Applied Soft Computing 71 (2018) 418–432

Contents lists available at ScienceDirect

Applied Soft Computing

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

A robust fuzzy least squares twin support vector machine for class imbalance learning

B. Richhariya, M. Tanveer*

Discipline of Mathematics, Indian Institute of Technology Indore, Simrol, Indore, 453552, India

ARTICLE INFO

Article history:
Received 21 November 2017
Received in revised form 1 March 2018
Accepted 1 July 2018
Available online 9 July 2018

Keywords:
Fuzzy membership
Least squares twin support vector machine
Class imbalance
Imbalance ratio
Outliers

ABSTRACT

Twin support vector machine is one of the most prominent techniques for classification problems. It has been applied in various real world applications due to its less computational complexity. In most of the applications on classification, there is imbalance in the number of samples of the classes which leads to incorrect classification of the data points of the minority class. Further, while dealing with imbalanced data, noise poses a major challenge in various applications. To resolve these problems, in this paper we propose a robust fuzzy least squares twin support vector machine for class imbalance learning termed as RFLSTSVM-CIL using 2-norm of the slack variables which makes the optimization problem strongly convex. In order to reduce the effect of outliers, we propose a novel fuzzy membership function specifically for class imbalance problems. Our proposed function gives the appropriate weights to the datasets and also incorporates the knowledge about the imbalance ratio of the data. In our proposed model, a pair of system of linear equations is solved instead of solving a quadratic programming problem (QPP) which makes our model efficient in terms of computation complexity. To check the performance of our proposed approach, several numerical experiments are performed on synthetic and real world benchmark datasets. Our proposed model RFLSTSVM-CIL has shown better generalization performance in comparison to the existing methods in terms of AUC and training time.

1. Introduction

In the last few decades, support vector machine (SVM) [1,2] has become a popular technique for classification problems. Due to its structural risk minimization (SRM) principle, SVM has a low VC (Vapnik-Chervonenkis) dimension and thus less number of optimizing parameters. In contrast to algorithms like artificial neural networks which suffer from the problem of local minima, SVM gives a global and unique solution by solving a convex optimization problem. Due to these benefits, SVM has been applied to various applications [3–7]. To reduce the training time of SVM, a least squares support vector machine (LSSVM) is proposed in [8] where a system of linear equation is solved instead of solving a QPP. Jayadeva et al. [9] proposed an efficient twin support vector machine (TWSVM) for classification problems to reduce the computational complexity of SVM. In TWSVM, two hyperplanes are constructed instead of one as in SVM and the optimization problem is to keep each of the hyperplanes closer to its own class and away from the other class. In [10], Kumar and Gopal proposed a least squares twin support vector machine (LSTSVM) where a pair of system of linear equations is solved and takes very less computation time as compared to SVM. One of the important applications of SVM is the classification of class imbalance datasets. In many applications like disease detection [11], fault detection [12], defective software modules detection [13] and others involving high imbalance in the data, the priority is the classification of the data points of the minority class. For example, in disease detection there are very less samples of people having the disease as compared to the number of healthy people.

To reduce the effect of noise, fuzzy based memberships have been used with SVM giving lesser weights to the outlier data points. In various applications such as bankruptcy problem [14] and object tracking [15], fuzzy based approach is used. In [16], Lin and Wang proposed a fuzzy support vector machine (FSVM) based on distance from the class centroid for each class. This reduces the effect of outliers in the classification because the outliers get relatively less weight for the classification as compared to the other points. For multiclass classification, a fuzzy least squares support vector machines (FLS-SVM) is proposed in [17]. A bilateral-weighted FSVM (B-FSVM) is proposed in [18] where membership of each sample is calculated by considering the sample as belonging to both the

* Corresponding author.
E-mail addresses: phd1701241001@iiti.ac.in (B. Richhariya), mtanveer@iiti.ac.in, tanveergouri@gmail.com (M. Tanveer).

https://doi.org/10.1016/j.asoc.2018.07.003
1568-4946/

classes with different membership values. To reduce the computation complexity of B-FSVM, Balasundaram and Tanveer proposed a proximal bilateral weighted fuzzy support vector machine where a system of linear equations is solved [19]. For dealing with outliers, An and Liang [20] proposed a fuzzy support vector machine based on within-class scatter for classification problems. A fuzzy support vector machine algorithm for classification based on a novel partition index maximization (PIM) fuzzy clustering method is proposed in [21]. For multilabel classification, fuzzy support vector machines are proposed [22] where a membership function is defined for each label set and the data point is classified on the basis of fuzzy membership value for each label. A clifford fuzzy support vector machine for classification is proposed in [23] based on clifford geometric algebra for multiclass classification. Tanveer et al. [24] proposed a robust energy based least squares twin support vector machine which includes the SRM principle and the energy parameter reduces the effect of noise in the data. Recently, a new fuzzy twin support vector machine (NFTSVM) formulation is proposed in [25] to deal with noisy data.

In class imbalance learning, due to a huge difference in the number of samples of the binary classes, SVM classifier gives more priority to the samples of the majority class and misclassifies the samples which are in minority. To give more weight to the minority class, different weights are assigned to the data points of both the classes. Since FSVM is not suitable for class imbalance learning, Batuwita and Palade [26] proposed FSVM-CIL with different settings of parameters and fuzzy membership functions. An improved one-class SVM for class imbalance is proposed in [27] using a conformal kernel transformation. A boosting algorithm for support vector machine [28] is proposed for countering the excessive bias in classifying imbalance data. FSVM for class imbalance in medical datasets is proposed [29] for incorporating the local information using a local within-class preserving scatter matrix. A scaling kernel function is proposed [30] for SVM in class imbalance learning. A hybrid sampling approach is presented for SVM [31] where an oversampling technique is combined with the undersampling technique. A weighted least squares projection twin support vector machine is proposed in [32] to include the local information about the data. A fuzzy least squares twin support vector machine is proposed [33] to deal with class imbalance datasets. Razzaghi et al. [34] proposed a multilevel framework of the cost-sensitive SVM for class imbalance healthcare data with missing values. Fuzzy total margin based support vector machine (FTM-SVM) is proposed with different settings in [35] for imbalance problems. A weighted K-means support vector machine for cancer prediction is proposed in [36] to circumvent the problem of imbalance in the data. A weighted least squares twin support vector machine (WLSTSVM) is proposed for binary classification in [37] and a weighted multiclass least squares twin support vector machine (WMLSTSVM) is proposed in [38]. In WMLSTSVM, the fuzzy membership function gives membership on the basis of number of samples in the two classes and thus is not capable of dealing with outliers. Recently, an entropy based fuzzy support vector machine (EFSVM) is proposed in [39]. In EFSVM, the data points of the majority class are given fuzzy membership based on their information entropy on the basis of proximity to the binary classes. In this fuzzy membership approach, the drawback is that the outlier data points of the majority class also get higher membership value. In this work, we propose a novel robust fuzzy least squares twin support vector machine for class imbalance learning (RFLSTSVM-CIL) for binary classification. To give appropriate membership to the majority class, we propose a new fuzzy membership function for imbalance datasets. In the previous work on fuzzy membership for imbalance data, the range of fuzzy membership is fixed for all the datasets which are having different imbalance ratios (IR). To overcome this drawback, our function uses the information about the imbalance ratio (IR) and gives appropriate range of the fuzzy membership to different datasets. Moreover, we present a novel 2-norm based robust fuzzy least squares twin support vector machine for class imbalance learning (RFLSTSVM-CIL). To justify the effectiveness of our proposed approach, several numerical experiments are performed on synthetic and real world benchmark datasets.

In this paper, all vectors are taken as column vectors. The inner product of two vectors is represented as: $a^t b$ where a and b are the vectors in $n-$ dimensional real space R^n and a^t is the transpose of a. $||a||$ and $\left|\left|X\right|\right|$ represent the 2-norm of a vector a and a matrix X respectively. The identity matrix and the vector of ones of dimension m are denoted by I and e respectively. The minority class is termed as positive class with label '1' and majority class is termed as negative class with class label '-1'. The imbalance ratio (IR) is calculated as

$$\mathrm{IR} = \frac{\text{Number of samples of negative class}}{\text{Number of samples of positive class}}.$$

The rest of the paper is organized as follows: Section 2 gives the formulation of FTWSVM and LSTSVM. The proposed method RFLSTSVM-CIL and fuzzy membership function are discussed in Section 3. To check the effectiveness of the proposed method, several numerical experiments are performed on various synthetic and real world benchmark datasets in Section 4. Section 5 presents a discussion on the experimental results. In Section 6, we give the conclusions with future work.

2. Related work

We discuss the formulations of fuzzy twin support vector machine (FTWSVM) and least squares twin support vector machine (LSTSVM) in this section with the different fuzzy membership functions.

2.1. Fuzzy twin support vector machine (FTWSVM)

Jayadeva et al. [9] proposed a novel classification algorithm known as twin support vector machine (TWSVM). In TWSVM, two QPPs of smaller size are solved in comparison to traditional SVM and thus lead to lesser computational complexity. Here, for comparison of our proposed approach with twin support vector machine based models, we used the existing fuzzy memberships for the data points in TWSVM using the same approach as in [26,49,50]. In FTWSVM, twin hyperplanes are constructed based on the fuzzy membership values according to the fuzzy membership functions and has the advantage of less training cost as compared to SVM. The fuzzy memberships are based on distance from the centroid of the classes as in [26]. By using this approach, different weights are given to the data points which results in less effect of outliers on the final classifier.

The fuzzy membership functions for centroid based membership [26] are as follows

2.1.1. Centroid (linear)

In this, the fuzzy membership is assigned based on the distance of the data points from the centroid of its class. Here, the decaying function is linear in nature. The fuzzy membership function is given as

$$mem = 1 - \frac{d_{cen}}{\max(d_{cen}) + \delta}, \tag{1}$$

where d_{cen} is the Euclidean distance of each data point from the centroid of its class and δ is a small positive integer to ensure that denominator is not 0.

2.1.2. Centroid (exponential)

The decaying function is exponential in nature and the fuzzy membership is assigned based on the distance of the data points from the centroid of its class. The fuzzy membership function is written as

$$mem = \frac{2}{1+\exp\left(\beta\, d_{cen}\right)}, \tag{2}$$

where d_{cen} is the Euclidean distance of each data point from the centroid of its class and β decides the scale of the exponential function.

Let us consider the input matrices X_1 and X_2 of size $p \times n$ and $q \times n$ where p is the number of data points belonging to 'class 1' and q denotes the number of data points belonging to 'class 2' such that total number of data samples are $m = p + q$, and n are the attributes of each data point.

The objective functions of FTWSVM in primal are written as

$$\min_{w_1,b_1,\xi} \frac{1}{2}\left|\left|K\left(X_1, D^t\right)w_1 + e_1 b_1\right|\right|^2 + C_1 s_2^t \xi$$

$$\text{s.t.}\ -(K(X_2, D^t)w_1 + e_2 b_1) + \xi \geq e_2,\ \xi \geq 0, \tag{3}$$

$$\min_{w_2,b_2,\eta} \frac{1}{2}\left|\left|K\left(X_2, D^t\right)w_2 + e_2 b_2\right|\right|^2 + C_2 s_1^t \eta$$

$$\text{s.t.}\ \left(K\left(X_1, D^t\right)w_2 + e_1 b_2\right) + \eta \geq e_1,\ \eta \geq 0, \tag{4}$$

where ξ, η represent slack variables; matrix D consists of $[X_1; X_2]$, e_1, e_2 are vectors of suitable dimension having all values as 1 and $K\left(x^t, D^t\right) = (k(x, x_1), \ldots, k(x, x_m))$ is a row vector in R^m, C_1, C_2 are penalty parameters; s_1, s_2 are vectors containing the membership values of data samples in the constraints.

The Lagrangian function of the problems (3) & (4) are written as

$$L_1 = \frac{1}{2}\left|\left|K\left(X_1, D^t\right)w_1 + e_1 b_1\right|\right|^2 + C_1 s_2^t \xi + \alpha_1^t\left(\left(K\left(X_2, D^t\right)\right.\right.$$
$$\left. w_1 + e_2 b_1\right) - \xi + e_2\Big) - \beta_1^t \xi, \tag{5}$$

$$L_2 = \frac{1}{2}\left|\left|K(X_2, D^t)w_2 + e_2 b_2\right|\right|^2 + C_2 s_1^t \eta + \alpha_2^t\left(\left(-K\left(X_1, D^t\right)\right.\right.$$
$$\left. w_2 - e_1 b_2\right) - \eta + e_1\Big) - \beta_2^t \eta, \tag{6}$$

where $\alpha_1 = \left(\alpha_{11}, \ldots, \alpha_{1q}\right)^t$, $\beta_1 = \left(\beta_{11}, \ldots, \beta_{1q}\right)^t$, $\alpha_2 = \left(\alpha_{21}, \ldots, \alpha_{2p}\right)^t$ and $\beta_2 = \left(\beta_{21}, \ldots, \beta_{2p}\right)^t$ are the vectors of Lagrange multipliers.

