Information Sciences 578 (2021) 659–682

Contents lists available at ScienceDirect

Information Sciences

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

Class imbalance learning using fuzzy ART and intuitionistic fuzzy twin support vector machines

Salim Rezvani, Xizhao Wang *

Big Data Institute, College of Computer Science and Software Engineering, Guangdong Key Lab. of Intelligent Information Processing, Shenzhen University, Shenzhen 518060, Guangdong, China

ARTICLE INFO

Article history:
Received 18 November 2019
Received in revised form 29 June 2021
Accepted 2 July 2021
Available online 6 July 2021

Keywords:
Class imbalance learning
Coordinate descent
Intuitionistic fuzzy number
Fuzzy ART
Twin support vector machine

ABSTRACT

The classification in imbalanced datasets is one of the main problems for machine learning techniques. Support vector machine (SVM) is biased to the majority class samples, and the minority class samples may incorrectly be considered as noise. Therefore, SVM has poor predictive accuracy for imbalanced datasets and generates inaccurate classification models. Existing class imbalance learning (CIL) techniques can make SVM less sensitive to class imbalance, but these methods suffer from issues related to noise and outliers. Moreover, despite the solid theoretical basis and good classification performance, SVM is not appropriate for the classification of large-scale datasets because the training complexity of SVM is closely related to the dataset size. Class imbalance learning (CIL) using Fuzzy adaptive resonance theory (ART) and intuitionistic fuzzy twin SVM (CIL-FART-IFTSVM), which can be applied to address the class imbalance issue in the presence of noise and outliers and large scale datasets, is proposed to overcome these substantial difficulties. In this method, we modify the distribution of the datasets using fuzzy adaptive resonance theory (Fuzzy ART) as a clustering method to overcome the imbalance problem. Then, after data reduction, IFTSVM is utilized to find excellent non-parallel hyperplanes in the generated data points. Finally, a coordinate descent system with shrinking by an active set is applied to reduce the computational complexity. Forty-five imbalanced datasets are considered to validate the performance of the proposed CIL-FART-IFTSVM method. The Friedman test and the bootstrap technique with 95% confidence intervals are applied to quantify the results statistically. The experimental results indicate that the method proposed in this paper has a better performance compared with other methods, and the training time is significantly better than that of other classifiers for large-scale datasets.

1. Introduction

Support vector machines (SVM), introduced by Vapnik [1], is a popular machine learning technique that has been successfully applied to data classification and function estimation problems in various areas.

In addition to SVM with two parallel hyperplanes, various classifiers with nonparallel hyperplanes such as the generalized eigenvalue proximal SVM (GEPSVM) [2] and twin SVM (TSVM) [3–5] have been developed. As represented in [3], TSVM is 4 times faster than SVM. SVM cannot locate an optimal hyperplane if the support vectors are polluted by noise, which gener-

* Corresponding author.
E-mail addresses: salim_rezvani@szu.edu.cn (S. Rezvani), xizhaowang@ieee.org (X. Wang).

https://doi.org/10.1016/j.ins.2021.07.010
0020-0255/

ates poor results. Similar to the typical SVM, TSVM is also limited by noise deterioration. Therefore, a fuzzy support vector machine (FSVM), which uses a degree of membership function for every training sample to represent their importance, was introduced in [6–8]. Due to this effective strategy, FSVM can partially solve the aforementioned problem but is still sensitive to the datasets that are polluted by noises and outliers, similar to traditional TSVM and SVM.

The intuitionistic fuzzy set (IFS) [9] is a more accurate extension of the fuzzy set. Fuzzy sets are specified by only the membership function, but an IFS is specified by a membership function and a nonmembership function. Values and ambiguities of the membership and nonmembership functions for an IFS are used in [10,11] to determine the value index and the ambiguity index. Rezvani and Wang [12] introduced IFTSVM, which combines the idea of the IFS with TSVM to alleviate the noise related to the polluted inputs.

Although SVM works efficiently with balanced datasets, for imbalanced datasets, SVM can generate inaccurate results [13–15]; that is, the SVM technique generates a model based on the majority class that has poor performance on the minority class. When these algorithms are applied to imbalanced datasets, they cannot achieve sufficient accuracy on both classes of the data. To minimize these issues, methods have been proposed to promote advisable performing classifiers for imbalanced datasets. These types of learning methods are commonly called class imbalance learning (CIL) techniques. Some applications suffer from CIL problem, including face recognition [16], fault diagnosis [17], anomaly detection [18], and e-mail foldering [19].

[20] proposed an FTSVM based on information entropy for CIL in which the idea of entropy-based FSVM for the imbalanced dataset motivated from [21,22] and the new fuzzy membership evaluation inspired from [23]. In [24], the authors indicated ensemble diversity has a positive impact on the classification of imbalanced data sets. They had two aims. First, the reason for diversity measured by Q-statistic can bring improved overall accuracy is described. Then, pattern analysis of single-class performance measures is extended. A new support vector machine was introduced in [25], which is called GSVM. This method created for bi-classification issues that balanced the accuracy between classes is the objective. The bias for GSVM is calculated by moving the original bias in the SVM to improve the geometric mean between the true positive rate and the true negative rate.

A variant of ELM for handling binary class imbalance problem suggested in [26], which is named class-specific extreme learning machine. This work differs from weighted ELM as it doesn't need to assign weights to the training instances. A weighted under-sampling (WU) scheme for SVM based on space geometry distance is proposed in [27] to enhance the classification performance to deal with the data imbalance problems. In WU-SVM, the majority samples are grouped into some sub-regions (SRs) and assigned different weights according to their Euclidean distance to the hyperplane. In [28] a solution that can effectively find an advisable hyperplane by automatically tuning the error cost for between-class samples is offered. This method has two main features: (1) it can evaluate how efficient an error cost is in terms of classification accuracy; and (2) it changes the error cost in the right direction if it is not efficient. A novel self-adaptive cost weights-based support vector machine cost-sensitive ensemble for imbalanced data classification is suggested in [29]. To guarantee the consistency of optimization objectives between weak learners and boosting scheme, this method not only utilizes cost-sensitive SVMs as basic weak learners but also simultaneously modifies the standard boosting scheme to cost-sensitive ones. In [30], the authors offered two efficient sampling techniques that improve data distributions. One re-balanced technique and the other technique is Gaussian Over-sampling. The first method enhances the SMOTE technique by adaptively selecting groups of Inner and Danger data from the minority class. The others combine dimension reduction with the Gaussian distribution, which makes the tail of the Gaussian distribution thinner.

Moreover, SVM is not appropriate for the classification of large datasets because SVM requires solving a quadratic programming (QP) problem to find a separation hyperplane, which leads to extreme computational complexity. Some scholars have attempted to find appropriate methods to use SVM to classify large-scale datasets. The available methods can generally be categorized into to approaches: reducing the training dataset [31,32] and modifying the SVM classifier [33,34]. The large QP problem is changed via sequential minimal optimization (SMO) [35] into a series of small QP problems. The projected conjugate gradient (PCG) scales somewhere between linear and cubic in the training set [36]. Clustering, for instance, hierarchical clustering [33] and parallel clustering [37], is one way to reduce the size of a dataset. Another technique to decrease the training data is to apply the geometric characteristics of SVM [38]. Identifying the maximum-margin hyperplane is equivalent to locating the nearest neighbors (NN) in the convex hulls of each class [39]. The NN problem (NPP) can then be reformulated to complete SVM classification [40].

To address problems such as CIL, noise/outliers, and large-scale datasets, we propose a new imbalance learning method as CIL-FART-IFTSVM. The contributions of this proposed method are as follows:

1) The proposed framework is designed to modify the distribution of a dataset. This framework not only finds samples that possibly support vectors but also perfectly reduces the imbalance ratio to overcome the problem of imbalanced datasets. Moreover, the proposed framework has specific advantages on large-scale datasets, for which the training time is significantly better than that of other techniques.

2) The proposed method significantly reduces the negative impact of noise and outliers because Fuzzy ART can select an appropriate subset of the original majority samples, therefore all the majority samples are easily recognized and trained with our model.

3) The proposed method performs statistically better on imbalanced datasets than do similar techniques.

This paper is organized as follows: Section II presents the details of IFS, SVM, FSVM, TSVM, and IFTSVM. Section III characterizes the structure of the proposed CIL-FART-IFTSVM model. Section IV discusses experimental results. Conclusions and suggestions for future research are presented in Section V.

2. Related work

In this Section, we first present the intuitionistic fuzzy set (IFS) and then define SVM, FSVM, TSVM, and IFTSVM. Finally, we discuss the structure of the fuzzy adaptive resonance theory (ART).

2.1. IFS

A fuzzy set A in a universe X (nonempty set) can be defined as [9,10]

$$A = \{(x, \mu_A(x))|x \in X\} \tag{1}$$

where $\mu_A(x)$ is the degree of membership of $x \in X$ and $\mu_A : X \to [0, 1]$. An IFS is defined as

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x), v_{\tilde{A}}(x))|x \in X\} \tag{2}$$

where $\mu_{\tilde{A}}(x)$ is the degree of membership function and $v_{\tilde{A}}(x)$ is the degree of nonmembership function of $x \in X$.

2.2. SVM

Normal SVM can be applied to binary classification. SVM searches for the optimal hyperplane $w^T x + b = 0$, where $b \in \mathcal{R}$ is the bias term and $w \in \mathcal{R}^n$ is the weight. This hyperplane can be applied to describe the label, positive or negative, of input sample x_i as follows [1]:

$$\begin{cases} (w.x_i + b) \geqslant 0, & if \quad y_i \ is \ positive, \\ (w.x_i + b) \leqslant 0, & if \quad y_i \ is \ negative. \end{cases} \tag{3}$$

In SVM for linear cases, after solving the following primal quadratic programming problem (QPP), an optimal hyperplane can be obtained:

$$\begin{cases} min \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i, \\ s.t. y_i (w^T x_i + b) \geqslant 1 - \xi_i, \xi_i \geqslant 0, i = 1, 2, \ldots, l. \end{cases} \tag{4}$$

where $\xi_i (i = 1, 2, \ldots, l)$ are slack variables, C is a penalty parameter, and l is the number of training samples.

2.3. FSVM

Let $\{(x_1, y_1, s_1), (x_2, y_2, s_2), .., (x_i, y_i, s_i)\}$ be a training set consisting of i samples with their related fuzzy membership functions (s_i), where $\sigma \leqslant s_i \leq 1$ and $\sigma > 0$ is sufficiently small. Let $z = \phi(x)$ be a mapping $\phi : \mathcal{R}^N \to \mathcal{Z}$ to a feature space. The optimal hyperplane can be obtained by solving following equation [6]:

$$min \frac{1}{2} w^T .w + C \sum_{i=1}^{l} s_i \xi_i$$

$$s.t. y_i (w.z_i + b) \geqslant 1 - \xi_i, \quad \xi_i \geqslant 0, i = 1, \ldots, l \tag{5}$$

where $s_i \xi_i$ is the measured error with different weightings.

The Lagrangian can be defined as follows:

$$maximize W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$s.t. \sum_{i=1}^{l} y_i \alpha_i = 0, \ 0 \leqslant \alpha_i \leqslant s_i C, \ i = 1, \ldots, l \tag{6}$$

and the Karush–Kuhn–Tucker (K.K.T.) conditions [41] are defined as:

$$\bar{\alpha}_i (y_i (\bar{w}.z_i + \bar{b}) - 1 + \bar{\xi}_i) = 0, \quad i = 1, \ldots, l \tag{7}$$

$$(s_i C - \bar{\alpha}_i)\xi_i = 0, \quad i = 1, \ldots, l \tag{8}$$

The point x_i with the corresponding $\bar{\alpha}_i > 0$ is recognized as a support vector (SV). FSVM can have two types of SV. The first, with $0 < \bar{\alpha}_i < s_i C$, lies on the margin of the hyperplane, and the second, with $\bar{\alpha}_i = s_i C$, is misclassified.

2.4. TSVM

In contrast to standard SVM, which uses just one hyperplane to separate the positive samples from the negative samples, TSVM [3] produces two nonparallel hyperplanes

$$w_{(1)}.x_i + b_{(1)} = 0, \quad w_{(2)}x_i + b_{(2)} = 0 \tag{9}$$

where $w_{(i)}$ is the weight and $b_{(i)}$ is a bias term of the i-th hyperplane. The two hyperplanes are obtained by solving the following QPPs:

$$\min_{w_{(1)},b_{(1)},\xi_2} \frac{1}{2}\left(Aw_{(1)} + e_1 b_{(1)}\right)^T \left(Aw_{(1)} + e_1 b_{(1)}\right) + p_1 e_2^T \xi_2$$

$$s.t. \quad -\left(Bw_{(1)} + e_2 b_{(1)}\right) + \xi_2 \geqslant e_2, \xi_2 \geqslant 0 \tag{10}$$

and

$$\min_{w_{(1)},b_{(1)},\xi_1} \frac{1}{2}\left(Bw_{(2)} + e_2 b_{(2)}\right)^T \left(Bw_{(2)} + e_2 b_{(2)}\right) + p_2 e_2^T \xi_1$$

$$s.t. \quad \left(Aw_{(2)} + e_1 b_{(2)}\right) + \xi_1 \geqslant e_1, \xi_1 \geqslant 0 \tag{11}$$

where A is class $+1$ and B is class -1, e_1 and e_2 are vectors of all ones, ξ_1 and ξ_2 are slack functions, and p_1 and p_2 are penalty parameters. The optimal parameters, i.e., (w_1^*, b_1^*) and (w_2^*, b_2^*), are obtained, and a input sample x can be achieved as follows:

$$f(x) = arg\ min_{i\in 1,2} \frac{|(w_i^*)^T x + b_i^*|}{\|w_i^*\|}. \tag{12}$$

2.5. IFTSVM

Rezvani and Wang [12] recently proposed IFTSVM to improve the influence and generalizability of FTSVM, which uses an intuitionistic fuzzy number to create a pair of membership and nonmembership functions for every training samples.

