Intuitionistic Fuzzy Weighted Least Squares Twin SVMs

M. Tanveer, *Senior Member, IEEE*, M. A. Ganaie, A. Bhattacharjee, and C. T. Lin, *Fellow, IEEE*

***Abstract*—Fuzzy membership is an effective approach used in twin support vector machines (SVMs) to reduce the effect of noise and outliers in classification problems. Fuzzy twin SVMs (TWSVMs) assign membership weights to reduce the effect of outliers, however, it ignores the positioning of the input data samples and hence fails to distinguish between support vectors and noise. To overcome this issue, intuitionistic fuzzy TWSVM combined the concept of intuitionistic fuzzy number with TWSVMs to reduce the effect of outliers and distinguish support vectors from noise. Despite these benefits, TWSVMs and intuitionistic fuzzy TWSVMs still suffer from some drawbacks as: 1) the local neighborhood information is ignored among the data points and 2) they solve quadratic programming problems (QPPs), which is computationally inefficient. To overcome these issues, we propose a novel intuitionistic fuzzy weighted least squares TWSVMs for classification problems. The proposed approach uses local neighborhood information among the data points and also uses both membership and nonmembership weights to reduce the effect of noise and outliers. The proposed approach solves a system of linear equations instead of solving the QPPs which makes the model more efficient. We evaluated the proposed intuitionistic fuzzy weighted least squares TWSVMs on several benchmark datasets to show the efficiency of the proposed model. Statistical analysis is done to quantify the results statistically. As an application, we used the proposed model for the diagnosis of Schizophrenia disease.**

***Index Terms*—Fuzzy membership, intuitionistic fuzzy number (IFN), least squares twin SVM, support vector machines (SVMs).**

Manuscript received 23 April 2021; revised 15 September 2021 and 13 February 2022; accepted 28 March 2022. Date of publication 10 June 2022; date of current version 16 June 2023. This work was supported in part by the Department of Science and Technology, Science and Engineering Research Board (SERB) funded Research Projects, Government of India, through Early Career Research Award Scheme under Grant ECR/2017/000053, through Ramanujan Fellowship Scheme, under Grant SB/S2/RJN-001/2016, and through Interdisciplinary Cyber Physical Systems (ICPS) Scheme under Grant DST/ICPS/CPS-Individual/2018/276; and in part by the Council of Scientific and Industrial Research (CSIR), New Delhi, India, through the Extra Mural Research (EMR) Scheme under Grant 22(0751)/17/EMR-II and SERB through Mathematical Research Impact-Centric Support (MATRICS) Scheme under Grant MTR/2021/000787. This article was recommended by Associate Editor Y. Tan. *(Corresponding author: M. Tanveer.)*

M. Tanveer, M. A. Ganaie, and A. Bhattacharjee are with the Department of Mathematics, Indian Institute of Technology Indore, Indore 453552, India (e-mail: mtanveer@iiti.ac.in; phd1901141006@iiti.ac.in; avijit.bhattacharjee1996@gmail.com).

C. T. Lin is with the Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: chintenglin@gmail.com; chin-teng.lin@uts.edu.au).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2022.3165879.

Digital Object Identifier 10.1109/TCYB.2022.3165879

I. Introduction

Support vector machines (SVMs) [1] have found applications across different domains, such as regression [2], [3], fault prediction [4], medical prediagnosis [5], anomaly detection [6], and multiview learning [7]. SVMs use maximum margin concept to minimize the generalization error. Different approaches, such as generalized SVM [8], incremental learning SVM [9], semisupervised learning SVM [10], and multiclass SVM [11] have been proposed to improve the performance of SVM. SVM implements the structural risk minimization principle to achieve better generalization performance. SVM solves a quadratic programming problem (QPP) to obtain the optimal hyperplane, however, solving QPP limits the applicability of the SVMs due to higher training time. To overcome computational issues, several approaches have been put forth, such as generalized eigenvalue proximal SVM (GEPSVM) [12] and twin SVM (TWSVM) [13]. In GEPSVM and TWSVM, two nonparallel hyperplanes are generated such that each hyperplane is closer to the samples of one class and as far as possible from the samples of the other class. Unlike solving a single large QPP in SVM, TWSVM solves two smaller size QPPs leading to better efficiency of the TWSVM. TWSVM is approximately four times faster than SVM [13]. Unlike SVM which implements the structural risk minimization principle, TWSVM implements the empirical risk minimization principle [14], [15]. Hence, invertibility issues need to be handled explicitly while applying TWSVM to classification problems.

Standard SVM is not feasible when support vectors are mixed with noise or the dataset is contaminated by outliers. To reduce the effect of noise, truncated hinge loss SVM [16] was proposed. SVM with the pinball loss function [17] maximized the quantile distance to reduce the effect of noise and outliers. To retain the sparsity of the models, ε-insensitive pinball loss SVM [17] ignored the samples in the insensitive zone. Generalized pinball loss SVMs [18] redefined the insensitive zone to overcome the issues of noise and outliers. In TWSVM variants multiple approaches, such as general pinball loss TWSVM [19], L_1 norm least squares TWSVM [20], twin parametric margin SVM [21], and twin parametric margin SVM with pinball loss [22] have been proposed to handle the issues of noise and outliers. In addition to different loss functions, fuzzy membership weights are used in fuzzy SVM (FSVM) [23]–[26] and fuzzy TWSVMs [27] to reduce the effect of noise and outliers. A novel fuzzy membership incorporated imbalance ratio for assigning the fuzzy weights in a model known as robust fuzzy least squares TWSVM for class imbalance learning (RFLSTSVM-CIL) [28]. Maximum margin of twin spheres SVM [29] has been used to handle the imbalance problems. According to a recent study [30], robust energy-based least squares TWSVM (RELSTSVM) [31] emerged as the best classifier among the TWSVM-based models. For details of the TWSVM models, we refer the readers to [32].

SVMs suffer from large computational complexity, hence, coordinate descent SVM (CDSVM) [33] uses the coordinate descent method to speed up the optimization. CDSVM assumes that all the samples are equally important for optimization which may not be feasible in presence of noise and outliers. To give appropriate weights, coordinate descent fuzzy TWSVM (CDFTWSVM) [34] appropriately weighted the samples via fuzzy membership to reduce the effect of noise and outliers. Fuzzy membership functions focus on the distance of each sample from the class center and may get confused while distinguishing among the support vectors and outliers. Hence, dual membership-based FSVM [35] are formulated. Although dual membership FSVM showed better performance compared to FSVM, however, the samples nearby the class center were assigned lower membership as compared to the samples far away from the class center [35]. Using intuitionistic fuzzy number (IFN) [36] and kernel function [37], each training sample is assigned a weight for better generalization performance. Intuitionistic fuzzy TWSVMs (IFTWSVMs) [38] assigns a pair of nonmembership and membership functions to reduce the effect of noise and correctly identify the noise from support vectors. However, IFTWSVM suffers from several drawbacks: 1) it ignores the local neighborhood structure of the training data samples; 2) it has large computational complexity as it solves two QPPs; and 3) it requires an external toolbox to solve the QPPs. To overcome these issues, we propose intuitionistic fuzzy weighted least squares TWSVM (IFW-LSTSVM). IFW-LSTSVM uses intraclass local weights via k-nearest neighbor (KNN)-based weighting technique to incorporate the local neighborhood structure and solves a system of linear equations instead of solving the QPPs which leads to a faster solution. The proposed IFW-LSTSVM uses the membership function to measure the distance of each sample from the class center and the nonmembership function to measure the relation between the number of samples in the neighborhood and number of inharmonic samples. The advantages of the proposed IFW-LSTSVM model are listed as follows.