Now, applying the Karush-Kuhn-Tucker (K.K.T.) necessary and sufficient conditions, the Wolfe duals of Eqs. (5) and (6) are written as

$$\min_{\alpha_1} \frac{1}{2}\alpha_1^t T\left(S^t S\right)^{-1} T^t \alpha_1 - e_2^t \alpha_1$$

$$s.t.\ 0 \leq \alpha_1 \leq s_2 C_1, \tag{7}$$

$$\min_{\alpha_2} \frac{1}{2}\alpha_2^t S\left(T^t T\right)^{-1} S^t \alpha_2 - e_1^t \alpha_2$$

$$s.t.\ 0 \leq \alpha_2 \leq s_1 C_2, \tag{8}$$

where $S = \left[K\left(X_1, D^t\right) \quad e_1\right]$ and $T = \left[K\left(X_2, D^t\right) \quad e_2\right]$.

We compute the non-linear hyperplanes $K\left(x^t, D^t\right)w_1 + b_1 = 0$ and $K\left(x^t, D^t\right)w_2 + b_2 = 0$ by computing the values of w_1, w_2, b_1 and b_2 as

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -(S^t S + \delta I)^{-1} T^t \alpha_1 \text{ and } \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (T^t T + \delta I)^{-1} S^t \alpha_2, \tag{9}$$

where δ is a small positive value to remove the ill conditioning of the matrices for finding the inverse.

The classifier is written as

$$class\, i = \min |K(x^t,\ D^t)w_i + b_i| \quad \text{for}\ i = 1, 2. \tag{10}$$

2.2. Least squares twin support vector machine (LSTSVM)

In order to reduce the computation cost of TWSVM, Kumar and Gopal [10] proposed a least squares twin support vector machine (LSTSVM) where two hyperplanes are obtained by solving a pair of system of linear equations. The optimization problem for non-linear LSTSVM is written as

$$\min_{w_1,b_1,\xi} \frac{1}{2}\left|\left|K\left(\mathrm{X}_1, D^t\right)w_1 + eb_1\right|\right|^2 + \frac{C_1}{2}\xi^t \xi$$

$$s.t.\ -\left(K\left(\mathrm{X}_2, D^t\right)w_1 + eb_1\right) + \xi = e, \tag{11}$$

And

$$\min_{w_2,b_2,\eta} \frac{1}{2}||K(\mathrm{X}_2, D^t)w_2 + eb_2||^2 + \frac{C_2}{2}\eta^t \eta$$

$$s.t.\ \left(K\left(\mathrm{X}_1, D^t\right)w_2 + eb_2\right) + \eta = e, \tag{12}$$

where ξ, η represent slack variables, matrix D consists of $[X_1; X_2]$, $K(x^t, D^t) = (k(x, x_1), \ldots, k(x, x_m))$ is a row vector in R^{m}, $C_1, C_2 > 0$ are penalty parameters and e is the vector of ones of suitable dimension.

Using the equality constraints of (11) and (12) in their objective functions, we get

$$\min_{w_1,b_1} \frac{1}{2}||K(\mathrm{X}_1, D^t)w_1 + eb_1||^2 + \frac{C_1}{2}||K(\mathrm{X}_2, D^t)w_1 + eb_1 + e||^2 \tag{13}$$

and

$$\min_{w_2,b_2} \frac{1}{2}||K(\mathrm{X}_2, D^t)w_2 + eb_2||^2 + \frac{C_2}{2}||-K(\mathrm{X}_1, D^t)w_2 - eb_2 + e||^2. \tag{14}$$

Taking the gradient of (13) with respect to w_1 and b_1 and equating to 0, we get

$$K(\mathrm{X}_1, D^t)^t(K(\mathrm{X}_1, D^t)w_1 + eb_1) + C_1 K(\mathrm{X}_2, D^t)^t(K(\mathrm{X}_2, D^t)w_1$$
$$+ eb_1 + e) = 0e, \tag{15}$$

$$e^t(K(\mathrm{X}_1, D^t)w_1 + eb_1) + C_1 e^t(K(\mathrm{X}_2, D^t)w_1$$
$$+ eb_1 + e) = 0. \tag{16}$$

Combining (15) and (16) in matrix form, w_1 and b_1 are obtained as

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -\left(V^t V + \frac{1}{C_1}U^t U\right)^{-1} V^t e \tag{17}$$

where $U = [K(\mathrm{X}_1, D^t)\ e]$ and $V = [K(\mathrm{X}_2, D^t)\ e]$. Similarly, for the other hyperplane the parameters are computed as

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = \left(U^t U + \frac{1}{C_2}V^t V\right)^{-1} U^t e. \tag{18}$$

For reducing the computation time of finding the inverse, Sherman–Morrison–Woodbury (SMW) formula [41] is used for the Eqs. (17) and (18) and inverses of smaller dimensions are solved.

Case 1. $p < q$

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -\left(Y - YU^t\left(C_1 I + UYU^t\right)^{-1} UY\right) V^t e, \tag{19}$$

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = C_2\left(Y - YU^t\left(\frac{I}{C_2} + UYU^t\right)^{-1} UY\right) U^t e, \tag{20}$$

where $Y = \left(V^t V\right)^{-1}$.

Using the regularization term εI, where $\varepsilon > 0$ for the possible ill conditioning of $\left(V^t V\right)^{-1}$ and rewritten as

$$Y = \frac{1}{\varepsilon}\left(I - V^t\left(\varepsilon I + VV^t\right)^{-1} V\right)$$

Case 2. $q < p$

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -C_1\left(Z - ZV^t\left(\frac{I_2}{C_1} + VZ^t V\right)^{-1} VZ\right) V^t e, \tag{21}$$

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = \left(Z - ZV^t\left(C_2 I + VZV^t\right)^{-1} VZ\right) U^t e, \tag{22}$$

where $Z = \left(U^t U\right)^{-1}$ which is rewritten using SMW formula as

$$Z = \frac{1}{\varepsilon}\left(I - U^t\left(\varepsilon I + UU^t\right)^{-1} U\right)$$

A new data sample $x \in R^n$ is assigned class label of the hyperplane which is nearer to it on the basis of the perpendicular distances from the hyperplanes $K(x^t, D^t)w^1 + b_1 = 0$ and $K(x^t, D^t)w_2 + b_2 = 0$ using (10).

3. Proposed robust fuzzy least squares twin support vector machine for class imbalance learning (RFLSTSVM-CIL)

3.1. Proposed fuzzy membership function

The existing fuzzy membership functions which have been used with different variants of SVM suffer from some drawbacks. For example, the centroid based fuzzy membership function only gives membership values on the basis of the distance from its centroid but it does not take into account whether the data point is near to the region of its own class or the other class.

Similarly, in other fuzzy functions used for class imbalance problems [39], this idea of proximity to its class centroid in combination with the information about the other class is not considered. Also, the information about the extent of imbalance in the data is not utilised in the previous works. In our approach we include the information about the imbalance ratio (IR) which control the range of the membership values. Motivated by the works of [40,25], we propose a new fuzzy membership function for class imbalance problem.

For negative class:

$$mem = \left(\frac{1}{1+IR}\right) + \left(\frac{IR}{1+IR}\right)\left(\frac{\exp\left(C_0\left((d_1 - d_2)/d - d_2/r_2\right)\right) - \exp\left(-2C_0\right)}{\exp\left(C_0\right) - \exp\left(-2C_0\right)}\right), \tag{23}$$

where IR is the imbalance ratio, d_1 is Euclidean distance from centroid of positive class and d_2 is Euclidean distance from centroid of negative class, d is the distance between the centroid of the binary classes, r_2 is the maximum distance of the data points of negative class from its centroid and C_0 decides the scale of the exponential function.

The membership is assigned as 1 to all the data points of the positive class which is having the lesser number of samples [39].

In our proposed fuzzy membership function for the majority class, we consider two aspects of the majority class data points which are as follows:

(1) Proximity of the majority class data point to the centroid of the other classes.
(2) Proximity of the majority class data points to their own class.

3.1.1. Properties of proposed function

1 The membership value of the negative class ranges from $\left(\frac{1}{1+IR}\right)$ to 1 based on the position of the data points w.r.t. centroids of the two classes. The range of the membership value depends on the imbalance ratio (IR).
2 The membership value for the negative data point based on its proximity to the positive class depends on the variable $(d_1 - d_2)/d$ in the membership function.
3 The penalty for the outliers which are proximal to the negative class is taken care by d_2/r_2.
4 The membership value is equal to 1 when $d_2 = 0$ which makes $d_1 = d$.
5 The membership value of data point is equal to $\left(\frac{1}{1+IR}\right)$ when it is closest to the positive class centroid i.e., $d_1 = 0$ resulting in $d = r_2$, and farthest from the centroid of the negative class i.e., $d_2 = r_2$.
6 One can observe from Fig. 2 that if the outlier data point of the majority class (negative class) is in the positive class region, then the penalty on the membership value is higher as compared to when it is on its own side i.e. negative class.

In class imbalance problems, the objective is to classify the data points of the minority class more effectively. For achieving this property in an effective manner, our proposed fuzzy membership function gives the membership according to the following cases which is illustrated in Fig. 2 for an artificial binary dataset:

Case 1. Negative data point closer to the centroid of its own class: *High membership*

Case 2. Negative data point away from its own centroid but relatively closer to its own centroid as compared to the other class: *Low membership*

Case 3. Negative data point away from its own centroid and relatively closer to the positive class centroid: *Very low membership*

3.2. Linear RFLSTSVM-CIL

In our proposed approach, the 2-norm of the weighted slack vector is used for giving fuzzy membership values to the data points in the constraints of the optimization problem.

The objective functions of linear RFLSTSVM-CIL are written as

$$\min_{w_1, b_1, \xi} \frac{1}{2}||X_1 w_1 + eb_1||^2 + \frac{C_1}{2}(S_2\xi)^t(S_2\xi)$$

$$s.t. \ -(X_2 w_1 + eb_1) + \xi = e, \tag{24}$$

$$\text{and} \min_{w_2, b_2, \eta} \frac{1}{2}||X_2 w_2 + eb_2||^2 + \frac{C_2}{2}(S_1\eta)^t(S_1\eta)$$

$$\text{s.t. } (X_1 w_2 + eb_2) + \eta = e, \tag{25}$$

where X_1 and X_2 are matrices of class 1 (minority) and class 2 (majority) containing p and q number of samples respectively. S_1 is identity matrix of dimension p and S_2 is diagonal matrix of dimension q containing the fuzzy membership values at the diagonal places. The slack variables are represented by ξ, η with $C_1, C_2 > 0$ as the penalty parameters and e is the vector of ones of appropriate dimension.

Using the equality constraints of (24) in its objective function, the QPP is written as

$$\min_{w_1,b_1,\xi} \frac{1}{2}||X_1w_1 + eb_1||^2 + \frac{C_1}{2}||S_2(X_2w_1 + eb_1 + e)||^2. \tag{26}$$

Taking the gradient of (26) with respect to w_1 and b_1 and equating to 0, we get

$$X_1^t(X_1w_1 + eb_1) + C_1(S_2X_2)^t(S_2(X_2w_1 + eb_1 + e)) = 0e, \tag{27}$$

$$e^t(X_1w_1 + eb_1) + C_1(S_2e)^t(S_2(X_2w_1 + eb_1 + e)) = 0. \tag{28}$$

Eqs. (27) and (28) can be written in the matrix form as,

$$\begin{bmatrix} (S_2X_2)^t(S_2X_2) & (S_2X_2)^t(S_2e) \\ (S_2e)^t(S_2X_2) & (S_2e)^t(S_2e) \end{bmatrix} \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \frac{1}{C_1}\begin{bmatrix} X_1^tX_1 & X_1^te \\ e^tX_1 & p \end{bmatrix} \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} (S_2X_2)^t(S_2e) \\ (S_2e)^t(S_2e) \end{bmatrix} = 0e,$$

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} (S_2X_2)^t(S_2X_2) + \frac{1}{C_1}X_1^tX_1 & (S_2X_2)^t(S_2e) + \frac{1}{C_1}X_1^te \\ (S_2e)^t(S_2X_2) + \frac{1}{C_1}e^tX_1 & (S_2e)^t(S_2e) + \frac{1}{C_1}p \end{bmatrix}^{-1} \begin{bmatrix} -(S_2X_2)^t(S_2e) \\ -(S_2e)^t(S_2e) \end{bmatrix},$$

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = \left[\begin{bmatrix} (S_2X_2)^t \\ (S_2e)^t \end{bmatrix} [(S_2X_2)\ (S_2e)] + \frac{1}{C_1}\begin{bmatrix} X_1^t \\ e^t \end{bmatrix} [X_1\ e]\right]^{-1} \begin{bmatrix} -(S_2X_2)^t(S_2e) \\ -(S_2e)^t(S_2e) \end{bmatrix}. \tag{29}$$

One can write (29) in the following form

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -(T^tT + \frac{1}{C_1}R^tR)^{-1} T^tS_2e, \tag{30}$$

where $R = [X_1\ e]$ and $T = [S_2X_2\ S_2e]$.