IFTSVM for a linear kernel can be defined as follows:

$$\min_{w_1,b_1,\xi_2} \frac{1}{2}\|Aw_1 + e_1 b_1\|^2 + \frac{1}{2}C_1\|w_1\|^2 + C_2 s_2^T \xi_2$$

$$subject\ to \ -(Bw_1 + e_2 b_1) + \xi_2 \geqslant e_2, \xi_2 \geqslant 0 \tag{13}$$

and

$$\min_{w_2,b_2,\xi_1} \frac{1}{2}\|Bw_2 + e_2 b_2\|^2 + \frac{1}{2}C_3\|w_2\|^2 + C_4 s_1^T \xi_1$$

$$subject\ to\ (Aw_2 + e_1 b_2) + \xi_1 \geqslant e_1, \xi_1 \geqslant 0 \tag{14}$$

where C_1, C_2, C_3 and C_4 are positive penalty parameters, ξ_1 and ξ_2 are slack variables, e_1 and e_2 are vectors of all ones, $s_1 \in \mathcal{R}^{l_+}$ is the scores of the positive class, and $s_2 \in \mathcal{R}^{l_-}$ is the scores of the negative class.

Using K.K.T. conditions and Eq. (13), the Wolfe dual can be obtained as:

$$\max_{\alpha} e_2^T \alpha - \frac{1}{2}\alpha^T G_2 \left(H_1^T H_1 + C_1 I\right)^{-1} G_2^T \alpha$$

$$subject\ to\ 0 \leqslant \alpha \leqslant C_2 s_2 \tag{15}$$

Furthermore, for Eq. (14), the Wolfe dual can be obtained as:

$$\max_{\beta} e_1^T \beta - \frac{1}{2}\beta^T G_1 \left(G_2^T G_2 + C_3 I\right)^{-1} H_1^T \beta$$

$$subject\ to\ 0 \leqslant \beta \leqslant C_4 s_1 \tag{16}$$

A new pattern of x can be classified as a member of the positive class or negative class as follows:

$$x \in W_k, k = \underset{i=1,2}{arg\ min}\left\{\frac{|w_1^T x + b_1|}{\|w_1\|}, \frac{|w_2^T x + b_2|}{\|w_2\|}\right\} \tag{17}$$

where $|.|$is the absolute value.

The kernel function for the nonlinear case is defined as follows:

$$k\left(x, X^T\right)w_1 + b_1 = 0, \quad k\left(x, X^T\right)w_2 + b_2 = 0, \tag{18}$$

where $k(x_1, x_2) = (\phi(x_1), \phi(x_2))$ is a kernel function. The primal issue of nonlinear IFTSVM is defined as:

$$\underset{w_1, b_1, \xi_2}{min}\ \frac{1}{2}\|k\left(A, X^T\right)w_1 + e_1 b_1\|^2 + \frac{1}{2}C_1\|w_1\|^2 + C_2 s_2^T \xi_2$$

$$subject\ to\ -\left(k\left(B, X^T\right)w_1 + e_2 b_1\right) + \xi_2 \geqslant e_2, \xi_2 \geqslant 0 \tag{19}$$

and

$$\underset{w_2, b_2, \xi_1}{min}\ \frac{1}{2}\|k\left(B, X^T\right)w_2 + e_2 b_2\|^2 + \frac{1}{2}C_3\|w_2\|^2 + C_4 s_1^T \xi_1$$

$$subject\ to\ \left(k\left(A, X^T\right)w_2 + e_1 b_2\right) + \xi_1 \geqslant e_1, \xi_1 \geqslant 0 \tag{20}$$

With the Lagrangian method and the K.K.T. conditions, the corresponding Wolfe dual can be obtained as:

$$\underset{\alpha}{maxe}_2^T \alpha - \frac{1}{2}\alpha^T G_2^*\left(H_1^{T*}H_1^* + C_1 I\right)^{-1} G_2^{T*}\alpha$$

$$subject\ to\ 0 \leqslant \alpha \leqslant C_2 s_2 \tag{21}$$

and

$$\underset{\beta}{maxe}_1^T \beta - \frac{1}{2}\beta^T G_1^*\left(G_2^{T*}G_2^* + C_3 I\right)^{-1} H_1^{T*}\beta$$

$$subject\ to\ 0 \leqslant \beta \leqslant C_4 s_1 \tag{22}$$

A new pattern of x can be classified as a member of the positive class or negative class as follows:

$$k = \underset{i=1,2}{arg\ min}\left\{\frac{|w_1^T k\left(x, X^T\right) + b_1|}{\sqrt{w_1^T k\left(A, X^T\right)w_1}}, \frac{|w_2^T k\left(x, X^T\right) + b_2|}{\sqrt{w_2^T k\left(B, X^T\right)w_2}}\right\}. \tag{23}$$

2.6. Fuzzy Adaptive Resonance Theory (Fuzzy ART)

ART is a neural theory of cognitive information processing that states that fast learning is a resonant phenomenon in neural circuits [42]. Fuzzy ART inherits the benefits of ART, including fast and stable learning and incremental clustering. The Fuzzy ART method is typically applied in the incremental learning model of a self-organizing neural network (SONN) [43]. The Fuzzy ART architecture consists of two-layer nodes or neurons, the feature representation field F_1, and the category representation field F_2, as shown in Fig. 1. The two layers are connected via adaptive weights w_j, emanating from node j in layer F_2. In other words, those nodes are connected by the bottom-up-weight vector w_{ij} and top-down-weight vector w_{ji}.

The ability to automatically obtain the number of clusters is a benefit of Fuzzy ART clustering. Generally, Fuzzy ART is accurate for well-separated data. In Fuzzy ART, as shown in Fig. 2, a low vigilance level, which is a value used to scale the cluster size, results in small specific categories, including patterns from different data distributions or classes (Fig. 2(a)), while a high vigilance level results in more categories (Fig. 2(b)).

In fact, Fuzzy ART is a combination of fuzzy logic and ART network that applies the fuzzy operators min($\wedge$) and max($\vee$).

$$y_j = \frac{|I \wedge w_j|}{(\alpha + |w_j|)} \tag{24}$$

where I is an input vector, w_j is a weight vector, and $\alpha > 0$ is the parameter to be selected. Each input I is an m-dimensional vector $I = (i_1, i_2, \ldots, i_m)$, where each component $i_i (i = 1, 2, \ldots, m)$ is in the interval $[0, 1]$. I (the input to the network) is normalized by adding A, which is the actual input, to its complement $1 - A$ [44]. An original vector $A = (a_1, \ldots, a_k)$ is coded into

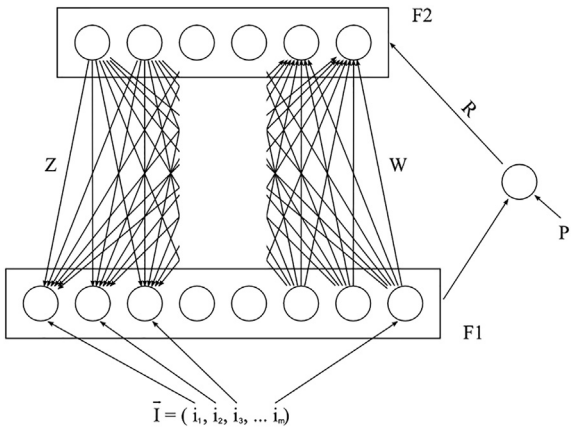


Fig. 1. Basic ART structure.

an input pattern I by adding the complements of its elements to the original vector. This doubles the dimension of all input patterns and prototypes

$$I = (A, A') = (a_1, \ldots, a_k, 1 - a_1, \ldots, 1 - a_k) \quad a_i \in [0, 1] \quad \forall\, i$$

The L_1-$norm^2$ of complement encoded vectors of the same dimension is constant, independent of the values of their elements

$$|I| = \sum_{i=1}^{2k} i_i = \sum_{i=1}^{k} a_i + \sum_{i=1}^{k} 1 - a_i = \sum_{i=1}^{k} a_i + k - \sum_{i=1}^{k} a_i = k = m/2$$

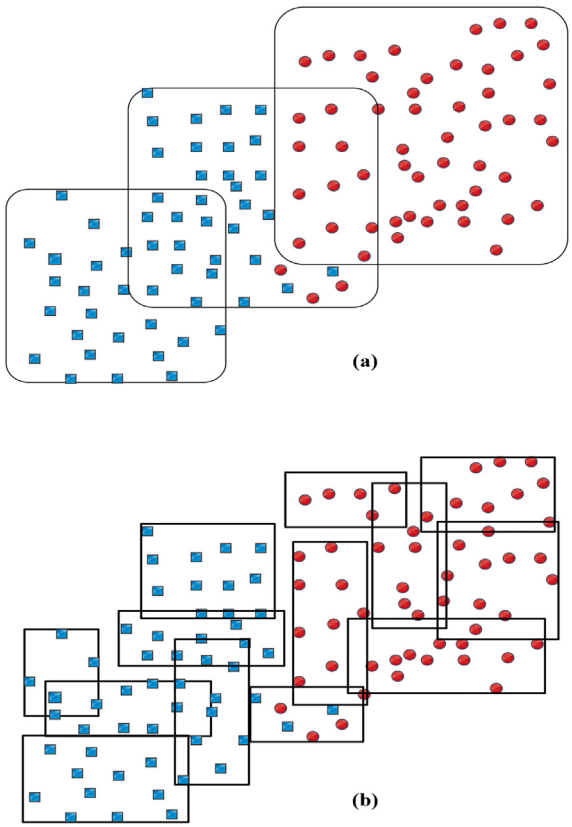


Fig. 2. Geometric explanation of Fuzzy ART for low vigilance level data (a) and high vigilance level data (b).

Each category j in the weight vector corresponds to a vector $w_j = (w_{j1}, w_{j2}, \ldots, w_{jm})(j = 1, 2, \ldots, n)$ of adaptive weight. The number of potential categories n is arbitrary. Initially

$$w_{j1} = w_{j1} = \ldots = w_{jm} = 1$$

The Fuzzy ARTs dynamics is identified by selection parameter $\alpha > 0$ to break the tie when more than one prototype vector is a fuzzy subset of the input pattern,

$$y_j = \max_j \{y_j\}$$

The normalized input pattern I is used to compute y in Eq. (24).

The winner, which is denoted by y_j with J as the winning node index, and an expectation is reflected in layer F_1, as shown in Eq. (24), has to pass the vigilance test. ρ, a value between 0 and 1, is a vigilance parameter set by the user, and the number of existing prototypes is k.

$$\rho \leqslant \frac{|I \wedge w_j|}{k} \tag{25}$$

If the test is passed, resonance occurs. Input I joins cluster J, and w_J (the winning prototype vector) is updated via the following equation,

$$w_J^{new} = \beta\left(I \wedge w_J^{old}\right) + (1 - \beta) w_J^{old} \tag{26}$$

where β is a Fuzzy ART parameter, called learning rate, which may assume values in the interval $(0, 1]$. If $\beta = 1$ the learning is called fast learning.

On the other hand, if the vigilance criterion is not met, a reset signal is sent back to layer F_2 to shut off the current winning neuron, which will remain disabled for the entire duration of the presentation of this input pattern, and a new competition is performed among the rest of the neurons. This new expectation is then projected into layer F_1, and this process repeats until the vigilance criterion is met. In the case that an uncommitted neuron is selected for coding, a new uncommitted neuron is created to represent a potential new cluster.

3. Proposed method

In this section, we modify the distribution of a dataset using the Fuzzy ART as a clustering method to overcome the class imbalance problem. Then, the IFTSVM [12] technique is utilized to find an excellent non-parallel hyperplane in the generated data points. Finally, a coordinate descent system with shrinking by an active set is applied to address the computational complexity. The proposed method, CIL-FART-IFTSVM, can be applied to solve the class imbalance problem in the presence of noise and outliers and for large-scale datasets.

