A Wasserstein Distance-Based Cost-Sensitive Framework for Imbalanced Data Classification

Rui FENG, Hongbing JI, Zhigang ZHU, Lei WANG*

School of Electronic Engineering, Xidian University, Xi'an, China

ruif@stu.xidian.edu.cn, hbji@xidian.edu.cn, zgzhu@xidian.edu.cn, leiwang@mail.xidian.edu.cn

Submitted April 21, 2023 / Accepted July 22, 2023 / Online first August 7, 2023

Abstract. *Class imbalance is a prevalent problem in many real-world applications, and imbalanced data distribution can dramatically skew the performance of classifiers. In general, the higher the imbalance ratio of a dataset, the more difficult it is to classify. However, it is found that standard classifiers can still achieve good classification results on some highly imbalanced datasets. Obviously, the class imbalance is only a superficial characteristic of the data, and the underlying structural information is often the key factor affecting the classification performance. As implicit prior knowledge, structural information has been validated to be crucial for designing a good classifier. This paper proposes a Wasserstein-based cost-sensitive support vector machine (CS-WSVM) for class imbalance learning, incorporating prior structural information and a cost-sensitive strategy. The Wasserstein distance is introduced to model the distribution of majority and minority samples to capture the structural information, which is employed to weight the majority and minority samples. Comprehensive experiments on synthetic and real-world datasets, especially on the radar emitter signal dataset, demonstrated that CS-WSVM can achieve outstanding performance in imbalanced scenarios.*

Keywords

Imbalanced classification, cost-sensitive, structural information, Wasserstein distance, radar emitter signal

1. Introduction

In the field of machine learning, most of the traditional classification algorithms are based on the premise that the number of samples in considered classes is roughly similar [1]. During the learning process, these methods train classifiers by maximizing classification accuracy and treating all samples equally. However, in most real-world classification problems, the data distribution is skewed, i.e., some classes have more samples than others. For example, in the IFF (Identification Friend or Foe) problem [2], the number of friend targets (majority classes) is more than that of enemy targets (minority classes). There are more legitimate credit card users than fraudulent credit card users in the case of credit card fraud detection [3]. The same situation also occurs in many practical applications, such as emitter identification [4], network intrusion detection [5], and industrial fault detection [6]. In the imbalanced case, traditional classifiers usually fail to achieve good performance, especially for the minority class, as they are designed to generate simple assumptions based on overall accuracy. Such a model is not practical in real life because what is more critical to experts is often the prediction accuracy of the interested class (i.e., the minority class).

In recent years, many approaches have been developed to handle the class imbalance problem, which can be grouped into two categories: data-level methods and algorithm-level methods.

Data-level methods attempt to balance the data distribution by adopting sampling techniques [1]. Studies have shown that a balanced dataset is more conducive to enhancing the global classification performance compared to an imbalanced dataset [7–9]. The various re-sampling methods include removing samples from the majority class (under-sampling), generating new samples for the minority class (over-sampling), and the integration of two techniques [48], [49]. Although these approaches do help to adjust the ratio of minority and majority samples and improve the classification accuracy of the whole data, there are still obvious drawbacks: the under-sampling methods often lose valuable information while over-sampling methods are prone to generate redundant data and result in model overfitting.

Algorithm-level methods solve the class imbalance problem by designing a model suitable for the imbalanced data. Research efforts in this area mainly include cost-sensitive learning [10], [11], ensemble learning [12], [13], and other improved algorithms such as improved versions of decision trees [14], k-nearest neighbor [15], [16], and support vector machine [13], [17]. Table 1 provides an overview of methods used to deal with imbalanced datasets. Engaging readers may refer to [18–25] for a comprehensive survey of the methods for imbalanced data classification.

DOI: 10.13164/re.2023.0451

Data-level methods	Algorithm-level methods
Under-sampling	Cost-sensitive learning
Over-sampling	Ensemble learning
Hybrid-sampling	Other improved algorithms

Tab. 1. Overview of the categories of methods to deal with imbalanced datasets.

In general, the higher the imbalance ratio, the more difficult the classification task. However, in some applications, it is found that the standard learning model can still achieve a good classification result on some highly unbalanced datasets. The class imbalance is not the only factor that weakens classification performance; other factors, such as noisy samples, class overlap, and the dataset's structural information, can also significantly affect classification performance. As an implicit prior knowledge, structure information has been proven to be crucial for designing a good classification model. Therefore, the classifier needs to pour more attention to the underlying data structural information, i.e., the data distribution information.

The cost-sensitive method, as one of the mainstream methods to solve the imbalance classification problem, has been widely used for its flexibility in integrating with many standard classifiers [11, 13, 26–30]. Specifically, it assigns a larger cost to minority class samples and a lower cost to majority class samples, hoping to reduce the impact of the data imbalance. However, most existing cost-sensitive methods simply assign different weights to different classes of samples based on the number of positive and negative class samples, without considering the structural information of the data. In this paper, we propose a structured cost-sensitive framework to handle the imbalanced data classification problem, namely the Wasserstein distance-based cost-sensitive support vector machine (CS-WSVM). Instead of directly weighting majority and minority samples according to the imbalance ratio, a new distance is introduced to model the distribution of majority and minority classes, and weighting the majority and minority samples based on the distribution information, thus promoting SVM in a cost-sensitive framework. Additionally, distribution information is also introduced into the standard SVM object function in the form of regular terms. The mainstream and benefits of the proposed framework are summarized as follows:

(i) We proposed a new strategy to capture the underlying data structural information and thus guide the design of the classifier.

(ii) We constructed a series of Wasserstein distance-guided data clusters, so that the original imbalanced data became balanced at the clustering level.

(iii) According to the Wasserstein distance between different clusters, we defined *well-classified*, *hard-classified,* and *regular samples*.

The rest of the paper is organized as follows. In Sec. 2, the standard SVM is briefly described. Section 3 presents the proposed CS-WSVM, including the linear and nonlinear cases. The experimental results on toy and real-world problems are shown in Sec. 4. Finally, some conclusions are given in Sec. 5.

2. Support Vector Machine

As one of the most popular classification methods, SVM and its variants [31–35] have been widely used in image classification, face recognition, voice recognition, and other applications. The basic idea of SVM is to find a hyperplane that can separate the two-class data points with a maximal margin, which is based on the minimization of the hinge loss function that assigns the same penalty parameters to all samples.

Given the training samples set $\mathbf{X} = \{(\mathbf{x}_1,y_1), (\mathbf{x}_2,y_2),\ldots, (\mathbf{x}_n,y_n)\}$, $y_i \in \{-1,+1\}$, the standard SVM is to find a hyperplane $f = \boldsymbol{\omega}^{\mathrm{T}}\mathbf{X} + b$, which separates the samples of different classes with a margin of $2/\|\boldsymbol{\omega}\|$. The object function can be formulated as

$$\begin{aligned} &\min \frac{1}{2}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega} \\ &\text{s.t.} \quad y_i(\boldsymbol{\omega}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1, i = 1,2,\cdots,n. \end{aligned} \tag{1}$$

For the linear non-separable cases, the soft-margin SVM is posed,

$$\begin{aligned} &\min \frac{1}{2}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega} + C\sum_{i=1}^{n}\xi_i \\ &\text{s.t.} \quad y_i(\boldsymbol{\omega}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1 - \xi_i\,, \\ &\qquad \xi_i \geq 0, i = 1,2,\cdots,n \end{aligned} \tag{2}$$

where ξ_i is the slack variable that measures the degree of misclassification of the data $\mathbf{x}_i$, $\sum_{i=1}^{n}\xi_i$ is an upper bound on the number of training errors. C is a trading-off parameter, a large C corresponding to assigning a higher penalty to errors ($C = \infty$ leads to a hard-margin SVM).

Consequently, following the above analysis, it is clear that SVM can obtain an optimal hyperplane in the balanced classification problem. However, for the imbalanced case, due to the lack of minority class samples, the information provided to the classifier by the minority class samples is very limited. In order to obtain a high-accuracy result, SVM must ensure the majority class samples are correctly classified as much as possible during the training process, so that the hyperplane is shifted to the minority class, which leads to an unsatisfied classification result. Consequently, various approaches that modify SVM to achieve cost-sensitive have been proposed [36–39], such as BP-SVM (Biased Penalties Support Vector Machine) [39] and CS-SVM (Cost-Sensitive Support Vector Machine) [37]. The former introduces different penalty parameters C_+ and C_- for the positive and negative samples during training, and the latter extends the SVM hinge loss to optimize the classifier concerning class imbalance or class cost. CS4VM (Cost-Sensitive Semi-Supervised Support Vector Machine) [38] and Cos-LapSVM [40] (Cost-Sensitive Laplacian

Support Vector Machine) have been proposed to deal with the semi-supervised imbalanced classification problem.

3. Wasserstein Distance-Based Cost-Sensitive Support Vector Machine

In this section, we introduced a novel classifier named Wasserstein distance-based cost-sensitive support vector machine (CS-WSVM). Different from the traditional method of weighting positive and negative samples according to the imbalance ratio, a different strategy is adopted to set the cost for positive and negative samples. Specifically, the proposed CS-WSVM weights positive and negative samples according to their distribution. First, by using clustering techniques, a series of Wasserstein distance-guided data clusters were constructed. After clustering, the original imbalanced data becomes balanced at the clustering level. Moreover, the Wasserstein distance between each negative and positive cluster can be used to assign different costs to different samples. Then the optimization problem was obtained by embedding the distribution information and the different misclassification costs of each sample into the object function. Moreover, this algorithm can also be extended to a nonlinear version by a kernel trick, as in many kernel-based methods.

3.1 Construct the Wasserstein Distance-Guided Data Clusters

In this step, a cost-sensitive training set is constructed based on the Wasserstein distance. Wasserstein distance, also known as earth mover's distance, was first proposed by Rubner as a metric between two distributions [41]. It is defined as the minimal cost needed to transform one distribution into the other. The Wasserstein distance is based on solving the transportation problem through linear optimization. Rubner explains this theory through a cargo transportation example.