Similarly, for the other hyperplane the parameters are computed by the following formula

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (R^tR + \frac{1}{C_2}T^tT)^{-1} R^tS_1e, \tag{31}$$

where $R = [S_1X_1\ S_1e]$ and $T = [X_2\ e]$.

For reducing the computation time of finding the inverse, SMW formula [41] is used for the Eqs. (30) and (31) and inverses of smaller dimensions are solved.

If we give the membership values for both the classes according to our fuzzy membership function then our proposed RFLSTSVM-CIL can also be applied to datasets with no class imbalance. Further, if the membership values of both the classes are set to 1, then our proposed RFLSTSVM-CIL reduces to the standard LSTSVM. So, we can say that LSTSVM is a special case of our RFLSTSVM-CIL.

3.3. Non-linear RFLSTSVM-CIL

The formulation of the non-linear RFLSTSVM-CIL is written as

$$\min_{w_1,b_1,\xi} \frac{1}{2}||K(X_1,D^t)w_1 + eb_1||^2 + \frac{C_1}{2}(S_2\xi)^t(S_2\xi)$$

$$\text{s.t. } -(K(X_2,D^t)w_1 + eb_1) + \xi = e, \tag{32}$$

$$\text{and } \min_{w_2,b_2,\eta} \frac{1}{2}||K(X_2,D^t)w_2 + eb_2||^2 + \frac{C_2}{2}(S_1\eta)^t(S_1\eta)$$

$$\text{s.t. } \left(K(X_1,D^t)\, w_2 + eb_2\right) + \eta = e, \tag{33}$$

where matrix D consists of $[X_1; X_2]$, $K(X_1,D^t)$, $K(X_2,D^t)$ are the kernel matrices of class 1 and 2 respectively. S_1 is identity matrix of dimension p and S_2 is diagonal matrix of dimension q containing the fuzzy membership values at the diagonal places. The slack variables are represented by ξ, η with $C_1, C_2 > 0$ as the penalty parameters and e is the vector of ones of suitable dimension.

Using the equality constraints of (32) in the objective function of (32), the QPP is written as

$$\min_{w_1,b_1,\xi} \frac{1}{2}||K(X_1,D^t)w_1 + eb_1||^2 + \frac{C_1}{2}||S_2(K(X_2,D^t)w_1 + eb_1 + e)||^2. \tag{34}$$

Taking the gradient of (34) with respect to w_1 and b_1 and equating to 0, we get

$$K(X_1,D^t)^t(K(X_1,D^t)w_1 + eb_1) + C_1(S_2K(X_2,D^t))^tS_2(K(X_2,D^t)w_1 + eb_1 + e) = 0e, \tag{35}$$

$$e^t(K(X_1,D^t)w_1 + eb_1) + C_1(S_2e)^tS_2(K(X_2,D^t)w_1 + eb_1 + e) = 0. \tag{36}$$

Similarly as in the linear case, one can write in the following form

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -(T^tT + \frac{1}{C_1}R^tR)^{-1} T^tS_2e, \tag{37}$$

where $R = [K(X_1,D^t)\ e]$ and $T = [S_2K(X_2,D^t)\ S_2e]$.

Similarly, for the other hyperplane the parameters are computed as

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (R^tR + \frac{1}{C_2}T^tT)^{-1} R^tS_1e, \tag{38}$$

where $R = [S_1K(X_1,D^t)\ S_1e]$ and $T = [K(X_2,D^t)\ e]$.

For reducing the computation time of finding the inverse, SMW formula [41] is used for the Eqs. (37) and (38) and inverses of smaller dimensions are solved.

The class of a new data sample $x \in R^n$ is predicted based on the perpendicular distances from the hyperplanes $K(x^t, D^t)w_1 + b_1 = 0$ and $K(x^t, D^t)w_2 + b_2 = 0$ and the class label of the nearer hyperplane is assigned to it.

3.4. Computational complexity

Our proposed approach of RFLSTSVM-CIL incorporates the 2-norm of the slack variable with fuzzy membership values in the formulation of LSTSVM. Similar to LSTSVM, our proposed RFLSTSVM-CIL solves two systems of linear equations which involve the inversion of matrices.

In the formulation of LSTSVM, the calculation of two inverses of size $(m+1)$ is required where $m = p + q$, p and q are number of

Table 1
Imbalance ratio (IR) of synthetic imbalance datasets for all the samples and for the training data.

Dataset (Train size, Test size)	Imbalance ratio (All samples)	Imbalance ratio (Training samples)
04clover5z-600-5-60-BI (200×2, 400×2)	5	5.06
03subcl5-600-5-30-BI (200×2, 400×2)	5	4.71
03subcl5-600-5-50-BI (200×2, 400×2)	5	4.41
03subcl5-600-5-60-BI (200×2, 400×2)	5	5.06
Crossplane_400 (119×2, 281×2)	7	6
Crossplane_450 (134×2, 316×2)	8	7.93

data points of positive and negative class. So, to reduce the computation of the inverses the Sherman–Morrison–Woodbury (SMW) formula [41] is used, where three inverses of smaller size are solved as shown in Eq. 19–22. In case of our proposed algorithm the size of the invertible matrices are same as in LSTSVM, so there is no computation overhead in terms of solving the optimization problem as compared to LSTSVM. Further, in comparison to SVM and TWSVM our proposed RFLSTSVM-CIL is computationally efficient as it is calculating the solution of linear equations as in LSTSVM.

The additional computation involved in our proposed method is the calculation of the fuzzy membership function. In comparison to the existing fuzzy based approaches, our fuzzy membership function is efficient in terms of computation cost. The traditional fuzzy based functions have the time complexity of $O(m)$. This is because these functions calculate the fuzzy membership of all the data points based on measures like distance from centroid with linear and exponential decay functions. Our proposed fuzzy based function has the time complexity as $O(q)$ where $q < m$ and q is the number of samples of the negative class. This is due to the fact that our proposed function calculates fuzzy values only for the majority class and assigns the membership value as 1 for the minority class.

4. Numerical experiments

The performance of the proposed approach is compared with several existing algorithms on various synthetic and real world imbalanced datasets. EFSVM [39], TWSVM [9], $\text{FTWSVM}_{\text{lin}}$ & $\text{FTWSVM}_{\text{exp}}$ [9,26,49,50], universum twin support vector machine (UTSVM) [42] and LSTSVM [10] are compared with the proposed method RFLSTSVM-CIL in terms of accuracy and training time. UTSVM incorporates the notion of prior information about the data. So we have compared our proposed RFLSTSVM-CIL with UTSVM in case of imbalanced data. To show the effectiveness of our proposed fuzzy membership function, we show the comparison of our proposed RFLSTSVM-CIL with the novel fuzzy function to the proposed algorithm using existing fuzzy functions for assigning the weights. The centroid based fuzzy membership functions with linear ($\text{FLSTSVM}_{\text{lin}}$-CIL) and exponential decay ($\text{FLSTSVM}_{\text{exp}}$-CIL) are also the proposed algorithms using the existing fuzzy membership functions.

All computations are performed on a PC running on Windows 10 OS with 64 bit, 3.60 GHz Intel® core^{TM} i7-7700 processor having 16 GB of RAM under MATLAB R2008b environment. MOSEK optimization toolbox is used to solve the quadratic programming problems which are taken from http://www.mosek.com. For the selection of the optimal parameters, 5-fold cross-validation is used for all the methods. We used Gaussian kernel in all the methods, since Gaussian kernel defines a function space that is larger than that of the linear kernel or the polynomial kernel and thus pro-

Table 2
Performance comparison of proposed RFLSTSVM-CIL on AUC and training time with EFSVM, TWSVM, $\text{FTWSVM}_{\text{lin}}$, $\text{FTWSVM}_{\text{exp}}$, UTSVM, LSTSVM, $\text{FLSTSVM-CIL}_{\text{lin}}$ and $\text{FLSTSVM-CIL}_{\text{exp}}$ using Gaussian kernel for classification on synthetic imbalanced datasets. Bold value indicates highest AUC for the dataset.

Dataset (Train size, Test size)	EFSVM AUC (%) ±SD (C, μ, K) Time (s)	TWSVM AUC (%) ±SD (C, μ) Time (s)	$\text{FTWSVM}_{\text{lin}}$ AUC (%) ±SD (C, μ) Time (s)	$\text{FTWSVM}_{\text{exp}}$ AUC (%) ±SD (C, μ, β) Time (s)	UTSVM AUC(%) ±SD (C, μ, ε) Time (s)	LSTSVM AUC (%) ±SD (C, μ) Time (s)	$\text{FLSTSVM-CIL}_{\text{lin}}$ AUC (%) ±SD (C, μ) Time (s)	$\text{FLSTSVM-CIL}_{\text{exp}}$ AUC (%) ±SD (C, μ, β) Time (s)	RFLSTSVM-CIL AUC (%) ±SD (C, μ, C_0) Time (s)
04clover5z-600-5-60-BI (200×2, 400×2)	69.38 ± 9.52 (10^-5, 2^5, 7) 1.1949	61.15 ± 8.13 (10^-3, 2^5) 0.1408	63.36 ± 8.93 (10^-3, 2^5) 0.1469	60.93 ± 6.36 (10^0, 2^5, 0.1) 0.1269	64.33 ± 8.37 (10^-3, 2^5, 0.3) 0.1532	72.28 ± 9.5 (10^-5, 2^5) 0.1007	72.33 ± 5.52 (10^-5, 2^5) 0.1115	67.35 ± 14.88 (10^-5, 2^5, 0.5) 0.156	**75.54 ± 8.33** (10^-4, 2^5, 0.5) 0.1122
03subcl5-600-5-30-BI (200×2, 400×2)	71.26 ± 6.1 (10^2, 2^5, 3) 1.2436	65.83 ± 6.76 (10^-4, 2^5) 0.1442	64.66 ± 6.63 (10^-4, 2^5) 0.1566	60.25 ± 4.43 (10^5, 2^5, 0.3) 0.1392	67.62 ± 7.79 (10^0, 2^5, 0.3) 0.1736	66.72 ± 5.55 (10^-4, 2^5) 0.1111	72.64 ± 5.55 (10^-4, 2^5) 0.1297	**75.45 ± 10** (10^-5, 2^5, 0.3) 0.1537	72.8 ± 2.86 (10^-2, 2^5, 1.5) 0.1264
03subcl5-600-5-50-BI (200×2, 400×2)	58.65 ± 8.84 (10^-5, 2^5, 3) 1.1921	51.86 ± 4.87 (10^-4, 2^5) 0.1448	52.95 ± 4.28 (10^-3, 2^5) 0.1505	49.57 ± 1.73 (10^0, 2^5, 0.1) 0.1342	56.31 ± 10.4 (10^-2, 2^5, 0.6) 0.1592	54.76 ± 8.58 (10^-4, 2^5) 0.1102	55.7 ± 7.65 (10^-5, 2^5) 0.1267	**60.35 ± 6.35** (10^-5, 2^5, 0.1) 0.1255	58.1 ± 7.11 (10^-5, 2^5, 0.5) 0.1283
03subcl5-600-5-60-BI (200×2, 400×2)	69.73 ± 5.75 (10^0, 2^5, 3) 1.1877	66.18 ± 6.78 (10^-1, 2^5) 0.1498	65.91 ± 6.18 (10^-1, 2^5) 0.1504	62.98 ± 2.35 (10^2, 2^5, 0.1) 0.131	66.84 ± 8.16 (10^-1, 2^5, 0.3) 0.1624	71.04 ± 5.67 (10^-4, 2^5) 0.1073	**72.75 ± 5.9** (10^-4, 2^5) 0.1237	71.19 ± 7.63 (10^-5, 2^5, 0.3) 0.1463	72.07 ± 5.8 (10^-4, 2^5, 2) 0.1281
Crossplane_400 (119×2, 281×2)	92.1 ± 9.14 (10^0, 2^-4, 3) 0.6101	95.43 ± 6.97 (10^-5, 2^-1) 0.0743	93.97 ± 5.58 (10^-2, 2^0) 0.0693	95.65 ± 3.24 (10^-2, 2^0, 1) 0.0712	95.43 ± 6.97 (10^-5, 2^-1, 0.1) 0.0719	93.99 ± 1.65 (10^-3, 2^-1) 0.0547	**98.58 ± 1.49** (10^-4, 2^-1) 0.0702	94.59 ± 2.12 (10^-3, 2^-1, 0.7) 0.0545	97.18 ± 1.5 (10^-1, 2^2, 2.5) 0.0552
Crossplane_450 (134×2, 316×2)	96.23 ± 6.2 (10^0, 2^-5, 3) 0.7515	97.32 ± 3.68 (10^-5, 2^1) 0.0864	97.32 ± 3.68 (10^-5, 2^1) 0.0839	97.32 ± 3.68 (10^-5, 2^1, 0.1) 0.0852	97.32 ± 3.68 (10^-5, 2^1, 0.1) 0.089	98.96 ± 1.12 (10^-5, 2^-3) 0.0686	99.05 ± 1.42 (10^-5, 2^-3) 0.0904	98.96 ± 1.12 (10^-5, 2^-3, 0.1) 0.0702	**99.13 ± 0.88** (10^-4, 2^-3, 1.5) 0.0791

vides better classification accuracy. The Gaussian kernel is defined as $K(a, b) = \exp\left(-\frac{1}{2\mu^2}||a-b||^2\right)$ where vector $a, b \in R^m$ and μ is the kernel parameter. The area under the receiver operating characteristics (ROC) curve (AUC) [43] is used for the comparison of the methods which is calculated as

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2},$$

where TP_{rate} is the true positive rate of data points belonging to positive class and FP_{rate} is the false positive rate of the data points of negative class.