3.1. Class Imbalance Learning Using Fuzzy ART and Intuitionistic Fuzzy Twin Support Vector Machines (CIL-FART-IFTSVM)

Let T be the training set for IFTSVM:

$$T = \{x_1, y_1, \mu_1, v_1\}, \{x_2, y_2, \mu_2, v_2\}, \ldots, \{x_l, y_l, \mu_l, v_l\}.$$

where μ_i and v_i are the degrees of the membership and nonmembership functions of x_i, respectively. For IFTSVM, the score function can be defined as:

$$s_i = \begin{cases} \mu_i & v_i = 0, \\ 0 & \mu_i \leqslant v_i, \\ \frac{1 - v_i}{2 - \mu_i - v_i} & others. \end{cases} \tag{27}$$

The role of IFTSVM classification is to detect the optimal hyperplane that maximizes the margin among the classes. As shown in Fig. 3, CIL-FART-IFTSVM can be organized into three steps:

(1) Data selection with Fuzzy ART,

(2) Declustering to modify the distribution of datasets,

(3) IFTSVM classification with Coordinate descent to reduce the computational complexity.

The description of each step is given as follows.

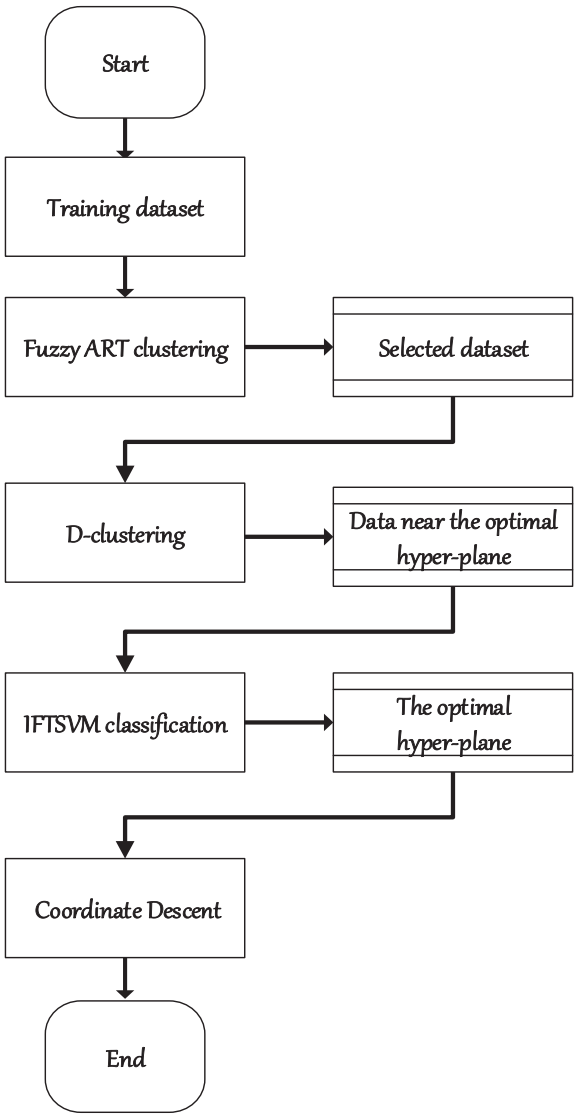


Fig. 3. CIL-FART-IFTSVM flowchart.

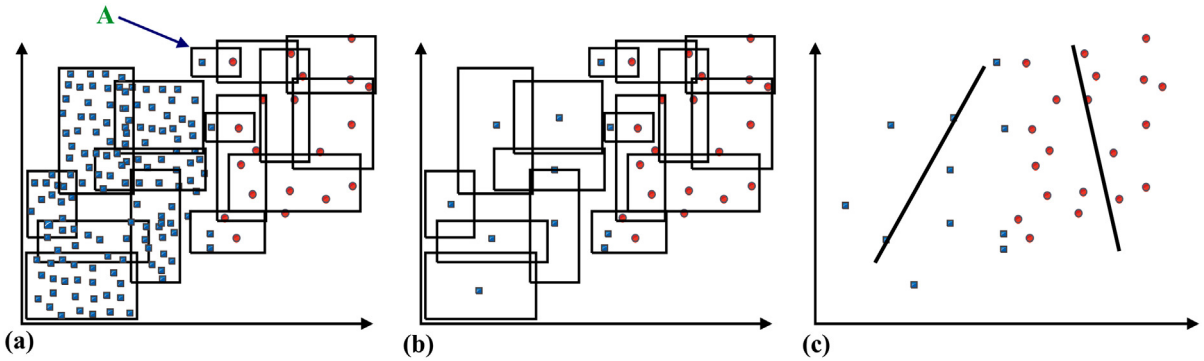


Fig. 4. An example of (1) Data selection with Fuzzy ART, (2) Declustering to modify the distribution of datasets, (3) IFTSVM classification.

3.1.1. Data selection with Fuzzy ART

To implement CIL-FART-IFTSVM, we must first select data from imbalanced datasets as the input of CIL-FART-IFTSVM. Suppose, as shown in Fig. 4, that the majority class is negative samples (the blue squares) and the minority class is positive samples (the red circles). We apply Fuzzy ART clustering as the data selection method. The goal of Fuzzy ART clustering is to find N clusters or partitions, i.e., F_i $(i = 1, 2, \ldots, N)$ from T, where $N < l, F_i \neq \phi$, and $\cup_{i=1}^{N} F_i = T$. The obtained clusters are as follows:

(i) clusters with only positive samples, defined by F_+, i.e., $F_+ = \{\cup F_i | y = +1\}$,

(ii) clusters with only negative samples, defined by F_-, i.e., $F_- = \{\cup F_i | y = -1\}$,

(iii) clusters with both types of samples (+/-) or mix-labeled, defined by $F_\pm$, i.e., $F_\pm = \{\cup F_i | y = \pm 1\}$.

Fig. 4(a) shows the clusters after Fuzzy ART, where the clusters with only red circles are positive samples (i.e., F_+), the clusters with only blue squares are negative samples (i.e.. F_-), and clusters such as A consist of both samples (i.e.. $F_\pm$). We illustrate the set of the centers of the clusters in F_+ and F_- are C_+ and C_-, respectively, i.e.,

$$C_+ = \{\cup C_i | y = +1\} \ \textit{positive samples center} \tag{28}$$

$$C_- = \{\cup C_i | y = -1\} \ \textit{negative samples center} \tag{29}$$

The class center [12] of each class can be measured by

$$C^{\pm} = \frac{1}{l_\pm} \sum_{y_i = \pm 1} x_i$$

where l_+ is the total number of positive and l_- is the total number of negative samples.

3.1.2. Declustering to modify the distribution of datasets

We suggest restoring data to the training dataset by including the data in the clusters, we call this procedure declustering. To address imbalanced datasets, we focus on only the majority samples (the blue squares). Thus, more data close to the hyperplane can be found via declustering, and the dataset used in this step is the union of C_-, F_+ and $F_\pm$, i.e., $C_- \cup F_+ \cup F_\pm$. The declustering outcomes of the support vectors are presented in Fig. 4 (b). In fact, we delete only those clusters consisting of all the negative samples (majority) and maintain only the centers of those samples. Additionally, we keep all the data of the positive samples (minority) and mix-labeled clusters ($F_\pm$) as the training data in this step. The declustering process not only finds samples that possibly support vectors but also modifies the distribution of the datasets to overcome the imbalance, which improves the accuracy.

3.1.3. IFTSVM classification with coordinate descent method to reduce computational complexity

In this step, the recovered dataset (reduced dataset from the second step) is taken as a new training set. We use IFTSVM classification to obtain the decision hyperplanes to find excellent nonparallelism in the generated data points.

Our Wolfe dual IFTSVM equations require a pair strictly convex QPPs (Eqs. 15 and 16 or 21 and 22), but they can be solved similarly. For instance, by defining $W = \left(H_1^T H_1 + C_1 I\right)^{-1} G_2^T$ and $\overline{W} = G_2 W$, Eq. (15) can be simplified as a quadratic explanation:

$$\min_{\alpha}(\alpha) = \frac{1}{2}\alpha^T \overline{W} \alpha - e_2^T \alpha$$

$$\textit{subject to } 0 \leqslant \alpha \leqslant C_2 s_2 \tag{30}$$

To solve the above equation, a coordinate descent strategy with active set shrinking is approved to address the computational complexity. The pseudocode is presented in Algorithm 1. Readers can find more theoretical details in [45,46]. In our algorithm, $\nabla_i^{proj} g(\alpha)$ is a projected gradient defined as

$$\nabla_i^{proj} g(\alpha) = \begin{cases} min\left(0, \nabla_i^{proj} g(\alpha)\right) & \alpha_i = 0, \\ \nabla_i^{proj} g(\alpha) & 0 < \alpha_i < C_2 s_2, \\ max\left(0, \nabla_i^{proj} g(\alpha)\right) & \alpha_i = C_2 s_2. \end{cases} \tag{31}$$

where $\nabla_i g$ denotes the $i - th$ component of gradient ∇g.

Algorithm 1 CIL-FART-IFTSVM with active set shrinking

Compute $W = \left(H_1^T H_1 + C_1 I\right)^{-1} G_2^T$ and $\overline{W}_{ii} = G_{2i} W_i$
Set $A \leftarrow \{1, \ldots, l_2\}$
Given ϵ and initialized $\alpha \leftarrow 0, u_1 \leftarrow 0$
Initialized $\overline{N} \leftarrow \infty$ and $\overline{n} \leftarrow -\infty$
while do
  Initialize $N \leftarrow -\infty$ and $n \leftarrow \infty$
  for all $i \in A$ (randomly and exclusively selected) **do**
    Compute $\nabla_i g(\alpha) = -G_{2i} u_1 - 1$
    Assign temporally $\nabla_i^{proj} g(\alpha) \leftarrow 0$
    if $\alpha_i = 0$ **then**
      if $\nabla_i^{proj} g(\alpha) > \overline{N}$, **then** $A = A \setminus \{i\}$ **end if**
      if $\nabla_i^{proj} g(\alpha) < 0$, **then** $\nabla_i^{proj} g(\alpha) \leftarrow \nabla_i g(\alpha)$
      end if
    else if $\alpha_i = C_2 s_{i2}$ **then**
      if $\nabla_i^{proj} g(\alpha) < \overline{n}$, **then** $A = A \setminus \{i\}$ **end if**
      if $\nabla_i^{proj} g(\alpha) > 0$, **then** $\nabla_i^{proj} g(\alpha) \leftarrow \nabla_i g(\alpha)$
      end if
    else
      $\nabla_i^{proj} g(\alpha) \leftarrow \nabla_i g(\alpha)$
    end if
    $N \leftarrow max\left(N, \nabla_i^{proj} g(\alpha)\right)$
    $n \leftarrow min\left(n, \nabla_i^{proj} g(\alpha)\right)$
    if $\nabla_i^{proj} g(\alpha) \neq 0$ **then**
      $\overline{\alpha}_i \leftarrow \alpha_i$
      $\alpha_i \leftarrow min(max(\alpha_i - \nabla_i g(\alpha)/\overline{W}_{ii}, 0), C_2 s_{i2})$
      $u_{1i} \leftarrow u_{1i} - W_i(\alpha_i - \overline{\alpha}_i)$
    end if
  end for
  if $N - n < \epsilon$ **then**
    if $A = \{1, \ldots, l_2\}$, **break**
  else
    $A \leftarrow \{1, \ldots, l_2\}$, $\overline{N} \leftarrow \infty, \overline{n} \leftarrow -\infty$
    if $N \leq 0$, **then** $\overline{N} \leftarrow \infty$. **else** $\overline{N} \leftarrow N$ **end if**
    if $N \geqslant 0$, **then** $\overline{n} \leftarrow -\infty$. **else** $\overline{n} \leftarrow n$ **end if**
  end if
end while

4. Complexity analysis of the CIL-FART-IFTSVM

In this section, the big-O notation [47] is used for the analysis of on-time complexity of the CIL-FART-IFTSVM. The analysis is divided into two parts: Fuzzy ART and IFTSVM. Let M and N denote the number of input features and number of prototype nodes, respectively. As reported in [48], Fuzzy ART records the worst-case time complexity of $N^2 + MN$, which is asymptotically equivalent to $O\left(N^2\right)$, when $N \to \infty$ and $M \to \infty$.

In IFTSVM, let n be the total number of training samples and $m = n/2$ be the number of samples in each class. The IFTSVM measures the degrees of membership and nonmembership functions to compute the score value of each sample. Therefore, the time complexity of the IFTSVM [12] is $O\left(2 \times (n/2)^3\right)$.

Therefore, the worst-case time complexity of CIL-FART-IFTSVM can be determined by combining the time complexity of Fuzzy ART $O\left(N^2\right)$ and IFTSVM $O\left(2 \times (n/2)^3\right)$, which is $O\left(N^2\right) + O\left(2 \times (n/2)^3\right)$ when all variables extend to infinity.

5. Experiment and results

To validate the efficacy and generalizability of CIL-FART-IFTSVM, we conducted experiments on 44 imbalanced datasets with an imbalance ratio (IR) from 1 to 129.44 from the KEEL imbalanced datasets [49] and UCI machine learning repository [50] for binary classification. Additionally, one real dataset [51] is used to detect credit card fraud. Table 1 provides details of the imbalanced datasets. For all datasets, 5-fold cross-validation is used. The bootstrap technique [52] with 95% confidence intervals is utilized to statistically quantify the results. A significant advantage of bootstrap is its simplicity. It is a straightforward way to derive estimates of standard errors and confidence intervals for complex estimators of the distribution, such as percentile points, proportions, odds ratio, and correlation coefficients. Bootstrap is also an appropriate way to control and check the stability of the results. Although for most problems, it is impossible to know the true confidence interval, bootstrap is asymptotically more accurate than the standard intervals obtained using sample variance and assumptions of normality. Bootstrapping is also a convenient method that avoids the cost of repeating the experiment to get other groups of sample data. In this technique, Bootstrap is set to repeat 5000 times our results to find the mean and standard deviations.