Suppose there are two distributions $\mathbf{P} = \{(\mathbf{p}_i,\omega_{p_i})\}^{m}{}_{i=1}$ and $\mathbf{Q} = \{(\mathbf{q}_j,\omega_{q_j})\}^{n}{}_{j=1}$, where $\mathbf{p}_i$ is the supplier, ω_{p_i} is the quantity of goods it owns, $\mathbf{q}_j$ is the warehouse, and ω_{q_j} is the quantity of goods it can receive. $\mathbf{D} = [d_{ij}]$ is the ground distance matrix where d_{ij} is the ground distance between $\mathbf{p}_i$ and $\mathbf{q}_j$. Then the Wasserstein distance can be expressed as the following linear optimization problem: we hope to find a flow $\mathbf{F} = [f_{ij}]$ that minimizes the overall transportation cost, where f_{ij} is the flow from $\mathbf{p}_i$ to $\mathbf{q}_j$.

$$
\begin{aligned}
&WORK(\mathbf{P},\mathbf{Q},\mathbf{F}) = \sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij} \\
&\text{s.t.}\quad f_{ij} \ge 0 \quad 1 \le i \le m, 1 \le j \le n \\
&\qquad \sum_{j=1}^{n} f_{ij} \le \omega_{p_i}, \quad 1 \le i \le m, \\
&\qquad \sum_{i=1}^{m} f_{ij} \le \omega_{q_j}, \quad 1 \le j \le n, \\
&\qquad \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = \min\left(\sum_{i=1}^{m}\omega_{p_i}, \sum_{i=1}^{n}\omega_{q_j}\right), \\
&\qquad d_{ij} = \left|\mathbf{p}_i - \mathbf{q}_j\right| .
\end{aligned}
\tag{3}
$$

Once the transportation problem is solved, the Wasserstein distance can be normalized as:

$$
\mathbf{W}(\mathbf{P},\mathbf{Q}) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} . \tag{4}
$$

For a typical binary imbalanced classification problem, given a training set $\mathbf{X} = \{(\mathbf{x}_i,y_i), i = 1,2,\ldots,N\}$, where $\mathbf{x}_i$ is the i-th sample in the training set and y_i is the label corresponding to $\mathbf{x}_i$. Define all negative samples in $\mathbf{X}$ as majority class, denoted as $\mathbf{N} = \{(\mathbf{x}_i,y_i|y_i = -1, i = 1,2,\ldots,N)\}$; define all positive samples in $\mathbf{X}$ as minority class, denoted as $\mathbf{P} = \{(\mathbf{x}_i,y_i|y_i = +1, i = 1,2,\ldots,N)\}$. Different strategies would be used for the minority and majority classes. Here we adopt the hierarchical clustering [43] technique to divide the majority class set $\mathbf{N}$ into several clusters $\mathbf{N}_1$, $\mathbf{N}_2$,…, $\mathbf{N}_c$. For each cluster, we calculate the Wasserstein distance $\mathbf{W}_{\mathbf{N}_i,\mathbf{P}}$ between $\mathbf{N}_i$ and the minority class set $\mathbf{P}$, as well as the Wasserstein distance $\mathbf{W}_{\mathbf{N},\mathbf{P}}$ between $\mathbf{N}$ and $\mathbf{P}$. If $\mathbf{W}_{\mathbf{N}_i,\mathbf{P}} > \mathbf{W}_{\mathbf{N},\mathbf{P}}$, we define the samples in the cluster $\mathbf{N}_i$ as well-classified samples, the other samples in majority class will be defined as regular samples. Smaller weights are assigned to well-classified samples, and normal weights are assigned to regular samples, respectively. Meanwhile, the minority class samples are defined as hard-classified samples and are assigned larger weights. Figure 1 illustrates the construction of the Wasserstein distance-guided data clusters.

Considering that each sample has different misclassification cost, the sample set can be expressed as $(\mathbf{x}_1,y_1,co_1)$, $(\mathbf{x}_2,y_2,co_2)$,…, $(\mathbf{x}_n,y_n,co_n)$, where co_i is the misclassification cost of $\mathbf{x}_i$.

3.2 CS-WSVM for Linear Case

After calculating the Wasserstein distance, we can obtain the underlying data structure information. Meanwhile, after clustering, a cost-sensitive training set will be constructed. Accordingly, the CS-WSVM model can be formulated as

$$
\begin{aligned}
&\min\ \frac{1}{2}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega} + \frac{\lambda}{2}\boldsymbol{\omega}^{\mathrm{T}}\mathbf{W}_{\mathrm{d}}\boldsymbol{\omega} + C\left(\sum_{i=1}^{n} co_i\xi_i\right) \\
&\text{s.t.}\quad y_i(\boldsymbol{\omega}^{\mathrm{T}}\mathbf{x}_i + b) \ge \Omega_i - \xi_i, \xi_i \ge 0 \\
&\qquad \Omega_i = \begin{cases} 1 & y_i = +1 \\ \dfrac{1}{co_i} & y_i = -1 \end{cases}
\end{aligned}
\tag{5}
$$

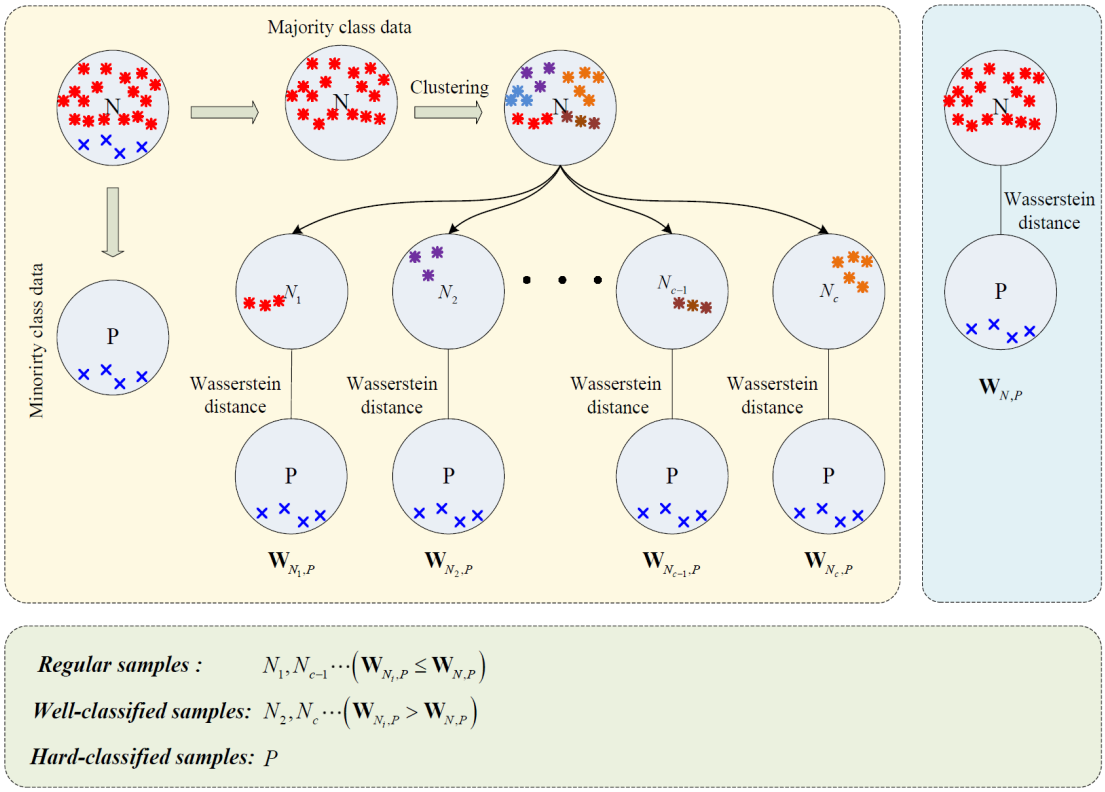


Fig. 1. Construct the Wasserstein distance-guided data clusters.

where $\mathbf{W}_{\mathrm{d}}$ is the Wasserstein distance matrix between the two kinds of distributions. λ is the parameter that regulates the relative importance of the distance information within the two distributions. co_i is the misclassification cost of $\mathbf{x}_i$. The value of co_i for different samples can be defined as

$$co_i = \begin{cases} C_- \Big/ \left[1+\log\left(p+2\right)\right], & \mathbf{x}_i \in \text{well-classified samples}, \\ C_-, & \mathbf{x}_i \in \text{regular samples}, \\ pC_-, & \mathbf{x}_i \in \text{hard-classified samples}. \end{cases} \tag{6}$$

In this case, cost-sensitivity is controlled by the parameters co_i and Ω_i. co_i means that a large weight would be assigned to the minority class sample, i.e. the hard-classified sample during training. While, the parameter Ω_i ensures that CS-WSVM does not simply over train on the minority class.

Incorporating the constraints into the object function, we can rewrite (5) as a primal Lagrangian

$$\begin{aligned} L(\boldsymbol{\omega},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\mu}) = {} & \frac{1}{2}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega} + \frac{\lambda}{2}\boldsymbol{\omega}^{\mathrm{T}}\mathbf{W}_{\mathrm{d}}\boldsymbol{\omega} + C\left(\sum_{i=1}^{n} co_i\xi_i\right) \\ & + \sum_{i=1}^{n}\alpha_i\left[\Omega_i - \xi_i - y_i\left(\boldsymbol{\omega}^{\mathrm{T}}\mathbf{x}_i + b\right)\right] - \sum_{i=1}^{n}\mu_i\xi_i \end{aligned} \tag{7}$$

where α and μ is the Lagrange multiplier. We set the partial derivative of $L(\boldsymbol{\omega},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\mu})$ with respect to $\boldsymbol{\omega}$, b and $\boldsymbol{\xi}$ equal to zero, respectively.

$$\frac{\partial L}{\partial \boldsymbol{\omega}} = \left(\mathbf{I} + \lambda\mathbf{W}_{\mathrm{d}}\right)\boldsymbol{\omega} - \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i = 0, \tag{8}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n}\alpha_i y_i = 0, \tag{9}$$

$$\frac{\partial L}{\partial \xi_i} = C\cdot co_i - \alpha_i - \mu_i = 0. \tag{10}$$

Substituting (8), (9), and (10) into (7), we obtain

$$\begin{aligned} & \frac{1}{2}\boldsymbol{\omega}^{\mathrm{T}}\left(\mathbf{I}+\lambda\mathbf{W}_{\mathrm{d}}\right)\boldsymbol{\omega} + C\left(\sum_{i=1}^{n} co_i\xi_i\right) \\ & + \sum_{i=1}^{n}\alpha_i\left[\Omega_i - \xi_i - y_i\left(\boldsymbol{\omega}^{\mathrm{T}}\mathbf{x}_i + b\right)\right] - \sum_{i=1}^{n}\mu_i\xi_i \\ & = \frac{1}{2}\sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i^{\mathrm{T}}\left(\mathbf{I}+\lambda\mathbf{W}_{\mathrm{d}}\right)^{-1}\sum_{j=1}^{n}\alpha_j y_j \mathbf{x}_j \\ & + \sum_{i=1}^{n}\alpha_i\Omega_i - \sum_{i=1}^{n}\alpha_i y_i \sum_{j=1}^{n}\alpha_j y_j \mathbf{x}_j^{\mathrm{T}}\left(\mathbf{I}+\lambda\mathbf{W}_{\mathrm{d}}\right)^{-1}\mathbf{x}_i \\ & = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^{\mathrm{T}}\left(\mathbf{I}+\lambda\mathbf{W}_{\mathrm{d}}\right)^{-1}\mathbf{x}_j + \sum_{i=1}^{n}\alpha_i\Omega_i. \end{aligned} \tag{11}$$

Then, we transform the primal into the dual problem

$$\begin{aligned}\max\quad & -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^{\mathrm{T}}(\mathbf{I}+\lambda\mathbf{W}_{\mathrm{d}})^{-1}\mathbf{x}_j+\sum_{i=1}^{n}\alpha_i\Omega_i\\ \text{s.t.}\quad & \sum_{i=1}^{n}\alpha_i y_i=0\\ & 0\le\alpha_i\le C\cdot co_i, i=1,\cdots,n\\ & \Omega_i=\begin{cases}1 & y_i=+1\\ \dfrac{1}{co_i} & y_i=-1\end{cases}\end{aligned} \tag{12}$$

By using QP technique, we can obtain the solution α_i. Then, the derived classifier function can be formulated as follows, which is used to predict the class labels for testing data $\mathbf{x}$

$$\begin{aligned}Class\ \mathbf{x}&=\operatorname{sgn}\left[\boldsymbol{\omega}^{\mathrm{T}}\mathbf{x}+b\right]\\&=\operatorname{sgn}\left[\sum_{i=1}^{n}\alpha_i y_i\mathbf{x}_i^{\mathrm{T}}\left(\mathbf{I}+\lambda\mathbf{W}_{\mathrm{d}}\right)^{-1}\mathbf{x}+b\right].\end{aligned} \tag{13}$$