The AUC is calculated as mean AUC with standard deviation for five iterations on the testing data. In each iteration, one part is used for testing and the remaining data for training. The time is calculated in seconds and averaged over five iterations. The value of the penalty parameter is taken as $C = C_1 = C_2 = C_u$ from the set $\{10^{-5}, \ldots, 10^5\}$ where C_u is used in UTSVM [42] and μ is taken from the set $\{2^{-5}, \ldots, 2^5\}$ for all the cases. For RFLSTSVM-CIL, C_0 is chosen from the set $\{0.5, 1, 1.5, 2, 2.5\}$. For EFSVM, β is considered as 0.05, l is taken as 10 and the value of k is chosen from $\{3, 5, 7, 9, 11\}$. For FTWSVM$_{lin}$, δ is taken as 0.01 and for FTWSVM$_{exp}$, β is chosen from the set $\{0.1, 0.3, 0.5, 0.7, 1\}$. In UTSVM, ε is taken from the set $\{0.1, 0.3, 0.5, 0.6\}$ and number of universum samples i.e u is taken as 10% of the training data. In this work, the AUC is calculated in terms of percentage for all the algorithms.

4.1. Synthetic datasets

To analyse the performance of our proposed method, we performed experiments on different synthetic datasets. We used 6 synthetic datasets to test the performance of our proposed approach. The datasets containing noise are taken from KEEL imbalanced dataset repository [44,45] having 2 classes where the data points are randomly and uniformly distributed in the two-dimensional space (both attributes are real valued). The noisy datasets are namely 04clover5z-600-5-60-BI, 03subcl5-600-5-30-BI, 03subcl5-600-5-50-BI and 03subcl5-600-5-60-BI with the disturbance ratio as 60%, 30%, 50% and 60% respectively [45].

We also performed experiments on Crossplane (XOR) dataset [46] generated with different number of samples and imbalance ratios as shown in Table 1. For the generation of the datasets, randomized values of data points are used in the equation of a line i.e, $y = kx + b$ to generate the dataset. The parameters for slope and intercept i.e., k and b are chosen as 0.7 and 0.1 for negative class and $-$ 0.6 and 1 for positive class. Fig. 1 shows the distribution of data points in the Crossplane dataset with 400 samples. AUC values and training time are shown in Table 2 for Gaussian kernel with the corresponding ranks in Table 3 for the performance comparison of the proposed RFLSTSVM-CIL with EFSVM, TWSVM, FTWSVM$_{lin}$, FTWSVM$_{exp}$, UTSVM, LSTSVM, FLSTSVM$_{lin}$-CIL and FLSTSVM$_{exp}$-CIL on the synthetic datasets.

4.2. Real world datasets

Numerical experiments are performed on several real world imbalanced datasets taken from KEEL imbalanced datasets [44] and UCI repository [47] for binary classification. The class imbalance ratios of the various real world datasets are shown in Table 4. The performance of the proposed RFLSTSVM-CIL is compared with EFSVM, TWSVM, FTWSVM$_{lin}$, FTWSVM$_{exp}$, UTSVM, LSTSVM, FLSTSVM$_{lin}$-CIL and FLSTSVM$_{exp}$-CIL in terms of AUC values and training time in Table 5 . The corresponding rank table is shown in Table 6.

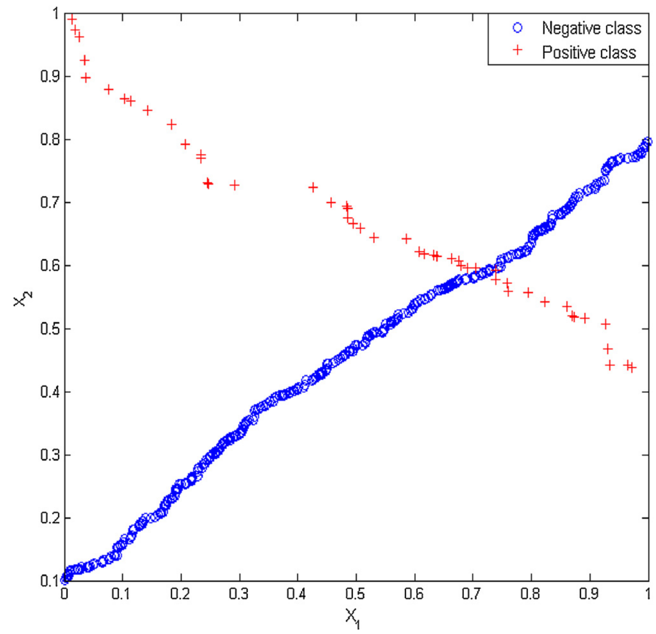


Fig. 1. Crossplane artificial imbalance dataset (Crossplane_400) with imbalance ratio (IR) as 7.

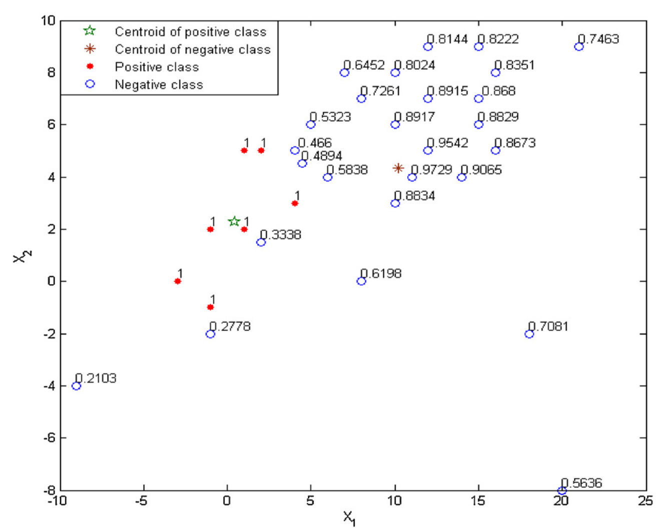


Fig. 2. Plot of artificial dataset ($IR = 3.57$) showing membership values of the data points based on the proposed membership function for $C_0 = 0.5$.

One can notice from Table 6 that our proposed approach is having the least rank with less training time since our approach solves a pair of system of linear equations. To check the statistical significance of our proposed RFLSTSVM-CIL, we use Friedman test with the corresponding post-hoc test [48] for the 9 algorithms using 31 binary class datasets. Here, we assume that all the methods are equivalent under null hypothesis. The Friedman statistic is computed for the ranks on AUC values from Table 6 as

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

where k is the number of methods and N is the number of datasets.

Table 3
Rank comparison on the basis of AUC of proposed RFLSTSVM-CIL with EFSVM, TWSVM, $FTWSVM_{lin}$, $FTWSVM_{exp}$, UTSVM, LSTSVM, FLSTSVM-CIL_{lin} and FLSTSVM-CIL_{exp} using Gaussian kernel for classification on synthetic imbalanced datasets.

Dataset	EFSVM	TWSVM	$FTWSVM_{lin}$	$FTWSVM_{exp}$	UTSVM	LSTSVM	FLSTSVM-CIL_{lin}	FLSTSVM-CIL_{exp}	RFLSTSM-CIL
04clover5z-600-5-60-BI	4	8	7	9	6	3	2	5	1
03subcl5-600-5-30-BI	4	7	8	9	5	6	3	1	2
03subcl5-600-5-50-BI	2	8	7	9	4	6	5	1	3
03subcl5-600-5-60-BI	5	7	8	9	6	4	1	3	2
Crossplane_400	9	4.5	8	3	4.5	7	1	6	2
Crossplane_450	9	6.5	6.5	6.5	6.5	3.5	2	3.5	1
Average Rank	5.5	6.8333	7.4167	7.5833	5.3333	4.9167	2.3333	3.25	1.8333

Table 4
Imbalance ratio (IR) of real world datasets for all the samples and for the training data.

Dataset (Train size, Test size)	Imbalance ratio (All samples)	Imbalance ratio (Training samples)
Cmc (700×9, 773×9)	0.75	1.46
Ecoli-0-1_vs_2-3-5 (120×7, 124×7)	9.17	9
Ecoli-0-1_vs_5 (120×6, 120×6)	11	16.14
Ecoli-0-1-4-7_vs_5-6 (150×6, 182×6)	12.28	10.54
Ecoli-0-2-3-4_vs_5 (100×7, 102×7)	9.1	6.69
Ecoli-0-2-6-7_vs_3-5 (110×7, 114×7)	9.18	7.46
Ecoli-0-3-4-6_vs_5 (100×7, 105×7)	9.25	8.09
Ecoli-0-4-6_vs_5 (100×6, 103×6)	9.15	11.5
Ecoli-0-6-7_vs_3-5 (110×7, 112×7)	9.09	12.75
Ecoli-0-6-7_vs_5 (110×6, 110×6)	10	9
Ecoli4 (150×7, 186×7)	15.8	17.75
Glass-0-4_vs_5 (50×9, 42×9)	9.22	6.14
Glass2 (100×9, 114×9)	11.59	13.29
Ripley (600×2, 650×2)	1	1.08
Yeast-0-2-5-6_vs_3-7-8-9 (500×8, 504×8)	9.14	11.5
Yeast-0-3-5-9_vs_7-8 (250×8, 256×8)	9.12	11.5
Yeast-0-5-6-7-9_vs_4 (250×8, 278×8)	9.35	8.26
Ecoli-0-1-4-6_vs_5 (150×6, 130×6)	13	9.71
Ecoli2 (150×7, 186×7)	8.6	6.89
Vowel (500×10, 488×10)	9.98	9.87
Ecoli3 (150×7, 186×7)	8.6	6.89
Abalone9-18 (350×7, 381×7)	16.4	18.44
Vehicle 1 (400×18, 446×18)	2.9	3.3
Vehicle2 (400×18, 446×18)	2.88	2.48
Pima-Indians (300×8, 468×8)	1.87	1.63
Yeast3 (500×8, 984×8)	8.1	7.2
Yeast1vs7 (200×8, 259×8)	14.3	15.67
Yeast2vs8 (250×8, 233×8)	23.15	19.83
Ecoli0137vs26 (180×7, 131×7)	4.76	4.63
Australian-Credit (300×14, 390×14)	1.25	1.13
Monk2 (300×7, 301×7)	1.92	2.06

$$\chi_F^2 = \frac{12 \times 31}{9 \times (9+1)} \Bigg[(5.1129^2 + 6.1613^2 + 5.2258^2 + 6.2742^2 + 6.9677^2 + 3.6613^2 + 3.9677^2 + 4.8871^2 + 2.7419^2) - \frac{9 \times (9+1)^2}{4} \Bigg] \cong 61.4887,$$

$$F_F = \frac{(31-1) \times 61.4887}{31 \times (9-1) - 61.4887} \cong 9.8903,$$

where F_F is distribution according to the F -distribution with $(9-1, (9-1) \times (31-1)) = (8, 240)$ degrees of freedom with 9 methods and 31 datasets. The critical value of $F(8, 240)$ is 1.9771 for the level of significance at $\alpha = 0.05$. Since the value of $F_F = 9.8782 > 1.9771$, so we reject the null hypothesis. Further, the Nemenyi post-hoc test is performed for pair-wise comparison of methods. The significant difference between the methods is checked by computing the critical difference (CD) at $p = 0.10$ which should differ by at least $2.855\sqrt{\frac{9\times(9+1)}{6\times 31}} \approx 1.986$.

The difference between the averages ranks of EFSVM, TWSVM, $FTWSVM_{lin}$, $FTWSVM_{exp}$, UTSVM and $FLSTSVM_{exp}$-CIL with RFLSTSVM-CIL are $(5.1129 - 2.7419 = 2.371)$, $(6.1613 - 2.7419 = 3.4194)$, $(5.2258 - 2.7419 = 2.4839)$, $(6.2742 - 2.7419 = 3.5323)$, $(6.9677 - 2.7419 = 4.2258)$ and $(4.8871 - 2.7419 = 2.1452)$ respectively which are greater than 1.986, so we conclude that our RFLSTSVM-CIL is significantly better than EFSVM, TWSVM, $FTWSVM_{lin}$, $FTWSVM_{exp}$, UTSVM and $FLSTSVM_{exp}$-CIL for class imbalance problems.