The CIL-FART-IFTSVM parameters are fixed as follows: c_i $(i = 1, 2, 3, 4)$ are properly investigated in the grids $\left\{2^i | i = -10, -9, \ldots, 9, 10\right\}$ by setting $C_1 = C_3, C_2 = C_4$. Moreover, a Gaussian kernel function is used in the nonlinear cases, i.e., $\mathcal{K}(x_1, x_2) = exp\left(-\|x_1 - x_2\|^2/\sigma^2\right)$ and $\sigma \in \left\{2^{\sigma_{min}:\sigma_{max}}\right\}$ with $\sigma_{min} = -10, \sigma_{max} = 10$. All the experiments are implemented in a MATLAB 2018a environment on a PC with an Intel(R) Core i5 processor (3.30 GHz) and 12 GB RAM.

The geometric mean of sensitivity and specificity (G-Mean) [23] is used to validate the classification performance of the algorithms on imbalanced datasets. A larger G-Mean indicates better performance. We performed the Friedman test [53] to statistically compare 14 algorithms on 40 datasets. The experiments were performed on imbalanced datasets with SVM [1], least squares SVM (LSSVM) [54], FSVM [6], TSVM [3], FTSVM [55], coordinate descent FTSVM (CDFTSVM) [8], entropy-based FSVM (EFSVM) [21], entropy-based FTSVM for CIL (EFTWSVM-CIL) [20], SVM with synthetic minority oversampling technique (SVM-SMOTE) [56], tree-based AdaBoost Algorithm (AdaBoost) [57], SVM with one-sided selection (SVM-OSS) [58], EasyEnsemble [59], SVM with random undersampling (SVM-RUS) [60], and the NN classifier (1-NN) [61].

5.1. Statistical comparison results

The Friedman test [53] was performed to compare the performance of the proposed CIL-FART-IFTSVM method and other existing CIL methods. In the Friedman test, we rank the algorithms applied to each dataset individually: the algorithm with

Table 1
Details of the Imbalanced Datasets.

Dataset	Positive	Negative	Instance	Dimension	Im. Ratio	Dataset	Positive	Negative	Instance	Dimension	Im. Ratio
Ripley	650	650	1,250	3	1	Ecoli 0-1-vs-2-3-5	24	220	244	8	9.17
Cleveland	160	137	297	14	1.17	Yeast 0-5-6-7-9-vs-4	51	477	528	9	9.35
EEG Eye State	6723	8257	14,980	15	1.23	Connect-4	6449	61,108	67,557	43	9.48
Australian	383	307	690	14	1.25	Vowel	90	898	988	11	9.98
CMC	629	844	1,473	10	1.34	Ecoli 0-6-7-vs-5	20	200	220	7	10
Ionosphere	126	225	351	34	1.79	Led7digit 0-2-4-5-6-7-8-9-vs-1	37	406	443	8	10.97
Wisconsin	239	444	683	10	1.86	Ecoli 0-1-vs-5	20	220	240	7	11
Pima	268	500	768	9	1.87	Ecoli 0-1-4-7-vs-5-6	25	307	332	7	12.28
Yeast 1	429	1,055	1,484	9	2.64	Ecoli 0-1-4-6-vs-5	20	260	280	7	13
Vehicle 2	218	628	864	19	2.88	Shuttle c0-vs-c4	123	1,706	1,829	10	13.87
Vehicle 1	217	629	864	19	2.90	Glass 4	201	13	214	10	15.46
Adult	34,014	11,208	45,222	15	3.03	Ecoli 4	20	316	336	8	15.80
Transfusion	178	570	748	5	3.20	Yeast 1-4-5-8-Vs-7	30	663	693	8	22.10
Wpbc	47	151	198	34	3.22	Glass 5	9	205	214	10	22.78
Segment	329	1979	2,308	19	6.02	Yeast 2-Vs-8	20	462	482	9	23.10
Yeast 3	163	1,321	1,484	9	8.10	Covertype	20,510	560,502	581,012	55	27.33
Page blocks	560	4,913	5,473	11	8.78	Yeast 4	51	1,433	1,484	9	28.10
Yeast 2-vs-4	463	51	514	9	9.08	Yeast 1-2-8-9-Vs-7	30	917	947	9	30.57
Ecoli 0-2-3-4-vs-5	20	182	202	8	9.10	Yeast 5	44	1,440	1,484	8	32.73
Yeast 0-3-5-9-vs-7-8	50	456	506	9	9.12	Ecoli 0-1-3-7-vs-2-6	7	274	281	8	39.14
Yeast 0-2-5-6-vs-3-7-8-9	99	905	1,004	9	9.14	Yeast 6	35	1,449	1,484	9	41.40
Ecoli 0-4-6-vs-5	20	183	203	8	9.15	Abalone 19	32	4,142	4,174	8	129.44

the best performance is ranked first, next best is ranked 2, etc. Suppose that n is the number of imbalanced datasets, k is the number of algorithms, and r_i^j is the rank of the j^{th} algorithm on the i^{th} dataset. This test evaluates the average rank of algorithms $R_j = \frac{1}{n}\sum_{i=1}^{n} r_i^j$. Under the null hypothesis, the ranks (R_j) are equal, i.e., all the algorithms are equivalent. The Friedman test

$$\chi_F^2 = \frac{12n}{k(k+1)}\left[\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{32}$$

is based on a χ_F^2 distribution with $(k-1)$ and $(k-1)(n-1)$ degrees of freedom when n and k are sufficiently large. In [62], the authors showed that the Friedman χ_F^2 produced a better statistic with pessimistic behavior

$$F_F = \frac{(n-1)\chi_F^2}{n(k-1)-\chi_F^2} \tag{33}$$

5.2. Imbalanced datasets

Our experiments with imbalanced datasets are organized as follows:

5.2.1. Parameter effect

The influence of various σ (kernel parameter) and C (trade-off) is studied using the Vowel dataset. The purpose is to identify the optimal parameters, i.e., C for linear kernel functions and C and σ for nonlinear kernel functions, that produce the best accuracy. First, C, which varies in $[-10, 10]$, is optimized for the linear kernel function. Clearly, for $C > 0$ (Fig. 5), CIL-FART-IFTSVM generates better results, especially when $C = 1$. Next, for nonlinear kernel functions, C and σ are optimized. C and σ can vary in $[-10, 10]$. CIL-FART-IFTSVM with $C = 1$ and $\sigma = -1$ performs best (Fig. 6).

In this paper, we empirically set the vigilance value (ρ) more than 0.7 for small datasets and less than 0.3 for large datasets. For large data sets (small data sets, respectively), the values of $\{0.1, 0.2, 0.25, 0.3\}$ ($\{0.7, 0.8, 0.85, 0.9\}$, respectively) are chosen. Then, we randomly repeat the data sets ten times for each vigilance value individually to discover which one has the best performance. Finally, the best value is chosen for cross-validation. Twenty small and four large data sets are considered to validate the best vigilance value. Fig. 7 and 8 show the best vigilance values for the small and large data sets, respectively.

5.2.2. In the second experiment, we focus on the classification performance of CIL-FART-IFTSVM, TSVM, FTSVM, EFSVM, and EFTWSVM-CIL to study the robustness of CIL-FART-IFTSVM to noise and outliers.

Table 2 shows the G-Mean for a linear kernel function along with the standard deviation (SD). The proposed method has the best performance on all datasets except for Yeast 0-5-6-7-9-vs-4, Led7digit 0-2-4-5-6-7-8-9-vs-1, Ecoli 0-1-4-6-vs-5, and

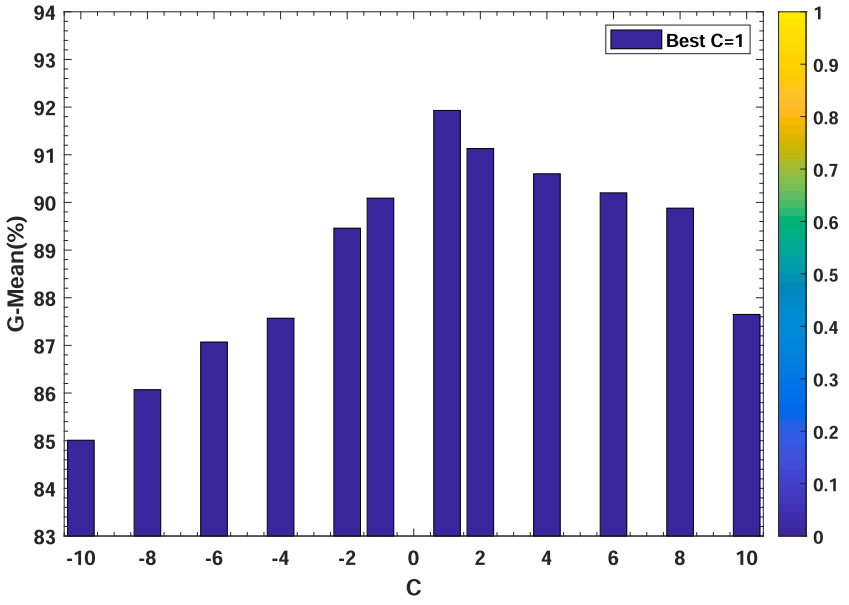


Fig. 5. G-Mean (%) of CIL-FART-IFTSVM with linear kernel for Vowel dataset.

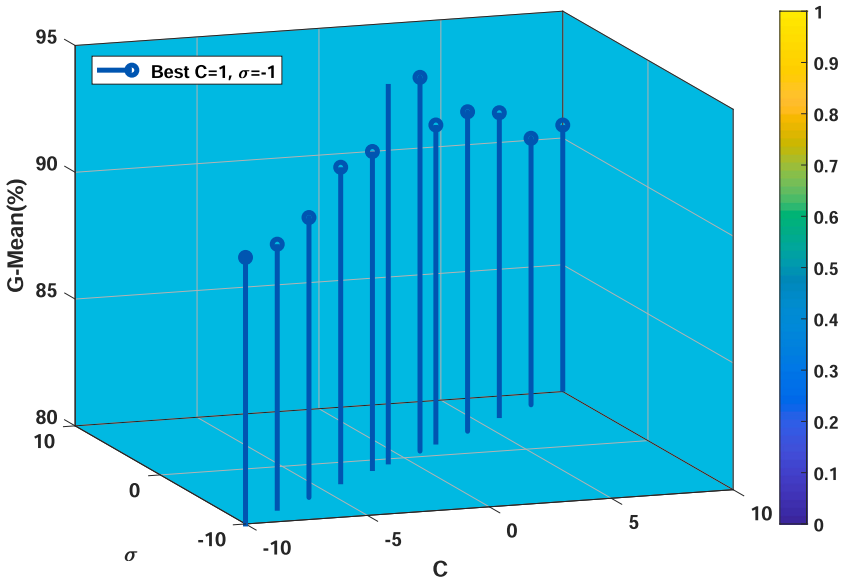


Fig. 6. G-Mean (%) of CIL-FART-IFTSVM with nonlinear kernel for Vowel dataset.

Yeast 5. However, the EFTWSVM-CIL technique achieves the highest G-Mean rate for Yeast 0-5-6-7-9-vs-4 and Yeast 5. FTWSVM technique achieves the highest G-Mean rate for Ecoli 0-1-4-6-vs-5.

In some datasets like Pima or Transfusion, we have an improvement of the G-Mean outcomes presented by five techniques: i) From 66.74% to 76.05% applying the CIL-FART-IFTSVM in Pima which is 1% better than the second best method (EFTWSVM-CIL), and ii) From 56.90% to 76.14% using the CIL-FART-IFTSVM in Transfusion which is 25% better than the second best method (EFTWSVM-CIL).