1) Unlike TWSVM and IFTWSVM models which involve external toolbox for solving the QPPs, the proposed IFW-LSTSVM model solves a system of linear equations which does not require any external toolbox.
2) TWSVM, IFTWSVM, RFLSTSVM-CIL, and RELSTSVM models ignore the local neighborhood information, however, the proposed IFW-LSTSVM model exploits the local neighborhood information to improve the generalization performance.
3) The RFLSTSVM-CIL model uses the imbalance ratio and the distance of the samples for calculating the fuzzy membership weights, however, the proposed IFW-LSTSVM model uses distance of the samples as well as the heterogeneity of the samples for calculating fuzzy score values.
4) TWSVM and RELSTSVM models assume that all the samples are equally important for generating the optimal hyperplanes, however, such an assumption may not hold in real-world scenarios. Hence, the proposed IFW-LSTSVM model uses fuzzy score values to give appropriate weights to the samples.
5) TWSVM and RFLSTSVM-CIL models minimize the empirical risk which may lead to the issues of overfitting. Hence, the proposed IFW-LSTSVM minimizes the structural risk to avoid the issues of overfitting and improve the generalization performance.

The remainder of this article is organized as follows: Section II discusses the formulation of IFTWSVM, Section III discusses the concept of Intuitionistic fuzzy sets, the weight generation method, and the formulation of the proposed IFW-LSTSVM model for linear and nonlinear cases. Section IV discusses the computational complexity of the proposed IFW-LSTSVM models and Section V evaluates the performance of the proposed IFW-LSTVM models and the baseline models. Finally, we conclude this article in Section VI.

II. Related Work

We discuss the formulation of intuitionistic fuzzy TWSVMs [38] in this section.

A. Intuitionistic Fuzzy Twin Support Vector Machines

The main idea of IFTWSVM [38] is triggered by the IFN [37] and fuzzy TWSVM [27] concepts. Unlike fuzzy-based SVMs, the IFTWSVM generates the optimal separating hyperplanes via solving two QPPs of smaller size instead of a single large QPP. To subsidize the effect of outliers, the samples are weighted based on their distance from the centroid as well as the heterogeneity of the samples. IFTWSVM with nonlinear kernel finds the hyperplanes given by

$$K(x, A^t)w_1 + b_1 = 0, \text{ and } K(x, A^t)w_2 + b_2 = 0 \tag{1}$$

where x is a data sample, A is the training set, w_i and b_i are the weights and biases, respectively, for $i = 1, 2$.

The primal formulation of nonlinear IFTWSVM [38] is defined as

$$\begin{aligned} \min_{w_1,b_1,\xi} \quad & \frac{1}{2}\left\|K(X_1, A^t)w_1 + e_1 b_1\right\|^2 + \frac{C_1}{2}\|w_1\|^2 + C_2 S_2^t \xi \\ \text{s.t.} \quad & -\big(K(X_2, A^t)w_1 + e_2 b_1\big) + \xi \geq e_2, \xi \geq 0 \end{aligned} \tag{2}$$

and

$$\begin{aligned} \min_{w_2,b_2,\eta} \quad & \frac{1}{2}\left\|K(X_2, A^t)w_2 + e_2 b_2\right\|^2 + \frac{C_3}{2}\|w_2\|^2 + C_4 S_1^t \eta \\ \text{s.t.} \quad & \big(K(X_1, A^t)w_2 + e_1 b_2\big) + \eta \geq e_1, \eta \geq 0. \end{aligned} \tag{3}$$

Here, $C_1, C_2, C_3, C_4 > 0$ are hyperparameters; e_1 and e_2 are vector of ones of appropriate dimensions; X_1 and X_2 are the samples corresponding to the positive and negative class, respectively; $A^t = [X_1; X_2]^t$; ξ and η are slack variables; and S_i is the score values of the IFNs, for $i = 1, 2$.

The first term in each objective function makes the hyperplanes proximal to the samples of the corresponding class, the second term minimizes the structural risk, and the third term ensures that the weighted samples are at least at a distance of 1 from the samples of other class.

The corresponding Wolfe dual of (2) is given as

$$\begin{aligned}&\max_{\alpha} \ e_2^t\alpha - \frac{1}{2}\alpha^t P\left(R^t R + C_1 I\right)^{-1} P^t \alpha \\ &\text{s.t. } \ 0 \le \alpha \le C_2 S_2\end{aligned} \tag{4}$$

where α is the vector of Lagrange multipliers, $R = [K(X_1, A^t), e_1]$ and $P = [K(X_2, A^t), e_2]$.

Similarly, one can obtain the Wolfe dual of the problem (3) as

$$\begin{aligned}&\max_{\beta} \ e_1^t\beta - \frac{1}{2}\beta^t R\left(P^t P + C_3 I\right)^{-1} R^t \beta \\ &\text{s.t. } \ 0 \le \beta \le C_4 S_1\end{aligned} \tag{5}$$

where β is the vector of Lagrange multipliers.

One can obtain $u_1 = [w_1 \ \ b_1]^t$ and $u_2 = [w_2 \ \ b_2]^t$ as

$$u_1 = -\left(R^t R + C_1 I\right)^{-1} P^t \alpha, \quad u_2 = \left(P^t P + C_3 I\right)^{-1} R^t \beta.$$

The testing data samples are assigned the class labels based on their proximity to the optimal hyperplanes of each class. Thus, new test data point $x \in \mathbb{R}^n$ is assigned the class label as follows: $\arg\min_{i=1,2}(|w_i^t K(x, A^t) + b_i| / \sqrt{w_i^t K(A, A^t) w_i})$.

III. Proposed Work

Motivated by IFTWSVM [38] and least squares TWSVM [39], we propose a novel IFW-LSTSVM model for the classification problems.

1) Intuitionistic Fuzzy Set [38]: Consider a non empty set χ, a fuzzy set Z in universe χ is defined as

$$Z = \{(z, \mu_Z(z)) | z \in \chi\} \tag{6}$$

where $\mu_Z : \chi \to [0, 1]$ and $\mu_Z(z)$ is the degree of membership of $z \in \chi$. Unlike fuzzy sets which consider the membership for each sample, the intuitionistic fuzzy sets considers both the membership and nonmembership for each sample. Mathematically, an intuitionistic fuzzy set is given as

$$\tilde{Z} = \left\{\left(z, \mu_{\tilde{Z}}(z), \nu_{\tilde{Z}}(z)\right) | z \in \chi\right\} \tag{7}$$

where $\mu_{\tilde{Z}}(z)$ and $\nu_{\tilde{Z}}(z)$ are the degrees of membership and nonmembership values of $z \in \chi$, respectively, $\mu_{\tilde{Z}} : \chi \to [0, 1], \nu_{\tilde{Z}} : \chi \to [0, 1]$, and $0 \le \mu_{\tilde{Z}}(z) + \nu_{\tilde{Z}}(z) \le 1$. The degree of hesitation of $z \in \chi$ is given as

$$\pi_{\tilde{Z}}(z) = 1 - \mu_{\tilde{Z}}(z) - \nu_{\tilde{Z}}(z). \tag{8}$$

An IFN is defined as $\theta = (\mu_\theta, \nu_\theta)$, where $\mu_\theta \in [0, 1], \nu_\theta \in [0, 1]$, and $0 \le \mu_\theta + \nu_\theta \le 1$. $\theta^- = (0, 1)$ and $\theta^+ = (1, 0)$ are the smallest and largest IFNs, respectively. For a given $\theta = (\mu_\theta, \nu_\theta)$, the score value for the IFN is given as

$$S(\theta) = \mu_\theta - \nu_\theta. \tag{9}$$

However, for some IFNs the score function is impossible to determine. Thus, to avoid this problem the following function is used:

$$h(\theta) = \mu_\theta + \nu_\theta. \tag{10}$$

From (8) and (10), we obtain

$$h(\theta) + \pi(\theta) = 1. \tag{11}$$

If $S(\theta_1) = S(\theta_2)$ and $h(\theta_1) < h(\theta_2)$ then $\theta_1 < \theta_2$.

