and C.J. Harris are with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK (E-mails: sqc@ecs.soton.ac.uk; cjh@ecs.soton.ac.uk).

This work was supported by UK EPSRC.

978-1-4244-9636-5/11/$26.00 ©2011 IEEE

and model selection procedure. A representative work of this approach [31] proposed a regularized orthogonal weighted least square (ROWLS) kernel estimator using the orthogonal forward selection (OFS) based on the model selection criterion of maximizing the leave-one-out area under the curve (LOO-AUC) of receiver operating characteristics (ROC).

For balanced data sets, the prevalent approach for constructing the RBF and other sparse kernel classifiers is to assign a fixed common variance for every kernel and to select input data as the candidate centers for RBF kernels by minimizing the leave-one-out (LOO) misclassification rate in the efficient OFS procedure [32]. This approach has its root in regression application [33]-[36]. There are two limitations with this "fixed" RBF kernel approach. Firstly, kernels cannot be flexibly tuned, as the position of each kernel is restricted to the input data and the shape of each kernel is fixed rather than determined by the model learning procedure. Secondly, the common kernel variance has to be determined via cross validation, which inevitably increases the computational cost. The previous studies [37]-[39] constructed the tunable RBF classifier based on the OFS procedure using a global search optimization algorithm [40] to optimize the RBF kernels one by one. This tunable RBF kernel approach was observed to produce sparser classifiers with better performance but higher computational complexity in classifier construction, in comparison with the standard fixed kernel approach. Recently, the particle swarm optimization (PSO) algorithm [41] was adopted to minimize the LOO misclassification rate in the OFS construction of tunable RBF classifier [42][43]. PSO [41] is an efficient population-based stochastic optimization technique inspired by social behaviour of bird flocks or fish schools, and it has been successfully applied to wide-ranging optimization applications [44]-[48]. Owing to the efficiency of PSO, the tunable RBF modeling approach advocated in [42][43] offers significant advantages in terms of better generalization performance and smaller classifier size as well as lower complexity in learning process, compared with the standard fixed kernel approach. This PSO aided tunable RBF classifier, therefore, offers the state-of-the-art for balanced data sets. When dealing with highly imbalanced problems, however, its performance may degrade.

Against this background, our novel contribution is to combine the SMOTE algorithm and the PSO aided RBF classifier to deal with two-class imbalanced classification problems effectively. The SMOTE is first applied to generate synthetic instances in the positive class to balance the training data set. Using the resulting balanced data set, the tunable RBF classifier is then constructed by applying the PSO to minimize the LOO misclassification rate in the computationally efficient OFS procedure. Experimental results obtained demonstrate that the proposed method is competitive to other existing state-of-the-arts methods for two-class imbalanced problems. The rest of the paper is organized as follows. Section II introduces the tunable RBF model for two-class classification and the OFS procedure based on the LOO misclassification rate, while Section III presents the PSO algorithm for tuning the RBF kernels by minimizing the LOO misclassification rate. Section IV introduces the SMOTE method and presents the proposed combined SMOTE and PSO based RBF algorithm. The effectiveness of our approach is demonstrated by numerical examples in Section V, and our conclusions are given in Section VI.

II. Tunable RBF modeling for classification

Consider the two-class data set $D_N = \{\mathbf{x}_k, y_k\}_{k=1}^N$, where $y_k = \{\pm 1\}$ denotes the class label for the feature vector $\mathbf{x}_k \in \Re^m$, while there are N_+ positive instances and N_- negative instances, with $N = N_+ + N_-$. We use the data set D_N to construct the RBF classifier of the form:

$$\begin{aligned} \hat{y}_k^{(M)} &= \sum_{i=1}^{M} w_i g_i(\mathbf{x}_k) = \mathbf{g}_M^{\mathrm{T}}(k)\mathbf{w}_M \\ \tilde{y}_k^{(M)} &= \mathrm{sgn}\big(\hat{y}_k^{(M)}\big) \end{aligned} \tag{1}$$

where M is the number of RBF kernels, $\hat{y}_k^{(M)}$ is the output of the M-term classifier with the M kernels, $g_i(\bullet)$ for $1 \le i \le M$, and $\tilde{y}_k^{(M)}$ denotes the corresponding estimated class label for $\mathbf{x}_k$, while $\mathbf{w}_M = \begin{bmatrix} w_1 & w_2 \cdots w_M \end{bmatrix}^{\mathrm{T}}$ is the weight vector and $\mathbf{g}_M^{\mathrm{T}}(k) = \begin{bmatrix} g_1(\mathbf{x}_k) & g_2(\mathbf{x}_k) \cdots g_M(\mathbf{x}_k) \end{bmatrix}$. In this study, we use the Gaussian kernel function

$$g_i(\mathbf{x}) = \exp\left(-(\mathbf{x} - \mathbf{c}_i)^{\mathrm{T}} \mathbf{\Sigma}_i^{-1} (\mathbf{x} - \mathbf{c}_i)\right) \tag{2}$$

where $\mathbf{c}_i \in \Re^m$ is the center vector of the ith RBF kernel and $\mathbf{\Sigma}_i = \mathrm{diag}\{\sigma_{i,1}^2, \sigma_{i,2}^2, \cdots, \sigma_{i,m}^2\}$ is the diagonal covariance matrix of the ith kernel. Hence, the position of each kernel, $\mathbf{c}_i$, and coverage of each kernel, $\mathbf{\Sigma}_i$, are both considered as the parameters to be determined in kernel modeling.

From (1), the RBF classifier over D_N can be written in the matrix form as

$$\mathbf{y} = \mathbf{G}_M \mathbf{w}_M + \mathbf{e}^{(M)} \tag{3}$$

where $\mathbf{e}^{(M)} = \begin{bmatrix} e_1^{(M)} & e_2^{(M)} \cdots e_N^{(M)} \end{bmatrix}^{\mathrm{T}}$ is the error vector with the M-term modeling error $e_k^{(M)} = y_k - \hat{y}_k^{(M)}$, $\mathbf{y} = \begin{bmatrix} y_1 & y_2 \cdots y_N \end{bmatrix}^{\mathrm{T}}$ is the desired class label vector, and the kernel matrix $\mathbf{G}_M = \begin{bmatrix} \mathbf{g}_1 & \mathbf{g}_2 \cdots \mathbf{g}_M \end{bmatrix}$ with $\mathbf{g}_l = \begin{bmatrix} g_l(\mathbf{x}_1) & g_l(\mathbf{x}_2) \cdots g_l(\mathbf{x}_N) \end{bmatrix}^{\mathrm{T}}$ for $1 \le l \le M$. Note that $\mathbf{g}_l$ is the lth column of $\mathbf{G}_M$ while $\mathbf{g}_M^{\mathrm{T}}(k)$ is the kth row of $\mathbf{G}_M$.