3.3 CS-WSVM for Nonlinear Case

We can also apply the kernel trick [42] in CS-WSVM to improve the classification performance for real-world linearly non-separable datasets. We use a mapping Φ which can map the data to a higher (or infinite) dimensional Euclidean space $\mathcal{H}$, i.e., Φ: $\mathcal{R}^{\mathrm{d}}\mapsto\mathcal{H}$. The hyperplane in $\mathcal{H}$ can be defined as

$$f(\mathbf{X})=\boldsymbol{\omega}^{\mathrm{T}}\Phi(\mathbf{X})+b. \tag{14}$$

Similar to (5), the kernel CS-WSVM model can be formulated as

$$\begin{aligned}&\min\frac{1}{2}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega}+\frac{\lambda}{2}\boldsymbol{\omega}^{\mathrm{T}}\Phi(\mathbf{X})\mathbf{W}_{\mathrm{d}}^{\Phi}\Phi(\mathbf{X})^{\mathrm{T}}\boldsymbol{\omega}+C\left(\sum_{i=1}^{n}co_i\xi_i\right)\\ &\text{s.t.}\quad y_i\left(\boldsymbol{\omega}^{\mathrm{T}}\Phi(\mathbf{x}_i)+b\right)\ge\Omega_i-\xi_i,\ \xi_i\ge0\\ &\Omega_i=\begin{cases}1, & y_i=+1\\ \dfrac{1}{co_i}, & y_i=-1\end{cases}\end{aligned} \tag{15}$$

where $\mathbf{W}_{\mathrm{d}}^{\Phi}$ is the Wasserstein distance of the two kinds of distributions in the kernel space. $\mathbf{W}_{\mathrm{d}}^{\Phi}$ can be calculated by the following equation

$$\mathbf{W}_{\mathrm{d}}^{\Phi}\left(\Phi(\mathbf{P}),\Phi(\mathbf{Q})\right)=\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}d_{\Phi(\mathbf{p}_i)\Phi(\mathbf{q}_j)}f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n}f_{ij}} \tag{16}$$

where $d_{\Phi(\mathbf{p}_i)\Phi(\mathbf{q}_j)}$ is the Euclidean distance of feature $\Phi(\mathbf{p}_i)$ and $\Phi(\mathbf{q}_j)$ in the high-dimensional space, which can be expressed as

$$\begin{aligned}&d_{\Phi(\mathbf{p}_i)\Phi(\mathbf{q}_j)}\\&=\sqrt{\left\|\Phi(\mathbf{p}_i)-\Phi(\mathbf{q}_j)\right\|^2}\\&=\sqrt{\Phi(\mathbf{p}_i)^{\mathrm{T}}\Phi(\mathbf{p}_i)-\Phi(\mathbf{p}_i)^{\mathrm{T}}\Phi(\mathbf{q}_j)-\Phi(\mathbf{q}_j)^{\mathrm{T}}\Phi(\mathbf{p}_i)+\Phi(\mathbf{q}_j)^{\mathrm{T}}\Phi(\mathbf{q}_j)}.\end{aligned} \tag{17}$$

However, since $\mathcal{H}$ is a higher (or infinite) dimensional space, we cannot obtain the formulation of Φ explicitly. If there were a "kernel function" K such that $K(\mathbf{x}_i,\mathbf{x}_j)=\Phi(\mathbf{x}_i)^{\mathrm{T}}\Phi(\mathbf{x}_j)$, we would only need to use $\mathbf{K}_{ij}=K(\mathbf{x}_i,\mathbf{x}_j)$ in the algorithm, so Equation (17) can be rewritten as:

$$\begin{aligned}&d_{\Phi(\mathbf{p}_i)\Phi(\mathbf{q}_j)}\\&=\sqrt{\left\|\Phi(\mathbf{p}_i)-\Phi(\mathbf{q}_j)\right\|^2}\\&=\sqrt{K(\mathbf{p}_i,\mathbf{p}_i)-K(\mathbf{p}_i,\mathbf{q}_j)-K(\mathbf{q}_j,\mathbf{p}_i)+K(\mathbf{q}_j,\mathbf{q}_j)}.\end{aligned} \tag{18}$$

The Lagrangian form of (15) can be written as

$$\begin{aligned}&L(\boldsymbol{\omega},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\mu})\\&=\frac{1}{2}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega}+\frac{\lambda}{2}\boldsymbol{\omega}^{\mathrm{T}}\Phi(\mathbf{X})\mathbf{W}_{\mathrm{d}}^{\Phi}\Phi(\mathbf{X})^{\mathrm{T}}\boldsymbol{\omega}+C\left(\sum_{i=1}^{n}co_i\xi_i\right)\\&\quad+\sum_{i=1}^{n}\alpha_i\left[\Omega_i-\xi_i-y_i(\boldsymbol{\omega}^{\mathrm{T}}\Phi(\mathbf{x}_i)+b)\right]-\sum_{i=1}^{n}\mu_i\xi_i.\end{aligned} \tag{19}$$

The dual problem is

$$\begin{aligned}&\max\\&-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\Phi(\mathbf{x}_i)^{\mathrm{T}}\left(\mathbf{I}+\lambda\Phi(\mathbf{X})\mathbf{W}_{\mathrm{d}}^{\Phi}\Phi(\mathbf{X})^{\mathrm{T}}\right)^{-1}\Phi(\mathbf{x}_j)\\&+\sum_{i=1}^{n}\alpha_i\Omega_i\\&\text{s.t.}\quad 0\le\alpha_i\le C\cdot co_i, i=1,\cdots,n\\&\sum_{i=1}^{n}\alpha_i y_i=0\\&\Omega_i=\begin{cases}1 & y_i=+1\\ \dfrac{1}{co_i} & y_i=-1\end{cases}\end{aligned} \tag{20}$$

According to the Woodbury's formula [47]

$$\left(\mathbf{I}+\mathbf{UBV}\right)^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{UB}\left(\mathbf{B}+\mathbf{BVA}^{-1}\mathbf{UB}\right)^{-1}\mathbf{BVA}^{-1}, \tag{21}$$

we have

$$\begin{aligned}&\left[\mathbf{I}+\lambda\Phi(\mathbf{X})\mathbf{W}_{\mathrm{d}}^{\Phi}\Phi(\mathbf{X})^{\mathrm{T}}\right]^{-1}\\&=\mathbf{I}-\lambda\Phi(\mathbf{X})\mathbf{W}_{\mathrm{d}}^{\Phi}\left[\mathbf{W}_{\mathrm{d}}^{\Phi}+\lambda\mathbf{W}_{\mathrm{d}}^{\Phi}\Phi(\mathbf{X})^{\mathrm{T}}\Phi(\mathbf{X})\mathbf{W}_{\mathrm{d}}^{\Phi}\right]\\&=\mathbf{I}-\lambda\Phi(\mathbf{X})\mathbf{W}_{\mathrm{d}}^{\Phi}\left[\mathbf{W}_{\mathrm{d}}^{\Phi}+\lambda\mathbf{W}_{\mathrm{d}}^{\Phi}\mathbf{K}\mathbf{W}_{\mathrm{d}}^{\Phi}\right]^{-1}\mathbf{W}_{\mathrm{d}}^{\Phi}\Phi(\mathbf{X})^{\mathrm{T}}\\&=\mathbf{I}-\lambda\Phi(\mathbf{X})\mathbf{P}\Phi(\mathbf{X})^{\mathrm{T}}\end{aligned} \tag{22}$$

where $\mathbf{P} = \mathbf{W}_{\mathrm{d}}^{\Phi}[\mathbf{W}_{\mathrm{d}}^{\Phi} + \lambda \mathbf{W}_{\mathrm{d}}^{\Phi}\mathbf{K}\ \mathbf{W}_{\mathrm{d}}^{\Phi}]\ \mathbf{W}_{\mathrm{d}}^{\Phi}$. Let $\mathbf{K}_{i:}$ denote the i-th row of $\mathbf{K}$, $\mathbf{K}_{:j}$ denote the j-th column of $\mathbf{K}$, then the dual problem can be cast as

$$\begin{aligned} \max\ & -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \left[\mathbf{K}_{ij} - \lambda \mathbf{K}_{i:}\mathbf{P}\mathbf{K}_{:j}\right] + \sum_{i=1}^{n}\alpha_i\Omega_i \\ \text{s.t.}\ & 0 \le \alpha_i \le C \cdot co_i, i = 1, \cdots, n \\ & \sum_{i=1}^{n}\alpha_i y_i = 0 \end{aligned} \tag{23}$$

$$\Omega_i = \begin{cases} 1 & y_i = +1 \\ \dfrac{1}{co_i} & y_i = -1 \end{cases}$$

Once the solution α are obtained from the above convex optimization problem, we can get the hyperplane. The label of a new data point $\mathbf{x} \in R^n$ is determined as

$$\begin{aligned} & Class\ \mathbf{x} \\ & = \mathrm{sgn}\left[\boldsymbol{\omega}^{\mathrm{T}}\Phi(\mathbf{x}) + b\right] \\ & = \mathrm{sgn}\left[\sum_{i=1}^{n}\alpha_i y_i \mathbf{K}(\mathbf{x}_i, \mathbf{x}) - \lambda\sum_{i=1}^{n}\alpha_i y_i \mathbf{K}_{i:}\mathbf{P}\mathbf{K}(\mathbf{X}, \mathbf{x}) + b\right]. \end{aligned} \tag{24}$$

4. Experiments

In this section, we conducted a series of experiments on both synthetic and real-world datasets. We compared the performance of the proposed CS-WSVM with standard SVM and some representative cost-sensitive methods, including BP-SVM and CS-SVM, as well as the well-known resampling method SVM+SMOTE (Synthetic Minority Over-sampling Technique) [44] and SVM+ADASYN (Adaptive Synthetic) [48]. Many imbalanced datasets have been published in the field of imbalance classification. We first evaluated the effectiveness of the proposed CS-WSVM on synthetic datasets as a toy example to illustrate the impact of introducing data distribution information and cost-sensitive terms on classification. Then we evaluated the performance of these methods on benchmark UCI machine learning datasets [46] and KEEL datasets [45], respectively. Finally, we apply CS-WSVM to the radar emitter identification task, a typical IFF problem. All the experiments were carried out on a PC with a 3.50 GHz CPU and 48 GB RAM.

4.1 Evaluation Matrix

Traditionally, we use the accuracy and error rate to evaluate classifier performance. For a basic binary classification problem, the confusion matrix is shown in Tab. 2. We define the minority class as the positive class and the majority class as the negative class.

	Predicted positive	**Predicted negative**
Actual positive	True positive (TP)	False negative (FN)
Actual negative	False positive (FP)	True negative (TN)

Tab. 2. Confusion matrix of the binary classification problems.