Using (9), the score function is given as

$$H(\theta) = \frac{1 - \nu(\theta)}{2 - \mu(\theta) - \nu(\theta)}. \tag{12}$$

The relationship between membership and nonmembership is defined as follows:

$$S(\theta_1) < S(\theta_2) \Rightarrow H(\theta_1) < H(\theta_2) \tag{13}$$

$$S(\theta_1) = S(\theta_2), h(\theta_1) < h(\theta_2) \Rightarrow H(\theta_1) < H(\theta_2). \tag{14}$$

2) Intra Class Weights: We incorporate intraclass local structure using the KNN-based weighting technique [40]. Given any pair of points (x_i, x_j) in the same class, the weight matrix can be defined as

$$W_{ij} = \begin{cases} \exp \frac{-\|x_i - x_j\|^2}{\sigma}, & \text{if } x_i \text{ is } k \text{ nearest neighbour of } x_j \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where σ is the kernel parameter.

Using (15) the intraclass weights can be found as:

$\rho_i^{(c)} = \sum_{j=1}^{k} W_{ij}$, where $\rho_i^{(c)}$ represents the weight of the ith sample of the cth class.

A. Linear IFW-LSTSVM

Linear IFW-LSTSVM generates two nonparallel hyperplanes for the binary class problems. Mathematically, the hyperplanes given as

$$w_1^t x + b_1 = 0 \quad and \quad w_2^t x + b_2 = 0. \tag{16}$$

To solve the binary class problems, we generate proximal plane for each class such that the samples of a given class are proximal to the samples of the corresponding plane. Mathematically, the optimization problems of the proposed linear IFW-LSTSVM are given as

$$\begin{aligned}&\min_{w_1, b_1, \xi} \ \frac{1}{2}\|\rho_1(X_1 w_1 + e_1 b_1)\|^2 + \frac{C_1}{2}\|S_2\xi\|^2 + \frac{C_2}{2}\left(\|w_1\|^2 + b_1^2\right) \\ &\text{s.t. } \ -(X_2 w_1 + e_2 b_1) + \xi = e_2\end{aligned} \tag{17}$$

and

$$\begin{aligned}&\min_{w_2, b_2, \eta} \ \frac{1}{2}\|\rho_2(X_2 w_2 + e_2 b_2)\|^2 + \frac{C_3}{2}\|S_1\eta\|^2 + \frac{C_4}{2}\left(\|w_2\|^2 + b_2^2\right) \\ &\text{s.t. } \ (X_1 w_2 + e_1 b_2) + \eta = e_1.\end{aligned} \tag{18}$$

Here, ρ_1 and ρ_2 are intraclass weight matrices for respective classes.

The first term in each optimization problem makes the plane proximal to the weighted samples of the corresponding class, the second term ensures that the weighted samples are atleast at a distance of 1 from the samples of the other class, and the last term minimizes the structural risk.

Substituting the equality constraints in the corresponding objective function of (17), we have

$$\min_{w_1,b_1,\xi} \frac{1}{2}\|\rho_1(X_1w_1+e_1b_1)\|^2 + \frac{C_1}{2}\|S_2(X_2w_1+e_2b_1+e_2)\|^2 + \frac{C_2}{2}\left(||w_1||^2+b_1^2\right). \quad (19)$$

Taking the gradient of (19) with respect to (w.r.t.) w_1, b_1 and then equating to zero, we obtain

$$(\rho_1X_1)^t\rho_1(X_1w_1+e_1b_1)+C_1(S_2X_2)^t \times(S_2(X_2w_1+e_2b_1+e_2))+C_2w_1=0$$
$$(\rho_1e_1)^t\rho_1(X_1w_1+e_1b_1)+C_1(S_2e_2)^t \times(S_2(X_2w_1+e_2b_1+e_2))+C_2b_1=0.$$

Rewriting the above two equations in matrix form, we obtain

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -C_1\left(C_1T^tT+R^tR+C_2I\right)^{-1}T^tS_2e_2 \quad (20)$$

where $R=\begin{bmatrix}\rho_1X_1 & \rho_1e_1\end{bmatrix}$, $T=\begin{bmatrix}S_2X_2 & S_2e_2\end{bmatrix}$, and I is identity matrix of appropriate dimension.

In a similar way, we obtain the second hyperplane as

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = C_3\left(C_3R^tR+T^tT+C_4I\right)^{-1}R^tS_1e_1 \quad (21)$$

where $R=\begin{bmatrix}S_1X_1 & S_1e_1\end{bmatrix}$ and $T=\begin{bmatrix}\rho_2X_2 & \rho_2e_2\end{bmatrix}$.

From (20) and (21), it is clear that the proposed IFW-LSTSVM solves system of equations to obtain the optimal separating hyperplanes which does not require any external toolbox.

A new test data point $x\in\mathbb{R}^n$ obtains the class label from the decision function given by: $\text{argmin}_{i=1,2}(|w_i^tx+b_i|/\|w_i\|)$.

B. Nonlinear IFW-LSTSVM

The data samples may not be linearly separable, hence, we follow the kernel trick to project the data into higher dimensional space. We use Gaussian kernel to project the data into higher dimensional space. Mathematically, IFW-LSTSVM with nonlinear kernel finds the hyperplanes given by

$$K(x,A^t)w_1+b_1=0 \quad \text{and} \quad K\left(x,A^t\right)w_2+b_2=0$$

here $K(,)$ denotes the kernel function.

The optimization problems of the proposed nonlinear IFW-LSTSVM are given as

$$\begin{aligned}\min_{w_1,b_1,\xi}\ & \frac{1}{2}\left\|\rho_1\left(K(X_1,A^t)w_1+e_1b_1\right)\right\|^2+\frac{C_1}{2}||S_2\xi||^2 + \frac{C_2}{2}\left(||w_1||^2+b_1^2\right)\\ \text{s.t.}\ & -\left(K(X_2,A^t)w_1+e_2b_1\right)+\xi=e_2 \end{aligned} \quad (22)$$

and

$$\begin{aligned}\min_{w_2,b_2,\eta}\ & \frac{1}{2}\left\|\rho_2\left(K(X_2,A^t)w_2+e_2b_2\right)\right\|^2+\frac{C_3}{2}||S_1\eta||^2 + \frac{C_4}{2}\left(||w_2||^2+b_2^2\right)\\ \text{s.t.}\ & \left(K(X_1,A^t)w_2+e_1b_2\right)+\eta=e_1. \end{aligned} \quad (23)$$

Here, each term of the objective function have the similar motive as given in the linear case. Moreover, each symbol has the same meaning as given in the linear case.