Consider the orthogonal decomposition $\mathbf{G}_M = \mathbf{P}_M \mathbf{A}_M$, where

$$\mathbf{A}_M = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,M} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{M-1,M} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{4}$$

$$\mathbf{P}_M = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 \cdots \mathbf{p}_M \end{bmatrix} \tag{5}$$

and the columns in (5) satisfy $\mathbf{p}_i^{\mathrm{T}} \mathbf{p}_j = 0$ for $i \ne j$. The RBF classifier (3) can alternatively be represented as:

$$\mathbf{y} = \mathbf{P}_M \boldsymbol{\theta}_M + \mathbf{e}^{(M)} \tag{6}$$

where $\boldsymbol{\theta}_M = \begin{bmatrix} \theta_1 & \theta_2 \cdots \theta_M \end{bmatrix}^{\mathrm{T}}$ satisfies $\boldsymbol{\theta}_M = \mathbf{A}_\mathbf{M} \mathbf{w}_M$. The space spanned by the original model bases $\mathbf{g}_i$, $1 \le i \le M$, is identical to that spanned by $\mathbf{p}_i$, $1 \le i \le M$.

The OFS procedure constructs the RBF kernels one by one by minimizing the LOO misclassification rate [42][43]. Specifically, at the nth stage, the nth RBF kernel ($\mathbf{p}_n$ and θ_n) is determined. Define the LOO-model output of the n-term RBF model constructed from the LOO data set $D_N \backslash (\mathbf{x}_k, y_k)$, calculated at $\mathbf{x}_k$, as $\hat{y}_k^{(n,-k)}$. Further define the associated LOO decision variable as

$$s_k^{(n,-k)} = \mathrm{sgn}(y_k)\hat{y}_k^{(n,-k)} = y_k \hat{y}_k^{(n,-k)} \tag{7}$$

Then the LOO misclassification rate is defined by [32]

$$J_{\mathrm{LOO}}^{(n)} = \frac{1}{N}\sum_{k=1}^{N} \mathcal{I}_d\big(s_k^{(n,-k)}\big) \tag{8}$$

in which the indicator function $\mathcal{I}_d(s)$ is defined as

$$\mathcal{I}_d(s) = \begin{cases} 1, & s \le 0 \\ 0, & s > 0 \end{cases} \tag{9}$$

By making use of Sherman-Morrison-Woodbury theorem [49] as well as the orthogonal property, the LOO decision variable can be efficiently calculated according to [32][42][43]

$$s_k^{(n,-k)} = \frac{\psi_k^{(n)}}{\eta_k^{(n)}} \tag{10}$$

in which $\psi_k^{(n)}$ and $\eta_k^{(n)}$ can be computed recursively by:

$$\psi_k^{(n)} = \psi_k^{(n-1)} + y_k\theta_n p_n(k) - \frac{p_n^2(k)}{\mathbf{p}_n^{\mathrm{T}}\mathbf{p}_n + \lambda} \tag{11}$$

$$\eta_k^{(n)} = \eta_k^{(n-1)} - \frac{p_n^2(k)}{\mathbf{p}_n^{\mathrm{T}}\mathbf{p}_n + \lambda} \tag{12}$$

where $p_n(k)$ is the kth element of $\mathbf{p}_n$ and $\lambda \ge 0$ is a small regularization parameter.

At the nth stage of the OFS procedure, the nth RBF kernel, namely, its center vector $\mathbf{c}_n$ and diagonal covariance matrix $\mathbf{\Sigma}_n$, are determined by minimizing $J_{\mathrm{LOO}}^{(n)}$. The construction terminates at the size of M when $J_{\mathrm{LOO}}^{(M+1)} \ge J_{\mathrm{LOO}}^{(M)}$ [32][42][43].

III. PSO for optimizing RBF parameters

Let $\boldsymbol{\mu} = [\mu(1)\ \mu(2)\cdots\mu(2m)]^{\mathrm{T}}$ be the $2m$-dimensional vector that contains $\mathbf{c}_n$ and $\mathbf{\Sigma}_n$. Then, as defined in the previous section, the problem of determining the nth RBF kernel's parameters at the nth OFS stage is to solve the following optimization problem

$$\hat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu}\in\mathbf{\Gamma}} J_{\mathrm{LOO}}^{(n)} \tag{13}$$

where $\mathbf{\Gamma}$ defines the $2m$-dimensional search space for the parameter vector $\boldsymbol{\mu}$, which is specified by the two values $\mathbf{\Gamma}_{\min} = \big[\Gamma_{1,\min}\ \Gamma_{2,\min}\cdots\Gamma_{2m,\min}\big]^{\mathrm{T}}$ and $\mathbf{\Gamma}_{\max} = \big[\Gamma_{1,\max}\ \Gamma_{2,\max}\cdots\Gamma_{2m,\max}\big]^{\mathrm{T}}$ as follows. The search space for $\mathbf{c}_n = \big[c_{n,1}\ c_{n,2}\cdots c_{n,m}\big]^{\mathrm{T}}$ is specified by the distribution of the training data $\big\{\mathbf{x}_k = \big[x_{k,1}\ \ x_{k,2}\cdots x_{k,m}\big]^{\mathrm{T}}\big\}_{k=1}^{N}$, namely,

$$c_{n,i} \in \big[x_{\min,i},\ x_{\max,i}\big] \triangleq \big[\Gamma_{i,\min},\ \Gamma_{i,\max}\big], 1 \le i \le m \tag{14}$$

with $x_{\min,i} = \min\limits_{1\le k\le N} x_{k,i}$ and $x_{\max,i} = \max\limits_{1\le k\le N} x_{k,i}$, while each element of $\mathbf{\Sigma}_n$ is limited in the range

$$\begin{aligned}\sigma_{n,i}^2 \ &\in\ \big[\sigma_{\min}^2,\ \sigma_{\max}^2\big] \\ &\triangleq \big[\Gamma_{(i+m),\min},\ \Gamma_{(i+m),\max}\big],\ 1 \le i \le m\end{aligned} \tag{15}$$

When applying a PSO [41] to solve the optimisation (13), a swarm of the candidate particles $\big\{\boldsymbol{\mu}_i^{[l]}\big\}_{i=1}^{S}$ are "flying" in the search space $\mathbf{\Gamma}$ in order to find a solution $\hat{\boldsymbol{\mu}}$, where S is the size of the swarm and $l \in \{0, 1, \cdots, L\}$ denotes the lth movement of the swarm. Each particle $\boldsymbol{\mu}$ has a $2m$-dimensional velocity $\boldsymbol{\nu} = [\nu(1)\ \nu(2)\cdots\nu(2m)]^{\mathrm{T}}$ to direct its search, and the velocity $\boldsymbol{\nu} \in \mathbf{V}$ with the velocity space defined by $\mathbf{V} = \big[-\mathbf{V}_{\max},\ \mathbf{V}_{\max}\big]$, where $\mathbf{V}_{\max} = \big[V_{1,\max}\ V_{2,\max}\cdots V_{2m,\max}\big]^{\mathrm{T}} = \frac{1}{2}\big(\mathbf{\Gamma}_{\max} - \mathbf{\Gamma}_{\min}\big)$.