For imbalance classification problems, it is not reasonable to evaluate the performance of a classifier only by using accuracy, so we adopt *TNR, TPR, FPR, G-Mean,* ROC curve, and AUC to measure the effectiveness of a classifier. These metrics of *TNR, TPR, FPR,* and *G-Mean* are defined as:

$$TNR = \frac{TN}{TN + FP}, \tag{25}$$

$$TPR = \frac{TP}{TP + FN}, \tag{26}$$

$$FPR = \frac{FP}{TN + FP}, \tag{27}$$

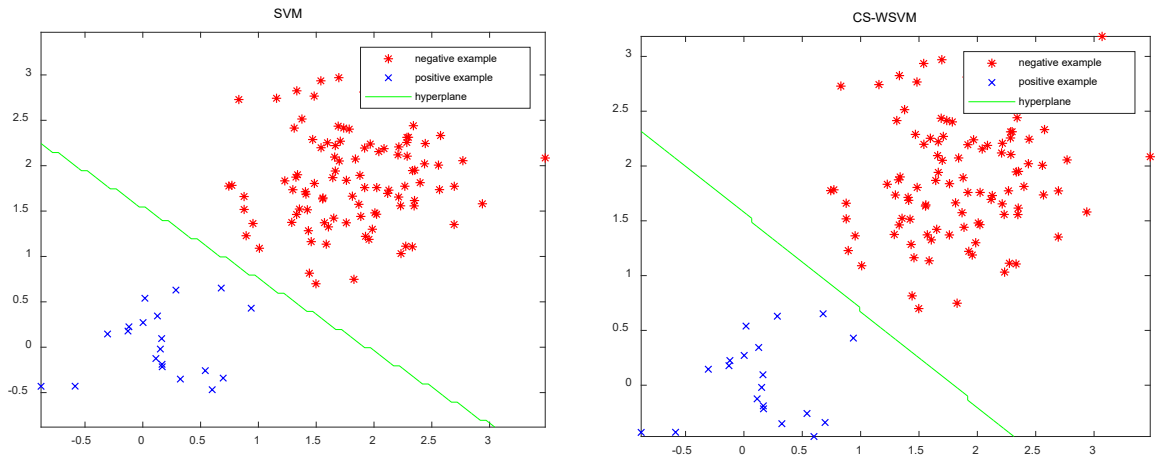
$$G\text{-}Mean = \sqrt{TPR \times TNR}\ . \tag{28}$$

Intuitively, *TNR* represents how many negative class samples are labeled correctly, namely, the majority class recognition accuracy. *TPR* measures the proportion of true positive samples correctly classified, that is, the minority class recognition accuracy. *FPR* indicates the proportion of all negative samples that are incorrectly predicted by the model, i.e., the false positive rate. *G-Mean* comprehensively considered the classification accuracy of the two types of samples. Compared with accuracy, *G-Mean* can effectively measure the classifier's performance on imbalanced datasets. In general, the higher the value of *G-Mean*, the better the classifier's performance is considered.

The ROC (Receiver Operating Characteristic) curve is the plot of the *TPR* against the *FPR*, and it is a performance measurement for the classifiers at various threshold settings. Generally, the closer the ROC curve is to the upper left, the better the performance of the classifier. However, the ROC curve does not give a quantitative evaluation of the performance of the classifier, so researchers often use the AUC (Area Under the ROC Curve) to evaluate the classifier's performance. The larger the AUC value, the better the classifier's performance.

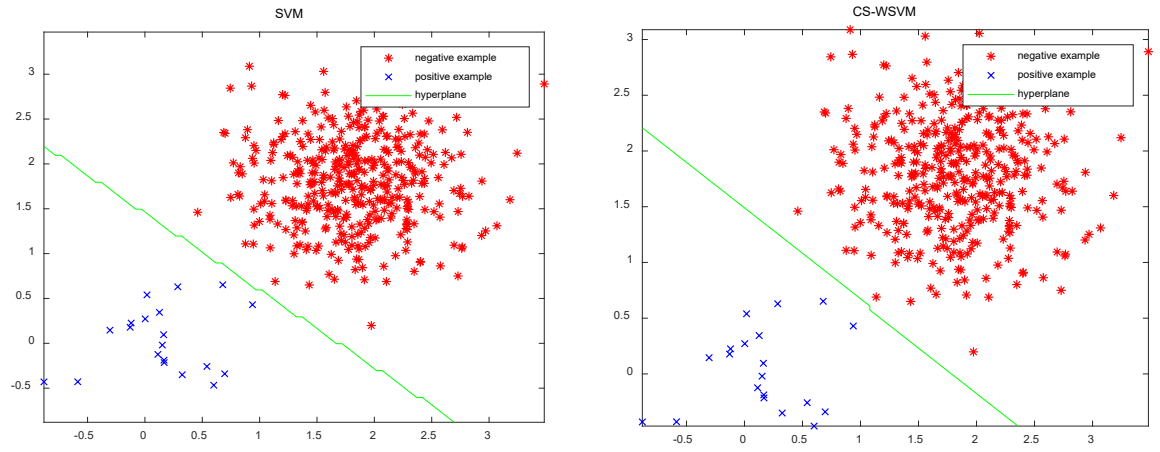
4.2 Experiments on Synthetic Dataset

In this subsection, we conducted the experiment on three synthetic datasets, i.e., two Non-overlapped datasets and one Overlapped dataset with imbalance ratios of 1:5, 1:20, and 1:5, respectively. The Non-overlapped dataset consists of two groups of randomly generated Gaussian distributions. The Overlapped dataset consists of four groups of randomly generated Gaussian distributions. Figure 2 and Figure 3 show the hyperplanes of SVM and CS-WSVM on Non-overlapped datasets with different imbalance ratios (Non-overlapped dataset1 and Non-overlapped dataset2), respectively. Figure 6 shows the hyperplanes of SVM and CS-WSVM on the Overlapped dataset, where red * indicates negative samples (majority class), blue x indicates positive samples (minority class), and solid green lines indicate the classification hyperplane.



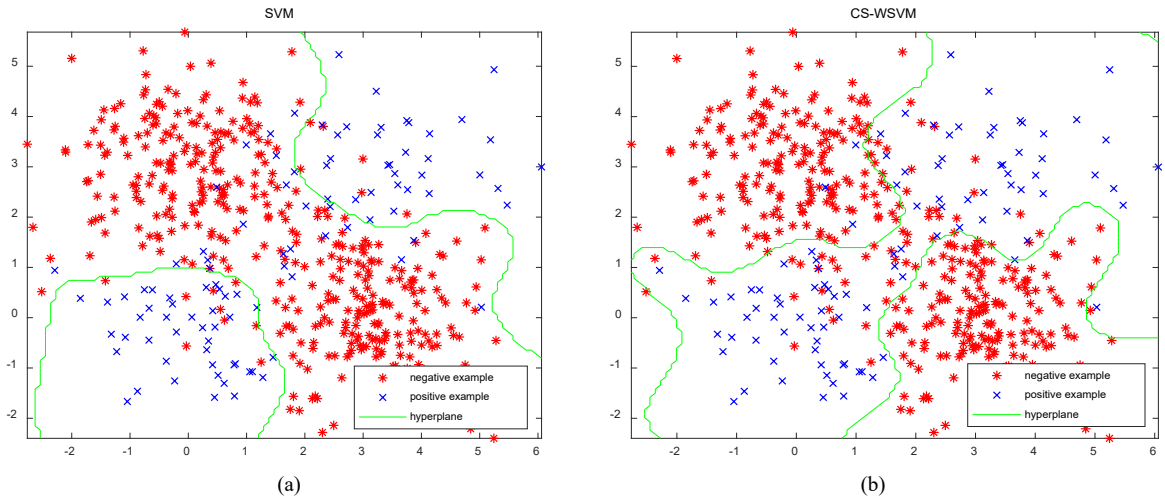
(a) (b)

Fig. 2. Classification result of SVM and CS-WSVM on the Non-overlapped dataset1 (Imbalance ratio = 1 : 5). (a) The discriminant boundary of SVM. (b) The discriminant boundary of CS-WSVM.



(a) (b)

Fig. 3. Classification result of SVM and CS-WSVM on the Non-overlapped dataset2 (Imbalance ratio = 1 : 20). (a) The discriminant boundary of SVM. (b) The discriminant boundary of CS-WSVM.



(a) (b)

Fig. 4. Classification result of SVM and CS-WSVM on Overlapped dataset (Imbalance ratio = 1 : 5). (a) The discriminant boundary of SVM. (b) The discriminant boundary of CS-WSVM.

As can be seen from Fig. 2 and Fig. 3, for the simple and separated Non-overlapped datasets, both SVM and CS-WSVM can obtain an optimal hyperplane, and the increase of the imbalance ratio has little effect on the decision boundary. This indicates that the class imbalance is not the only factor that weakens the classification performance.

From Fig. 4, we can see that the boundary is unwillingly skewing to the minority class samples due to the lack of minority class samples, resulting in a large number of minority class samples being misclassified. This is because SVM is accuracy-oriented, and it is optimized by reducing the overall error rate during the training process, so it must ensure the majority classes are correctly classified. In contrast, thanks to capturing the distance information between the two distributions, and assigning different weights to different samples, our CS-WSVM can obtain a more reasonable hyperplane while ensuring the minority class samples are correctly classified.

4.3 Experiments on Real-World Classification Problems

A. UCI Dataset

In the real-world problems, a total of 18 UCI balanced and imbalanced datasets are used to evaluate the performance of our CS-WSVM. For each dataset, we randomly split the samples by 50% for training and 50% for testing, and this process is repeated ten times to achieve a more stable result. In the linear case, for SVM and SVM-SMOTE, the parameter C is selected from the set $[10^{-3},\ldots,10^{3}]$. For BP-SVM and CS-SVM, we define $co_+ = pco_-$, where co_+ and co_- are penalty parameters for the positive and negative samples, respectively. For CS-WSVM, we define

$$\left\{co_i = co_- \,\middle|\, i \in \text{regular samples}\right\},$$

$$\left\{co_i = \frac{1}{\log(2+p)} co_- \,\middle|\, i \in \text{well-classified samples}\right\},$$

$$\left\{co_i = pco_- \,\middle|\, i \in \text{hard-classified samples}\right\}$$

and, co_- is selected from the set $[10^{-3},\ldots,10^{3}]$, the parameter p is selected from the set $[1,1.5,2,\ldots,8]$, the regularization parameter of CS-WSVM is selected from the set $[2^{-8},\ldots,2^{8}]$. In the nonlinear case, the RBF kernel $K\left(\mathbf{x}_i,\mathbf{x}_j\right)=\exp\left(-\frac{1}{2\sigma^2}\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2\right)$ will be used for all SVM algorithms. The width of the RBF kernel is selected from the set $[2^{-8},\ldots,2^{8}]$. We compared the G-Mean of all the algorithms. The G-Mean results with a linear kernel and the G-Mean result with an RBF kernel are illustrated in Tab. 3 and Tab. 4, respectively. The box plots of all classifiers are presented in Fig. 5.