From Table 2, the Friedman test statistic is calculated for a linear kernel function under the null hypothesis when $n = 18$ and $k = 5$:

$$\chi_F^2 = \frac{12 \times 18}{5 \times (5+1)} \left[\left(1.47^2 + 2.53^2 + 4.25^2 + 3.11^2 + 3.64^2\right) - \frac{5 \times (5+1)^2}{4} \right] = 32.73$$

and

$$F_F = \frac{(18-1) \times 32.73}{18 \times (5-1) - 32.73} = \frac{606.05}{36.35} = 14.17$$

F_F is obtained from the F-distribution with $(k-1) = (5-1) = 4$ and $(k-1)(n-1) = (5-1)(18-1) = 68$ degrees of freedom. The critical value of $F(4, 68)$ at $\alpha = 0.05$ is 2.51; thus, we reject the null hypothesis and state that the compared algorithms are not equivalent at $\alpha = 0.05$, i.e., $F_F = 14.17 > 2.51$. Table 2 shows differences in average ranks of the compared algorithms and the average rank of the CIL-FART-IFTSVM for the 18 imbalanced datasets with respect to the G-Mean. At this time, we applied the Bonferroni-Dunn test [63] to compare CIL-FART-IFTSVM with the other imbalanced learning algorithms. We computed the critical difference (CD), which is defined in Eq. (15):

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6n}}. \tag{34}$$

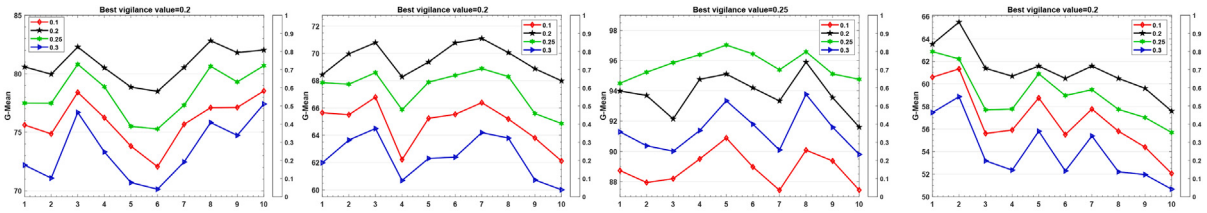
where q_α (critical values) are taken from [64]. We can calculate the critical difference, $CD = 2.498\sqrt{\frac{5(5+1)}{6\times 18}} = 1.32$. The difference between the average ranks of CIL-FART-IFTSVM and EFSVM (FTWSVM and TWSVM, respectively) is $4.25 - 1.47 = 2.78$ ($3.11 - 1.47 = 1.64$ and $3.64 - 1.472 = 2.17$, respectively). The differences are greater than 1.32; therefore, there is a significant difference between CIL-FART-IFTSVM and EFSVM (FTWSVM and TWSVM, respectively). However, the difference between the average ranks of CIL-FART-IFTSVM and EFTWSVM-CIL is less than 1.32, i.e., $2.53 - 1.47 = 1.06 < 1.32$. Thus, we cannot conclude that CIL-FART-IFTSVM is significantly different from EFTWSVM-CIL. However, for almost all the datasets, CIL-FART-IFTSVM performed better than EFTWSVM-CIL (see the results and ranks in Table 2).

Table 3 shows the G-Mean for a non-linear kernel function along with the standard deviation (SD). One can observe that the proposed technique almost achieves the highest G-Mean rate. However, the EFTWSVM-CIL technique achieves the highest G-Mean rate for Yeast 2-vs-4 and Ecoli 4 datasets. Therefore, it is obvious to conclude that CIL-FART-IFTSVM has better generalization capability than other learning techniques.

Fig. 7. Best vigilance values for selected small data sets.

In some datasets like Yeast 2-vs-4 or Yeast 5, we have an improvement of the G-Mean outcomes presented by five techniques: i) From 86.51% to 88.97% applying the EFTWSVM-CIL in Yeast 2-vs-4 which is 1% better than the second best method (CIL-FART-IFTSVM), and ii) From 72.82% to 94.36% using the CIL-FART-IFTSVM in Yeast 5 which is 5% better than the second best method (EFTWSVM-CIL).

From Table 3, the Friedman statistic test is calculated for nonlinear kernel function under the null hypothesis when $n = 22$ and $k = 5$:



(a) Adult data set (b) Connect-4 data set (c) Covertype data set (d) EEG Eye State data set

Fig. 8. Best vigilance values for selected large data sets.

Table 2
G-Mean (%) and ranks for imbalanced datasets with a linear kernel function with standard deviation (SD).

dataset	Im. Ratio	TWSVM	FTWSVM	EFSVM	EFTWSVM-CIL	CIL-FART-IFTSVM
Ripley	1	85.68	85.87	80.99	86.08	**88.48±1.31**
Wisconsin	1.86	93.31	94.20	95.06	93.79	**98.32±0.87**
Pima	1.87	73.02	72.85	66.74	75.25	**76.05±2.97**
Vehicle 2	2.88	89.69	87.60	86.29	89.92	**93.70±2.05**
Vehicle 1	2.90	65.99	67.75	55.16	67.37	**74.14±3.05**
Transfusion	3.20	56.90	56.95	56.95	60.82	**76.14±3.62**
Wpbc	3.22	62.45	61.77	66.98	64.39	**69.71±1.42**
Yeast 2-vs-4	9.08	84.59	86.49	78.45	84.59	**88.21±3.38**
Ecoli 0-2-3-4-vs-5	9.10	94.68	96.19	91.61	96.32	**97.07±4.48**
Ecoli 0-1-vs-2-3-5	9.17	91.98	91.95	72.85	91.95	**94.59±7.45**
Yeast 0-5-6-7-9-vs-4	9.35	75.84	78.45	71.66	**79.29**	78.32±3.03
Vowel	9.98	87.41	86.41	79.56	87.41	**91.08±6.68**
Led7digit 0-2-4-5-6-7-8-9-vs-1	10.97	89.29	87.92	**91.89**	89.50	89.66±7.57
Ecoli 0-1-vs-5	11	**94.31**	**94.31**	86.84	**94.31**	**94.31±6.67**
Ecoli 0-1-4-6-vs-5	13	92.80	**92.96**	83.22	92.85	92.20±6.42
Shuttle c0-vs-c4	13.87	99.96	99.98	99.32	99.98	**100±00**
Yeast 5	32.73	73.09	73.94	86.37	**96.94**	94.69±5.42
Ecoli 0-1-3-7-vs-2-6	39.14	98.09	98.16	89.05	98.01	**98.53±0.90**
Ave. Rank	–	3.64	3.11	4.25	2.53	**1.47**
Differences	–	2.17	1.64	2.78	1.06	N/A

Table 3
G-Mean (%) and ranks for imbalanced datasets with a nonlinear kernel function with standard deviation (SD).

dataset	Im. Ratio	TWSVM	FTWSVM	EFSVM	EFTWSVM-CIL	CIL-FART-IFTSVM
Cleveland	1.17	82.19	81.93	83.85	83.67	**84.42±7.88**
Australian	1.25	83.86	85.64	83.32	85.91	**86.87±3.14**
CMC	1.34	64.17	63.73	62.86	64.01	**66.80±2.41**
Ionosphere	1.79	89.67	90.03	82.07	89.68	**95.26±3.12**
Pima	1.87	70.80	71.48	73.91	73.55	**74.73±1.89**
Vehicle 2	2.88	93.51	93.29	93.12	92.96	**95.24±1.40**
Transfusion	3.20	63.09	64.76	62.69	64.87	**75.31±4.63**
Wpbc	3.22	62.00	62.01	59.89	63.84	**69.07±3.87**
Yeast 2-vs-4	9.08	86.88	86.51	86.92	**88.97**	88.30±3.46
Ecoli 0-2-3-4-vs-5	9.10	92.71	92.10	93.06	92.71	**98.77±1.18**
Yeast 0-3-5-9-vs-7-8	9.12	69.13	68.12	70.67	71.43	**77.33±7.01**
Yeast 0-2-5-6-vs-3-7-8-9	9.14	71.59	75.53	73.17	75.53	**84.76±10.61**
Ecoli 0-4-6-vs-5	9.15	89.01	89.07	87.79	89.07	**96.75±4.38**
Ecoli 0-1-vs-2-3-5	9.17	93.53	92.68	91.45	92.73	**96.79±4.36**
Yeast 0-5-6-7-9-vs-4	9.35	72.16	70.99	75.41	74.51	**80.66±4.04**
Ecoli 0-6-7-vs-5	10	88.20	88.23	88.42	88.22	**90.67±6.65**
Ecoli 0-1-vs-5	11	90.02	90.00	84.93	90.02	**92.67±7.46**
Ecoli 0-1-4-7-vs-5-6	12.28	93.01	93.27	93.27	93.27	**94.21±5.17**
Shuttle c0-vs-c4	13.87	99.38	99.58	99.43	99.62	**100±0.00**
Glass 4	15.46	86.02	86.61	85.79	87.57	**90.84±8.93**
Ecoli 4	15.80	95.42	95.51	93.89	**95.68**	94.43±6.61
Yeast 5	32.73	80.83	89.16	72.82	89.33	**94.36±3.69**
Ave. Rank	–	3.82	3.50	3.91	2.59	**1.18**
Differences	–	2.64	2.32	2.73	1.41	N/A

$$\chi_F^2 = \frac{12 \times 22}{5 \times (5+1)} \left[\left(1.18^2 + 2.59^2 + 3.91^2 + 3.50^2 + 3.82^2 \right) - \frac{5 \times (5+1)^2}{4} \right] = 46.03$$

and

$$F_F = \frac{(22-1) \times 46.03}{22 \times (5-1) - 46.03} = \frac{1034.04}{38.76} = 23.03$$

F_F is obtained from the F-distribution with $(k-1) = (5-1) = 4$ and $(k-1)(n-1) = (5-1)(22-1) = 84$ degrees of freedom. The critical value of $F(4, 84)$ at $\alpha = 0.05$ is 2.48. We can reject the null hypothesis because the compared algorithms are not equivalent at $\alpha = 0.05$, i.e., $F_F = 23.03 > 2.48$. Table 3 shows the differences between the average ranks of the compared algorithms and the average rank of CIL-FART-IFTSVM for the 22 imbalanced datasets with respect to the G-Mean. At this time, we applied the Bonferroni-Dunn test to compare the CIL-FART-IFTSVM with the other imbalanced learning algorithms in this experiment. We can calculate the critical difference, $CD = 2.498\sqrt{\frac{5(5+1)}{6\times 22}} = 1.19$. The difference between the average ranks of CIL-FART-IFTSVM and EFTWSVM-CIL (EFSVM, FTWSVM, and TWSVM, respectively) is $2.59 - 1.18 = 1.41$ ($3.91 - 1.18 = 2.73, 3.50 - 1.18 = 2.32$, and $3.82 - 1.18 = 2.64$, respectively). The differences are greater than 1.19; thus, CIL-FART-IFTSVM is significantly better than all the other methods in this experiment.

5.2.3. In the third experiment, we evaluate the performance of the proposed CIL-FART-IFTSVM on low and medium imbalance datasets by comparing with CIL-FART-IFTSVM, EasyEnsemble, SVM-RUS, FSVM, SVM-SMOTE, SVM-OSS, SVM, 1-NN, and AdaBoost.

Table 4 shows the G-Mean rate for linear and non-linear kernel functions along with the standard deviation (SD) on 8 low imbalance datasets. The CIL-FART-IFTSVM technique is better in most linear and nonlinear cases. The linear case of CIL-FART-IFTSVM technique is better for Wisconsin, Pima, and Yeast 3 datasets, while the nonlinear CIL-FART-IFTSVM technique achieves the highest G-Mean rate for Wisconsin, Pima, Vehicle 2, Vehicle 1, Segment, and Yeast 3 datasets. However, the EasyEnsemble technique achieves the highest G-Mean rate for Yeast 1 and Page blocks datasets.

In some datasets like Yeast 1, Wisconsin, or Yeast 3, we have an enhancement of the G-Mean outcomes presented by nine techniques: i) From 64.39% to 73.40% applying the EasyEnsemble in Yeast 1 which is 1% better than the second best method (CIL-FART-IFTSVM$_{non-Linear}$), ii) From 92.30% to 98.32% using the CIL-FART-IFTSVM$_{non-Linear}$ in Wisconsin which is 2% better than the second best method (FSVM), and iii) From 78.29% to 93.26% applying the CIL-FART-IFTSVM$_{non-Linear}$ in Yeast 3 which is 3% better than the second best method (EasyEnsemble).

From Table 4 (Ave. Rank), the Friedman statistic is calculated on low imbalance datasets for the nonlinear kernel under the null hypothesis when $n = 8$ and $k = 9$:

$$\chi_F^2 = \frac{12 \times 8}{9 \times (9+1)} \left[\left(1.25^2 + 7.12^2 + 5.25^2 + 2.25^2 + 6.12^2 + 5.50^2 + 7.12^2 + 4.75^2 + 5.62^2 \right) - \frac{9 \times (9+1)^2}{4} \right]$$

and

$$F_F = \frac{(8-1) \times 34.70}{8 \times (9-1) - 34.70} = \frac{242.9}{29.3} = 8.29$$

F_F is obtained from the F-distribution with $(k-1) = (9-1) = 8$ and $(k-1)(n-1) = (9-1)(8-1) = 56$ degrees of freedom. The critical value of $F(8, 56)$ at $\alpha = 0.05$ is 2.11. Thus, we can reject the null hypothesis that the compared algorithms are equivalent at $\alpha = 0.05$, i.e., $F_F = 8.29 > 2.11$. Table 4 shows the differences between the average ranks of the compared algorithms and the average rank of CIL-FART-IFTSVM for the 8 imbalanced datasets with respect to the G-Mean. We then applied the Bonferroni-Dunn test to compare CIL-FART-IFTSVM with the other imbalanced learning algorithms. The critical difference $CD = 2.72\sqrt{\frac{9(9+1)}{6\times 8}} = 3.72$. The difference between the average ranks of CIL-FART-IFTSVM and FSVM (SVM-OSS, SVM-RUS, SVM, AdaBoost, and 1-NN, respectively) is $5.62 - 1.25 = 4.37$ ($7.12 - 1.25 = 5.87, 5.50 - 1.25 = 4.25, 6.12 - 1.25 = 4.87, 5.25 - 1.25 = 4$, and $7.12 - 1.25 = 5.87$, respectively). The differences are greater than 3.72; therefore, there is a significant difference between CIL-FART-IFTSVM and FSVM (SVM-OSS, SVM-RUS, SVM, AdaBoost, and 1-NN, respectively). However, the difference between the average ranks of CIL-FART-IFTSVM and SVM-SMOTE (EasyEnsemble, respectively) is $4.75 - 1.25 = 3.5$ ($2.25 - 1.25 = 1$, respectively), which is less than 3.72. Thus, there is no significant difference between CIL-FART-IFTSVM and SVM-SMOTE (EasyEnsemble, respectively). Still, for almost all the datasets, CIL-FART-IFTSVM performed better than SVM-SMOTE and EasyEnsemble (see the results in Table 4).