To start the PSO, the candidate particles $\big\{\boldsymbol{\mu}_i^{[0]}\big\}_{i=1}^{S}$ are initialized randomly within $\mathbf{\Gamma}$, and the velocity for each candidate particle is initialized to zero, namely, $\big\{\boldsymbol{\nu}_i^{[0]} = \mathbf{0}\big\}_{i=1}^{S}$. The cognitive information $\boldsymbol{pb}_i^{[l]}$ and the social information $\boldsymbol{gb}^{[l]}$ record the best position visited by the particle i and the best position visited by the entire swarm, respectively, during the l movements. For notational convenience, we denote the LOO cost calculated on $\boldsymbol{pb}_i^{[l]}$ as $J_{\mathrm{LOO}}^{(n)}\big(\boldsymbol{pb}_i^{[l]}\big)$ and the LOO cost calculated on $\boldsymbol{gb}^{[l]}$ as $J_{\mathrm{LOO}}^{(n)}\big(\boldsymbol{gb}^{[l]}\big)$. The cognitive information $\boldsymbol{pb}_i^{[l]}$ and the social information $\boldsymbol{gb}^{[l]}$ are used to update the velocities and positions according to

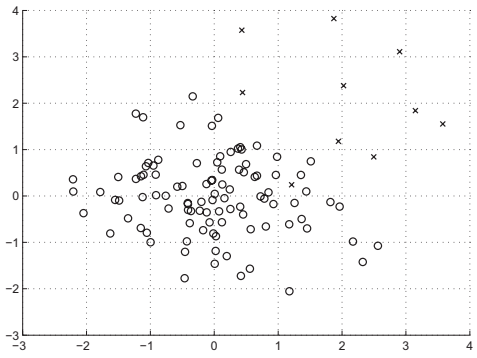
$$\begin{aligned}\boldsymbol{\nu}_i^{[l+1]} \ &= \ a\cdot\boldsymbol{\nu}_i^{[l]} + rand()\cdot b\cdot\big(\boldsymbol{pb}_i^{[l]} - \boldsymbol{\mu}_i^{[l]}\big) \\ &\quad + rand()\cdot c\cdot\big(\boldsymbol{gb}^{[l]} - \boldsymbol{\mu}_i^{[l]}\big)\end{aligned} \tag{16}$$

$$\boldsymbol{\mu}_i^{[l+1]} \ = \ \boldsymbol{\mu}_i^{[l]} + \boldsymbol{\nu}_i^{[l+1]} \tag{17}$$

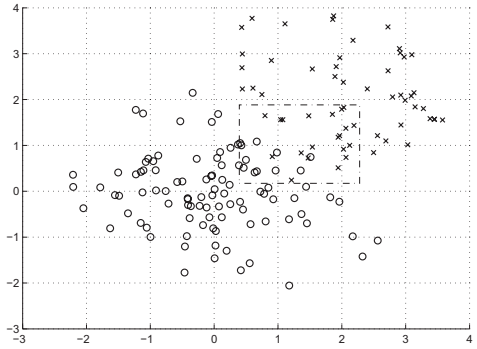
where a denotes the inertia weight, $rand()$ is the random number uniformly distributed in $[0,\ 1]$, b and c are the two acceleration coefficients. Experimental results given in [43] show that a better performance can be achieved by using $a = rand()$ instead of a constant inertia weight. Adopting the time varying acceleration coefficients (TVAC) [44], in which $b = 2.5 - (2.5-0.5)\cdot l/L$ and $c = 0.5 + (2.5-0.5)\cdot l/L$, can often enhance the performance of PSO. The search space $\mathbf{\Gamma}$ and the velocity space $\mathbf{V}$ are used to confine $\boldsymbol{\mu}_i^{[l+1]}$ and $\boldsymbol{\nu}_i^{[l+1]}$ derived from (16) and (17), respectively. If $\boldsymbol{\nu}_i^{[l+1]}$ becomes too close to $\mathbf{0}$, a random re-initialization is needed, which may take the form $\boldsymbol{\nu}_i^{[l+1]} = \pm 0.1\cdot rand()\cdot\mathbf{V}_{max}$. The detailed PSO aided OFS algorithm can be found in [43].

IV. Combined SMOTE and PSO optimized RBF for imbalanced classification

The SMOTE [21] over-samples the positive class by creating synthetic instances by a specified over-sampling ratio of the original minority data size, $\beta\%$. Based on each minority data sample, denoted by $\mathbf{x}_o$, $\beta\%$ synthetic data points are generated by randomly selecting data points on the lines linking $\mathbf{x}_o$ with some of its K nearest neighbors, where K is predetermined. Depending on the required SMOTE amount $\beta\%$, one out of the K nearest positive class data samples are randomly selected several times. For example, if $\beta\% = 600\%$



(a)



(b)

Fig. 1. (a) Original training data space, and (b) training data space after SMOTE over-sampling the positive class by 500% of its original size, where x denotes positive class instance while ○ denotes negative class instance

and $K = 5$, then one out of five nearest neighbors of $\mathbf{x}_o$ is randomly chosen repeatedly for six times. Each time a random kth neighbor is selected to create a line linking $\mathbf{x}_o$ to this neighbor, and then a single synthetic instance is created by randomly selecting a point on the line. Thus any synthetic instance $\mathbf{x}_s$ is given by

$$\mathbf{x}_s = \mathbf{x}_o + \delta \cdot \left(\mathbf{x}_o^{\{t\}} - \mathbf{x}_o\right) \tag{18}$$

where $\mathbf{x}_s$ denotes one synthetic instance, $x_o^{\{t\}}$ is the tth nearest neighbors of $\mathbf{x}_o$ in the positive class, and $\delta \in [0,\ 1]$ is a random number. The procedure is repeated for all the positive data points.

A major problem caused by imbalanced data sets is that most classifiers tend to attribute the positive class instances within the decision region to the negative class, due to insufficient positive-class training instances in the decision region. As a result, the trained decision boundary tends to be far away from the negative class and too close to the positive class. The contribution of SMOTE is to enhance the significance of the small and specific region belonging to the positive class in the decision region, which leads to the better generalization for classifiers. Fig. 1(a) shows a simulated imbalanced data set, the details of which are specified in Section V. After the SMOTE over-sampling the positive class by 500% of its original size, the instances from the positive class become more significant in the decision region (the area specified by dash-dot line), as shown in Fig. 1(b), compared with the original data set. Consequently, the trained decision boundary tends to be further away from the positive class.

We combine this SMOTE with the PSO optimized RBF classifier described in Section III to create a powerful algorithm for combating two-class imbalanced problems. This combined SMOTE and PSO aided RBF is detailed below.

1) Algorithm initialization
   (a) Specify the balanced degree $\beta\%$ and the value of K. Create the new training data set $\tilde{D}_N$ by appending the generated positive training data points to the original training data set via the SMOTE.
   (b) Specify the search space $\mathbf{\Gamma}$ and the velocity space $\mathbf{V}$. Specify the values of L and S.
   (c) Set $J_{\rm LOO}^{(0)} = 1$, $\psi_k^{(0)} = 0$, and $\eta_k^{(0)} = 1$.
2) Construct the nth RBF kernel
   (a) PSO initialization: Randomly initialize $\{\boldsymbol{\mu}_i^{[0]}\}_{i=1}^S$ in $\mathbf{\Gamma}$, and set $\{\boldsymbol{\nu}_i^{[0]} = \mathbf{0}\}_{i=1}^S$.
   (b) For $0 \le l < L$:
   (b.i) Construct the candidates $\mathbf{g}_n^{\{i\}}$ from $\boldsymbol{\mu}_i^{[l]}$, for $1 \le i \le S$. Then, for $1 \le i \le S$ and $1 \le j < n$, compute:

$$a_{j,n}^{\{i\}} = \begin{cases} 1, & n = 1 \\ \frac{\mathbf{p}_j^{\rm T}\mathbf{g}_n^{\{i\}}}{\mathbf{p}_j^{\rm T}\mathbf{p}_j}, & n > 1 \end{cases}$$

$$\mathbf{p}_n^{\{i\}} = \begin{cases} \mathbf{g}_n^{\{i\}}, & n = 1 \\ \mathbf{g}_n^{\{i\}} - \sum_{j=1}^{n-1} a_{j,n}^{\{i\}}\mathbf{p}_j, & n > 1 \end{cases}$$

$$\theta_n^{\{i\}} = \frac{\left(\mathbf{p}_n^{\{i\}}\right)^{\rm T}\mathbf{y}}{\left(\mathbf{p}_n^{\{i\}}\right)^{\rm T}\mathbf{p}_n^{\{i\}} + \lambda}$$