Dataset	Ratio	SVM	BP-SVM	CS-SVM	SVM-SMOTE	SVM-ADASYN	CS-WSVM
Sonar	1.14	76.68±3.55	76.68±3.55	76.19±3.45	75.48±3.41	76.81±2.68	**76.91±4.19**
Breast	2.42	55.98±4.74	63.95±4.12	67.11±3.16	67.64±3.55	67.63±1.74	**69.19±2.54**
Cryotherapy	1.14	83.18±4.09	87.60±4.09	88.05±6.26	85.33±5.86	86.55±6.36	**89.27±4.24**
Fertility	7.33	30.08±27.04	48.66±13.32	49.59±7.38	48.31±19.81	57.14±6.48	**57.58±10.17**
Wdbc	1.68	95.48±1.03	95.78±1.66	**95.73±10.9**	95.56±1.15	95.42±1.34	95.01±0.94
Ionosphere	1.78	83.44±1.73	83.99±2.22	84.13±2.35	84.99±1.98	84.21±2.19	**85.55±2.17**
Hepatitis	5.15	69.10±11.69	69.51±11.83	72.96±7.60	72.37±9.52	72.55±6.83	**75.69±10.68**
Spectf	3.85	65.07±7.38	69.06±7.04	75.83±2.43	71.12±3.75	70.64±3.58	**76.82±1.96**
Pima	1.86	72.69±3.08	72.69±3.08	72.69±2.96	74.05±2.34	74.36±2.22	**74.40±1.38**
Heart	1.18	82.99±2.50	82.99±2.50	82.99±2.50	81.25±3.07	81.85±3.02	**83.44±2.84**
Liver	1.38	66.78±2.67	67.05±2.94	66.65±2.68	64.44±2.75	67.17±2.71	**67.42±2.87**
Bupa	1.38	**66.21±2.63**	**66.21±2.63**	65.60±3.11	62.41±4.01	66.07±3.90	66.02±3.27
Monk2	1.92	0	43.94±3.75	50.27±3.36	**57.15±3.87**	53.31±3.38	54.12±19.27
Haberman	2.78	32.03±27.80	55.75±3.22	61.37±4.07	59.97±4.03	59.96±4.67	**62.48±5.06**
Bcc	1.23	71.21±7.67	72.57±6.00	72.64±5.13	72.95±3.87	72.54±6.71	**73.81±6.87**
Wpbc	3.21	63.38±4.22	65.31±8.23	65.24±6.11	**66.58±5.57**	66.51±5.26	65.67±7.11
Planning	2.5	25.19±14.86	41.29±7.72	36.85±6.22	44.11±6.13	**44.85±6.47**	43.09±6.32
Vote	1.59	94.41±1.21	95.05±1.49	94.48±1.55	94.78±1.03	95.12±1.00	**95.24±1.03**

Tab. 3. G-Mean (mean±std.) comparison with linear kernel. The bold value indicates the best G-Mean on each dataset.

Dataset	Ratio	SVM	BP-SVM	CS-SVM	SVM-SMOTE	SVM-ADASYN	CS-WSVM
Sonar	1.14	**87.56±2.51**	**87.56±2.51**	87.11±2.50	86.72±1.82	**87.56±2.51**	85.48±1.62
Breast	2.42	27.60±4.65	63.70±5.21	63.97±4.76	68.56±2.10	68.11±2.22	**69.77±2.35**
Cryotherapy	1.14	84.00±4.29	86.01±4.89	85.33±6.16	84.70±4.69	86.24±4.01	**88.90±5.24**
Fertility	7.33	5.77±18.26	47.82±19.23	33.82±19.63	44.93±20.03	56.27±5.93	**58.36±8.20**
Wdbc	1.68	94.22±1.42	94.22±1.42	**94.33±0.93**	93.56±1.13	93.86±0.98	93.70±1.50
Ionosphere	1.78	94.65±1.62	94.62±1.63	94.83±1.54	94.17±1.21	94.38±1.47	**95.05±1.02**
Hepatitis	5.15	68.37±10.89	69.05±12.66	63.57±13.44	68.07±18.62	68.68±12.93	**75.47±8.55**
Spectf	3.85	53.34±12.67	69.22±5.68	75.58±3.41	72.77±3.11	76.66±3.35	**77.63±2.71**
Pima	1.86	72.32±2.39	72.57±2.03	73.17±2.19	74.16±2.65	74.68±2.14	**74.68±3.27**
Heart	1.18	83.09±2.56	83.09±2.56	83.38±1.74	81.61±2.98	82.62±2.31	**83.44±3.17**
Liver	1.38	69.95±3.11	69.95±3.11	70.42±3.17	69.00±4.05	69.84±3.18	**70.59±4.12**
Bupa	1.38	69.70±2.31	69.97±2.56	70.42±2.96	69.65±2.31	**70.62±2.13**	70.36±2.99
Monk2	1.92	89.92±5.89	**92.65±6.30**	92.59±2.04	92.28±6.96	91.62±2.48	87.75±6.76
Haberman	2.78	47.56±3.22	58.66±3.48	56.87±2.91	63.19±4.69	63.74±4.00	**64.38±4.22**
Bcc	1.23	68.60±5.01	68.59±4.99	67.00±8.38	67.19±6.03	66.54±6.08	**69.15±4.66**
Wpbc	3.21	63.78±6.21	64.39±6.03	61.65±5.83	62.23±6.78	67.17±4.13	**67.93±6.39**
Planning	2.5	48.55±4.69	49.11±5.11	46.53±4.35	46.74±5.01	46.28±6.79	**52.05±4.63**
Vote	1.59	95.01±1.02	95.23±0.99	94.92±1.10	95.14±0.94	95.24±0.63	**95.40±0.59**

Tab. 4. G-Mean (mean±std.) comparison with RBF kernel. The bold value indicates the best G-Mean on each dataset.

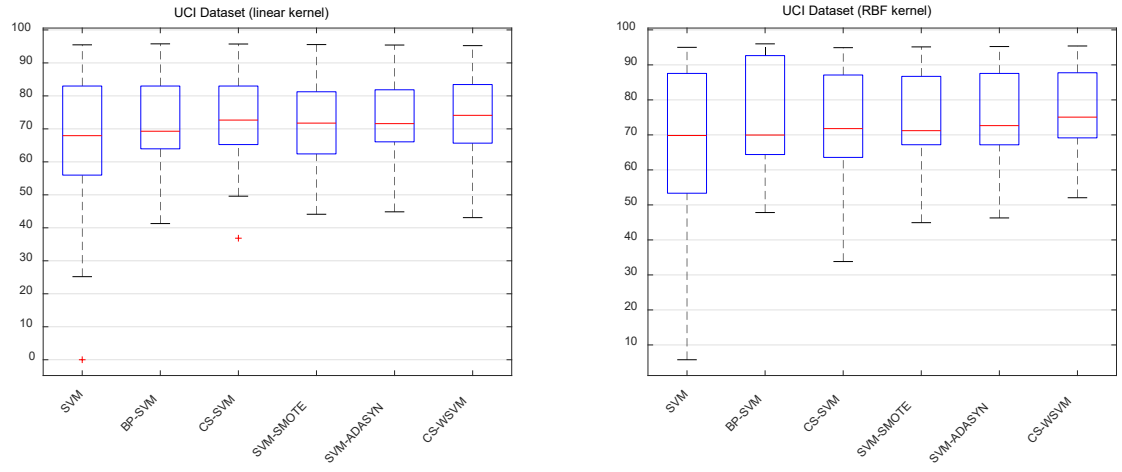


Fig. 5. The box plots for all classifiers on UCI Dataset. (a) Linear kernel. (b) RBF kernel.

From the results, we can draw the following conclusions:

(1) It can be seen from the experimental results that the imbalanced data distribution will affect the performance of the classifier, and the imbalanced classification method can effectively improve the performance of the classifier. As the improved algorithms of SVM, BP-SVM, CS-SVM, and CS-WSVM are devoted to modifying SVM in order to achieve cost-sensitivity, SVM-SMOTE adopts the over-sampling technique to deal with imbalanced datasets. However, the proposed CS-WSVM performs better than other methods on most datasets.

(2) For balanced datasets, due to the consideration of data distribution information, CS-WSVM can obtain better results than other methods. According to the well-known "No Free Lunch" Theorem, introducing data distribution information can indeed improve the classifier performance. The outstanding performance of CS-WSVM on balanced

datasets further validates the necessity of distribution information for the classifier design.

(3) For the three cost-sensitive methods, BP-SVM introduces different penalty parameters C_+ and C_- for the positive and negative samples during training, CS-SVM extends the SVM hinge loss to optimize the classifier with respect to class imbalance or class cost, CS-WSVM not only considers the data underlying structure information but also assigns different penalty parameters to different samples, so CS-WSVM is better than other methods in most cases. Specifically, for the linear case, CS-WSVM performs significantly better than SVM, BP-SVM, CS-SVM, and SVM-SMOTE on Breast, Cryotherapy, Fertility, and Hepatitis datasets. For kernel cases, CS-WSVM consistently outperforms other imbalanced classification approaches considerably on the Cryotherapy, Fertility, Hepatitis, Spectf, Wpbc, and Planning datasets.

B. KEEL Dataset

To further validate the effectiveness of our CS-WSVM, we conducted an experiment on the KEEL imbalanced dataset. KEEL (Knowledge Extraction Based on Evolutionary Learning) is an open-source Java software tool that can be used for a large number of different knowledge data discovery tasks. It contains a wide variety of classical knowledge extraction algorithms, preprocessing techniques, computational intelligence-based learning algorithms, hybrid models, statistical methodologies for contrasting experiments, and so forth. It provides a series of imbalanced datasets for classification. The experiments were conducted on 11 KEEL datasets. All the imbalance ratios of these datasets are higher than 5, with the highest up to 72.69.

Table 5 and Figure 6(a) illustrate the G-Mean results and box plots for all compared methods with linear kernels, respectively. As far as the results are concerned, CS-WSVM outperforms other methods on most of the datasets, especially on the highly imbalanced dataset. It weights the positive and negative samples based on the distribution information, thus reducing the bias of well-classified classes, and ultimately reducing the impact of data imbalance on the classifier. This shows that, benefiting from the consideration of the data distribution information and the cost-sensitive term, CS-WVSM can handle the imbalanced problem more effectively, improving performance.

Dataset	Ratio	SVM	BP-SVM	CS-SVM	SVM-SMOTE	SVM-ADASYN	CS-WSVM
Ecoli2	5.46	83.21±4.58	87.42±4.06	89.92±1.87	**89.93±1.83**	88.47±1.41	89.25±2.43
Ecoli4	15.8	94.84±3.35	93.94±3.941	93.31±4.30	93.63±4.13	93.20±1.41	**95.63±1.94**
Glass4	15.47	78.95±8.58	84.95±10.08	84.95±10.08	81.70±7.80	89.13±7.04	**90.51±3.21**
Yeast3	8.1	86.10±2.20	88.13±0.94	89.82±1.71	89.73±0.76	90.55±1.29	**90.59±0.83**
Yeast2vs4	9.08	84.66±4.84	86.59±3.50	87.32±3.75	88.41±3.32	87.75±2.70	**88.51±3.79**
Segment0	6.02	99.45±0.26	99.65±0.35	99.43±0.39	99.63±0.39	99.61±0.45	**99.68±0.25**
Vowel0	9.98	91.56±2.07	92.15±3.55	90.28±2.22	92.60±2.20	**95.70±1.82**	93.64±1.87
New_thyroid2	5.14	95.67±3.24	97.00±2.40	95.41±2.73	**97.76±2.05**	97.82±2.42	97.26±1.51
Yeast6	41.4	11.48±24.25	78.78±6.63	87.63±2.98	81.34±4.77	87.68±2.43	**88.74±2.79**
winequality-red-3_vs_5	68.1	33.51±25.02	48.85±19.33	44.61±24.95	42.31±24.58	61.55±9.74	**69.30±13.40**
Abalone20vs8_9_10	72.69	49.22±28.15	82.86±10.25	68.95±12.08	83.91±8.21	91.48±3.90	**91.51±5.55**

Tab. 5. G-Mean (mean±std.) comparison with linear kernel. The bold value indicates the best G-Mean on each dataset.

Dataset	Ratio	SVM	BP-SVM	CS-SVM	SVM-SMOTE	SVM-ADASYN	CS-WSVM
Data1	5.46	68.32±3.12	70.96±4.93	72.50±3.67	76.29±5.47	77.05±5.40	**78.83±3.38**
Data2	15.8	60.41±8.82	65.43±5.12	75.59±4.41	72.85±5.15	71.90±7.22	**78.21±1.44**
Data3	15.47	51.70±14.03	58.37±10.85	81.71±5.48	66.90±16.90	81.35±4.94	**84.84±2.27**
Data4	8.1	64.54±13.65	72.00±8.76	85.03±5.34	84.56±5.39	85.40±3.80	**89.20±3.67**

Tab. 6. G-Mean (mean±std.) comparison with linear kernel. The bold value indicates the best G-Mean on each dataset.