Similar to the linear case, we have

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -C_1(C_1T^tT+R^tR+C_2I)^{-1}T^tS_2e_2 \quad (24)$$

where $R=[\rho_1K(X_1,A^t), \quad \rho_1e_1]$ and $T=[S_2K(X_2,A^t), \quad S_2e_2]$.

In a similar way, the hyperplane corresponding to the other class is given as

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = C_3(C_3R^tR+T^tT+C_4I)^{-1}R^tS_1e_1 \quad (25)$$

where $R=[S_1K(X_1,A^t), S_1e_1]$ and $T=[\rho_2K(X_2,A^t), \rho_2e_2]$.

A new test data point $x\in\mathbb{R}^n$ obtains it is class label from the decision function given by: $\arg\min_{i=1,2}(|w_i^tK(x,A^t)+b_i|/\sqrt{w_i^tK(A,A^t)w_i})$.

The algorithm for IFW-LSTSVM is briefly given in Algorithm 1.

Algorithm 1 IFW-LSTSVM

1: Input: The training samples X_1, X_2.
2: Output: The weight vectors and bias for each class i.e., w_i, b_i for $i=1,2$.
3: Calculate ρ_i for samples of each class, for $i=1,2$.
4: Calculate score values S_i for samples of each class, for $i=1,2$, respectively.
5: Solve the equations (20) and (21) or equations (24) and (25) to obtain optimal hyperplanes for each class of linear and nonlinear cases, respectively.
6: Return the optimal hyperplanes w_i, b_i for each class, for $i=1,2$.

IV. Computational Complexity of the Proposed IFW-LSTSVM

In this section, we discuss the computational complexity of the proposed IFW-LSTSVM model.

A. Computational Complexity

We use big-O notation to analyse the time complexity of our model. Let n be the total number of training samples and $m=(n/2)$ be the number of samples present in each class. To compute the degree of membership, the proposed IFW-LSTSVM model involves computation of class center, computation of class radius, the distance of each sample from the class center, and measure of the degree of membership of each sample which require $O(1)+O(1)+O(m)+O(m)$ operations. To measure the degree of nonmembership, $O(m)+O(m)$ operations are required. Hence, the proposed IFW-LSTSVM model uses $O(m)$ operations for assigning the score values. Calculating the KNN, we can compute the intraclass weights in $O(2m^2\log(m))$ [40]. Similar to LSTSVM, the proposed IFW-LSTSVM model involves computation of matrix inverse of the order of m. In comparison with LSTSVM, no computation overhead is required while solving the optimization problems of the proposed IFW-LSTSVM model.

V. Experimental Results

In this section, we analyze the experimental results of the proposed IFW-LSTSVM model and the given baseline models (here, TWSVM, RFLSTSVM-CIL, IFTWSVM, and RELSTSVM).

A. Experimental Setup

For performance evaluation of the proposed IFW-LSTSVM and the baseline models, we use benchmark datasets from the fishereies dataset [41], UCI [42], and KEEL [43] repository. The experiments are done on a machine with MATLAB R2017a on system with 2 Intel Xeon Processor, 128 GB of RAM, and 4 TB of secondary storage. The dataset is divided into 70 : 30 ratio for training and testing the models, respectively. We used the grid search method to tune the hyper-parameters via five-fold cross-validation method. For IFTWSVM and the proposed IFW-LSTSVM model, we set the parameters as $C_1 = C_3$ and $C_2 = C_4$, and for TWSVM we used $C_1 = C_2$. For classification of nonlinear data, we used Gaussian kernel ($K(x_1, x_2) = \exp(-||x_1 - x_2||^2/\sigma^2)$), where σ is the kernel parameter. The hyperparameters corresponding to different models are chosen from the following range: $\sigma = [2^{-5}, 2^{-4}, \ldots, 2^4, 2^5]$, $C_i = [10^{-5}, 10^{-4}, 10^{-3}, \ldots, 10^3, 10^4, 10^5]$, for $i = 1, 2, 3, 4$. Also, we set range of $C_0 = [0.5, 1, 1.5, 2, 2.5]$ for RFLSTSVM-CIL and $E_1 = E_2 = 0.6$ for the RELSTSVM model. We used Z-score normalization to normalize the datasets.

We evaluated the performance of the proposed IFW-LSTSVM and the given baseline models using different metrics, such as area under the receiver operating characteristic (ROC) curve (AUC) or accuracy, sensitivity, specificity, and other measures which are defined as

$$\text{Accuracy, AUC} = \frac{TP + TN}{TP + FP + TN + FN} \tag{26}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN}, \text{Specificity} = \frac{TN}{TN + FP} \tag{27}$$

$$\text{F-measure} = \frac{2 \times \text{ Precision } \times \text{ Recall}}{\text{Precision} + \text{Recall}} \tag{28}$$

$$\text{G-mean} = \sqrt{\text{Precision } \times \text{ Recall}} \tag{29}$$

where TP, TN, FP, and FN are the true positive, true negative, false positive, and false negative, respectively.

The F-measure and G-mean analysis of the given classification models and the proposed IFW-LSTSVM model on KEEL and UCI datasets with Gaussian kernel is given in Fig.-s1 and Fig.-s2, respectively (Here, s1 and s2 denote the 1 and 2 index of the supplementary file). One can see that the proposed model is competitive to the given baseline classification models.

As an application, we used the proposed model for the diagnosis of schizophrenia patients. The data used in this study is obtained from the center for biomedical research excellence (COBRE) (http://fcon_1000.projects.nitrc.org/indi/retro/cobre.html). The data involves 72 subjects of Schizophrenia (38.1 ± 13.9 years old, range 18-65 years) and 74 subjects of healthy control (35.8 ± 11.5 years old, range 18-65 years). Image processing was done via CAT12 package (http://www.neuro.uni-jena.de/cat/) implemented in statistical parametric mapping (SPM) toolbox version 12 (https://www.fil.ion.ucl.ac.uk/spm/software/spm12/). The 3-D $T1$-weighted MRI scans were parcellated into gray matter (GM), white matter (WM), and cerebrospinal fluid, skull, scalp, and air cavities. With high-dimensional diffeomorphic anatomic registration through the exponentiated Lie algebra algorithm (DARTEL), the GM images were normalized into Montreal Neurological Institute (MNI) space. With 8-mm full-width-half-maximum Gaussian kernel, the smoothed GM images were generated. Since the dataset size is relatively Small, hence we divided the data into 50:50 ratio of training and testing. We used 10-fold cross validation to obtain the optimal parameters.

TABLE I
Diagnosis of Schizophrenia

Model	(AUC, Time(s))	(Sens., Spec.)	(F-measure, G-mean)
TWSVM	(69.96, 1.9)	(81.58, 58.33)	(73.81, 74.15)
RFLSTSVM-CIL	(71.27, 0.26)	(84.21, 58.33)	(75.29, 75.72)
IFTWSVM	(68.79, 1.94)	(73.68, 63.89)	(70.89, 70.94)
RELSTSVM	(71.27, 0.19)	(84.21, 58.33)	(75.29, 75.72)
IFW-LSTSVM	(**72.81**, 0.41)	(78.95, 66.67)	(75, 75.09)

Sens. denotes Sensitivity, Spec. denotes Specificity and bold face denotes best performance.