   (b.ii) For $1 \le i \le S$ and $1 \le k \le N$, compute:

$$\psi_k^{(n)}\{i\} = \psi_k^{(n-1)} + y(k)\theta_n^{\{i\}}p_n^{\{i\}}(k) - \frac{\left(p_n^{\{i\}}(k)\right)^2}{\left(\mathbf{p}_n^{\{i\}}\right)^{\rm T}\mathbf{p}_n^{\{i\}} + \lambda}$$

$$\eta_k^{(n)}\{i\} = \eta_k^{(n-1)} - \frac{\left(p_n^{\{i\}}(k)\right)^2}{\left(\mathbf{p}_n^{\{i\}}\right)^{\rm T}\mathbf{p}_n^{\{i\}} + \lambda}$$

   Then, for $1 \le i \le S$, calculate the LOO costs:

$$J_{\rm LOO}^{(n)}\{i\} = \frac{1}{N}\sum_{k=1}^{N}\mathcal{I}_d\left(\frac{\psi_k^{(n)}\{i\}}{\eta_k^{(n)}\{i\}}\right)$$

   (b.iii) For $1 \le i \le S$:
   If $J_{\rm LOO}^{(n)}\{i\} < J_{\rm LOO}^{(n)}\left(\boldsymbol{pb}_i^{[l]}\right)$: $\boldsymbol{pb}_i^{[l]} = \boldsymbol{\mu}_i^{[l]}$ and $J_{\rm LOO}^{(n)}\left(\boldsymbol{pb}_i^{[l]}\right) = J_{\rm LOO}^{(n)}\{i\}$
   Then find $i_* = \arg\min_{1\le i\le S} J_{\rm LOO}^{(n)}\left(\boldsymbol{pb}_i^{[l]}\right)$
   If $J_{\rm LOO}^{(n)}\left(\boldsymbol{pb}_{i_*}^{[l]}\right) < J_{\rm LOO}^{(n)}\left(\boldsymbol{gb}^{[l]}\right)$: $\boldsymbol{gb}^{[l]} = \boldsymbol{pb}_{i_*}^{[l]}$ and $J_{\rm LOO}^{(n)}\left(\boldsymbol{gb}^{[l]}\right) = J_{\rm LOO}^{(n)}\left(\boldsymbol{pb}_{i_*}^{[l]}\right)$
   (b.iv) For $1 \le i \le S$:

$$\boldsymbol{\nu}_i^{[l+1]} = a \cdot \boldsymbol{\nu}_i^{[l]} + rand() \cdot b \cdot \left(\boldsymbol{pb}_i^{[l]} - \boldsymbol{\mu}_i^{[l]}\right) + rand() \cdot c \cdot \left(\boldsymbol{gb}^{[l]} - \boldsymbol{\mu}_i^{[l]}\right)$$

   If $\nu_i^{[l+1]}(j) = 0$: $\nu_i^{[l+1]}(j) = \pm 0.1 \cdot rand() \cdot V_{j,\max}$
   If $\nu_i^{[l+1]}(j) > V_{j,\max}$: $\nu_i^{[l+1]}(j) = V_{j,\max}$
   If $\nu_i^{[l+1]}(j) < -V_{j,\max}$: $\nu_i^{[l+1]}(j) = -V_{j,\max}$
   Then for $1 \le i \le S$:
   $\boldsymbol{\mu}_i^{[l+1]} = \boldsymbol{\mu}_i^{[l]} + \boldsymbol{\nu}_i^{[l+1]}$
   If $\mu_i^{[l+1]}(j) > \Gamma_{j,\max}$: $\mu_i^{[l+1]}(j) = \Gamma_{j,\max}$
   If $\mu_i^{[l+1]}(j) < \Gamma_{j,\min}$: $\mu_i^{[l+1]}(j) = \Gamma_{j,\min}$

(c) Termination of PSO: $\boldsymbol{gb}^{[L]}$ provides $\mathbf{c}_n$ and $\boldsymbol{\Sigma}_n$ with the associated LOO cost $J_{\mathrm{LOO}}^{(n)} = J_{\mathrm{LOO}}^{(n)}\left(\boldsymbol{gb}^{[L]}\right)$. The algorithm also generates $a_{j,n}$ for $1 \le j < n$, $\mathbf{p}_n$ and θ_n as well as $\psi_k^{(n)}$ and $\eta_k^{(n)}$ for $1 \le k \le N$.

3) OFS termination condition checking:
If $J_{\mathrm{LOO}}^{(n)} < J_{\mathrm{LOO}}^{(n-1)}$: $n = n + 1$, go to step 2);
Otherwise, $M = n - 1$, terminate the OFS procedure.

V. Experimental results

The effectiveness of the proposed SMOTE+PSO-OFS algorithm was investigated using a simulated imbalanced date set and three real imbalanced data sets. The first two real data sets were taken from [50], while the third real data set was from [51]. These three real data sets were chosen in the order of increasing imbalance. For each data set, the positive class was over-sampled at different rates $\beta\%$ of its original size using the SMOTE. For the synthetic data set, a separate test data set was used, while for the three real data sets, P-fold cross validation was used, to indicate the classifier generalization capability based on multiple specifications, including the true positive rate (TP%) and the false positive rate (FP%) [52], as well as the precision (Pr), the F-measure (F-meas) and the G-mean [53]. These criteria are commonly adopted as the performance metrics for evaluating imbalanced learning classifiers. The regularized least square parameter estimator (KRLS), the $\bar{K}$-nearest neighbor classifiers with $\bar{K} = 1$ and 3 ($1-NN$ and $3-NN$), as well as the LOO-AUC+OFS with different weight ρ were cited from [31] as the benchmarks for comparison in the synthetic and first two real data sets. For the third real data set, the weighted SVM with $(C+ = 1000, C- = -1000)$, the cost sensitive SVM (CS-SVM) with $(C+ = 1, C- = -0.1)$, the cost sensitive SUPANOVA with $(C+ = 1, C- = -0.1)$ and the LOO-AUC+OFS with different weight ρ were quoted from [31][51] as comparison.