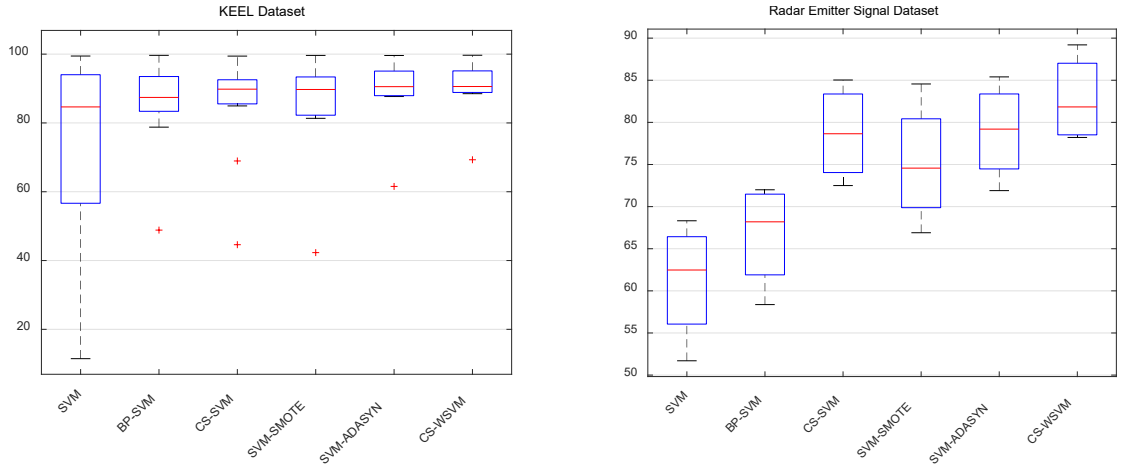


Fig. 6. The box plots for all classifiers on KEEL and Radar Emitter Signal Datasets. (a) KEEL Dataset. (b) Radar Emitter Signal Dataset.

C. Radar Emitter Signal Dataset

In order to test the effectiveness of our CS-WSVM in realistic applications, we also apply our method to specific emitter identification (SEI), a typical IFF problem, which plays an important role in modern electronic warfare. We adopted four radar emitter signal datasets (denoted as Data1, Data2, Data3, and Data4, respectively) to verify the effectiveness of the proposed algorithm.

The G-Mean results and box plots of CS-WSVM and other methods are presented in Tab. 6 and Fig. 6(b), respectively. As can be seen, BP-SVM, CS-SVM, SVM-SMOTE, and CS-WSVM all beat SVM by a considerable margin on the overall datasets. The performance of CS-WSVM is significantly better when the imbalance is more extreme.

Figure 7 shows the TPR and TNR results of CS-SVM and CS-WSVM on the radar emitter signal datasets. Comparing CS-SVM and CS-WSVM, due to capturing the data distribution information and considering the class separability, CS-WSVM obtains a more reasonable TPR and TNR than CS-SVM. It also shows that the introduction of prior knowledge can improve the classifier's performance. The outstanding performance of CS-WSVM further validates the necessity of distribution information as prior knowledge for the classifier design.

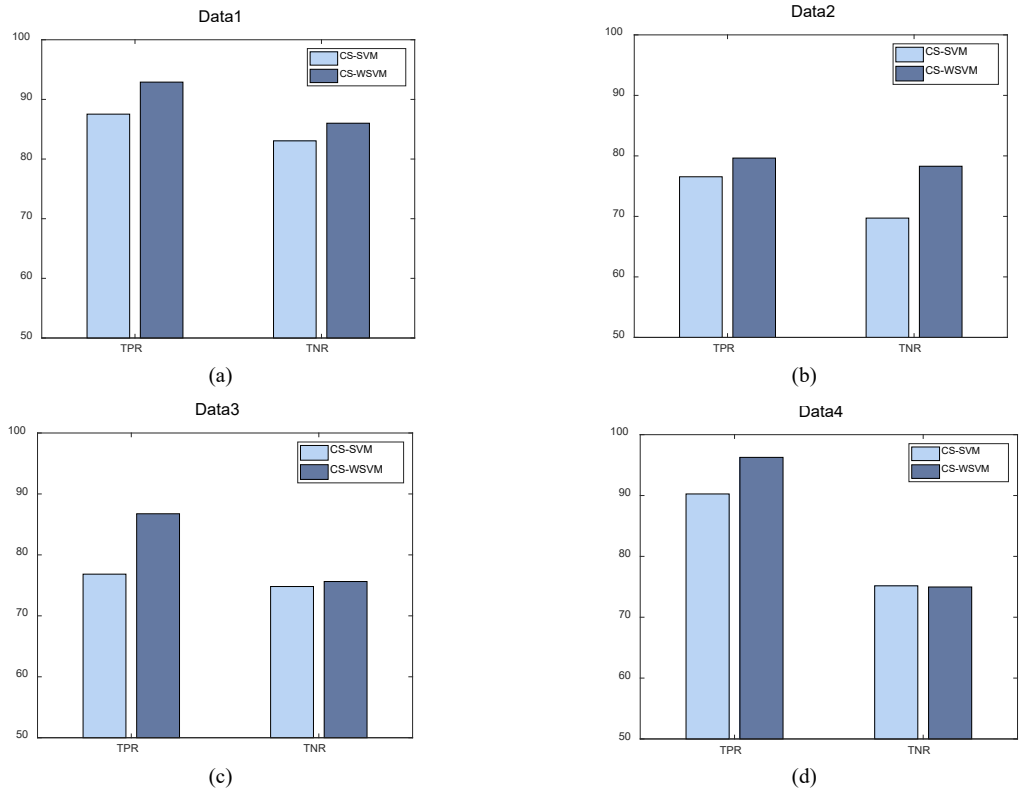
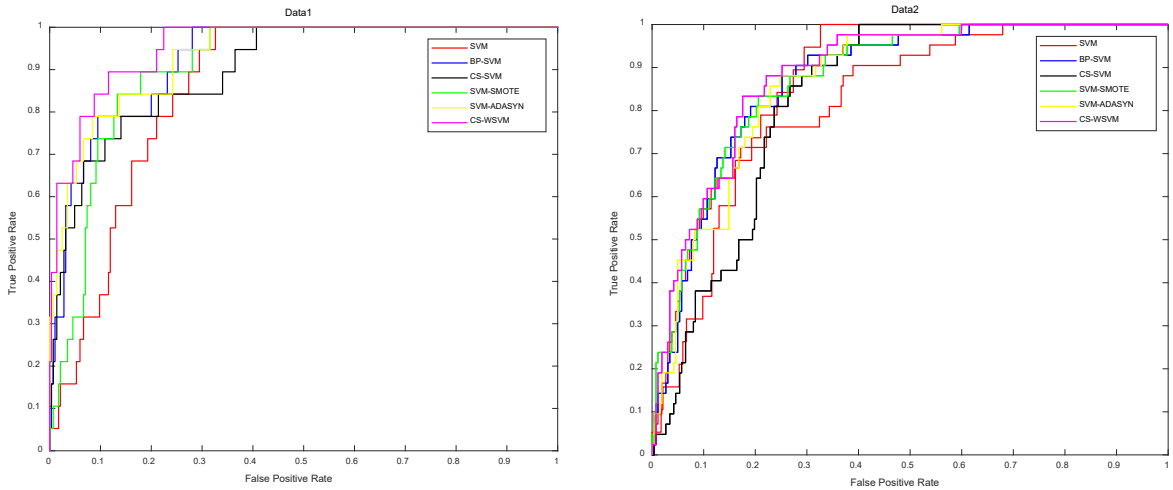


Fig. 7. TPR and TNR compared between CS-SVM and CS-WSVM on radar Emitter Signal Dataset. (a) Data1, (b) Data2, (c) Data3, (d) Data4.

The ROC curves of the different methods for Data1 and Data2 are given in Fig. 8, and the corresponding AUC values for each method are given in Tab. 8. The results show that our CS-WSVM has the largest area under the curve and has the best classification performance in the region with a low false positive rate (FPR).

The imbalanced ratio-performance curves for all classifiers on radar emitter signal datasets are depicted in Fig. 9. As can be seen from the figure, the performance of the classifier does not simply decrease with the growth of the imbalanced ratio. As we discuss in this manuscript, the imbalance is not the only reason affecting the classification performance, but it is also strongly related to the distribution characteristics of the dataset itself. As shown in Fig. 10, although Data4 has the highest imbalanced ratio, it has the best classification results due to its simple and separable data distribution.

Finally, we compared the training times of the different methods with a linear kernel on the radar emitter signal dataset, and the experimental result is presented in Tab. 8. We can see that, since CS-WVM involves clustering and the calculation of Wasserstein distance, it takes the longest training time, which is the shortcoming of this method at present. How to improve the running time of CS-WSVM is an important issue that we need to consider in the future.



(a) (b)

Fig. 8. The ROC curves of the Radar Emitter Signal Dataset: (a) Data1, (b) Data2.

Datasets	Ratio	SVM	BP-SVM	CS-SVM	SVM-SMOTE	SVM-ADASYN	CS-WSVM
Data1	7.00	0.8598	0.9245	0.9010	0.9064	0.9331	**0.9540**
Data2	11.82	0.8381	0.8720	0.8327	0.8744	0.8682	**0.8860**

Tab. 7. The AUC value of the different methods on the Radar Emitter Signal Dataset.

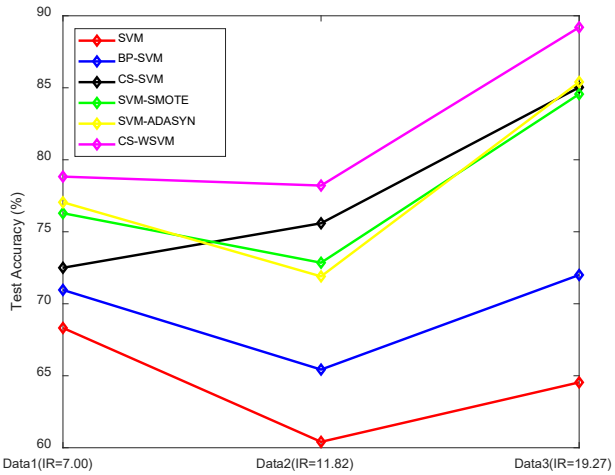
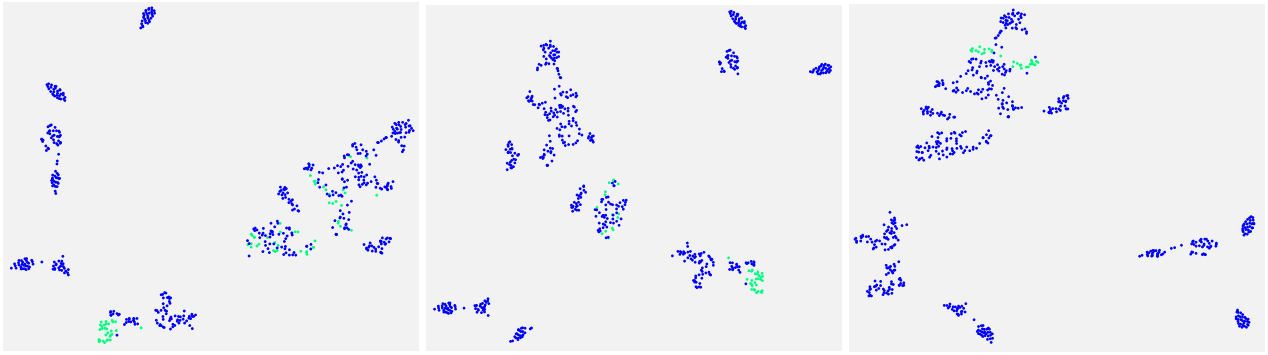


Fig. 9. The ratio-performance curves for all classifiers on Radar Emitter Signal Dataset.