Table 5 shows the G-Mean rate for linear and non-linear kernel functions along with the standard deviation (SD) on 25 medium imbalance datasets. The proposed technique is better in most linear and nonlinear cases. The linear CIL-FART-IFTSVM is better for Yeast 0-3-5-9-vs-7-8, Yeast 0-2-5-6-vs-3-7-8-9, Ecoli 0-4-6-vs-5, Ecoli 0-1-vs-2-3-5, Ecoli 0-6-7-vs-5, Led7digit 0-2-4-5-6-7-8-9-vs-1, Ecoli 0-1-vs-5, Shuttle c0-vs-c4, Ecoli 4, Glass 5, Yeast 5, and Ecoli 0-1-3-7-vs-2-6, while

Table 4
Ranks (only for nonlinear kernel) and G-MEAN (%) of the compared algorithms on low imbalanced datasets (Im. Ratio $\leqslant$ 9.0).

dataset	Im.Ratio	FSVM	SVM-SMOTE	SVM-OSS	SVM-RUS	SVM	EasyEnsemble	AdaBoost	1-NN	CIL-FART-IFTSVM Linear Non-linear
Wisconsin	1.86	96.28±3.86	94.59±2.30	94.56±1.55	94.31±3.54	94.76±1.45	94.53±2.17	92.53±5.39	92.30±3.00	**98.32±0.87 98.21±1.17**
Pima	1.87	71.23±3.33	72.44±5.51	70.66±6.28	72.59±1.61	71.00±5.30	74.36±2.65	72.22±3.99	65.59±3.46	**76.05±2.97 74.73±1.89**
Yeast 1	2.64	69.51±3.44	70.15±2.90	68.52±2.61	69.65±3.93	71.80±3.00	**73.40±4.80**	67.36±4.15	64.39±2.89	70.56±1.43 72.69±1.73
Vehicle 2	2.88	87.42±1.74	89.18±1.53	82.77±1.93	83.86±1.95	91.38±2.25	95.20±0.48	94.82±0.33	92.23±0.84	93.70±2.05 **95.24±1.40**
Vehicle 1	2.90	67.58±3.82	71.52±6.37	72.06±6.78	73.21±5.40	69.35±5.16	75.32±5.58	72.19±4.82	63.46±2.29	74.14±3.05 **75.37±3.80**
Segment	6.02	94.59±0.81	95.17±0.74	87.96±1.01	93.11±0.93	90.47±0.86	99.11±0.09	99.08±0.22	99.01±0.42	98.47±0.28 **99.18±0.08**
Yeast 3	8.10	87.51±3.14	89.83±1.69	87.09±2.26	87.55±2.13	86.74±5.67	90.38±3.45	84.98±2.39	78.29±2.44	**91.11±1.46 93.26±1.25**
Page blocks	8.78	79.98±3.95	78.24±7.17	73.28±3.00	78.15±2.82	73.18±6.44	**93.54±0.67**	90.37±0.57	89.84±1.58	90.42±1.66 91.55±1.55
Ave. Rank	–	5.62	4.75	7.12	5.50	6.12	2.25	5.25	7.12	**1.25**
Difference	–	4.37	3.50	5.87	4.25	4.87	1	4	5.87	N/A

Table 5
Ranks (only for nonlinear kernel) and G-Mean (%) of the compared algorithms on medium imbalanced datasets.

dataset	Im. Ratio	FSVM	SVM-SMOTE	SVM-OSS	SVM-RUS	SVM	EasyEnsemble	AdaBoost	1-NN	CIL-FART-IFTSVM Linear Non-linear
Yeast 2-vs-4	9.08	85.73±8.19	87.37±2.43	88.18±7.38	87.20±7.08	84.37±7.87	**91.46±4.29**	82.96±8.55	84.55±11.54	88.21±3.38 88.30±3.46
Ecoli 0-2-3-4-vs-5	9.10	95.36±4.51	95.53±4.04	96.24±4.16	97.93±2.12	96.29±5.26	95.64±4.26	95.42±5.03	91.21±5.02	97.07±4.48 **98.77±1.18**
Yeast 0-3-5-9-vs-7-8	9.12	68.01±1.42	72.80±0.91	65.62±1.25	71.49±0.88	67.57±1.16	74.71±1.03	65.47±0.94	67.57±1.13	**75.54±8.82 77.33±7.01**
Yeast 0-2-5-6-vs-3-7-8-9	9.14	76.48±8.80	79.70±6.21	79.49±5.16	77.74±5.65	75.45±5.58	78.75±5.89	70.85±4.04	75.47±3.55	**85.76±10.38 84.76±10.61**
Ecoli 0-4-6-vs-5	9.15	89.32±8.57	88.13±7.97	87.41±9.02	86.75±7.00	89.12±5.99	85.63±5.13	83.83±11.18	83.80±10.98	**96.60±0.48 96.75±4.38**
Ecoli 0-1-vs-2-3-5	9.17	91.90±8.88	92.74±11.16	91.85±12.25	91.93±10.19	92.06±12.33	91.41±9.40	83.42±9.54	83.75±10.12	**94.59±7.45 96.79±4.36**
Yeast 0-5-6-7-9-vs-4	9.35	78.39±8.09	79.61±5.53	79.44±7.31	80.13±5.31	77.06±7.63	78.12±6.55	66.87±7.28	69.49±7.70	78.32±3.03 **80.66±4.04**
Vowel	9.98	86.28±6.02	90.23±6.70	87.75±10.12	89.20±6.70	91.29±4.33	95.55±4.39	93.51±9.44	**99.83±0.20**	91.08±6.68 94.02±1.37
Ecoli 0-6-7-vs-5	10	88.13±6.40	88.00±5.50	87.89±6.44	86.70±5.31	86.55±4.86	84.22±5.64	76.07±4.30	83.14±4.73	**90.29±6.60 90.67±6.65**
Led7digit 0-2-4-5-6-7-8-9-vs-1	10.97	87.58±8.62	87.71±11.95	85.12±12.95	86.23±10.42	86.56±6.12	84.38±13.21	83.58±11.02	60.22±13.67	**89.66±7.57 89.03±7.46**
Ecoli 0-1-vs-5	11	90.48±3.95	92.27±3.87	90.41±4.58	90.65±2.81	89.71±3.85	86.00±6.97	83.47±8.65	86.71±7.79	**94.31±6.67 93.67±7.46**
Ecoli 0-1-4-7-vs-5-6	12.28	92.84±2.03	**95.30±2.54**	93.48±3.70	93.19±2.16	92.11±5.37	93.79±1.54	91.75±6.40	91.82±5.06	94.17±5.54 94.21±5.17
Ecoli 0-1-4-6-vs-5	13	91.90±1.48	92.04±1.53	**93.08±0.69**	91.00±1.51	91.09±1.53	90.15±1.02	79.35±2.18	87.86±1.59	92.20±6.42 91.74±5.52
Shuttle c0-vs-c4	13.87	99.86±0.11	99.89±0.05	99.86±0.7	99.87±0.05	99.81±0.07	99.93±0.04	81.09±0.41	99.79±0.65	**100±0.00 100±0.00**
Glass 4	15.46	85.58±9.96	88.60±9.42	86.39±11.18	89.53±3.34	**90.84±2.22**	83.84±10.99	84.96±9.69	90.19±8.49	85.65±11.83 **90.84±8.93**
Ecoli 4	15.80	93.05±5.11	94.30±4.70	93.44±5.96	94.23±3.97	94.43±1.93	90.17±4.83	84.03±13.16	88.40±4.22	**95.64±2.35** 94.43±6.61
Yeast 1-4-5-8-Vs-7	22.10	61.30±6.81	70.82±12.57	66.57±8.11	69.03±12.86	64.74±15.66	68.89±9.22	54.76±0.29	60.06±10.36	65.86±6.37 **72.23±4.70**
Glass 5	22.78	73.43±10.28	76.55±10.93	67.28±14.50	73.98±3.72	65.56±16.70	72.39±12.68	66.58±13.19	70.95±13.09	**81.22±8.07 81.44±6.03**
Yeast 2-Vs-8	23.10	71.94±6.61	72.96±6.41	70.73±4.49	73.63±5.09	74.96±7.93	75.54±6.13	70.06±4.89	69.99±4.78	72.40±14.44 **75.84±10.00**
Yeast 4	28.10	80.80±5.76	**85.25±1.11**	82.65±3.64	84.70±3.09	81.88±5.58	82.73±3.58	59.49±5.52	67.69±7.57	82.78±5.15 81.89±4.25
Yeast 1-2-8-9-Vs-7	30.57	66.22±4.54	74.11±4.54	66.84±6.07	**74.61±3.71**	63.89±11.44	72.36±9.58	60.73±6.83	56.82±3.66	72.86±4.85 **74.61±3.71**
Yeast 5	32.73	90.42±2.21	91.14±1.18	90.38±2.14	90.91±2.33	90.61±1.79	90.56±2.46	81.60±11.63	79.82±5.27	**94.69±5.42 94.36±3.69**
Ecoli 0-1-3-7-vs-2-6	39.14	98.24±3.42	98.29±3.44	94.15±3.07	93.73±3.18	95.97±1.66	84.98±2.89	72.21±3.40	93.94±3.06	**98.53±0.90 99.24±0.33**
Yeast 6	41.40	89.57±4.20	92.50±2.66	88.30±3.17	90.61±2.94	90.47±3.64	86.36±2.60	74.59±5.59	79.83±4.87	88.90±2.69 **92.72±4.27**
Abalone 19	129.44	54.74±2.54	65.25±1.85	53.64±2.04	66.45±1.56	49.77±4.87	**70.49±3.28**	49.83±0.24	51.48±0.80	65.87±4.25 66.28±2.38
Ave. Rank	–	5.26	2.96	5.18	3.98	5.34	4.92	8.40	7.38	**1.58**
Difference	–	3.68	1.38	3.60	2.40	3.76	3.34	6.82	5.80	N/A

the nonlinear CIL-FART-IFTSVM technique achieves the highest G-Mean rate for Ecoli 0-2-3-4-vs-5, Yeast 0-3-5-9-vs-7-8, Yeast 0-2-5-6-vs-3-7-8-9, Ecoli 0-4-6-vs-5, Ecoli 0-1-vs-2-3-5, Yeast 0-5-6-7-9-vs-4, Ecoli 0-6-7-vs-5, Led7digit 0-2-4-5-6-7-8-9-vs-1, Ecoli 0-1-vs-5, Shuttle c0-vs-c4, Glass 4, Yeast 1-4-5-8-Vs-7, Glass 5, Yeast 2-Vs-8, Yeast 1-2-8-9-Vs-7, Yeast 5, Ecoli 0-1-3-7-vs-2-6, and Yeast 6. However, EasyEnsemble (1-NN, SVM-SMOTE, SVM-OSS, and SVM, respectively) technique achieves the highest G-Mean rate for Yeast 2-vs-4 (Vowel, Ecoli 0-1-4-7-vs-5-6, Ecoli 0-1-4-6-vs-5, and Glass 4, respectively).

In some datasets like Yeast 0-2-5-6-vs-3-7-8-9, Ecoli 0-1-4-7-vs-5-6, or Led7digit 0-2-4-5-6-7-8-9-vs-1, we have an improvement of the G-Mean outcomes presented by nine techniques: i) From 70.85% to 85.76% applying the CIL-FART-IFTSVM$_{non-Linear}$ in Yeast 0-2-5-6-vs-3-7-8-9 which is 7% better than the second best method (SVM-SMOTE), ii) From 91.75% to 95.30% using the SVM-SMOTE in Ecoli 0-1-4-7-vs-5-6 which is 1% better than the second best method (CIL-FART-IFTSVM$_{non-Linear}$), and iii) From 60.22% to 89.66% applying the CIL-FART-IFTSVM$_{non-Linear}$ in Led7digit 0-2-4-5-6-7-8-9-vs-1 which is 2% better than the second best method (SVM-SMOTE).