5. Discussion

The proposed RFLSTVM-CIL uses the 2-norm of the slack variables with the fuzzy membership values as shown in Eq. (24) and (25). This makes the optimization problem strongly convex and gives globally optimal solution. For dealing with varying imbalance conditions, the novel fuzzy membership gives different ranges to the fuzzy membership values by using the information about the imbalance ratio (IR) of the data. The imbalance ratio of the synthetic and real datasets is shown in Table 1 and Table 4. By incorporating the imbalance ratio (IR) in fuzzy function proper range is set for the fuzzy membership values on different datasets. Moreover, the information about the proximity of data points to the two

Table 5
Performance comparison of proposed RFLSTSVM-CIL on average AUC and training time with EFSVM, TWSVM, FTWSVM$_{\text{lin}}$, FTWSVM$_{\text{exp}}$, UTSVM, LSTSVM, FLSTSVM-CIL$_{\text{lin}}$ and FLSTSVM-CIL$_{\text{exp}}$ using Gaussian kernel for classification on real world datasets. Bold value indicates highest AUC for the dataset.

Dataset (Train size, Test size)	EFSVM AUC (%) ±SD (C, μ, K) Time (s)	TWSVM AUC (%) ±SD (C, μ) Time (s)	FTWSVM$_{\text{lin}}$ AUC (%) ±SD (C, μ) Time (s)	FTWSVM$_{\text{exp}}$ AUC (%) ±SD (C, μ, β) Time (s)	UTSVM AUC (%) ±SD (C, μ, ε) Time (s)	LSTSVM AUC (%) ±SD (C, μ) Time (s)	FLSTSVM-CIL$_{\text{lin}}$ AUC (%) ±SD (C, μ) Time (s)	FLSTSVM-CIL$_{\text{exp}}$ AUC (%) ±SD (C, μ, β) Time (s)	RFLSTSVM-CIL AUC (%) ±SD (C, μ, C_0) Time (s)
Cmc (700×9, 773×9)	64.31 ± 29.12 (10^4, 2^2, 9) 4.6819	69.13 ± 34.63 (10^-2, 2^4) 0.4954	**70.68 ± 32.08** (10^-5, 2^5) 0.5306	68.56 ± 32.92 (10^-5, 2^4, 0.1) 0.5227	53 ± 4.05 (10^-1, 2^-1, 0.5) 0.6908	59.02 ± 17.43 (10^-2, 2^4) 0.3974	57.45 ± 14.58 (10^-2, 2^4) 0.4448	59.33 ± 15.84 (10^-2, 2^4, 0.3) 0.4398	60.66 ± 19.64 (10^-2, 2^4, 0.5) 0.4521
Ecoli-0-1_vs_2-3-5 (120×7, 124×7)	77.42 ± 21.94 (10^0, 2^5, 7) 0.1211	67.42 ± 21.77 (10^-5, 2^5) 0.0165	67.42 ± 21.77 (10^-3, 2^5) 0.0186	61.53 ± 21.48 (10^-4, 2^5, 0.1) 0.019	72.42 ± 26.33 (10^-5, 2^5, 0.1) 0.0195	69.55 ± 22.27 (10^-5, 2^4) 0.0102	**84.32 ± 11.65** (10^-5, 2^4) 0.011	75.76 ± 26.48 (10^-4, 2^5, 0.3) 0.0107	80.32 ± 13.29 (10^-5, 2^5, 0.5) 0.0113
Ecoli-0-1_vs_5 (120×6, 120×6)	78.71 ± 18.02 (10^-5, 2^1, 3) 0.1116	76.25 ± 18.84 (10^-5, 2^5) 0.0158	75.83 ± 19.18 (10^-5, 2^4) 0.0163	75.83 ± 19.18 (10^-5, 2^4, 0.1) 0.0159	72.88 ± 15.73 (10^-3, 2^4, 0.5) 0.0177	**83.75 ± 23.22** (10^-5, 2^5) 0.0092	81.21 ± 22.11 (10^-3, 2^5) 0.0102	74.97 ± 16.96 (10^-5, 2^4, 0.5) 0.0101	82.88 ± 23.74 (10^-4, 2^5, 0.5) 0.0117
Ecoli-0-1-4-7_vs_5-6 (150×6, 182×6)	83.33 ± 20.41 (10^0, 2^5, 5) 0.2588	76.67 ± 18.07 (10^-4, 2^4) 0.0311	68.33 ± 20.75 (10^-2, 2^5) 0.0381	76.34 ± 18.12 (10^3, 2^4, 1) 0.0302	60 ± 22.36 (10^-2, 2^3, 0.5) 0.036	**83.81 ± 19.86** (10^-3, 2^5) 0.0208	75.76 ± 18.22 (10^-5, 2^4) 0.0238	76.34 ± 18.12 (10^-5, 2^4, 1) 0.0225	**83.81 ± 19.86** (10^-3, 2^5, 0.5) 0.0234
Ecoli-0-2-3-4_vs_5 (100×7, 102×7)	**99.5 ± 1.12** (10^0, 2^4, 3) 0.0826	85 ± 22.36 (10^-1, 2^5) 0.0126	85 ± 22.36 (10^-1, 2^5) 0.0135	75 ± 25 (10^-4, 2^4, 0.3) 0.0121	74.21 ± 13.82 (10^-2, 2^3, 0.5) 0.0138	**99.5 ± 1.12** (10^-5, 2^5) 0.0067	98.42 ± 1.45 (10^-5, 2^5) 0.0102	90 ± 22.36 (10^-5, 2^4, 1) 0.0074	**99.5 ± 1.12** (10^-4, 2^5, 2.5) 0.0078
Ecoli-0-2-6-7_vs_3-5 (110×7, 114×7)	63.33 ± 12.64 (10^0, 2^5, 3) 0.1022	63.33 ± 12.64 (10^-3, 2^5) 0.0167	65.83 ± 16.24 (10^-3, 2^5) 0.0154	62.81 ± 12.07 (10^-5, 2^5, 0.1) 0.0163	61.95 ± 13.43 (10^-3, 2^3, 0.6) 0.0174	**66.13 ± 24.85** (10^-2, 2^5) 0.0105	63.92 ± 17.45 (10^-3, 2^5) 0.0096	59.07 ± 13.51 (10^-5, 2^3, 1) 0.0111	64.38 ± 16.87 (10^-3, 2^5, 1.5) 0.0118
Ecoli-0-3-4-6_vs_5 (100×7, 105×7)	**81.47 ± 20.68** (10^0, 2^5, 3) 0.0876	76.47 ± 17.91 (10^-5, 2^5) 0.0136	80.97 ± 20.22 (10^-5, 2^5) 0.0142	79.47 ± 21.11 (10^4, 2^5, 1) 0.0128	76.47 ± 17.91 (10^-5, 2^5, 0.1) 0.0151	80 ± 20.73 (10^-3, 2^5) 0.009	79.47 ± 21.11 (10^0, 2^5) 0.0095	78.47 ± 22.18 (10^-5, 2^5, 0.1) 0.0086	81 ± 21.59 (10^-4, 2^5, 1.5) 0.0114
Ecoli-0-4-6_vs_5 (100×6, 103×6)	84.44 ± 14.25 (10^-1, 2^4, 3) 0.0842	84.44 ± 15.04 (10^-4, 2^5) 0.0125	84.44 ± 15.04 (10^-3, 2^5) 0.0133	86.11 ± 19.04 (10^-5, 2^4, 0.1) 0.013	73.45 ± 19.73 (10^-2, 2^4, 0.6) 0.0141	88.36 ± 14.38 (10^-5, 2^5) 0.0072	88.92 ± 14.85 (10^-5, 2^5) 0.0098	**94.47 ± 6.53** (10^-5, 2^4, 0.7) 0.0093	88.92 ± 14.85 (10^-5, 2^5, 0.5) 0.0115
Ecoli-0-6-7_vs_3-5 (110×7, 112×7)	77.78 ± 15.49 (10^-5, 2^3, 3) 0.0996	73.28 ± 18.42 (10^-3, 2^4) 0.0141	80.78 ± 13.01 (10^-3, 2^4) 0.0164	78.78 ± 12.26 (10^-4, 2^4, 0.3) 0.0137	74.17 ± 3.12 (10^-1, 2^4, 0.6) 0.016	83.33 ± 13.48 (10^-5, 2^5) 0.0094	**83.83 ± 13.44** (10^-5, 2^5) 0.0089	80.83 ± 13.74 (10^-3, 2^5, 0.1) 0.011	82.83 ± 13.9 (10^-4, 2^5, 0.5) 0.0104
Ecoli-0-6-7_vs_5 (110×6, 110×6)	73 ± 16.9 (10^-5, 2^3, 5) 0.0962	59 ± 22.95 (10^-1, 2^5) 0.0141	59 ± 22.95 (10^-1, 2^5) 0.0148	69 ± 19.89 (10^2, 2^5, 0.3) 0.015	73.5 ± 24.47 (10^-2, 2^4, 0.6) 0.0154	**86 ± 12.94** (10^-3, 2^5) 0.0081	82 ± 21.17 (10^-4, 2^5) 0.0085	76.02 ± 23.65 (10^-5, 2^4, 0.1) 0.0109	81 ± 20.2 (10^-3, 2^5, 0.5) 0.0088
Ecoli4 (150×7, 186×7)	93.87 ± 8.62 (10^1, 2^-2, 3) 0.2707	96.34 ± 5.89 (10^0, 2^2) 0.0316	96.63 ± 5.97 (10^0, 2^2) 0.0397	96.34 ± 5.89 (10^0, 2^2, 0.7) 0.0347	93.87 ± 8.62 (10^-2, 2^-1, 0.6) 0.037	95.76 ± 7.11 (10^-3, 2^-1) 0.0226	**97.12 ± 1.03** (10^-2, 2^4) 0.0235	95.76 ± 7.11 (10^-3, 2^-1, 0.1) 0.0242	97.11 ± 1.07 (10^-5, 2^1, 0.5) 0.0236
Glass-0-4_vs_5 (50×9, 42×9)	68.57 ± 25.56 (10^1, 2^1, 3) 0.0163	60 ± 22.36 (10^-1, 2^4) 0.0066	68.57 ± 25.56 (10^-1, 2^4) 0.0071	60 ± 22.36 (10^-1, 2^4, 0.1) 0.007	68.57 ± 25.56 (10^0, 2^4, 0.6) 0.0058	**70 ± 27.39** (10^2, 2^0) 0.0013	**70 ± 27.39** (10^-5, 2^3) 0.0026	**70 ± 27.39** (10^-3, 2^3, 0.5) 0.0022	**70 ± 27.39** (10^0, 2^3, 0.5) 0.0016
Glass2 (100×9, 114×9)	53.33 ± 22.29 (10^5, 2^2, 3) 0.1039	51.15 ± 10.31 (10^-3, 2^3) 0.0145	47.79 ± 10.1 (10^-3, 2^3) 0.0153	46.88 ± 9.33 (10^-3, 2^3, 0.3) 0.0151	52.19 ± 15.25 (10^3, 2^3, 0.6) 0.0166	62.5 ± 17.4 (10^-3, 2^2) 0.0085	56.34 ± 18.37 (10^-3, 2^3) 0.0128	**66.81 ± 16.59** (10^-4, 2^-1, 1) 0.0122	58.16 ± 17.63 (10^-2, 2^3, 1.5) 0.0093
Ripley (600×2, 650×2)	91.27 ± 2.71 (10^0, 2^-2, 7) 3.2518	**91.54 ± 1.95** (10^-1, 2^-1) 0.3452	91.48 ± 2.14 (10^-1, 2^-1) 0.3485	**91.54 ± 2.58** (10^-1, 2^-1, 1) 0.3531	90.23 ± 1.91 (10^0, 2^-1, 0.1) 0.4037	90.64 ± 2.98 (10^0, 2^1) 0.2766	90.8 ± 2.4 (10^0, 2^-1) 0.3092	91.43 ± 2.13 (10^0, 2^-1, 0.1) 0.3033	91.03 ± 2.3 (10^0, 2^0, 1.5) 0.2999
Yeast-0-2-5-6_vs_3-7-8-9 (500×8, 504×8)	64.99 ± 5.44 (10^2, 2^0, 7) 2.0021	73.23 ± 8.09 (10^-5, 2^3) 0.2059	74.43 ± 5.1 (10^-2, 2^2) 0.2146	72.39 ± 7.12 (10^-5, 2^2, 0.5) 0.2155	74.42 ± 5.92 (10^1, 2^1, 0.6) 0.2759	**79.1 ± 7.74** (10^-1, 2^2) 0.1678	73.06 ± 6.8 (10^0, 2^3) 0.1802	**79.1 ± 7.74** (10^-1, 2^2, 0.1) 0.179	75.29 ± 8.19 (10^0, 2^4, 1.5) 0.1798
Yeast-0-3-5-9_vs_7-8 (250×8, 256×8)	55.81 ± 5.95 (10^3, 2^-2, 7) 0.5044	58.45 ± 6.08 (10^-2, 2^0) 0.0552	63.33 ± 3.58 (10^-2, 2^0) 0.0577	60.74 ± 8.25 (10^-2, 2^0, 0.1) 0.0586	56.19 ± 10.96 (10^2, 2^-2, 0.6) 0.0709	66.92 ± 6.84 (10^-1, 2^1) 0.0459	65.14 ± 6.26 (10^-1, 2^1) 0.0464	66.92 ± 6.84 (10^-1, 2^1, 0.1) 0.045	**73.15 ± 2.97** (10^0, 2^3, 2.5) 0.0463