Simulated imbalanced data set: The simulated data set was generated with the $m = 2$ features. The mean vector of the negative class was $[0\ 0]^{\mathrm{T}}$, while the mean vector of the positive class was $[2\ 2]^{\mathrm{T}}$. The covariance matrices of both the negative-class and positive-class instances were the same 2-dimensional identity matrix. The training data set contained 100 instances from the negative class and 10 instances from the positive class, as depicted in Fig. 1(a). The test data set contained 1000 instances from the negative class, and 100 instances from the positive class. The 5-nearest neighbor method was applied to generate synthetic training data in the SMOTE, with the over-sampling rate $\beta\%$ set to 0%, 100%, 500%, 1000%, 1500% and 2000%, respectively. For the SMOTE+PSO-OFS, the swarm size and the number of movements were set to $S = 10$ and $L = 20$. The test results obtained by the various classifiers are shown in Table I.

It can be seen from the results for the SMOTE+PSO-OFS listed in Table I that, as the over-sampling rate $\beta\%$ increases, typically TP% increases but FP% inevitably increases as well. A better tradeoff between TP% and FP% was achieved, however, at the over-sampling rate where the better G-mean and F-measure were obtained. Since the imbalance degree of the negative class to the positive class was $10:1$, the over-sampled positive instances made $\tilde{D}_N$ fully balanced at $\beta\% = 1000\%$. From Table I, it can be seen that the best test performance occurred at the over sampling rate around 500% to 1000%. The effect of the SMOTE on the decision boundary is shown in Fig. 2, where it can be seen that the decision boundary trained by the more balanced data set was pushed further away from the positive class. Compared with the other benchmark methods, the proposed SMOTE+PSO-OFS showed a competitive performance.

TABLE I
Test classification performance comparison for the synthetic data set

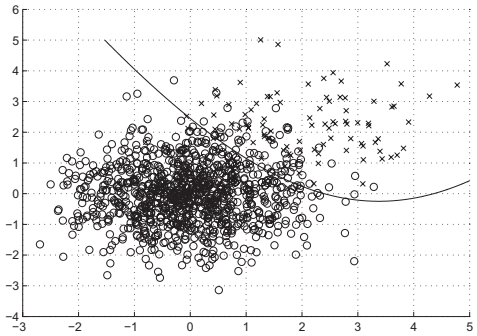
Method	TP%	FP%	Pr	G-mean	F-meas
KRLS with all data as centers	0.840	0.037	0.694	0.899	0.760
1-NN	0.830	0.047	0.638	0.899	0.722
3-NN	0.780	**0.022**	**0.780**	0.873	0.780
LOO-AUC+OFS ($\rho = 1$)	0.860	0.049	0.637	0.904	0.732
LOO-AUC+OFS ($\rho = 5$)	0.840	0.028	0.750	0.903	0.792
LOO-AUC+OFS ($\rho = 10$)	0.90	0.063	0.588	**0.918**	0.712
LOO-AUC+OFS ($\rho = 15$)	0.870	0.046	0.654	0.911	0.747
LOO-AUC+OFS ($\rho = 20$)	0.870	0.049	0.640	0.909	0.737
SMOTE+PSO-OFS ($\beta\% = 0\%$)	0.860	0.044	0.662	0.907	0.748
SMOTE+PSO-OFS ($\beta\% = 100\%$)	0.880	0.055	0.615	0.912	0.724
SMOTE+PSO-OFS ($\beta\% = 500\%$)	0.810	0.023	**0.780**	0.890	**0.794**
SMOTE+PSO-OFS ($\beta\% = 1000\%$)	0.890	0.053	0.627	**0.918**	0.736
SMOTE+PSO-OFS ($\beta\% = 1500\%$)	0.930	0.102	0.476	0.914	0.631
SMOTE+PSO-OFS ($\beta\% = 2000\%$)	**0.940**	0.110	0.461	0.915	0.618

Pima Indians diabetes data set [50]: The data set contained 768 instances from the two classes with 500 negative instances and 268 positive instances. The feature space dimension was $m = 8$. All the eight input features were normalized to the range $[0,\ 1]$ using the operation

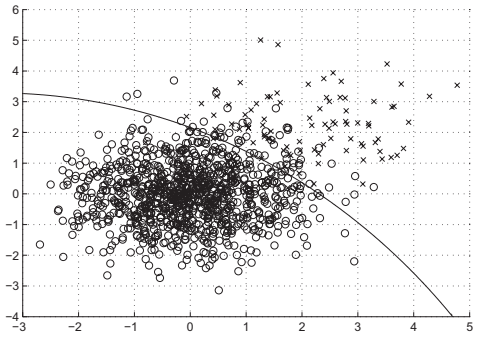
$$\bar{x}_{k,i} = \frac{x_{k,i} - x_{\min,i}}{x_{\max,i} - x_{\min,i}},\ 1 \le k \le N, 1 \le i \le m \qquad (19)$$

The 5-nearest neighbor scheme was applied to generate synthetic training data in the SMOTE. The over-sampling rate $\beta\%$ was set to 0%, 50%, 75%, 100%, 150%, 200%, 250% and 500%, respectively. The swarm size and the number of movements were set to $S = 10$ and $L = 20$ for the PSO. The 8-fold cross validation was used to investigate the test performance of a classifier. The 8-fold cross validation results for the various classifiers are shown in Table II.

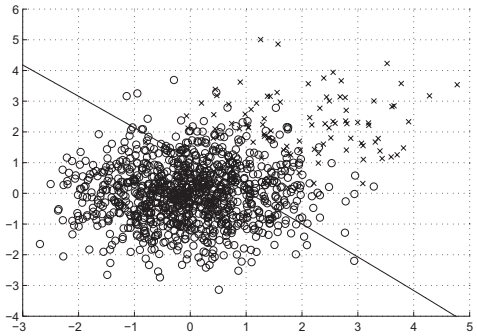
For the SMOTE+PSO-OFS, it can be seen that the best TP%, that is, the best detection capability for diabetes, occurred at $\beta\% = 500\%$, while the best FP% occurred at $\beta\% = 0\%$. But the best TP% was obtained at the expense of the worst FP%, and the best FP% was obtained at the expense of the worst TP%, as indicated by the poor values of the G-



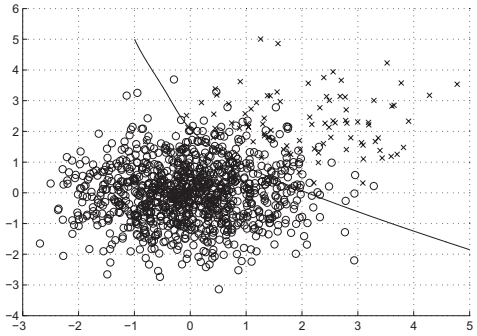
(a)



(b)



(c)



(d)

Fig. 2. Decision boundaries obtained by the SMOTE+PSO-OFS with different over-sampling rates: (a) $\beta\% = 0\%$, (b) $\beta\% = 100\%$, (c) $\beta\% = 1000\%$, and (d) $\beta\% = 2000\%$, where x denotes positive-class test instance while ○ denotes negative-class test instance.