B. Diagnosis of Schizophrenia

To show the application of the proposed model on real-world applications, we evaluated the proposed model for the diagnosis of the Schizophrenia disease. Table I gives the performance of the baseline models and the proposed IFW-LSTSVM model. One can see that the IFTWSVM achieved AUC with 68.79%, followed by TWSVM with 69.96%. Both RFLSTSVM-CIL and RELSTSVM models achieve 71.27% AUC. However, among the given models the proposed IFW-LSTSVM model achieved the highest 72.81% AUC. This demonstrates that the proposed IFW-LSTSVM model is better compared to baseline models.

C. Experimental Results and Discussion

In this section, we analyze the experimental results of the given baseline models and the proposed IFW-LSTSVM model with linear and Gaussian kernel on UCI and KEEL benchmark datasets.

We evaluate the experimental results statistically via the Friedman test and posthoc test. The Friedman Statistic is given by $\chi^2 = ([12 \times N]/[k \times (k+1)])[\sum_{j=1}^{k} R_j^2 - ([k(k+1)^2]/4)]$, where k is the number of algorithms and N is the number of datasets.

The better statistic F_F is given as follows: $F_F = ([(N-1)\chi_F^2]/[N(k-1) - \chi_F^2])$, where F_F is distributed according to F-distribution with $(N-1, (N-1)(k-1))$ degrees of freedom.

Under the Null hypothesis of the Friedman test, all the models are performing equally and their average ranks are equal. The null hypothesis is rejected if the calculated value of F_F is greater than the critical value of F-distribution table. If the null hypothesis is rejected, the Nemenyi posthoc test is used to give a pairwise comparison among the different models. We give this analysis for each of the linear and nonlinear kernel on the UCI and KEEL datasets as follows.

1) KEEL Datasets With Linear Kernel: The performance of the classification models on KEEL benchmark datasets with linear kernel is given in Table-sI. (Here, sI denotes the *I*th index of supplementary file). The Table gives the AUC, time in seconds, sensitivity and specificity of the models. One can see from Table-sI in the supplementary material, the performance of the proposed IFW-LSTSVM is better as compared to baseline models. Also, the average rank of the proposed IFW-LSTSVM is better compared to TWSVM and RFLSTSVM-CIL models. To analyze the performance of the TWSVM, RFLSTSVM-CIL, IFTWSVM, RELSTSVM, and proposed IFW-LSTSVM statistically, we use the Friedman test. The average ranks of the classification models TWSVM, RFLSTSVM-CIL, IFTWSVM, RELSTSVM, and the proposed IFW-LSTSVM are $3.13, 3.33, 2.78, 2.8,$ and 2.98, respectively. With simple calculations, we obtain $\chi_F^2 = 2.6768$ and $F_F = 0.6577$. F_F is distributed with $(k-1)$ and $(k-1)(N-1)$ degrees of freedom. From F-distribution table, with $k = 5$ and $N = 20$, the critical value $F_F(4, 76) = 2.495$. Since the calculated F_F value is smaller than the critical value of the F-distribution table, that is, $0.6577 < 2.495$, hence the Friedman test fails to show the significant difference between the proposed IFW-LSTSVM and the existing methods. However, one can see that the proposed IFW-LSTSVM achieved the highest average accuracy compared to the baseline methods.

2) KEEL Datasets With Gaussian Kernel: The performance of the classification models on KEEL benchmark datasets with Gaussian kernel is given in Table II. The Table gives the AUC, time in seconds, sensitivity, and specificity of the models. From Table II, one can see that the performance of the proposed IFW-LSTSVM is better as compared to baseline models. Compared to baseline models, the average rank of the proposed IFW-LSTSVM is better. The Friedman test is used to analyze the performance of the TWSVM, RFLSTSVM-CIL, IFTWSVM, RELSTSVM, and proposed IFW-LSTSVM statistically. The average ranks of the classification models TWSVM, IFTWSVM, and IFW-LSTSVM are $3.25, 3.55, 3.25, 2.73,$ and 2.23, respectively. With simple calculations, we obtain $\chi_F^2 = 9.2264$ and $F_F = 2.4769$. F_F is distributed with $(k-1)$ and $(k-1)(N-1)$ degrees of freedom. With $k = 5, N = 20$, and $F_F(4, 76) = 2.495$. Since $2.4769 < 2.495$, hence the Friedman test fails to detect the significant difference among the models. However, the proposed IFW-LSTSVM achieved better average accuracy and lower average rank as compared to the baseline models.

3) UCI Datasets With Linear Kernel: Table-sII in the supplementary material gives the experimental results of baseline models on UCI benchmark datasets with linear kernel. Table-sII in the supplementary material gives AUC, time in seconds, sensitivity, and specificity of the models. From Table-sII in the supplementary material, the performance of the proposed IFW-LSTSVM is better compared to baseline models. Also, the average rank of the proposed IFW-LSTSVM is lowest as compared to baseline models. Statistical analysis of the TWSVM, RFLSTSVM-CIL, IFTWSVM, RELSTSVM, and proposed IFW-LSTSVM is done via the Friedman test. The average ranks of the classification models TWSVM, RFLSTSVM-CIL, IFTWSVM, RELSTSVM, and the proposed IFW-LSTSVM are $4.04, 2.86, 2.71, 2.82,$ and 2.57, respectively. After simple calculations, we obtain $\chi_F^2 = 7.8546$ and $F_F = 2.1209$. F_F is distributed with $(k-1)$ and $(k-1)(N-1)$ degrees of freedom. With $k = 5$, $N = 14$, and $F_F(4, 52) = 2.545$. Since $2.1209 < 2.545$, hence the Friedman test fails to show the significant difference between the proposed IFW-LSTSVM and the existing methods. However, one can see that the proposed IFW-LSTSVM achieved the highest average accuracy and lower average rank compared to the baseline methods.

4) UCI Datasets With Gaussian Kernel: Table III gives the performance of the classification models on UCI benchmark datasets with Gaussian kernel. The table gives AUC, time in seconds, sensitivity, and specificity of the models. Except the RELSTSVM model, the average rank of the proposed IFW-LSTSVM is better. We use the Friedman test to evaluate the performance of the TWSVM, RFLSTSVM-CIL, IFTWSVM, RELSTSVM, and proposed IFW-LSTSVM statistically. The average ranks of the classification models TWSVM, RFLSTSVM-CIL, IFTWSVM, RELSTSVM, and proposed IFW-LSTSVM are $3.46, 3.71, 3.14, 2.29,$ and 2.39, respectively. With simple calculations, we obtain $\chi_F^2 = 8.6884$ and $F_F = 2.3873$. F_F is distributed with $(k-1)$ and $(k-1)(N-1)$ degrees of freedom. With $k = 5, N = 14$, and $F_F(4, 52) = 2.545$. Since $2.3873 < 2.545$, hence the Friedman test fails to show the significant difference between the proposed IFW-LSTSVM and the existing methods. From the results table, one can observe that the proposed IFW-LSTSVM achieved the highest average accuracy compared to the baseline models. Also, the average rank of the proposed IFW-LSTSVM models is better than baseline models except the RELSTSVM model.

D. Sensitivity of Hyperparameters

In this section, we analyze the effect of hyperparameters C_1 and C_2 on the generalization performance of the proposed IFW-LSTSVM. We evaluated the sensitivity on both UCI datasets and KEEL data in Figs. 1 and 2, respectively. In Fig. 1(a), one can see that the performance is better at lower values of C_1. However, in Fig. 1(b) the performance is better at higher values of C_1 and is better at lower values of C_2. Similarly, in Fig. 1(c) the performance of the proposed IFW-LSTSVM is better at middle range of C_1 values. Likewise in Fig. 2, one can see that varying the hyperparameters C_1 and C_2 leads to varying performance of the model. Hence, hyperparameters of the proposed IFW-LSTSVM models need to be chosen carefully for optimal generalization performance.