Yeast-0-5-6-7-9_vs_4 (250×8, 278×8)	66.73 ± 6.6 (10^3, 2^1, 3) 0.5954	65.35 ± 6.83 (10^-1, 2^3) 0.066	68.27 ± 10.72 (10^-1, 2^3) 0.0692	65.35 ± 6.83 (10^-1, 2^3, 0.1) 0.0681	50 ± 0 (10^1, 2^1, 0.1) 0.0797	**82.27 ± 12.89** (10^-3, 2^-1) 0.0497	72.92 ± 16.43 (10^0, 2^0) 0.0558	81.43 ± 14.11 (10^-3, 2^-1, 1) 0.0523	76.35 ± 15.84 (10^-2, 2^-1, 2) 0.0529
Ecoli-0-1-4-6_vs_5 (150×6, 130×6)	**100 ± 0** (10^-5, 2^3, 3) 0.131	95 ± 11.18 (10^-3, 2^5) 0.0182	95 ± 11.18 (10^-2, 2^5) 0.0192	95 ± 11.18 (10^3, 2^5, 0.3) 0.0252	**100 ± 0** (10^-3, 2^5, 0.5) 0.0205	**100 ± 0** (10^-2, 2^5) 0.0106	**100 ± 0** (10^-5, 2^4) 0.012	**100 ± 0** (10^4, 2^4, 0.5) 0.012	**100 ± 0** (10^-2, 2^5, 2.5) 0.0135
Ecoli2 (150×7, 186×7)	84.61 ± 22.4 (10^0, 2^-2, 11) 0.2656	82.37 ± 9.02 (10^-5, 2^4) 0.0319	85.12 ± 8.07 (10^-5, 2^4) 0.035	82.37 ± 9.02 (10^-5, 2^4, 0.1) 0.0323	82.37 ± 9.02 (10^-5, 2^4, 0.1) 0.0341	**86.42 ± 9.5** (10^-1, 2^5) 0.0281	83.7 ± 4.62 (10^-2, 2^2) 0.0236	**86.42 ± 9.5** (10^-1, 2^5, 0.1) 0.0233	85.14 ± 7.64 (10^-2, 2^1, 2.5) 0.0244
Vowel (500×10, 488×10)	85.1 ± 9.92 (10^1, 2^3, 11) 1.8297	95.48 ± 4.73 (10^-1, 2^5) 0.2214	93.93 ± 8.02 (10^-1, 2^5) 0.2191	95.48 ± 4.73 (10^-1, 2^5, 0.1) 0.2306	89.32 ± 12.11 (10^0, 2^4, 0.3) 0.2602	95.26 ± 7.21 (10^-1, 2^5) 0.1569	95.6 ± 5.52 (10^-2, 2^4) 0.169	93.12 ± 7.87 (10^0, 2^5, 0.7) 0.1709	**96.17 ± 6.34** (10^-1, 2^5, 0.5) 0.171
Ecoli3 (150×7, 186×7)	84.61 ± 22.4 (10^0, 2^-2, 11) 0.271	82.37 ± 9.02 (10^-5, 2^4) 0.0308	85.12 ± 8.07 (10^-5, 2^4) 0.0368	82.37 ± 9.02 (10^-5, 2^4, 0.1) 0.0333	82.37 ± 9.02 (10^-5, 2^4, 0.1) 0.034	**86.42 ± 9.5** (10^-1, 2^5) 0.0218	83.7 ± 4.62 (10^-2, 2^2) 0.024	**86.42 ± 9.5** (10^-1, 2^5, 0.1) 0.0236	85.14 ± 7.64 (10^-2, 2^1, 2.5) 0.0239
Abalone9-18 (350×7, 381×7)	67.62 ± 10.73 (10^5, 2^2, 7) 1.1422	71.19 ± 18.71 (10^-2, 2^0) 0.1277	66.87 ± 11.88 (10^-1, 2^-1) 0.1346	66.69 ± 10.65 (10^-1, 2^0, 0.7) 0.1384	66.41 ± 10.91 (10^-1, 2^0, 0.5) 0.1489	72.52 ± 17.89 (10^-1, 2^1) 0.0931	76.32 ± 17.83 (10^-1, 2^1) 0.1	73.95 ± 18.48 (10^-1, 2^1, 0.3) 0.0997	**79.53 ± 19.71** (10^0, 2^1, 1) 0.0996
Vehicle 1 (400×18, 446×18)	69.96 ± 3.23 (10^1, 2^5, 7) 1.5729	66.6 ± 5.59 (10^-5, 2^5) 0.1684	67.66 ± 4.1 (10^-2, 2^5) 0.1743	64.55 ± 6.25 (10^4, 2^5, 0.1) 0.1814	67.87 ± 4.59 (10^-2, 2^5, 0.3) 0.1912	**70.52 ± 6.59** (10^-3, 2^5) 0.138	67.43 ± 5.56 (10^-5, 2^4) 0.1511	66.18 ± 4.9 (10^-4, 2^5, 0.1) 0.1467	69.86 ± 7.57 (10^-3, 2^5, 2.5) 0.1467
Vehicle2 (400×18, 446×18)	92.96 ± 3.46 (10^1, 2^5, 11) 1.5622	85.38 ± 6.69 (10^-2, 2^5) 0.1721	86.56 ± 7.03 (10^-2, 2^5) 0.1795	83.45 ± 6.98 (10^4, 2^5, 0.1) 0.1946	91.2 ± 3.27 (10^-1, 2^5, 0.5) 0.1995	93.25 ± 4.28 (10^-1, 2^5) 0.1382	92.59 ± 4.5 (10^-1, 2^5) 0.1442	86.31 ± 1.98 (10^-3, 2^5, 0.1) 0.1469	**93.39 ± 5.45** (10^-2, 2^5, 0.5) 0.1469
Pima-Indians (300×8, 468×8)	71.47 ± 2.39 (10^-1, 2^5, 11) 1.6708	66.32 ± 3.08 (10^-5, 2^5) 0.1759	68.04 ± 4.15 (10^-5, 2^5) 0.1825	60.69 ± 4.35 (10^-1, 2^5, 0.1) 0.1851	50 ± 0 (10^0, 2^0, 0.3) 0.212	71.5 ± 1.8 (10^-4, 2^5) 0.1461	72.4 ± 2.89 (10^-4, 2^5) 0.1586	70.92 ± 1.15 (10^-3, 2^5, 0.3) 0.1567	**72.95 ± 4.14** (10^-3, 2^5, 2) 0.1554
Yeast3 (500×8, 984×8)	79.94 ± 4.81 (10^1, 2^-3, 5) 7.5527	86.51 ± 4.98 (10^-2, 2^2) 0.9194	85.68 ± 4.67 (10^-2, 2^2) 0.9642	86.45 ± 5.07 (10^-2, 2^2, 1) 0.9509	88.25 ± 2.8 (10^1, 2^-1, 0.5) 1.1206	90.91 ± 4.1 (10^-2, 2^0) 0.6599	**91.54 ± 4.56** (10^-1, 2^0) 0.72	90.62 ± 3.94 (10^-2, 2^0, 1) 0.7324	90.26 ± 3.81 (10^-1, 2^3, 1.5) 0.7326
Yeast1vs7 (200×8, 259×8)	**70.21 ± 22.98** (10^2, 2^-2, 3) 0.5025	68.92 ± 18.56 (10^-3, 2^2) 0.0569	63.89 ± 17.07 (10^-3, 2^1) 0.0612	68.92 ± 18.56 (10^-5, 2^2, 0.3) 0.0583	67.46 ± 17.9 (10^1, 2^-1, 0.5) 0.0683	59.54 ± 15.38 (10^-3, 2^3) 0.0425	66.08 ± 6.35 (10^-2, 2^2) 0.0456	58.7 ± 14.91 (10^-3, 2^3, 0.3) 0.0452	62.76 ± 10.89 (10^-2, 2^2, 0.5) 0.0456
Yeast2vs8 (250×8, 233×8)	58.12 ± 11.98 (10^2, 2^-2, 7) 0.4211	**61.67 ± 16.24** (10^-5, 2^2) 0.0468	**61.67 ± 16.24** (10^-5, 2^2) 0.05	**61.67 ± 16.24** (10^-5, 2^2, 0.3) 0.0546	**61.67 ± 16.24** (10^0, 2^-1, 0.3) 0.0601	56.09 ± 14.83 (10^-5, 2^-3) 0.0347	58.73 ± 14.92 (10^-5, 2^-3) 0.0368	56.74 ± 14.94 (10^-5, 2^-3, 1) 0.0373	58.72 ± 14.26 (10^-5, 2^-3, 0.5) 0.0377
Ecoli0137vs26 (180×7, 131×7)	**98.35 ± 2.71** (10^-5, 2^-3, 3) 0.1356	92.09 ± 10.82 (10^-5, 2^4) 0.0168	95 ± 11.18 (10^0, 2^0) 0.0185	95 ± 11.18 (10^0, 2^0, 1) 0.018	96.33 ± 4.65 (10^2, 2^4, 0.5) 0.0208	94.83 ± 4.29 (10^-1, 2^0) 0.0138	91.7 ± 3.6 (10^-1, 2^4) 0.0118	94.83 ± 4.29 (10^-1, 2^0, 0.1) 0.0117	96.25 ± 8.39 (10^0, 2^0, 1.5) 0.0122
Australian-Credit (300×14, 390×14)	86.39 ± 6.07 (10^3, 2^4, 3) 1.1943	86.44 ± 5.21 (10^-2, 2^3) 0.1245	86.64 ± 4.94 (10^-1, 2^3) 0.1281	86.53 ± 2.58 (10^-5, 2^4, 0.1) 0.127	85.82 ± 5.1 (10^-2, 2^3, 0.1) 0.1383	85.16 ± 4.57 (10^-1, 2^4) 0.1017	86.77 ± 6.01 (10^0, 2^5) 0.1096	86.08 ± 5.62 (10^0, 2^5, 0.7) 0.1095	**86.85 ± 5.92** (10^0, 2^5, 1) 0.1092
Monk2 (300×7, 301×7)	50.26 ± 1.69 (10^3, 2^2, 3) 0.6979	49.51 ± 1.3 (10^-1, 2^2) 0.0769	51.9 ± 4.26 (10^0, 2^2) 0.0831	51.03 ± 3.02 (10^5, 2^2, 0.1) 0.087	45.33 ± 6.26 (10^1, 2^3, 0.3) 0.0913	51.67 ± 2.75 (10^-1, 2^3) 0.0592	55.86 ± 6.32 (10^-1, 2^3) 0.0656	47.99 ± 6.49 (10^-3, 2^3, 0.1) 0.0631	**57.4 ± 6.05** (10^-1, 2^3, 0.5) 0.0656

classes is used in the proposed function which leads to better fuzzy membership for class imbalanced data.

For the experiments on synthetic datasets, one can observe in Table 2 that our proposed method RFLSTSVM-CIL is not performing better for all the synthetic datasets but having the least ranks in most of the datasets in Table 3 which justifies its robustness for different sets of data. For noisy data, our proposed method is having ranks as 1, 2, 3 and 2 out of 9 methods for 04clover5z-600-5-60-BI, 03subcl5-600-5-30-BI, 03subcl5-600-5-50-BI and 03subcl5-600-5-60-BI datasets respectively. Also, the training time of our proposed approach is lesser as compared to the existing algorithms in Table 2.

In real world datasets, our proposed RFLSTSVM-CIL is taking less training time with least rank as shown in Table 5 and 6 respectively. It is observable that the proposed algorithm with the existing fuzzy functions i.e., FLSTSVM$_{\text{lin}}$-CIL and FLSTSVM$_{\text{exp}}$-CIL also perform better in comparison to the traditional approaches. The average ranks of FLSTSVM$_{\text{lin}}$-CIL and FLSTSVM$_{\text{exp}}$-CIL are lesser in comparison to EFSVM, TWSVM, FTWSVM$_{\text{lin}}$, FTWSVM$_{\text{exp}}$ and UTSVM in Table 3 and Table 6. In comparison to the existing algorithms, our proposed RFLSTSVM-CIL takes more computation time in comparison to LSTSVM. This is due to the additional computation for calculating the fuzzy membership values in the proposed approach.