From Table 5 (Ave. Rank), the Friedman statistic is calculated on medium imbalance datasets for the nonlinear kernel under the null hypothesis when $n = 25$ and $k = 9$:

$$\chi_F^2 = \frac{12 \times 25}{9 \times (9+1)} \left[\left(1.58^2 + 7.38^2 + 8.40^2 + 4.92^2 + 5.34^2 + 3.98^2 + 5.18^2 + 2.96^2 + 5.26^2 \right) - \frac{9 \times (9+1)^2}{4} \right]$$

and

$$F_F = \frac{(25-1) \times 114.37}{25 \times (9-1) - 114.37} = \frac{2744.88}{85.63} = 32.05$$

F_F is obtained from the F-distribution with $(k-1) = (9-1) = 8$ and $(k-1)(25-1) = (9-1)(25-1) = 192$ degrees of freedom. The critical value of $F(8, 192)$ at $\alpha = 0.05$ is 1.987; thus, we can reject the null hypothesis that the compared algorithms are equivalent at $\alpha = 0.05$, i.e., $F_F = 32.05 \gg 1.987$. Table 5 shows the differences between the average ranks of the compared algorithms and the average rank of the CIL-FART-IFTSVM for the 16 imbalanced datasets with respect to the G-Mean. At this time, we applied the Bonferroni-Dunn test to compare CIL-FART-IFTSVM with the other imbalanced learning algorithms. The critical difference $CD = 2.72\sqrt{\frac{9(9+1)}{6\times 25}} = 2.11$. The difference between the average ranks of CIL-FART-IFTSVM and FSVM (SVM-OSS, SVM-RUS, SVM, EasyEnsemble, AdaBoost, and 1-NN, respectively) is $5.26 - 1.58 = 3.66$ ($5.18 - 1.58 = 3.60, 3.98 - 1.58 = 2.40, 5.34 - 1.58 = 3.76, 4.92 - 1.58 = 3.34, 8.40 - 1.55 = 6.82$, and $7.38 - 1.58 = 5.80$, respectively), which are greater than 2.11; therefore, there is a significant difference between CIL-FART-IFTSVM and FSVM (SVM-OSS, SVM-RUS, SVM, EasyEnsemble, AdaBoost, and 1-NN, respectively). However, the difference between the average rank of CIL-FART-IFTSVM and SVM-SMOTE is $2.96 - 1.58 = 1.38$, which is less than 2.63; therefore, there is no significant difference between CIL-FART-IFTSVM and SVM-SMOTE. Still, for almost all the datasets, CIL-FART-IFTSVM performed better than SVM-SMOTE (see the results in Table 5).

Table 6
G-Mean (%) of the compared algorithms.

dataset	Im. Ratio	UnderBagging	RUSBoost	SMOTEBagging	CBU	Best CIL-FART-IFTSVM
Wisconsin	1.86	94.97	95.69	95.74	**99.23**	98.32
Pima	1.87	75.12	72.07	74.57	75.26	**76.05**
Yeast 1	2.64	71.08	71.37	72.89	**74.10**	72.69
Vehicle 2	2.88	95.11	96.64	95.90	**98.42**	95.24
Vehicle 1	2.90	75.22	74.19	76.11	**82.15**	75.37
Segment	6.02	97.66	98.42	98.57	98.79	**99.18**
Yeast 3	8.10	93.10	91.32	93.22	**95.44**	93.26
Yeast 2-vs-4	9.08	91.48	90.71	87.23	**93.41**	88.30
Yeast 0-5-6-7-9-vs-4	9.35	76.88	78.92	80.40	**85.33**	80.66
Glass 4	15.46	84.64	90.83	86.77	84.72	**90.84**
Ecoli 4	15.80	88.41	93.54	92.63	94.36	**95.64**
Yeast 1-4-5-8-Vs-7	22.10	55.94	56.30	61.84	62.28	**72.23**
Yeast 2-Vs-8	23.10	74.77	77.51	77.01	**85.23**	75.84
Yeast 4	28.10	83.92	78.99	75.22	**85.01**	82.78
Yeast 1-2-8-9-Vs-7	30.57	67.30	71.87	65.62	69.03	**74.61**
Yeast 5	32.73	94.70	94.18	94.50	**96.93**	94.69
Ecoli 0-1-3-7-vs-2-6	39.14	72.84	79.68	83.08	80.64	**91.53**
Yeast 6	41.40	86.60	82.57	83.82	91.15	**92.72**

Table 7
Details of the large-scale imbalanced datasets.

Dataset	Positive	Negative	Dimension	Im. Ratio	**Instance**
EEG Eye State	6,723	8,257	15	1.23	**14,980**
Adult	34,014	11,208	15	3.03	**45,222**
Connect-4	6,449	61,108	43	9.48	**67,557**
Credit Card Dataset	2,653	97,347	16	36.7	**100,000**
Covertype	20,510	560,502	55	27.33	**581,012**

Table 8
G-Mean (%) with standard deviation (SD) and computational time (s) for imbalanced large-scale datasets

Dataset	SVM G-Mean Time(s)	LSSVM G-Mean Time(s)	TSVM G-Mean Time(s)	FSVM G-Mean Time(s)	CDFTSVM G-Mean Time(s)	CIL-FART-IFTSVM G-Mean Time(s)
EEG Eye State	49.61±0.00 8.27	59.88±1.34 39.16	56.84±1.11 4.42	58.34±1.43 10.26	58.49±1.32 **0.31**	**60.15**±1.20 0.34
Adult	71.54±1.99 25.73	70.11±0.43 2389.07	77.88±0.49 13.17	79.54±0.46 27.28	79.86±0.41 0.71	**81.41**±0.67 **0.68**
Connect-4	59.82±0.00 28.87	58.46±0.69 816.47	67.25±0.67 8.67	69.08±0.42 34.20	68.79±0.65 2.38	**69.23**±0.82 **1.33**
Covertype	56.95±9.34 1170.69	Out of memory	85.92±0.89 540.32	87.43±0.42 1404.82	95.34±0.11 67.78	**95.88**±0.11 **65.31**

Table 9
The experimental outcomes on Census-Income Database

Method	Minority Class F-Measure	Majority Class F-Measure
SBC	62.15	79.06
RT	46.47	38.58
AT	51.09	43.08
NearMiss-2	58.98	78.60
SBCNM-1	45.28	33.41
SBCNM-2	45.64	35.21
SBCNM-3	44.61	30.35
SBCMD	44.94	31.99
SBCMF	59.04	73.34
CIL-FART-IFTSVM$_{non-Linear}$	**62.61**	**81.41**
CIL-FART-IFTSVM$_{nonlinear}$	**64.87**	**83.54**

5.2.4. In the fourth experiment, we evaluate the performance of the proposed CIL-FART-IFTSVM with UnderBagging [57], RUSBoost [65], SMOTEBagging [66], and clustering-based undersampling (CBU) [67].

Table 6 shows the outcomes (G-Mean rate) of four existing under-sampling imbalance learning techniques on 18 imbalanced datasets. From these results, we can see that the proposed technique performed well on all datasets. The performance of the best CIL-FART-IFTSVM outperforms than UnderBagging, RUSBoost, and SMOTEBagging techniques, while it is comparable to the clustering-based undersampling (CBU) technique. As we can see, for eight out of eighteen datasets that were considered in this experiment, the results given by the best CIL-FART-IFTSVM technique are better than the results given by the CBU method, while CBU is better for the other ten out of eighteen datasets.

In some datasets like Wisconsin, Yeast 1-4-5-8-Vs-7, or Yeast 6, we have an improvement of the G-Mean outcomes presented by five techniques: i) From 94.97% to 99.23% applying the CBU in Wisconsin which is 0.92% better than the second best method (best CIL-FART-IFTSVM), ii) From 55.94% to 72.23% using the best CIL-FART-IFTSVM in Yeast 1-4-5-8-Vs-7 which is 13.77% better than the second best method (CBU), and iii) From 82.57% to 92.72% applying the CIL-FART-IFTSVM in Yeast 6 which is 1.69% better than the second best method (CBU).

The experimental results on imbalanced datasets demonstrate that the best CIL-FART-IFTSVM is of significant advantage compared with other algorithms in dealing with imbalanced datasets.

5.3. Large-scale datasets

SVM is not appropriate for classification of large-scale datasets because the training process is related to the size of the dataset. CIL-FART-IFTSVM, which can be applied to address the class imbalance issue in the presence of large-scale datasets, is proposed to overcome this problem. Based on the chart in Fig. 2, we delete the clusters that consist of all negative samples (majority) and select the centers of those samples. We maintain all the data of the mix-labeled clusters and positive samples clusters (minority) as training data in this step. Therefore, the reduced dataset (second step of Fig. 2) is taken as the new training set. To assess the performance of CIL-FART-IFTSVM, we considered 4 large-scale datasets and one real-world dataset. Table 7 shows the details of the large-scale datasets. Moreover, a coordinate descent system with shrinking by an active set is applied to reduce the computational complexity.

Table 10
Fraud detection rate (%) for real-world dataset

Group	Algorithm	FDR(%)
Rules	DTNB-X1	36.96
	Decision Table 1e	29.16
	One R	26.71
Bayes	Bayes Net	35.41
	Naive Bayes	26.03
	A2DE	35.34
Function	SGD	19.06
	Neura1 Network	21.03
	MLP Classifier	21.48
Lazy	IBk	42.27
	IB1	42.57
	KStar	48.30
	LWL	26.71
Meta	AdaBoost M1	26.71
	Attribute Selected Classifier	26.71
	Bagging	38.73
	Dagging	26.71
	Decorate	30.78
	END	30.67
	Filtered Classifier	28.37
	Logit Boost	26.22
	Multi Boost AB	26.71
	Multi Class Classifier	17.78
	Ordinal Class Classifier	30.67
	Random SubSpace	33.72
	Rotation Forest	49.17
	Random Committee	47.36
	Threshold Selector	25.14
	Randomizable Filtered	43.70
	Classification Via Regression	33.64
	MultiClass Classifier Updateable	19.06
	Iterative Classifier Optimizer	26.46
Tree	LMT	31.42
	LAD Tree	27.76
	J48	30.67
	REP Tree	40.73
	Random Tree	47.28
	Hoeffding Tree	27.58
CIL-FART-IFTSVM	Linear Kernel Function	**55.31**

5.3.1. Large scale UCI datasets

In the first experiment, we evaluate the efficacy of the proposed CIL-FART-IFTSVM on large-scale imbalanced datasets by comparing with SVM, LSSVM, TSVM, FSVM, and CDFTSVM. Four large-scale imbalanced datasets with sample sizes from 14,980 to 581,012 are employed. For the Connect-4 and Covertype datasets, we used the following description to transform the datasets into binary imbalanced datasets.

For the imbalanced version of the Connect-4 dataset, we considered class 2 as positive examples and classes 1 and 3 as negative examples.

For the imbalanced version of the Covertype dataset, we considered class 7 as positive examples and classes 1, 2, 3, 4, 5, and 6 as negative examples.

Table 8 shows that CIL-FART-IFTSVM performs superior to other methods on all datasets while maintaining the shortest training time. The time performance for the EEG Eye State dataset is not the best but is comparable to that of CDFTSVM and better than that of the others.

In the second experiment, we compare our technique with Cluster-based under-sampling approaches for imbalanced data distributions (SBC, SBCNM-1, SBCNM-2, SBCNM-3, SBCMD, SBCMF) [68], RT, AT, NearMiss-2 [69] that are under-sampling approaches in Census-Income Database. This dataset is collected from UCI Knowledge Discovery and extracted from the 1994 and 1995 current population surveys managed by the US Census Bureau. The total number of instances after cleaning the incomplete data is 30001, including 21465 majority class samples and 8536 minority class samples. We used 80% of the instances for training and 20% for testing to evaluate the performances of the classifiers. For this dataset, the five-fold cross-validation is repeated. The F-measures for our technique is chosen to compare with the other cluster-based under-

sampling methods. From Table 9, we can see that our technique CIL-FART-IFTSVM has the highest Minority Class F-measure and Majority Class F-Measure with other techniques. (See Table 10)

5.3.2. Real-world dataset

In recent years, credit card transactions have experienced remarkable progress with the development of electronic commerce and show great promise for improvement in the future. Therefore, the implementation of an effective fraud detection system has becomes essential for all card issuing authorities to avoid losses. Usually, only 1% of transactions are fraudulent while 99% are valid. Therefore, credit card transaction datasets are highly skewed and scattered. Hence, fraud detection or simple pattern matching techniques are not effective in detecting fraud.

To overcome these substantial difficulties, we apply CIL-FART-IFTSVM on the UCSD-FICO [51] dataset, which includes 100,000 transactions of 73,729 customers for 98 days. The dataset includes 20 features, including the class label. Of the 100,000 credit transactions, 97,346 (98.35%) are Class 0 (normal) and 2,654 (2.65%) were Class 1 (fraudulent). For this experiment, 10-fold cross-validation is performed. We focus on the fraud detection rate (FDR %) of CIL-FART-IFTSVM and other machine learning classifiers, such as Bayesian classifiers, rules, functions, meta-algorithms, lazy algorithms, and tree algorithms. The experimental results are obtained from [70]. For fraud detection, good performance means a high detection rate, i.e., true positives/positives (how many fraud cases can be detected correctly). Table 9 shows that the best result for Rules is 36.96 for DTNB-X1, for Bayes is 35.41 for Bayes Net, for Function is 21.48 for the MLP Classifier, for Lazy is 48.30 for KStar, for Meta is 49.17 for Rotation Forest, and for Tree is 47.28 for Random Tree. CIL-FART-IFTSVM (55.31) outperforms all other algorithms used in this experiment. Therefore, CIL-FART-IFTSVM can detect fraudulent transactions.