E. Effect of Number of Nearest Neighbors on the Performance of the Proposed IFW-LSTSVM Model

In this section, we analyze the effect of a number of nearest neighbors (k) on the performance of the proposed IFW-LSTSVM model. In Fig. 3(a), one can see that at higher values of k the performance is better compared to lower values of k. However, the reverse case is present in Fig. 3(b)

TABLE II
PERFORMANCE OF CLASSIFICATION MODELS ON KEEL DATASETS WITH GAUSSIAN KERNEL

Datasets (Dataset size)	TWSVM [13] (AUC,Time (s),Sens.,Spec.) (C_1, μ)	RFLSTSVM-CIL [28] (AUC,Time (s), Sens.,Spec.) (C_0, C_1, μ)	IFTWSVM [38] (AUC,Time (s), Sens., Spec.) (C_1, C_2, μ)	RELSTSVM [31] (AUC,Time (s), Sens., Spec.) (C_1, C_2, μ)	Proposed IFW-LSTSVM (AUC,Time (s),Sens.,Spec.) (C_1, C_2, k, μ)
aus	(84.2, 0.097, 84.04, 84.35)	(83.95, 0.039, 86.17, 81.74)	(84.82, 0.085, 86.17, 83.48)	(**87**, 0.048, 86.17, 87.83)	(85.5, 0.075, 84.04, 86.96)
(690×14)	(0.01, 32)	(1, 1, 32)	(1000, 10, 4)	(0.1, 1, 8)	(0.00001, 10, 4, 8)
brwisconsin	(96.05, 0.12, 93.33, 98.76)	(98.76, 0.039, 100, 97.52)	(98.76, 0.098, 100, 97.52)	(98.76, 0.05, 100, 97.52)	(**99.07**, 0.067, 100, 98.14)
(683×9)	(0.01, 8)	(1.5, 0.1, 16)	(1000, 100000, 4)	(0.00001, 100000, 4)	(0.01, 0.00001, 1, 16)
checkerboard_Data	(84.2, 0.095, 84.04, 84.35)	(83.95, 0.04, 86.17, 81.74)	(84.82, 0.081, 86.17, 83.48)	(**87**, 0.048, 86.17, 87.83)	(85.5, 0.073, 84.04, 86.96)
(690×14)	(0.01, 32)	(1, 1, 32)	(1000, 10, 4)	(0.1, 1, 8)	(0.00001, 10, 4, 8)
ecoli-0-1-4-7_vs_5-6	(**87.5**, 0.043, 75, 100)	(84.81, 0.01, 75, 94.62)	(86.42, 0.028, 75, 97.85)	(86.96, 0.008, 75, 98.92)	(**87.5**, 0.018, 75, 100)
(332×6)	(0.1, 32)	(1, 0.1, 32)	(10, 0.001, 0.5)	(0.1, 10, 2)	(0.00001, 0.00001, 1, 0.5)
glass2	(90.32, 0.03, 100, 80.65)	(88.71, 0.005, 100, 77.42)	(80.65, 0.023, 100, 61.29)	(80.65, 0.005, 100, 61.29)	(**95.97**, 0.01, 100, 91.94)
(214×9)	(10, 32)	(0.5, 0.1, 32)	(0.00001, 0.001, 32)	(0.00001, 0.01, 16)	(0.1, 0.00001, 4, 4)
haberman	(57.38, 0.023, 28, 86.76)	(**64.24**, 0.008, 52, 76.47)	(55.88, 0.031, 100, 11.76)	(61.38, 0.01, 36, 86.76)	(62.65, 0.012, 40, 85.29)
(306×3)	(0.1, 16)	(1.5, 0.01, 4)	(1000, 0.1, 0.125)	(0.1, 0.001, 32)	(0.00001, 0.001, 3, 4)
monk2	(90.16, 0.086, 92.19, 88.14)	(86.48, 0.026, 89.06, 83.9)	(87.06, 0.058, 95.31, 78.81)	(85.28, 0.037, 87.5, 83.05)	(**98.37**, 0.047, 98.44, 98.31)
(601×7)	(0.01, 4)	(0.5, 0.001, 4)	(100, 0.00001, 8)	(0.001, 0.00001, 8)	(0.1, 0.00001, 5, 8)
new-thyroid1	(94.44, 0.023, 88.89, 100)	(93.57, 0.006, 88.89, 98.25)	(94.44, 0.026, 88.89, 100)	(**99.12**, 0.005, 100, 98.25)	(94.44, 0.01, 88.89, 100)
(215×5)	(0.001, 1)	(0.5, 0.01, 1)	(0.0001, 0.0001, 4)	(0.01, 0.00001, 16)	(0.00001, 0.00001, 1, 4)
pima	(77.07, 0.128, 63.29, 90.85)	(62.77, 0.046, 98.73, 26.8)	(75.82, 0.091, 79.75, 71.9)	(**77.84**, 0.06, 75.95, 79.74)	(76.78, 0.08, 81.01, 72.55)
(768×8)	(0.1, 32)	(0.5, 0.1, 32)	(0.001, 1, 2)	(0.01, 1, 32)	(0.001, 0.1, 1, 2)
ripley	(90.74, 0.344, 87.43, 94.05)	(91.73, 0.118, 93.19, 90.27)	(92.03, 0.232, 91.62, 92.43)	(90.73, 0.151, 88.48, 92.97)	(**92.57**, 0.184, 91.62, 93.51)
(1250×2)	(1, 2)	(1.5, 10, 1)	(10, 100, 0.125)	(10, 100, 0.25)	(10000, 0.001, 3, 0.25)
shuttle-c0-vs-c4	(98.65, 0.793, 97.3, 100)	(98.65, 0.276, 97.3, 100)	(**100**, 0.602, 100, 100)	(98.65, 0.229, 97.3, 100)	(99.81, 0.403, 100, 99.61)
(1829×9)	(0.00001, 2)	(0.5, 0.0001, 16)	(0.0001, 0.00001, 16)	(0.00001, 0.00001, 4)	(0.00001, 0.00001, 1, 0.125)
sonar	(80.2, 0.021, 68.97, 91.43)	(80.2, 0.006, 68.97, 91.43)	(78.77, 0.022, 68.97, 88.57)	(**83.35**, 0.004, 72.41, 94.29)	(81.63, 0.011, 68.97, 94.29)
(208×60)	(0.001, 16)	(0.5, 0.001, 16)	(0.001, 0.01, 32)	(0.001, 0.00001, 16)	(0.001, 0.0001, 1, 16)
vehicle2	(**100**, 0.153, 100, 100)	(**100**, 0.063, 100, 100)	(98.19, 0.139, 100, 96.37)	(**100**, 0.048, 100, 100)	(99.48, 0.112, 100, 98.96)
(846×18)	(10, 16)	(0.5, 1, 8)	(0.001, 0.01, 32)	(1, 0.00001, 16)	(1, 0.00001, 2, 16)
votes	(**96.87**, 0.036, 96.3, 97.44)	(96.23, 0.019, 96.3, 96.15)	(95.58, 0.037, 96.3, 94.87)	(96.23, 0.011, 96.3, 96.15)	(96.23, 0.021, 96.3, 96.15)
(435×16)	(0.01, 16)	(0.5, 10, 32)	(1, 1, 32)	(0.01, 0.00001, 32)	(0.1, 0.01, 1, 16)
vowel	(79.63, 0.222, 59.26, 100)	(81.48, 0.089, 62.96, 100)	(94.82, 0.202, 92.59, 97.05)	(81.3, 0.081, 62.96, 99.63)	(**97.23**, 0.145, 96.3, 98.15)
(988×10)	(0.0001, 1)	(0.5, 0.0001, 2)	(0.00001, 0.001, 1)	(0.00001, 0.00001, 2)	(0.00001, 0.00001, 1, 1)
yeast-0-5-6-7-9_vs_4	(76.96, 0.087, 58.82, 95.1)	(75.15, 0.027, 70.59, 79.72)	(71.37, 0.066, 58.82, 83.92)	(73.26, 0.024, 64.71, 81.82)	(**79**, 0.043, 70.59, 87.41)
(528×8)	(1, 16)	(1, 0.01, 32)	(100, 10, 16)	(0.00001, 1000, 4)	(0.00001, 0.01, 5, 16)
yeast1	(99.12, 1.991, 98.56, 99.67)	(**100**, 0.676, 100, 100)	(**100**, 1.341, 100, 100)	(99.92, 0.778, 100, 99.84)	(94.13, 1.026, 93.14, 95.12)
(2968×8)	(0.01, 0.5)	(1, 0.0001, 0.5)	(0.001, 0.00001, 0.5)	(0.00001, 0.001, 0.25)	(0.01, 0.0001, 1, 1)
yeast2vs8	(70.42, 0.096, 50, 90.85)	(72.89, 0.017, 50, 95.77)	(**86.8**, 0.054, 75, 98.59)	(74.3, 0.017, 50, 98.59)	(75, 0.028, 50, 100)
(483×8)	(10, 8)	(1, 0.1, 32)	(10, 1, 16)	(0.00001, 10, 4)	(0.00001, 10, 3, 16)
yeast3	(89.67, 0.559, 86.15, 93.19)	(74.9, 0.194, 96.92, 52.88)	(81.75, 0.456, 67.69, 95.81)	(**93.63**, 0.168, 95.38, 91.88)	(91.36, 0.285, 87.69, 95.03)
(1484×8)	(1, 16)	(1.5, 1, 32)	(10000, 10000, 32)	(0.0001, 0.01, 16)	(0.001, 0.1, 6, 16)
yeast5	(90.25, 1.001, 83.33, 97.16)	(92.21, 0.192, 87.5, 96.93)	(**94.53**, 0.478, 91.67, 97.4)	(92.45, 0.167, 87.5, 97.4)	(84.83, 0.295, 70.83, 98.82)
(1484×8)	(1, 8)	(2.5, 0.001, 8)	(10000, 10000, 16)	(1, 10000, 16)	(10000, 100000, 2, 32)
Average Accuracy	86.69	85.53	87.13	87.39	**88.85**
Average Rank	3.25	3.55	3.25	2.73	**2.23**