TABLE II
8-FOLD CROSS VALIDATION CLASSIFICATION PERFORMANCE FOR PIMA INDIANS DIABETES DATA SET

Method	TP%	FP%	Pr	G-mean	F-meas
KRLS with all data as centers	0.56 ±0.05	0.14 ±0.04	0.68 ±0.07	0.69 ±0.03	0.61 ±0.04
1-NN	0.54 ±0.04	0.21 ±0.04	0.58 ±0.06	0.65 ±0.02	0.56 ±0.04
3-NN	0.58 ±0.06	0.17 ±0.06	0.65 ±0.07	0.69 ±0.04	0.61 ±0.04
LOO-AUC+OFS ($\rho = 1.0$)	0.58 ±0.03	0.13 ±0.05	0.70 ±0.09	0.71 ±0.03	0.63 ±0.05
LOO-AUC+OFS ($\rho = 1.5$)	0.68 ±0.06	0.20 ±0.07	0.65 ±0.08	0.73 ±0.04	0.66 ±0.05
LOO-AUC+OFS ($\rho = 2.0$)	0.73 ±0.05	0.24 ±0.07	0.62 ±0.07	0.74 ±0.04	0.67 ±0.05
LOO-AUC+OFS ($\rho = 2.5$)	0.77 ±0.05	0.31 ±0.06	0.57 ±0.05	0.73 ±0.03	0.66 ±0.07
SMOTE+PSO-OFS ($\beta\% = 0\%$)	0.57 ±0.04	**0.11** ±0.04	**0.73** ±0.10	0.71 ±0.03	0.64 ±0.06
SMOTE+PSO-OFS ($\beta\% = 50\%$)	0.70 ±0.07	0.19 ±0.09	0.67 ±0.07	0.75 ±0.03	0.68 ±0.04
SMOTE+PSO-OFS ($\beta\% = 75\%$)	0.73 ±0.12	0.23 ±0.19	0.68 ±0.14	0.73 ±0.06	0.69 ±0.04
SMOTE+PSO-OFS ($\beta\% = 100\%$)	0.79 ±0.07	0.25 ±0.10	0.64 ±0.06	**0.76** ±0.05	**0.70** ±0.04
SMOTE+PSO-OFS ($\beta\% = 150\%$)	0.81 ±0.07	0.29 ±0.09	0.60 ±0.06	**0.76** ±0.04	0.69 ±0.05
SMOTE+PSO-OFS ($\beta\% = 200\%$)	0.83 ±0.04	0.33 ±0.07	0.58 ±0.06	0.75 ±0.04	0.68 ±0.05
SMOTE+PSO-OFS ($\beta\% = 250\%$)	0.85 ±0.07	0.35 ±0.07	0.57 ±0.07	0.74 ±0.06	0.68 ±0.06
SMOTE+PSO-OFS ($\beta\% = 500\%$)	**0.91** ±0.05	0.44 ±0.06	0.52 ±0.05	0.71 ±0.04	0.67 ±0.05

mean and F-measure. The best tradeoff between TP% and FP% occurred around $\beta\% = 100\%$ to $= 150\%$, which enabled to detect as many positive diabetes patients as possible while ensuring the minimum incorrect diagnose of healthy people. Again, this best over-sampling rate made the enlarged data set fully balanced. The results of Table II also show that the test performance of the proposed SMOTE+PSO-OFS compare favourably with the other classifiers.

Haberman survival data set [50]: The data set contained 306 instances from the two classes with 225 negative instances and 81 positive instances. The feature space dimension was $m = 3$. All the three input features were normalized to the range $[0,\ 1]$ using the operation (19). The 5-nearest neighbor method was adopted to generate synthetic training data in the SMOTE. The over-sampling rate $\beta\%$ was set to 0%, 100%, 200%, 300% and 400%, respectively. The swarm size and the number of movements were chosen to be $S = 10$ and $L = 20$. The 3-fold cross validation was used to calculate test performance, and the results obtained for the various classifiers are shown in Table III. Compared with the other benchmark classifiers, the SMOTE+PSO-OFS demonstrated its competitive performance. For the SMOTE+PSO-OFS, the best tradeoff between TP% and FP% occurred around $\beta\% = 150\%$, which was again close to the imbalanced degree of the original data set.

ADI data set: The austempered ductile iron (ADI) material data set was obtained from a study on fatigue cracks from the graphite nodules within the microstructure in an automotive camshaft application [51]. This two-class data set contained

TABLE III
3-FOLD CROSS VALIDATION CLASSIFICATION PERFORMANCE FOR HABERMAN SURVIVAL DATA SET

Method	TP%	FP%	Pr	G-mean	F-meas
KRLS with all data as centers	0.33 ±0.05	0.11 ±0.01	**0.63** ±0.07	0.54 ±0.04	0.41 ±0.05
1-NN	0.32 ±0.03	0.21 ±0.02	0.36 ±0.01	0.50 ±0.02	0.38 ±0.02
3-NN	0.17 ±0.06	0.15 ±0.06	0.30 ±0.07	0.38 ±0.04	0.22 ±0.04
LOO-AUC+OFS ($\rho = 1$)	0.21 ±0.02	**0.05** ±0.01	0.61 ±0.05	0.45 ±0.02	0.31 ±0.03
LOO-AUC+OFS ($\rho = 2$)	0.38 ±0.08	0.13 ±0.02	0.51 ±0.02	0.57 ±0.05	0.44 ±0.06
LOO-AUC+OFS ($\rho = 3$)	0.62 ±0.08	0.27 ±0.03	0.45 ±0.05	0.67 ±0.05	0.52 ±0.06
LOO-AUC+OFS ($\rho = 4$)	0.67 ±0.02	0.42 ±0.08	0.36 ±0.03	0.62 ±0.03	0.47 ±0.02
SMOTE+PSO-OFS ($\beta\% = 0\%$)	0.23 ±0.04	0.07 ±0.06	0.57 ±0.01	0.44 ±0.05	0.31 ±0.05
SMOTE+PSO-OFS ($\beta\% = 100\%$)	0.44 ±0.09	0.15 ±0.06	0.52 ±0.09	0.61 ±0.07	0.48 ±0.09
SMOTE+PSO-OFS ($\beta\% = 200\%$)	0.63 ±0.06	0.23 ±0.06	0.50 ±0.07	**0.69** ±0.08	**0.55** ±0.09
SMOTE+PSO-OFS ($\beta\% = 300\%$)	0.80 ±0.09	0.58 ±0.07	0.34 ±0.05	0.57 ±0.09	0.47 ±0.05
SMOTE+PSO-OFS ($\beta\% = 400\%$)	**0.84** ±0.08	0.69 ±0.08	0.31 ±0.04	0.51 ±0.08	0.45 ±0.05

2923 instances in the feature space of dimension $m = 9$, with 2807 negative instances and 116 positive instances. As in [31][51], 700 negative-class instances and 90 positive-class instances were randomly selected from the original data set to form the 8-fold cross validation set. Initially, all the nine input features were normalized to within the range $[0,\ 1]$ using the operation (19). The SMOTE adopted the 5-nearest neighbor scheme to generate synthetic training data. The over-sampling rate $\beta\%$ was set to 0%, 100%, 300%, 500%, 800%, 1000%, 1500% and 2000%, respectively. The swarm size and the number of movements were set to $S = 10$ and $L = 20$ for the PSO. The 8-fold cross validation results obtained by the various classifiers are shown in Table IV. For the SMOTE+PSO-OFS, the best overall test performance was achieved at the over sampling rate of $\beta\% = 1500\%$, which is competitive to the performance of the CS-SVM, the SUPANOVA and the best LOO-AUC+OFS ($\rho = 15$).