6. Conclusion

This paper proposes CIL-FART-IFTSVM for the binary class imbalance problem. In this method, we used Fuzzy ART as a clustering method and IFTSVM to address the issues of class imbalance, outliers/noise, and large-scale dataset. To validate the performance of CIL-FART-IFTSVM and other existing CIL methods, we used 45 imbalanced datasets and considered a non-parametric statistical test. The experimental results indicate that the approach proposed in this paper outperforms the other methods, particularly in the presence of noise and outliers. For large-scale datasetss, CIL-FART-IFTSVM is significantly better than SVM, LSSVM, TSVM, FSVM, and CDFTSVM, and the training is significantly faster. In future work, it would be interesting to apply our method to decision trees.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Grants 61772344, 71371063, and 61732011) and in part by Basic Research Project of Knowledge Innovation Program in ShenZhen (JCYJ20180305125850156).

References

[1] V.N. Vapnik, Statistical learning theory, Wiley-Interscience, 1998.
[2] O.L. Mangasarian, E.W. Wild, Multisurface proximal support vector machine classification via generalized eigenvalues, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (1) (2006) 69–74.
[3] Khemchandani R. Jayadeva, S. Chandra, Twin support vector machines for pattern classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 29(5) (2007) 905–910.
[4] Y. Shao, C. Zhang, X. Wang, N. Deng, Improvements on twin support vector machines, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (6) (2011) 962–968.
[5] Z.M. Yang, H.J.Wu, C.N. Li, Y.H. Shao, Least squares recursive projection twin support vector machine for multi-class classification, International Journal of Machine Learning and Cybernetics 7(3) (2016) 411–426.
[6] C.F. Lin, S.D. Wang, Fuzzy support vector machines, IEEE Transactions on Neural Networks 13 (2) (2002) 464–471.
[7] R.K. Sevakula, N.K. Verma, Compounding general purpose membership functions for fuzzy support vector machine under noisy environment, IEEE Transactions on Fuzzy Systems 25 (6) (2017) 1446–1459.
[8] B. Gao, J. Wang, Y. Wang, C. Yang, Coordinate descent fuzzy twin support vector machine for classification, in: 2015 IEEE 14th International Conference on Machine Learning and Applications, 2015, pp. 7–12.
[9] T.Atanassov K., Intuitionistic fuzzy sets, Fuzzy Sets and Systems 20(1) (1986) 87–96.
[10] S. Rezvani, Ranking method of trapezoidal intuitionistic fuzzy numbers, Annals of Fuzzy Mathematics and Informatics 5 (3) (2010) 515–523.
[11] Rezvani. S, Wang. X, A new type-2 intuitionistic exponential triangular fuzzy number and its ranking method with centroid concept and euclidean distance, in: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2018, pp. 1–8.
[12] S. Rezvani, X. Wang, F. Pourpanah, Intuitionistic fuzzy twin support vector machines, IEEE Transactions on Fuzzy Systems 27 (11) (2019) 2140–2151, https://doi.org/10.1109/TFUZZ.2019.2893863.
[13] K. Veropoulos, C. Campbell, N. Cristianini, Controlling the sensitivity of support vector machines (1999) 55–60.
[14] G. Wu, E.Y. Chang, Kba: kernel boundary alignment considering imbalanced data distribution, IEEE Transactions on Knowledge and Data Engineering 17 (6) (2005) 786–795.
[15] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.

[16] Y. Liu, Y. Chen, Face recognition using total margin-based adaptive fuzzy support vector machines, IEEE Transactions on Neural Networks 18 (1) (2007) 178–192.
[17] Z. Yang, W.H. Tang, A. Shintemirov, Q.H. Wu, Association rule mining-based dissolved gas analysis for fault diagnosis of power transformers, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 39 (6) (2009) 597–610.
[18] W. Khreich, E. Granger, A. Miri, R. Sabourin, Iterative boolean combination of classifiers in the roc space: An application to anomaly detection with hmms, Pattern Recognition 43 (8) (2010) 2732–2752.
[19] P. Bermejo, J. Gmez, J. Puerta, Improving the performance of naive bayes multinomial in e-mail foldering by introducing distribution-based balance of datasets, Expert Systems with Applications 38 (3) (2011) 2072–2080.
[20] D. Gupta, B. Richhariya, P. Borah, A fuzzy twin support vector machine based on information entropy for class imbalance learning, Neural Computing and Applications (2018) 1–12.
[21] Q. Fan, Z. Wang, L. Dongdong, G. Daqi, Z. Hongyuan, Entropy-based fuzzy support vector machine for imbalanced datasets, Knowledge-Based Systems 115 (1) (2017) 87–99.
[22] S. Rezvani, X. Wang, Erratum to entropy-based fuzzy support vector machine for imbalanced datasets, Knowledge-Based Systems 192 (15) (2020) 105287.
[23] R. Batuwita, V. Palade, Fsvm-cil: Fuzzy support vector machines for class imbalance learning, IEEE Transactions on Fuzzy Systems 18 (3) (2010) 558–571.
[24] S. Wang, X. Yao, Relationships between diversity of classification ensembles and single-class performance measures, IEEE Transactions on Knowledge and Data Engineering 25 (1) (2013) 206–219.
[25] L.G. Abril, H. Nunez, C. Angulo, F. Velasco, Gsvm: An svm for handling imbalanced accuracy between classes in bi-classification problems, Applied Soft Computing 17 (1) (2014) 23–31.
[26] B.S. Raghuwanshi, S. Shukla, Class-specific extreme learning machine for handling binary class imbalance problem, Neural Networks 105 (2018) 206–217.
[27] Q. Kang, L. Shi, M. Zhou, X. Wang, Q. Wu, Z. Wei, A distance-based weighted undersampling scheme for support vector machines and its application to imbalanced classification, IEEE Transactions on Neural Networks and Learning Systems 29 (9) (2018) 4152–4165, https://doi.org/10.1109/TNNLS.2017.2755595.
[28] B. Cao, Y. Liu, C. Hou, J. Fan, B. Zheng, J. Yin, Expediting the accuracy-improving process of svms for class imbalance learning, IEEE Transactions on Knowledge and Data Engineering (2020) 1, https://doi.org/10.1109/TKDE.2020.2974949.
[29] X. Tao, Q. Li, W. Guo, C. Ren, C. Li, R. Liu, J. Zou, Self-adaptive cost weights-based support vector machine cost-sensitive ensemble for imbalanced data classification, Information Sciences 487 (2019) 31–56.
[30] T. Pan, J. Zhao, W. Wu, J. Yang, Learning imbalanced datasets based on smote and gaussian distribution, Information Sciences 512 (2020) 1214–1233.
[31] G. Folino, C. Pizzuti, G. Spezzano, Gp ensembles for large-scale data classification, IEEE Transactions on Evolutionary Computation 10 (5) (2006) 604–616.
[32] G. Huang, K. Mao, C. Siew, D.S. Huang, Fast modular network implementation for support vector machines, IEEE Transactions on Neural Networks 16 (6) (2005) 1651–1663.
[33] H. Yu, J. Yang, J. Han, Classifying large data sets using svms with hierarchical clusters, Proceedings of the 9th ACM SIGKDD, Washington, DC, 2003.
[34] M. Awad, L. Khan, F. Bastani, L.Yen I., An effective support vector machine svms performance using hierarchical clustering, in: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI04), 2004, pp. 663–667.
[35] J. Platt, Fast training of support vector machines using sequential minimal optimization, Advances in Kernel Methods – Support Vector Learning, MIT Press, 1998.
[36] R. Collobert, S. Bengio, Svmtorch: support vector machines for large regression problems, Journal of Machine Learning Research 1 (2001) 143–160.
[37] C. Pizzuti, D. Talia, P-autoclass: scalable parallel clustering for mining large data sets, IEEE Transactions on Knowledge and Data Engineering 15 (3) (2003) 629–641.
[38] V.H. Franc, An iterative algorithm learning the maximal margin classifier, Pattern Recognition 36 (9) (2003) 1985–1996.
[39] K.P. Bennett, E.J. Bredensteiner, Duality and geometry in svm classifiers, in: Proc 17th International Conf on Machine Learning, 2000, pp. 57–64.
[40] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy, A fast iterative nearest point algorithm for support vector machine classifier design, IEEE Transactions on Neural Networks 11 (1) (2000) 124–136.
[41] R. Duda, P. Hart, D. Stork, Pattern Classification Second edition, Wiley, Hoboken, 2001.
[42] S. Grossberg, How does a brain build a cognitive code?, Psychological Review 1 (1980) 1–51
[43] G.A. Carpenter, S. Grossberg, D.B. Rosen, Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system, Neural Networks 4 (1991) 759–771.
[44] T. Frank, K.F. Kraiss, T. Kuhlen, Comparative analysis of fuzzy art and art-2a network clustering performance, IEEE Transactions on Neural Networks 9 (3) (1998) 544–559.
[45] K.W. Chang, C.J. Hsieh, C.J. Lin, Coordinate descent method for large-scale 12-loss linear support vector machines, The Journal of Machine Learning Research 9 (2008) 1369–1398.
[46] Y.H. Shao, N.Y. Deng, A coordinate descent margin based-twin support vector machine for classification, Neural Networks 25 (2012) 114–121.
[47] T.H. Cormen, Introduction to Algorithms, MIT Press, Cambridge, MA, USA, 2009.
[48] F. Pourpanah, C.J. Tan, C.P. Lim, J. Mohamad-Saleh, A q-learning-based multi-agent system for data classification, Applied Soft Computing 52 (2017) 519–531.
[49] J. Alcala-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. Garcia, L. Sanchez, F. Herrera, Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework, Journal of Multiple-Valued Logic and Soft Computing 17 (2–3) (2011) 255–287.
[50] D. Dheeru, E. Karra Taniskidou, UCI machine learning repository, 2017. URL: https://archive.ics.uci.edu/ml.
[51] http://wwwpaymentscardsandmobilecom, 2005.
[52] B. Efron, Bootstrap methods: Another look at the jackknife, Annals of Statistics 7 (1) (1979) 1–26.
[53] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, The Annals of Mathematical Statistics 11 (1) (1940) 86–92.
[54] J. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Processing Letters 9 (3) (1999) 293–300.
[55] L. Kai, M. Hongyan, A fuzzy twin support vector machine algorithm, International Journal of Application or Innovation in Engineering and Management (IJAIEM) 2 (3) (2013) 459–465.
[56] K. Bowyer, N. Chawla, L. Hall, W. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (1) (2002) 321–357.
[57] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches, IEEE Transactions on Systems, Man, and Cybernetics, Part C 42 (4) (2012) 463–484.
[58] M. Kubat, S. Matwin, Addressing the curse of imbalanced training sets: one-sided selection, International Conference on Machine Learning 97 (1997) 179–186.
[59] X. Liu, J. Wu, Z. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Transactions on Systems, Man, and Cybernetics, Part B 39 (2) (2009) 539–550.
[60] G. Batista, R.C.P., M.C M., A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD Explorations Newsletter (2004) 1:20–29.
[61] I. Tomek, Two modifications of cnn, IEEE Transactions on Systems Man and Communications 6 (1976) 769–772.

[62] R. Iman, J. Devenport, Approximations of the critical region of the friedman statistics, Communications in Statistics-Theory and Methods 9 (6) (1980) 571–595.
[63] O.J. Dunn, Multiple comparisons among means, Journal of the American Statistical Association 56 (293) (1961) 52–64.
[64] J. Demar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (1) (2006) 1–30.
[65] C. Seiffert, T.M. Khoshgoftaar, J. Van Hulse, A. Napolitano, Rusboost: A hybrid approach to alleviating class imbalance, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 40 (1) (2010) 185–197.
[66] S. Wang, X. Yao, Diversity analysis on imbalanced data sets by using ensemble models, IEEE Symposium on Computational Intelligence and Data Mining (2009) 324–331.
[67] L. Wei-Chao, T. Chih-Fong, H. Ya-Han, J. Jing-Shang, Clustering-based undersampling in class-imbalanced data, Information Sciences 409–410 (2017) 17–26.
[68] S.J. Yen, Y.S. Lee, Cluster-based under-sampling approaches for imbalanced data distributions, Expert Systems with Applications 36 (3, Part 1) (2009) 5718–5727, https://doi.org/10.1016/j.eswa.2008.06.108.
[69] J. Zhang, I. Mani, knn approach to unbalanced data distributions: A case study involving information extraction, in: Proceedings of the ICML2003 Workshop on Learning from Imbalanced Datasets, 2003.
[70] M.S. Mahmud, P. Meesad, S. Sodsee, An evaluation of computational intelligence in credit card fraud detection, International Computer Science and Engineering Conference (ICSEC) 2016 (2016) 1–6, https://doi.org/10.1109/ICSection 2016.7859947.