(a) (b) (c)

Fig. 10. The T-SNE visualization of Radar Emitter Signal Dataset: (a) Data1(IR = 7.00), (b) Data2 (IR = 11.82), (c) Data4 (IR = 19.27).

Datasets	Ratio	SVM	BP-SVM	CS-SVM	SVM-SMOTE	SVM-ADASYN	CS-WSVM
Data1	7.00	0.0216	0.0240	0.0269	0.0268	0.1377	0.6774
Data2	11.82	0.0214	0.0198	0.0272	0.0384	0.1095	1.1272
Data3	19.27	0.0179	0.0192	0.0280	0.0248	0.0928	0.6527
Data4	19.27	0.0235	0.0218	0.0188	0.0318	0.0945	0.7037

Tab. 8. The training times of the different methods with liner kernel on the Radar Emitter Signal Dataset.

4.4 Discussion

Experiments conducted on two-dimensional synthetic datasets show that the classification difficulty of imbalanced data does not depend entirely on its degree of imbalance. The classifier can obtain a good classification result for a simple and separated dataset despite the significant difference in the number of samples between the two classes. It is evident that the class imbalance is only a superficial feature of the data and is not the primary reason that weakens the classifier's performance. However, the inherent structural features of the data set are the key factors that affect the classification. Therefore, for the classification problem of imbalanced data, it is more important to design a reasonable scheme for the specific problem from the inherent structural characteristics of the dataset.

The experiments on real-world datasets show that the proposed CS-WSVM can effectively classify imbalanced datasets, especially those with complex structures, and it can also obtain better classification results than the conventional SVM for balanced datasets.

5. Conclusion

In this paper, a novel cost-sensitive approach for imbalanced classification task is proposed. Unlike the existing cost-sensitive SVM, the proposed method incorporates prior structural information and cost-sensitive strategy, thus generalizing the SVM in a cost-sensitive framework while considering the underlying data structural information. Specifically, a new distance is imported to model the distribution of positive and negative samples, and the dataset is divided into well-classified samples, hard-classified samples, and regular samples according to this distance. Then the model can assign different misclassification costs to the different types of samples that we defined above. In addition, the structural information is also introduced into the standard SVM object function in the form of regular terms. We conducted extensive experiments on UCI benchmark datasets and a series of real-world tasks. The experimental results show that the proposed CS-WSVM significantly improves the performance over the standard SVM, especially for the highly imbalanced dataset.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62276204, 62203343) and the Fundamental Research Funds for the Central Universities (Grant No. ZYTS23139). The authors would like to thank the anonymous reviewers for their critical and constructive review of the manuscript.

References

[1] HE, H., GARCIA, E. A. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 2009, vol. 21, no. 9, p. 1263–1284. DOI: 10.1109/TKDE.2008.239

[2] SVYD, I., OBOD, I., MALTSEV, O., et al. Method of increasing the identification friend or foe systems information security. In *3rd International Conference on Advanced Information and Communications Technologies (AICT)*. Lviv (Ukraine), 2019, p. 434–438. DOI: 10.1109/AIACT.2019.8847853

[3] LI, Z., HUANG, M., LIU, G., et al. A hybrid method with dynamic weighted entropy for handling the problem of class imbalance with

overlap in credit card fraud detection. *Expert Systems with Applications*, 2021, vol. 175, p. 1–10. DOI: 10.1016/j.eswa.2021.114750

[4] TAN, K., ZHANG, L., YAN, W., et al. A semi-supervised emitter identification method for imbalanced category (in Chinese). *Journal of Radars*, 2022, vol. 11, no. 4, p. 713–727. DOI: 10.12000/JR22043

[5] BAGUI, S., LI, K. Resampling imbalanced data for network intrusion detection datasets. *Journal of Big Data*, 2021, vol. 8, no. 1, p. 1–41. DOI: 10.1186/s40537-020-00390-x

[6] PAN, T., CHEN, J., XIE, J., et al. Deep feature generating network: A new method for intelligent fault detection of mechanical systems under class imbalance. *IEEE Transactions on Industrial Informatics*, 2021, vol. 17, no. 9, p. 6282–6293. DOI: 10.1109/TII.2020.3030967

[7] KOZIARSKI, M., WOŹNIAK, M., KRAWCZYK, B. Combined cleaning and resampling algorithm for multi-class imbalanced data with label noise. *Knowledge-Based Systems*, 2020, vol. 204, p. 1 to 16. DOI: 10.1016/j.knosys.2020.106223

[8] LI, M., XIONG, A., WANG, L., et al. ACO resampling: Enhancing the performance of oversampling methods for class imbalance classification. *Knowledge-Based Systems*, 2020, vol. 196, p. 1–17. DOI: 10.1016/j.knosys.2020.105818

[9] CHARTE, F., RIVERA, A. J., DEL JESUS, M. J., et al. REMEDIAL-HwR: Tackling multilabel imbalance through label decoupling and data resampling hybridization. *Neurocomputing*, 2019, vol. 326–327, p. 110–122. DOI: 10.1016/j.neucom.2017.01.118

[10] YAN, J., HAN, S. Classifying imbalanced data sets by a novel re-sample and cost-sensitive stacked generalization method. *Mathematical Problems in Engineering*, 2018, vol. 2018, p. 1–13. DOI: 10.1155/2018/5036710

[11] ZHANG, C., TAN, K. C., LI, H., et al. A cost-sensitive deep belief network for imbalanced classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2019, vol. 30, no. 1, p. 109–122. DOI: 10.1109/TNNLS.2018.2832648

[12] BI, J., ZHANG, C. An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme. *Knowledge-Based Systems*, 2018, vol. 158, p. 81–93. DOI: 10.1016/j.knosys.2018.05.037

[13] TAO, X., LI, Q., GUO, W., et al. Self-adaptive cost weights-based support vector machine cost-sensitive ensemble for imbalanced data classification. *Information Sciences*, 2019, vol. 487, p. 31–56. DOI: 10.1016/j.ins.2019.02.062

[14] BOONCHUAY, K., SINAPIROMSARAN, K., LURSINSAP, C. Decision tree induction based on minority entropy for the class imbalance problem. *Pattern Analysis and Applications*, 2017, vol. 20, no. 3, p. 769–782. DOI: 10.1007/s10044-016-0533-3

[15] ZHANG, S. Cost-sensitive KNN classification. *Neurocomputing*, 2020, vol. 391, p. 234–242. DOI: 10.1016/j.neucom.2018.11.101

[16] ZERAATKAR, S., AFSARI, F. Interval–valued fuzzy and intuitionistic fuzzy–KNN for imbalanced data classification. *Expert Systems with Applications*, 2021, vol. 184, p. 1–16. DOI: 10.1016/j.eswa.2021.115510

[17] RICHHARIYA, B., TANVEER, M. A reduced universum twin support vector machine for class imbalance learning. *Pattern Recognition*, 2020, vol. 102, p. 1–19. DOI: 10.1016/j.patcog.2019.107150

[18] HOU, X., ZHANG, T., JI, L., et al. Combating highly imbalanced steganalysis with small training samples using feature selection. *Journal of Visual Communication and Image Representation*, 2017, vol. 49, p. 243–256. DOI: 10.1016/j.jvcir.2017.09.016

[19] CHAWLA, N. V. Data mining for imbalanced datasets: An overview. In MAIMON, O., ROKACH, L. (eds.) *Data Mining and Knowledge Discovery Handbook*. Boston (MA, USA): Springer, 2009, p. 875–886. DOI: 10.1007/978-0-387-09823-4_45

[20] BUDA, M., MAKI, A., MAZUROWSKI, M. A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 2018, vol. 106, p. 249–259. DOI: 10.1016/j.neunet.2018.07.011

[21] FERNÁNDEZ, A., GARCÍA, S., GALAR, M., et al. *Learning from Imbalanced Data Sets*. Cham: Springer, 2018, ch. 11, p. 279 to 303. DOI: 10.1007/978-3-319-98074-4_11

[22] THABTAH, F., HAMMOUD, S., KAMALOV, F., et al. Data imbalance in classification: Experimental evaluation. *Information Sciences*, 2020, vol. 513, p. 429–441. DOI: 10.1016/j.ins.2019.11.004

[23] VAN HULSE, J., KHOSHGOFTAAR, T. Knowledge discovery from imbalanced and noisy data. *Data & Knowledge Engineering*, 2009, vol. 68, no. 12, p. 1513–1542. DOI: 10.1016/j.datak.2009.08.005

[24] CERF, L., GAY, D., SELMAOUI-FOLCHER, N., et al. Parameter-free classification in multi-class imbalanced data sets. *Data & Knowledge Engineering*, 2013, vol. 87, p. 109–129. DOI: 10.1016/j.datak.2013.06.001

[25] WANG, W., WANG, J. N., HU, F. L., et al. SCA-CGAN: A new side-channel attack method for imbalanced small samples. *Radioengineering*, 2023, vol. 32, no. 1, p. 124–135. DOI: 10.13164/re.2023.0124

[26] DOMINGOS, P. MetaCost: A general method for making classifiers cost-sensitive. In *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Diego (CA, USA), 1999, p. 155–164. DOI: 10.1145/312129.312220

[27] LI, F., ZHANG, X., ZHANG, X., et al. Cost-sensitive and hybrid-attribute measure multi-decision tree over imbalanced data sets. *Information Science*, 2018, vol. 422, p. 242–256. DOI: 10.1016/j.ins.2017.09.013

[28] LOYOLA-GONZALEZ, O., MARTINEZ-TRINIDAD, J. F. C. O., CARRASCO-OCHOA, J. A., et al. Cost-sensitive pattern-based classification for class imbalance problems. *IEEE Access*, 2019, vol. 7, p. 60411–60427. DOI: 10.1109/ACCESS.2019.2913982

[29] JIA, J., ZHAI, L., REN, W., et al. An effective imbalanced jpeg steganalysis scheme based on adaptive cost-sensitive feature learning. *IEEE Transactions on Knowledge and Data Engineering*, 2022, vol. 34, no. 3, p. 1038–1052. DOI: 10.1109/tkde.2020.2995070

[30] LEE, H. K., KIM, S. B. An overlap-sensitive margin classifier for imbalanced and overlapping data. *Expert Systems with Applications*, 2018, vol. 98, p. 72–83. DOI: 10.1016/j.eswa.2018.01.008

[31] NIRANJAN, M. Support vector machines: A tutorial overview and critical appraisal. In *IEE Colloquium on Applied Statistical Pattern Recognition*. Birmingham (UK), 1999. DOI: 10.1049/ic:19990359

[32] HOU, S., ZHOU, Y., LIU, H., et al. Wavelet support vector machine algorithm in power analysis attacks. *Radioengineering*, 2017, vol. 26, no. 3, p. 890–902. DOI: 10.13164/re.2017.0890