VI. CONCLUSIONS

The RBF classifier performs well on balanced or slightly imbalanced data sets, and our previous work has provided an efficient and tunable RBF classifier optimized by the PSO. For highly imbalanced data sets, however, the performance of the tunable RBF classifier may no longer be satisfactory. In order to combat challenging imbalanced classification problems, many approaches have been proposed, which aim to reduce the influence from the underlying imbalanced distribution. In particular, the SMOTE is effective to increase the significance of the positive class in the decision region. In this contribution, we have proposed a powerful and efficient algorithm for solving two-class imbalanced problems, referred to as the SMOTE+PSO-RBF, by combining the SMOTE and the PSO optimized RBF classifier. The experimental results presented in this study have demonstrated that the proposed SMOTE+PSO-RBF offers a very competitive solution to other existing state-of-the-arts methods for combating imbalanced classification problems.

TABLE IV
8-FOLD CROSS VALIDATION CLASSIFICATION PERFORMANCE FOR ADI DATA SET

Method	TP%	FP%	Pr	G-mean	F-meas
SVM (C+=1000,C-=1000)	0.34	0.10	0.30	0.55 ±0.03	0.32
CS-SVM (C+=1,C-=0.1)	0.72	0.23	0.29	0.74 ±0.02	0.42
SUPANOVA (C+=1,C-=0.1)	0.80	0.53	0.18	0.64 ±0.03	0.29
LOO-AUC+OFS ($\rho = 1$)	0.21 ±0.03	**0.01** ±0.01	0.67 ±0.08	0.46 ±0.03	0.32 ±0.04
LOO-AUC+OFS ($\rho = 5$)	0.55 ±0.09	0.14 ±0.02	0.33 ±0.02	0.68 ±0.05	0.41 ±0.04
LOO-AUC+OFS ($\rho = 10$)	0.71 ±0.05	0.22 ±0.03	0.30 ±0.01	0.74 ±0.02	0.42 ±0.01
LOO-AUC+OFS ($\rho = 15$)	0.83 ±0.02	0.29 ±0.02	0.27 ±0.01	**0.76** ±0.01	0.40 ±0.02
LOO-AUC+OFS ($\rho = 20$)	0.88 ±0.03	0.36 ±0.04	0.24 ±0.02	0.75 ±0.02	0.37 ±0.02
SMOTE+PSO-OFS ($\beta\% = 0\%$)	0.20 ±0.04	**0.01** ±0.01	**0.70** ±0.09	0.44 ±0.04	0.30 ±0.03
SMOTE+PSO-OFS ($\beta\% = 100\%$)	0.30 ±0.07	0.04 ±0.02	0.53 ±0.09	0.55 ±0.05	0.39 ±0.03
SMOTE+PSO-OFS ($\beta\% = 300\%$)	0.51 ±0.07	0.11 ±0.03	0.38 ±0.04	0.67 ±0.03	**0.43** ±0.02
SMOTE+PSO-OFS ($\beta\% = 500\%$)	0.72 ±0.09	0.23 ±0.06	0.29 ±0.03	0.74 ±0.02	0.41 ±0.03
SMOTE+PSO-OFS ($\beta\% = 800\%$)	0.77 ±0.07	0.28 ±0.08	0.27 ±0.03	0.74 ±0.02	0.40 ±0.03
SMOTE+PSO-OFS ($\beta\% b = 1000\%$)	0.82 ±0.04	0.29 ±0.04	0.27 ±0.02	**0.76** ±0.01	0.41 ±0.01
SMOTE+PSO-OFS ($\beta\% = 1500\%$)	**0.89** ±0.04	0.35 ±0.04	0.25 ±0.02	**0.76** ±0.02	0.39 ±0.02
SMOTE+PSO-OFS ($\beta\% = 2000\%$)	0.88 ±0.02	0.35 ±0.03	0.24 ±0.02	0.75 ±0.02	0.38 ±0.02

REFERENCES

[1] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligence Data Analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[2] G. M. Weiss, "Mining with rarity: A unifying framework," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 7–19, 2004.

[3] N. Japkowicz, "Concept-learning in the presence of between-class and within-class imbalances," in *Advances in Artificial Intelligence*, E. Stroulia and S. Matwin, Eds., vol. 2056, pp. 67–77. Springer-Verlag: Berlin, 2001.

[4] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, and A. Geissbuhler, "Learning from imbalanced data in surveillance of nosocomial infection," *Artificial Intelligence in Medicine*, vol. 37, pp. 7–18, 2006.

[5] R. B. Rao, S. Krishnan, and R. S. Niculescu, "Data mining for improved cardiac care," *ACM SIGKDD Explorations Newsletter*, vol. 8, no. 1, pp. 3–10, 2006.

[6] C. Yu, L. Chou, and D. Chang, "Predicting protein-protein interactions in unbalanced data using the primary structure of proteins," *BMC Bioinformatics*, vol. 11, no. 1, pp. 167–177, 2010.

[7] F. Provost, T. Fawcett, and R. Kohavi, "The case against accuracy estimation for comparing induction algorithms," in *Proc. 15th Int. Conf. Machine Learning* (Madison, USA), July 24-27, 1998, pp. 445–453.

[8] T. Fawcett and F. Provost, "Adaptive fraud detection," *Data Mining and Knowledge Discovery*, vol. 1, pp. 291–316, 1997.

[9] D. A. Cieslak, N. V. Chawla, and A. Striegel, "Combating imbalance in network intrusion datasets," in *Proc. 2006 IEEE Int. Conf. Granular Computing* (Atlanta, USA), May 10-12, 2006, pp. 732–737.

[10] G. M. Weiss and H. Hirsh, "Learning to predict rare events in event squences," in *Proc. 4th Int. Conf. Knowledge Discovery and Data Mining* (New York, USA), Aug. 27-31, 1998, pp. 359–363.

[11] F. Provost, "Machine learning from imbalanced data sets 101," *AAAI Workshop on Learning from Imbalanced Data Sets*, 2000.

[12] H. He and A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[13] R. Barandela, J. S. Sánchez, V. García, and E. Rangel, "Strategies for learning in class imbalance problems," *Pattern Recognition*, vol. 36, pp. 849–851, 2003.

[14] V. García, J. S. Sánchez, R. A. Mollineda, R. Alejo, and J. M. Sotoca, "The class imbalance problem in pattern classification and learning," in *II Congreso Español de Informática*, 2007, pp. 283–291.

[15] N. V. Chawla, D. A. Cieslak, L. O. Hall, and A. Joshi, "Automatically countering imbalance and its empirical relationship to cost," *Data Min. Knowl. Discov.*, vol. 17, no. 2, pp. 225–252, 2008.

[16] G. M. Weiss, K. McCarthy, and B. Zabar, "Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs?," in *Proc. 2007 Int. Conf. Data Mining*, 2007, pp. 35–41.

[17] G. M. Weiss and F. Provost, "Learning when training data are costly: The effect of class distribution on tree induction," *Artificial Intelligence Research*, vol. 19, pp. 315–354, 2003.

[18] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *J. Chemical Information and Modeling*, vol. 20, no. 1, pp. 18–36, 2004.

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

[20] C. Drummond and R. C. Holte, "C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling," in *2003 Int. Conf. Machine Learning – Workshop Learning from Imbalanced Datasets II* (Washington DC, USA), Aug. 21, 2003, pp. 1–8.