TABLE III
PERFORMANCE OF CLASSIFICATION MODELS ON UCI DATASETS WITH GAUSSIAN KERNEL

Datasets (Dataset size)	TWSVM [13] (AUC,Time (s),Sens.,Spec.) (C_1, μ)	RFLSTSVM-CIL [28] (AUC,Time (s), Sens.,Spec.) (C_0, C_1, μ)	IFTWSVM [38] (AUC,Time (s), Sens., Spec.) (C_1, C_2, μ)	RELSTSVM [31] (AUC,Time (s), Sens., Spec.) (C_1, C_2, μ)	Proposed IFW-LSTSVM (AUC,Time (s),Sens.,Spec.) (C_1, C_2, k, μ)
bank	(72.85, 4.099, 62.03, 83.67)	(75.94, 1.712, 74.05, 77.83)	(76.57, 3.012, 67.72, 85.42)	(**78.37**, 2.134, 76.58, 80.17)	(76.51, 2.471, 71.52, 81.5)
(4521×15)	(0.00001, 32)	(2, 0.01, 16)	(0.001, 10, 16)	(0.001, 10, 8)	(0.00001, 1, 3, 32)
blood	(63.79, 0.109, 58.06, 69.51)	(64.28, 0.039, 37.1, 91.46)	(68.18, 0.083, 51.61, 84.76)	(**71.28**, 0.051, 64.52, 78.05)	(66.82, 0.074, 62.9, 70.73)
(748×3)	(0.1, 16)	(1, 1, 8)	(0.00001, 0.1, 8)	(0.01, 1, 16)	(1, 0.0001, 5, 2)
chess-krvkp	(98.53, 1.649, 98.08, 98.98)	(97.48, 0.648, 96.8, 98.17)	(79.3, 1.251, 98.93, 59.67)	(**99.16**, 0.95, 98.93, 99.39)	(98.95, 1.061, 98.72, 99.19)
(3196×35)	(0.1, 16)	(2, 1, 16)	(0.1, 1, 8)	(0.01, 0.0001, 32)	(1, 0.00001, 2, 32)
ilpd-indian-liver	(67.75, 0.06, 85.11, 50.39)	(66.39, 0.021, 80.85, 51.94)	(63.4, 0.048, 70.21, 56.59)	(66.11, 0.021, 70.21, 62.02)	(**68.42**, 0.034, 87.23, 49.61)
(583×8)	(1, 8)	(2, 0.1, 4)	(0.0001, 10, 4)	(0.1, 0.01, 32)	(1000, 100000, 1, 32)
mushroom	(**100**, 14.821, 100, 100)	(99.96, 7.121, 100, 99.92)	(**100**, 10.153, 100, 100)	(**100**, 9.531, 100, 100)	(99.84, 7.366, 99.92, 99.76)
(8124×20)	(0.00001, 2)	(0.5, 0.00001, 2)	(0.00001, 0.00001, 16)	(0.00001, 0.00001, 2)	(0.00001, 0.00001, 1, 2)
oocytes_merluccius_nucleus_4d	(80.86, 0.184, 75.25, 86.47)	(81.03, 0.067, 89.11, 72.95)	(67.08, 0.167, 94.06, 40.1)	(**83.87**, 0.096, 84.16, 83.57)	(83.34, 0.168, 80.2, 86.47)
(1022×40)	(100, 32)	(1, 1, 32)	(0.00001, 0.01, 32)	(0.1, 0.0001, 32)	(1, 0.01, 3, 16)
ringnorm	(98.03, 11.068, 98.62, 97.44)	(98.2, 5.555, 98.34, 98.06)	(98.5, 6.906, 97.97, 99.03)	(**98.73**, 7.24, 98.43, 99.03)	(97.87, 5.945, 99.08, 96.65)
(7400×19)	(0.00001, 2)	(0.5, 10000, 2)	(0.1, 10, 4)	(0.01, 10, 8)	(0.001, 0.01, 3, 2)
spambase	(93.41, 3.888, 93.08, 93.74)	(82.84, 1.641, 96.54, 69.14)	(91.37, 4.826, 93.65, 89.1)	(**94.12**, 2.301, 92.31, 95.94)	(94.01, 4.511, 92.31, 95.71)
(4601×56)	(0.01, 32)	(2, 0.01, 8)	(0.0001, 10, 16)	(0.01, 0.0001, 32)	(0.1, 0.001, 4, 32)
statlog-australian-credit	(52.38, 0.092, 38.57, 66.19)	(51.2, 0.032, 67.14, 35.25)	(50.31, 0.056, 14.29, 86.33)	(52.65, 0.044, 64.29, 41.01)	(**53.43**, 0.061, 45.71, 61.15)
(690×13)	(0.1, 16)	(1.5, 0.1, 32)	(100, 1, 2)	(10, 1000, 16)	(0.1, 10000, 2, 16)
statlog-german-credit	(67.8, 0.17, 61.04, 74.55)	(66.98, 0.068, 70.13, 63.84)	(**67.96**, 0.128, 64.94, 70.98)	(67.9, 0.081, 68.83, 66.96)	(69.7, 0.114, 67.53, 71.88)
(1000×23)	(0.1, 32)	(1.5, 0.01, 32)	(0.1, 100, 32)	(0.00001, 10, 32)	(0.1, 1, 6, 16)
statlog-heart	(81.4, 0.03, 88.89, 73.91)	(**82.79**, 0.006, 91.67, 73.91)	(81.7, 0.016, 91.67, 71.74)	(80.5, 0.006, 80.56, 80.43)	(**82.79**, 0.009, 91.67, 73.91)
(270×12)	(0.0001, 4)	(1.5, 0.0001, 4)	(1, 0.001, 16)	(1, 0.1, 32)	(10, 0.0001, 1, 4)
titanic	(70.94, 0.586, 51.89, 90)	(**72.67**, 0.26, 65.57, 79.78)	(69.76, 0.47, 49.53, 90)	(70.94, 0.406, 51.89, 90)	(**72.67**, 0.468, 65.57, 79.78)
(2201×2)	(0.1, 32)	(1.5, 100000, 0.5)	(0.01, 10000, 8)	(1, 0.00001, 0.03125)	(0.00001, 1, 6, 0.03125)
twonorm	(97.26, 11.314, 96.68, 97.83)	(97.21, 5.522, 96.68, 97.74)	(**97.71**, 6.776, 97.31, 98.1)	(97.62, 7.212, 97.13, 98.1)	(97.62, 6.424, 96.86, 98.37)
(7400×19)	(0.00001, 4)	(0.5, 0.0001, 4)	(10, 100, 16)	(0.00001, 10, 32)	(1000, 100000, 2, 16)
vertebral-column-2clases	(82.25, 0.029, 85.71, 78.79)	(77.65, 0.008, 75, 80.3)	(**83.28**, 0.018, 89.29, 77.27)	(79.92, 0.009, 75, 84.85)	(80.95, 0.012, 78.57, 83.33)
(310×5)	(0.01, 32)	(1, 10, 16)	(0.00001, 0.001, 32)	(1, 10, 8)	(1, 0.0001, 1, 4)
Average Accuracy	80.52	79.62	78.22	81.51	**81.64**
Average Rank	3.46	3.71	3.14	2.29	2.39