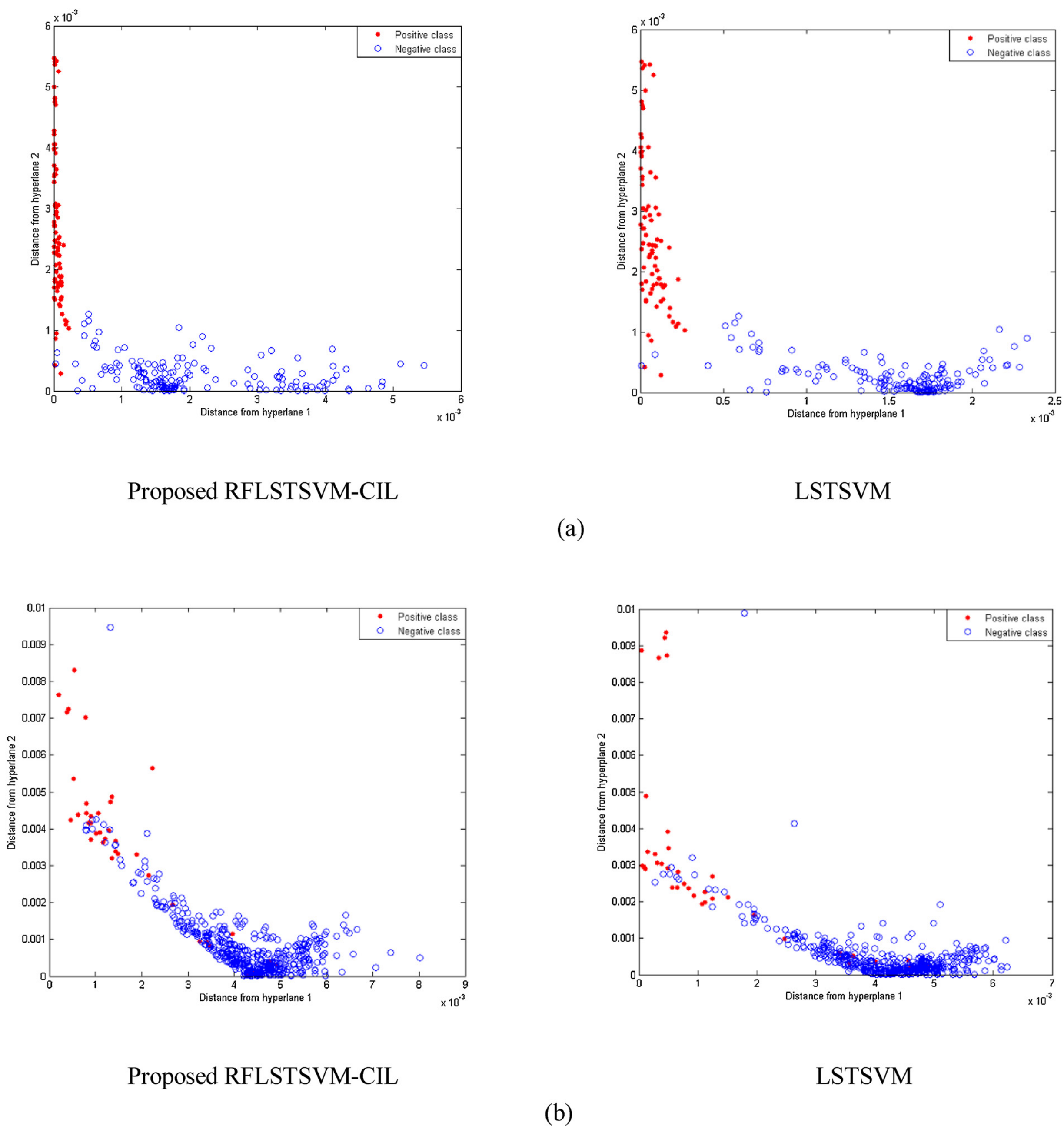
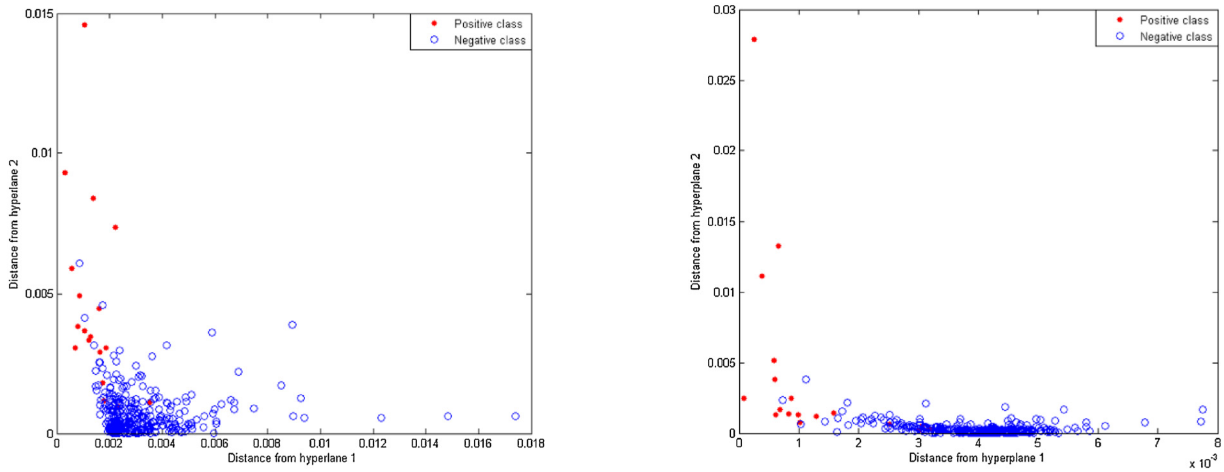


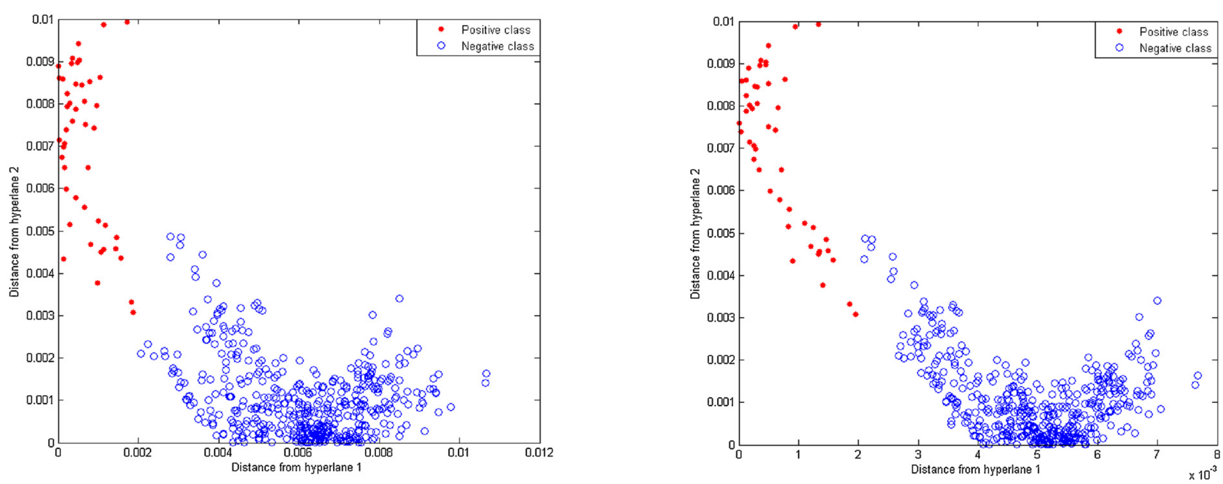
Fig. 3. Performance comparison of proposed RFLSTSVM-CIL with LSTSVM on the basis of distance from the hyperplanes for classification on (a) Monk2, (b) Yeast-0-2-5-6-vs_3-7-8-9, (c) Abalone9-18 and (d) Vowel dataset using Gaussian kernel.



Proposed RFLSTSVM-CIL

LSTSVM

(c)



Proposed RFLSTSVM-CIL

LSTSVM

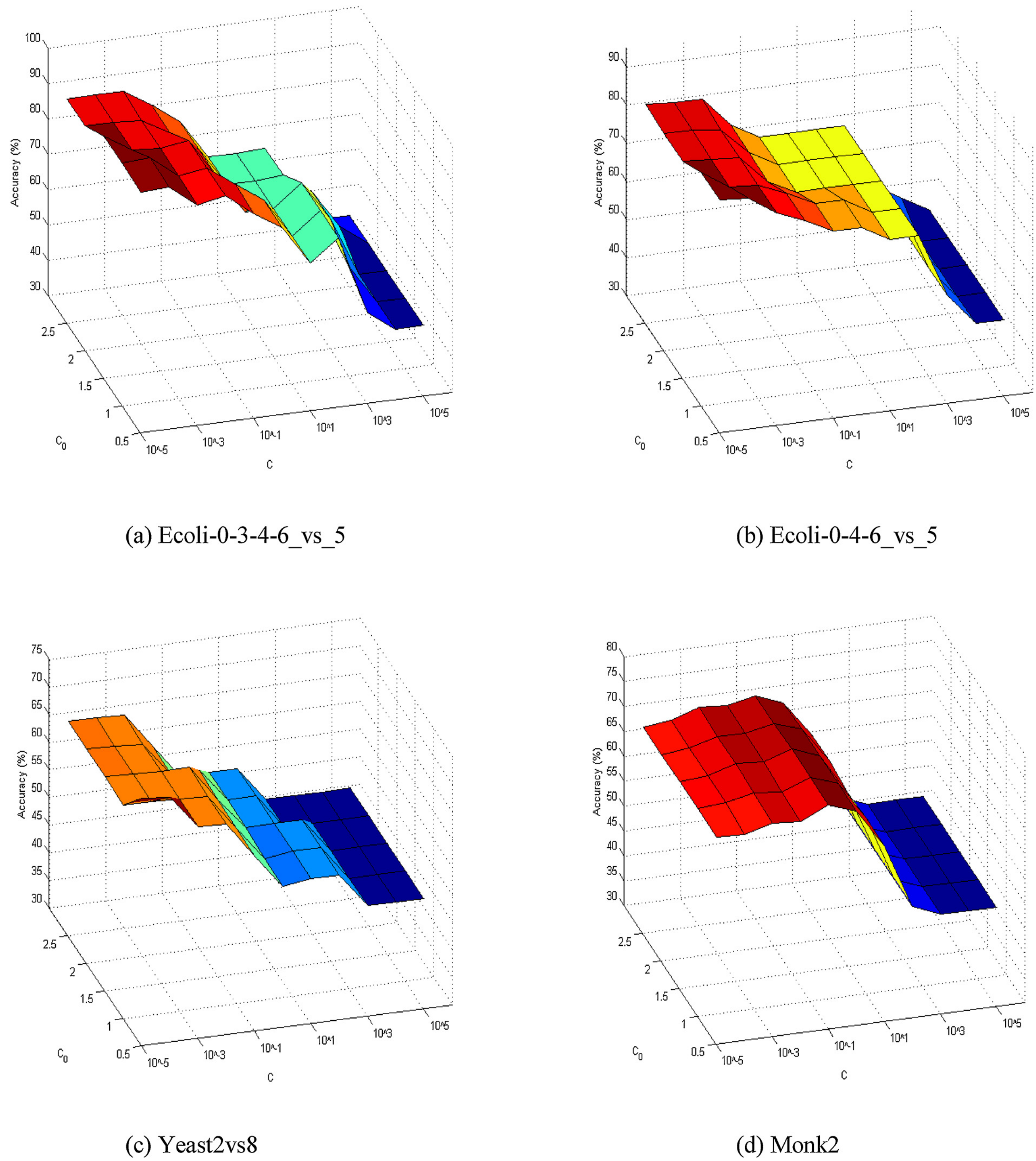
(d)

Fig. 3. (*Continued*)

The performance of the proposed method is compared with LSTSVM for showing the effect of our proposed fuzzy membership in Fig. 3 for Monk2, Yeast-0-2-5-6-vs_3-7-8-9, Abalone9-18 and Vowel datasets. The figures show the distance of the data points with the two hyperplanes. It is observable from Fig. 3(a)–(d) that the data points of the positive class are nearer to the positive class hyperplane and away from the negative class hyperplane. This justifies the fact the proposed fuzzy membership function is efficient in calculating the proper fuzzy membership value for imbalance datasets. The insensitivity analysis of the proposed RFLSTSVM-CIL to the parameters C and C_0 is shown in Fig. 4 for Ecoli-0-3-4-6_vs_5, Ecoli-0-4-6_vs_5, Yeast2vs8 and Monk2 datasets. It can be observed that RFLSTSVM-CIL performs better on lesser values of C as well as C_0.