[33] MUSHTAQ, M. T., KHAN, I., KHAN, M. S., et al. Signal detection for QPSK based cognitive radio systems using support vector machines. *Radioengineering*, 2015, vol. 24, no. 1, p. 192–198. DOI: 10.13164/re.2015.0192

[34] LUO, Z., WANG, X., WANG, L., et al. Hydrometeor classification for dual polarization radar based on multi-sample fusion SVM. *Radioengineering*, 2023, vol. 32, no. 1, p. 151–159. DOI: 10.13164/re.2023.0151

[35] XIE, Z., XU, Y., HU, Q. Uncertain data classification with additive kernel support vector machine. *Data & Knowledge Engineering*, 2018, vol. 117, p. 87–97. DOI: 10.1016/j.datak.2018.07.004

[36] DEVI, D., BISWAS, S. K., PURKAYASTHA, B. Learning in presence of class imbalance and class overlapping by using one-class SVM and undersampling technique. *Connection Science*, 2019, vol. 31, no. 2, p. 105–142. DOI: 10.1080/09540091.2018.1560394

[37] IRANMEHR, A., MASNADI-SHIRAZI, H., VASCONCELOS, N. Cost-sensitive support vector machines. *Neurocomputing*, 2019, vol. 343, p. 50–64. DOI: 10.1016/j.neucom.2018.11.099

[38] LI, Y., KWOK, J. T., ZHOU, Z. Cost-sensitive semi-supervised support vector machine. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2010, vol. 24, no. 1, p. 500–505. DOI: 10.1609/aaai.v24i1.7661

[39] KARAKOULAS, G., SHAWE-TAYLOR, J. Optimizing classifiers for imbalanced training sets. *Advances in Neural Information Processing Systems*, 1999, vol. 11, p. 253–259.

[40] QI, Z., TIAN, Y., SHI, Y., et al. Cost-sensitive support vector machine for semi-supervised learning. *Procedia Computer Science*, 2013, vol. 18, p. 1684–1689. DOI: 10.1016/j.procs.2013.05.336

[41] RUBNER, Y., TOMASI, C., GUIBAS, L. J. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 2000, vol. 40, p. 99–121. DOI: 10.1023/A:1026543900054

[42] SHIVASWAMY, P. K., JEBARA, J. Ellipsoidal kernel machines. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics.* San Juan (Puerto Rico), 2007, vol. 2, p. 484–491.

[43] JOHNSON, S. C. Hierarchical clustering schemes. *Psychometrika*, 1967, vol. 32, no. 3, p. 241–254. DOI: 10.1007/BF02289588

[44] CHAWLA, N. V., BOWYER, K. W., HALL, L. O., et al. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 2002, vol. 16, p. 321–357. DOI: 10.1613/jair.953

[45] ALCALÁ-FDEZ, J., SÁNCHEZ, L., GARCÍA, S., et al. KEEL: A software tool to assess evolutionary algorithms to data mining problems. *Soft Computing*, 2009, vol. 13, p. 307–318. DOI: 10.1007/s00500-008-0323-y

[46] *Machine Learning Repository UCI.* [Online] Available at: http://archive.ics.uci.edu/ml/datasets.html

[47] WOODBURY, M. A. *Inverting Modified Matrices.* Statistical Research Group, Memo. Rep. no. 42*,* 1950, p. 1–4.

[48] HE, H., BAI, Y., GARCIA, E. A., et al. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE International Joint Conference on Neural Networks*. Hong Kong, 2008, p. 1322–1328. DOI: 10.1109/ijcnn.2008.4633969

[49] SALAZAR, A., VERGARA, L., SAFONT, G. Generative adversarial networks and Markov random fields for oversampling very small training sets. *Expert Systems with Applications*, 2021, vol. 163, p. 1–12. DOI: 10.1016/j.eswa.2020.113819

Appendix A: Hierarchical Clustering

Hierarchical clustering attempts to partition datasets at different levels, then build a tree which all the leaves correspond to the given data point. The partitioning strategies are generally divided into bottom-up and top-down methods. The former considers each individual sample as a cluster and then merges the two closest clusters into one cluster. The key problem is how to calculate the distance between clusters. Here we use the ward's linkage distance.

Concretely, for clusters S and T, their ward's linkage can be calculated as

$$\mathbf{W}(S,T)=\frac{|S|\cdot|T|\cdot\left\|\mu_S-\mu_T\right\|^2}{|S|+|T|} \tag{A.1}$$

where μ_S and μ_T are the means of cluster S and T, respectively.

Initially, we define each sample is a cluster, the distance of $\mathbf{x}_i$ and $\mathbf{x}_j$ is $\mathbf{W}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^2/2$, we merge the two closest samples into a single cluster, then compute distances between the new cluster and each of the old clusters, repeat this process until all samples are clustered into a single cluster.

When two clusters S and T are being merged to a new cluster Y, the distance between Y and the other old cluster Z can be derived from $\mathbf{W}(S,T)$, $\mathbf{W}(S,Z)$, and $\mathbf{W}(T,Z)$ by

$$\mathbf{W}(Y,Z)=\frac{(|S|+|Z|)\mathbf{W}(S,Z)+(|T|+|Z|)\mathbf{W}(T,Z)-|Z|\mathbf{W}(S,T)}{|S|+|T|+|Z|}. \tag{A.2}$$

In the kernel space, the ward's linkage clustering is still applicable. The ward's linkage between $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$ can be calculated by

$$\begin{aligned}&\mathbf{W}\left(\Phi(\mathbf{x}_i),\Phi(\mathbf{x}_j)\right)\\&=\frac{\left\|\Phi(\mathbf{x}_i)-\Phi(\mathbf{x}_j)\right\|^2}{2}\\&=\frac{\Phi(\mathbf{x}_i)^{\mathrm{T}}\Phi(\mathbf{x}_i)-2\Phi(\mathbf{x}_i)^{\mathrm{T}}\Phi(\mathbf{x}_j)+\Phi(\mathbf{x}_j)^{\mathrm{T}}\Phi(\mathbf{x}_j)}{2}.\end{aligned} \tag{A.3}$$

However, due to the higher dimensions, Φ cannot be accurately represented, we need to define Φ as a dot product form: $\mathbf{K} = \Phi(\mathbf{X})^{\mathrm{T}}\,\Phi(\mathbf{X})$. The $\mathbf{W}(\Phi(\mathbf{x}_i),\Phi(\mathbf{x}_j))$ can be rewritten as

$$\begin{aligned}&\mathbf{W}(\Phi(\mathbf{x}_i),\Phi(\mathbf{x}_j))\\&=\frac{1}{2}\left[K(\mathbf{x}_i,\mathbf{x}_i)+K(\mathbf{x}_j,\mathbf{x}_j)-2K(\mathbf{x}_i,\mathbf{x}_j)\right].\end{aligned} \tag{A.4}$$

When two clusters S^φ and T^φ are being merged to a new cluster Y^φ, the distance between Y^φ and the other old cluster Z^φ can be calculated by

$$\begin{aligned}&\mathbf{W}(Y^\varphi,Z^\varphi)\\&=\frac{\left(\left|S^\varphi\right|+\left|Z^\varphi\right|\right)\mathbf{W}(S^\varphi,Z^\varphi)+\left(\left|T^\varphi\right|+\left|Z^\varphi\right|\right)\mathbf{W}(T^\varphi,Z^\varphi)}{\left|S^\varphi\right|+\left|T^\varphi\right|+\left|Z^\varphi\right|}\\&-\frac{\left|Z^\varphi\right|\mathbf{W}(S^\varphi,T^\varphi)}{\left|S^\varphi\right|+\left|T^\varphi\right|+\left|Z^\varphi\right|}.\end{aligned} \tag{A.5}$$

Appendix B: Datasets

In this paper, we conducted a series of experiments on three real-world datasets, including UCI machine datasets, KEEL datasets and radar emitter signal datasets. The details about each dataset are presented in Tab. B.1, Tab. B.2, Tab. B.3, respectively.

Dataset	Feature	Negative	Positive	Ratio
Sonar	60	111	97	1.14
Breast	9	196	81	2.42
Cryotherapy	6	48	42	1.14
Fertility	9	88	12	7.33
Wdbc	30	357	212	1.68
Ionosphere	34	225	126	1.78
Hepatitis	19	67	13	5.15
Spectf	44	212	55	3.85
Pima	8	500	268	1.86
Heart	13	164	139	1.18
Liver	6	200	145	1.38
Bupa	6	200	145	1.38
Monk2	6	395	206	1.92
Haberman	3	225	81	2.78
Bcc	9	64	52	1.23
Wpbc	33	151	47	3.21
Planning	12	130	52	2.5
Vote	16	267	168	1.59

Tab. B.1. Attributes of the UCI datasets. Feature is the number of features. Negative is the number of negative samples. Positive is the number of positive samples. Ratio is the class imbalance ratio.

Dataset	Feature	Negative	Positive	Ratio
Ecoli2	7	284	52	5.46
Ecoli4	7	316	20	15.8
Glass4	9	201	13	15.47
Yeast3	8	1321	163	8.1
Yeast2vs4	8	463	51	9.08
Segment0	19	1979	329	6.02
Vowel0	13	898	90	9.98
New_thyroid2	5	180	35	5.14
Yeast6	8	1449	35	41.4
winequality-red-3_vs_5	11	681	10	68.1
Abalone20vs8_9_10	7	1890	26	72.69

Tab. B.2. Attributes of the KEEL datasets. Feature is the number of features. Negative is the number of negative samples. Positive is the number of positive samples. Ratio is the class imbalance ratio.

Dataset	Feature	Negative	Positive	Ratio
Data1	125	532	76	5.46
Data2	125	532	45	15.8
Data3	50	578	30	15.47
Data4	125	578	30	8.1

Tab. B.3. Attributes of the Radar Emitter Signal datasets. Feature is the number of features. Negative is the number of negative samples. Positive is the number of positive samples. Ratio is the class imbalance ratio.

About the Authors…

Rui FENG was born in Shanxi. She received her B.S. degree and M.S. degree in the School of Electronic Engineering from Xidian University. She is now a Ph.D. candidate in the School of Electronic Engineering, Xidian University. Her research interests include deep learning, pattern recognition, and signal processing.

Hongbing JI (corresponding author) was born in Shaanxi. He received the B.S. degree in Radar Engineering, the M.S. degree in Circuit, Signals and Systems, and the Ph.D. degree in Signal and Information Processing from the Northern West Telecommunications Engineering College (now Xidian University), Xi'an, China, in 1983, 1989, and 1999, respectively. Since 1989, he has been with the School of Electronic Engineering, Xidian University. He is currently a Professor and an Advisor for Ph.D. students. His research interests include pattern recognition, radar signal processing, and multi-sensor information fusion.

Zhigang ZHU was born in Shandong. He received B.S. (2013) degree in the School of Communication and Electronic Engineering from Qingdao University of Technology. He received M.S. (2014) degree in Signal and Information Processing and Ph.D. (2020) degree in Pattern Recognition and Intelligent Systems from Xidian University, respectively. After graduation in 2020, he has been with the School of Electronic Engineering at Xidian University, a lecturer from 2020 to now. His primary areas of research include pattern recognition, radar signal processing, deep learning and action recognition.

Lei WANG was born in Anhui. He received his Ph.D. degree in Pattern Recognition and Intelligent System from Xidian University at 2011. Currently, he is an Associate Professor with the School of Electronic Engineering, Xidian University. His research interests include pattern recognition, signal and information processing, machine learning and computer vision.