[21] N. V. Chawla, K. W. Bowyer, and L. O. Hall, "Smote: Synthetic minority over-sampling technique," *J. Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[22] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. 17th Int. Joint Conf. Artificial Intelligence* (Seattle, USA), Aug. 4-10, 2001, pp. 973–978.

[23] K. M. Ting, "An instance-weighting method to induce cost-sensitive trees," *IEEE Trans. Knowledge and Data Engineering*, vol. 14, no. 3, pp. 659–665, 2002.

[24] M. A. Maloof, "Learning when data sets are imbalanced and when costs are unequal and unknown," in *2003 Int. Conf. Machine Learning – Workshop Learning from Imbalanced Datasets II* (Washington DC, USA), Aug. 21, 2003, pp. 1–8.

[25] K. McCarthy, B. Zabar, and G. Weiss, "Does cost-sensitive learning beat sampling for classifying rare classes?" in *Proc. 1st Int. Workshop Utility-Based Data Mining* (Chicago, USA), Aug. 21, 2005, pp. 69–77.

[26] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Proc. 15th European Conf. Machine Learning* (Pisa, Italy), Sept. 20-24, 2004, pp. 39–50.

[27] P. Kang and S. Cho, "Eus svms: Ensemble of under-sampled svms for data imbalance problems," in *Proc. 13th Int. Conf. Neural Information Processing* (Hong Kong, China), Oct. 3-6, 2006, pp. 837–846.

[28] B. X. Wang and N. Japkowicz, "Boosting support vector machines for imbalanced data sets," in *Proc. 17th Int. Conf. Foundations of Intelligent Systems* (Toronto, Canada), May 20-23, 2008, pp. 38–47.

[29] Y. Tang, Y.-Q. Zhang, Z. Huang, and X. Hu, "Granular svm-rfe gene selection algorithm for reliable prostate cancer classification on microarray expression data," in *Proc. 5th IEEE Symp. Bioinformatics and Bioengineering* (Minneapolis, USA), Oct. 19-21, 2005, pp. 290–293.

[30] Y. Tang and Y.-Q. Zhang, "Granular svm with repetitive undersampling for highly imbalanced protein homology prediction," in *Proc. 2006 IEEE Int. Conf. Granular Computing* (Atlanta, USA), May 10-12, 2006, pp. 457–460.

[31] X. Hong, S. Chen, and C. J. Harris, "A kernel-based two-class classifier for imbalanced data sets," *IEEE Trans. Neural Networks*, vol. 18, no. 1, pp. 28–41, 2007.

[32] X. Hong, S. Chen, and C. J. Harris, "A fast linear-in-the-parameters classifier construction algorithm using orthogonal forward selection to minimize leave-one-out misclassification rate," *Int. J. Systems Science*, vol. 39, no. 2, pp. 119–25, 2008.

[33] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.

[34] X. Hong, P. M. Sharkey, and K. Warwick, "Automatic nonlinear predictive model construction using forward regression and the press statistic," *IEE Proc. Control Theory & Appl.*, vol. 150, no. 3, pp. 245–254, 2003.

[35] X. Hong, P. M. Sharkey, and K. Warwick, "A robust nonlinear identification algorithm using press statistic and forward regression," *IEEE Trans. Neural Networks*, vol. 14, no. 2, pp. 454–458, 2003.

[36] S. Chen, X. Hong, C. J. Harris, and P. M. Sharkey, "Sparse modelling using orthogonal forward regression with PRESS statistic and regularization," *IEEE Trans. Systems, Man and Cybernetics, Part B*, vol. 34, no. 2, pp. 898–911, 2004.

[37] S. Chen, X. X. Wang, X. Hong and C. J. Harris, "Kernel classifier construction using orthogonal forward selection and boosting with Fisher ratio class separability measure," *IEEE Trans. Neural Networks*, vol. 17, no. 6, pp. 1652–1656, 2006.

[38] S. Chen, X. Hong and C. J. Harris, "Construction of RBF classifiers with tunable units using orthogonal forward selection based on leave-one-out misclassification rate," in *Proc. 2006 Int. Joint Conf. Neural Networks* (Vancouver, Canada), July 16-21, 2006, pp. 6390–6394.

[39] S. Chen, X. Hong, B. L. Luk, and C. J. Harris, "Construction of tunable radial basis function networks using orthogonal forward selection," *IEEE Trans. Systems, Man, and Cybernetics, Part B*, vol. 39, no. 2, pp. 457–466, 2009.

[40] S. Chen, X. X. Wang, and C. J. Harris, "Experiments with repeating weighted boosting search for optimization in signal processing applications," *IEEE Trans. Systems, Man and Cybernetics, Part B*, vol. 35, no. 4, pp. 682–693, 2005.

[41] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. Morgan Kaufmann, 2001.

[42] S. Chen, X. Hong, and C. J. Harris, "Radial basis function classifier construction using particle swarm optimisation aided orthogonal forward regression," in *Proc. 2010 Int. Joint Conf. Neural Networks* (Barcelona, Spain), July 18-23, 2010, pp. 3418–3423.

[43] S. Chen, X. Hong, and C. J. Harris, "Particle swarm optimization aided orthogonal forward regression for unified data modelling," *IEEE Trans. Evolutionary Computation*, vol. 14, no. 4, pp. 477–499, 2010.

[44] A. Ratnaweera and S. K. Halgamuge, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

[45] W.-F. Leong and G. G. Yen, "PSO-based multiobjective optimization with dynamic population size and adaptive local archives," *IEEE Trans. Systems, Man, and Cybernetics, Part B*, vol. 38, no. 5, pp. 1270–1293, 2008.

[46] S. Chen, X. Hong, B. L. Luk, and C. J. Harris, "Non-linear system identification using particle swarm optimisation tuned radial basis function models," *Int. J. Bio-Inspired Computation*, vol. 1, no. 4, pp. 246–258, 2009.

[47] M. Ramezani, M.-R. Haghifam, C. Singh, H. Seifi, and M. P. Moghaddam, "Determination of capacity benefit margin in multiarea power systems using particle swarm optimization," *IEEE Trans. Power Systems*, vol. 24, no. 2, pp. 631–641, 2009.

[48] S. Chen, W. Yao, H. R. Palally, and L. Hanzo, "Particle swarm optimisation aided MIMO transceiver designs," in: Y. Tenne and C.-K. Goh, Eds., *Computational Intelligence in Expensive Optimization Problems*. Springer-Verlag: Berlin, 2010, pp. 487–511.

[49] R. H. Myers, *Classical and Modern Regression with Applications* (2nd Edition). PWS-KENT: Boston, 1990.

[50] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," Department of Computer Science, University of California, Irvine, CA, 1998.
`http://archive.ics.uci.edu/ml/datasets.html`

[51] K. K. Lee, C. J. Harris, S. R. Gunn, and P. A. S. Reed, "Classification of imbalanced data with transparent kernel," in *Proc. 2001 Int. Joint Conf. Neural Networks* (Washington DC, USA), July 15-19, 2001, pp. 2410–2415.

[52] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, pp. 1145–1159, 1997.

[53] C. van Rijsbergen, *Information Retrieval*, Butterworths: London, U.K., 1979.