6. Conclusions and future work

In this work, we propose a novel robust model for fuzzy based least squares twin support vector machine for class imbalanced datasets based on 2-norm of the slack variable. This makes the optimization problem strongly convex and implies a globally unique solution. We also proposed a novel fuzzy membership function specifically for class imbalance learning, which gives different range of fuzzy membership values for different datasets. The different range of the fuzzy membership helps in giving proper weights to the data points in different imbalance scenarios. Our proposed approach has shown good generalization performance with noisy data as compared to the existing algorithms. From the experiments, it is clear that our proposed approach is having the least



(a) Ecoli-0-3-4-6_vs_5

(b) Ecoli-0-4-6_vs_5

(c) Yeast2vs8

(d) Monk2

Fig. 4. Insensitivity performance of proposed RFLSTSVM-CIL for classification to the user specified parameters (C, C_0) using Gaussian kernel.

ranks for most of the datasets on the basis of AUC which justifies its robustness to different distribution of data. Further, our RFLSTSVM-CIL takes less computation time as compared to the existing fuzzy based algorithms for parallel and non-parallel support vector machines which justifies its applicability to real world applications. In future, the procedure of parameter selection for the fuzzy membership function can be improved by using heuristic based approaches. Our fuzzy membership function can be applied to various applications involving class imbalance. The proposed approach can be extended to multiclass classification, since in most of the multiclass classification problems there is imbalance of data belonging to the different classes.

Acknowledgements

This work was supported by Science and Engineering Research Board (SERB) as Early Career Research Award grant no. ECR/2017/000053 and Department of Science and Technology as Ramanujan Fellowship grant no. SB/S2/RJN-001/2016. We gratefully acknowledge the Indian Institute of Technology Indore for providing facilities and support. We are thankful to the Ministry of Human Resource Development (MHRD), Govt. of India for providing Teaching Assistantship (TA) fellowship to Mr. Bharat Richhariya.

Table 6
Rank comparison on the basis of AUC of proposed RFLSTSVM-CIL with EFSVM, TWSVM, $FTWSVM_{lin}$, $FTWSVM_{exp}$, UTSVM, LSTSVM, $FLSTSVM-CIL_{lin}$ and $FLSTSVM-CIL_{exp}$ using Gaussian kernel for classification on real world datasets.

Dataset	EFSVM	TWSVM	$FTWSVM_{lin}$	$FTWSVM_{exp}$	UTSVM	LSTSVM	$FLSTSVM-CIL_{lin}$	$FLSTSVM-CIL_{exp}$	RFLSTSVM-CIL
Cmc	4	2	1	3	9	7	8	6	5
Ecoli-0-1_vs_2-3-5	3	7.5	7.5	9	5	6	1	4	2
Ecoli-0-1_vs_5	4	5	6.5	6.5	9	1	3	8	2
Ecoli-0-1-4-7_vs_5-6	3	4	8	5.5	9	1.5	7	5.5	1.5
Ecoli-0-2-3-4_vs_5	2	6.5	6.5	8	9	2	4	5	2
Ecoli-0-2-6-7_vs_3-5	5.5	5.5	2	7	8	1	4	9	3
Ecoli-0-3-4-6_vs_5	1	8.5	3	5.5	8.5	4	5.5	7	2
Ecoli-0-4-6_vs_5	7	7	7	5	9	4	2.5	1	2.5
Ecoli-0-6-7_vs_3-5	7	9	5	6	8	2	1	4	3
Ecoli-0-6-7_vs_5	6	8.5	8.5	7	5	1	2	4	3
Ecoli4	8.5	4.5	3	4.5	8.5	6.5	1	6.5	2
Glass-0-4_vs_5	6	8.5	6	8.5	6	2.5	2.5	2.5	2.5
Glass2	5	7	8	9	6	2	4	1	3
Ripley	5	1.5	3	1.5	9	8	7	4	6
Yeast-0-2-5-6_vs_3-7-8-9	9	6	4	8	5	1.5	7	1.5	3
Yeast-0-3-5-9_vs_7-8	9	7	5	6	8	2.5	4	2.5	1
Yeast-0-5-6-7-9_vs_4	6	7.5	5	7.5	9	1	4	2	3
Ecoli-0-1-4-6_vs_5	3.5	8	8	8	3.5	3.5	3.5	3.5	3.5
Ecoli2	5	8	4	8	8	1.5	6	1.5	3
Vowel	9	3.5	6	3.5	8	5	2	7	1
Ecoli3	5	8	4	8	8	1.5	6	1.5	3
Abalone9-18	6	5	7	8	9	4	2	3	1
Vehicle 1	2	7	5	9	4	1	6	8	3
Vehicle2	3	8	6	9	5	2	4	7	1
Pima-Indians	4	7	6	8	9	3	2	5	1
Yeast3	9	6	8	7	5	2	1	3	4
Yeast1vs7	1	2.5	6	2.5	4	8	5	9	7
Yeast2vs8	7	2.5	2.5	2.5	2.5	9	5	8	6
Ecoli0137vs26	1	8	4.5	4.5	2	6.5	9	6.5	3
Australian-Credit	6	5	3	4	8	9	2	7	1
Monk2	6	7	3	5	9	4	2	8	1
Average Rank	5.1129	6.1613	5.2258	6.2742	6.9677	3.6613	3.9677	4.8871	2.7419

References

[1] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (3) (1995) 273–297.
[2] S.R. Gunn, Support vector machines for classification and regression, ISIS technical report 14 (1998) 85–86.
[3] E. Osuna, R. Freund, F. Girosit, Training support vector machines: an application to face detection, in: In Computer Vision and Pattern Recognition, 1997. Proceedings of IEEE Computer Society Conference on (June, 1997), 1997, pp. 130–136.
[4] P.J. Phillips, Support vector machines applied to face recognition, In Advances in Neural Information Processing Systems (1999) 803–809.
[5] P. Michel, R. El Kaliouby, Real time facial expression recognition in video using support vector machines ACM (November, 2003). Proceedings of the 5th International Conference on Multimodal Interfaces (2003) 258–264.
[6] M. Schmidt, H. Gish, Speaker identification via support vector classifiers, Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Proceedings of IEEE International Conference on (May, 1996) 1 (1996) 105–108.
[7] L. Khan, M. Awad, B. Thuraisingham, A new intrusion detection system using support vector machines and hierarchical clustering, VLDB J.-Int. J. Very Large Data Bases 16 (4) (2007) 507–521.
[8] J.A.K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (1999) 293–300.
[9] Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) 29 (2007) 905–910.
[10] M.A. Kumar, M. Gopal, Least squares twin support vector machines for pattern classification, Expert Syst. Appl. 36 (4) (2009) 7535–7543.
[11] D. Tomar, S. Agarwal, Hybrid feature selection based weighted least squares twin support vector machine approach for diagnosing breast cancer, hepatitis, and diabetes, Adv. Artif. Neural Syst. (2015) 1.
[12] X. Zhang, D. Jiang, T. Han, N. Wang, W. Yang, Y. Yang, Rotating machinery fault diagnosis for imbalanced data based on fast clustering algorithm and support vector machine, J. Sens. 2017 (2017) 15, Article ID 8092691.
[13] D. Tomar, S. Agarwal, Prediction of defective software modules using class imbalance learning, Appl. Comput. Intell. Soft Comput. (2016) 6.
[14] A. Chaudhuri, K. De, Fuzzy support vector machine for bankruptcy prediction, Appl. Soft Comput. 11 (2) (2010) 2472–2486.
[15] S. Zhang, S. Zhao, Y. Sui, L. Zhang, Single object tracking with fuzzy least squares support vector machine, IEEE Trans. Image Process. 24 (2015) 5723–5738.
[16] C.F. Lin, S.D. Wang, Fuzzy support vector machines, IEEE Trans. Neural Netw. 13 (2) (2002) 464–471.
[17] D. Tsujinishi, S. Abe, Fuzzy least squares support vector machines for multiclass problems, Neural Netw. 16 (5) (2003) 785–792.
[18] Y. Wang, S. Wang, K.K. Lai, A new fuzzy support vector machine to evaluate credit risk, IEEE Trans. Fuzzy Syst. 13 (6) (2005) 820–831.
[19] S. Balasundaram, M. Tanveer, On proximal bilateral-weighted fuzzy support vector machine classifiers, Int. J. Adv. Intell. Paradig. 4 (3–4) (2012) 199–210.
[20] W. An, M. Liang, Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises, Neurocomputing 110 (2013) 101–110.
[21] Z. Wu, H. Zhang, J. Liu, A fuzzy support vector machine algorithm for classification based on a novel PIM fuzzy clustering method, Neurocomputing 125 (2014) 119–124.
[22] S. Abe, Fuzzy support vector machines for multilabel classification, Pattern Recognit. 48 (6) (2015) 2110–2117.
[23] R. Wang, X. Zhang, W. Cao, Clifford fuzzy support vector machines for classification, Advances in Applied Clifford Algebras 26 (2) (2016) 825–846.
[24] M. Tanveer, M.A. Khan, S.S. Ho, Robust energy-based least squares twin support vector machines, Appl. Intell. 45 (1) (2016) 174–186.
[25] S.G. Chen, X.J. Wu, A new fuzzy twin support vector machine for pattern classification, Int. J. Mach. Learn. Cybern. (2017) 1–12.
[26] R. Batuwita, V. Palade, FSVM-CIL: fuzzy support vector machines for class imbalance learning, IEEE Trans. Fuzzy Syst. 18 (3) (2010) 558–571.
[27] G. Cohen, M. Hilario, C. Pellegrini, One-class support vector machines with a conformal kernel. A case study in handling class imbalance Springer, Berlin, Heidelberg, Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR) (August, 2004) (2004) 850–858.
[28] B.X. Wang, N. Japkowicz, Boosting support vector machines for imbalanced data sets, Knowl. Inf. Syst. 25 (1) (2010) 1–20.
[29] X. Gu, T. Ni, H. Wang, New fuzzy support vector machine for the class imbalance problem in medical datasets classification, Sci. World J. (2014).
[30] Y. Zhang, P. Fu, W. Liu, G. Chen, Imbalanced data classification based on scaling kernel-based support vector machine, Neural Comput. Appl. 25 (3–4) (2014) 927–935.
[31] Q. Wang, A hybrid sampling SVM approach to imbalanced data classification, in: Abstract and Applied Analysis (June 2014), Hindawi Publishing Corporation, 2014.
[32] X. Hua, S. Ding, Weighted least squares projection twin support vector machines with local information, Neurocomputing 160 (2015) 228–237.
[33] J.S. Sartakhti, N. Ghadiri, H. Afrabandpey, N. Yousefnezhad, Fuzzy Least Squares Twin Support Vector Machines, Arxiv Preprint Arxiv:1505.05451, 2015.

[34] T. Razzaghi, O. Roderick, I. Safro, N. Marko, Multilevel weighted support vector machine for classification on healthcare data with missing values, PloS one 11 (5) (2016), e0155119.
[35] H.L. Dai, Class imbalance learning via a fuzzy total margin based support vector machine, Appl. Soft Comput. 31 (2015) 172–184.
[36] S. Kim, Weighted K-means support vector machine for cancer prediction, Springerplus 5 (1) (2016) 1162.
[37] D. Tomar, S. Singhal, S. Agarwal, Weighted least square twin support vector machine for imbalanced dataset, Int. J. Database Theory Appl. 7 (2) (2014) 25–36.
[38] D. Tomar, S. Agarwal, An effective weighted multi-class least squares twin support vector machine for imbalanced data classification, Int. J. Comput. Intell. Syst. 8 (4) (2015) 761–778.
[39] Q. Fan, Z. Wang, D. Li, D. Gao, H. Zha, Entropy-based fuzzy support vector machine for imbalanced datasets, Knowl.-Based Syst. 115 (2017) 87–99.
[40] J.M. Keller, D.J. Hunt, Incorporating fuzzy membership functions into the perceptron algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1985) 693–699.
[41] G.H. Golub, C.F. Van Loan, Matrix Computations, 3rd ed., The John Hopkins University Press, 1996.
[42] Z. Qi, Y. Tian, Y. Shi, Twin support vector machine with universum data, Neural Netw. 36 (2012) 112–119.
[43] J. Huang, C.X. Ling, Using AUC and accuracy in evaluating learning algorithms, IEEE Trans. Knowl. Data Eng. 17 (3) (2005) 299–310.
[44] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Mult.-Valued Logic Soft Comput. 17 (2011).
[45] K. Napierała, J. Stefanowski, S. Wilk, Learning from imbalanced data in presence of noisy and borderline examples Springer, Berlin, Heidelberg, International Conference on Rough Sets and Current Trends in Computing (June, 2010) (2010) 158–167.
[46] Y.H. Shao, C.H. Zhang, X.B. Wang, N.Y. Deng, Improvements on twin support vector machines, IEEE Trans. Neural Networks 22 (6) (2011) 962–968.
[47] P.M. Murphy, D.W. Aha, UCI Repository of Machine Learning Databases, University of California, Irvine, 1992 http://www.ics.uci.edu/∼mlearn.
[48] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (January) (2006) 1–30.
[49] K. Li, H. Ma, A fuzzy twin support vector machine algorithm, Int. J. Appl. Innov. Eng. Manag. (IJAIEM) 2 (3) (2013) 459–465.
[50] B.B. Gao, J.J. Wang, Y. Wang, C.Y. Yang, Coordinate Descent Fuzzy Twin Support Vector Machine for Classification IEEE, Machine Learning and Applications (ICMLA), 2015. IEEE 14th International Conference (December, 2015) (2015) 7–12.