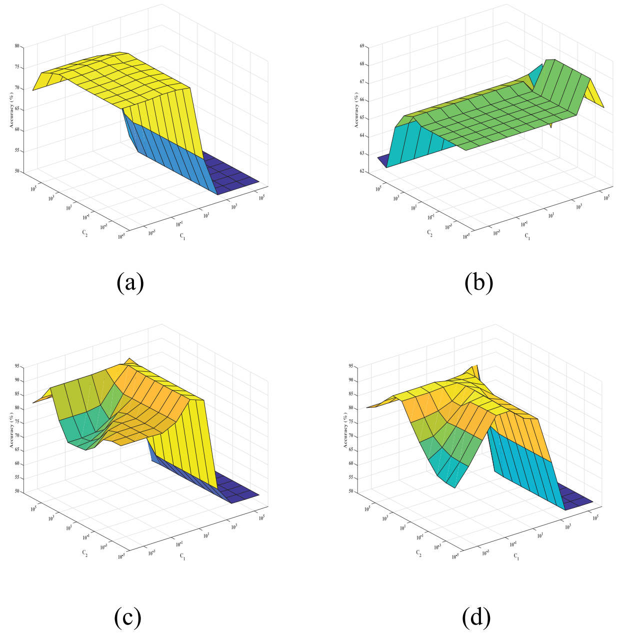

Fig. 1. Effect of parameters C_1 and C_2 on the performance of the proposed IFW-LSTSVM model on the UCI datasets. (a) Bank. (b) Blood. (c) Chess-krvkp. (d) Spambase.

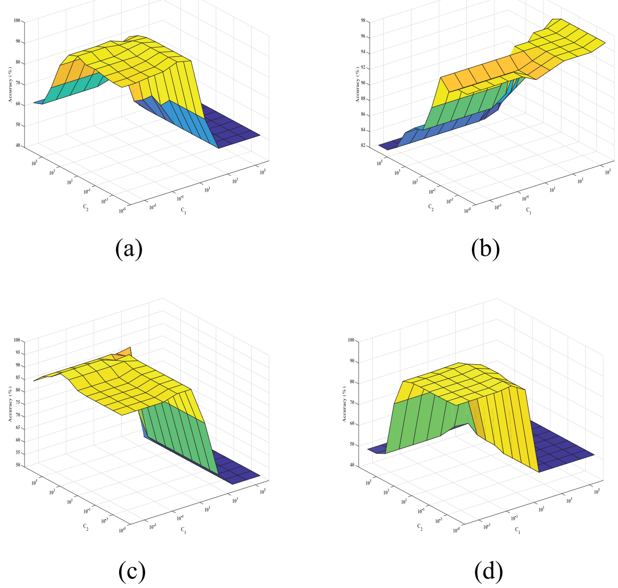

Fig. 2. Effect of parameters C_1 and C_2 on the performance of the proposed IFW-LSTSVM model on the KEEL datasets. (a) Vehicle2. (b) Votes. (c) Vowel. (d) Yeast5.

wherein the better performance is obtained at lower values of k and decreases with increase of k. From Fig. 3(c), one can see that the performance is varying at varying levels of k. In Fig. 3(d), one can observe that initially performance of the proposed IFW-LSTSVM model increases and then decreases with increase in k. However, the performance then again increases at higher values of k. Hence, the hyperparameter k needs to be chosen optimally for better performance of the proposed IFW-LSTSVM model.

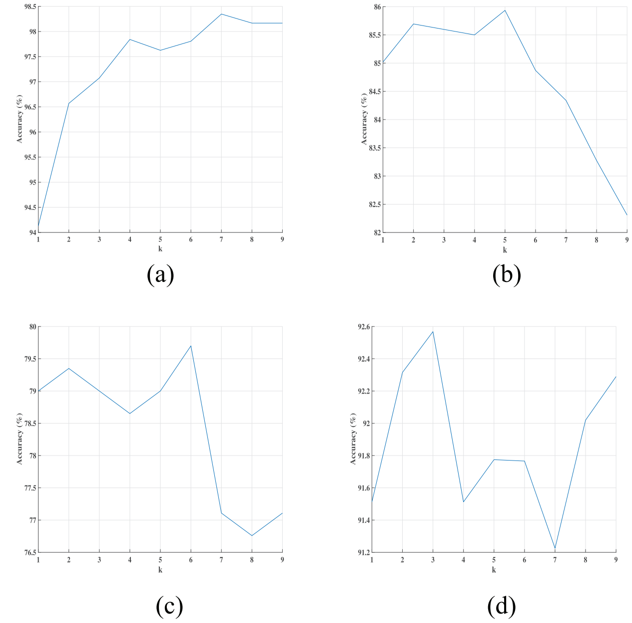

Fig. 3. Effect of parameter k on the performance of the proposed IFW-LSTSVM model. (a) Yeast1. (b) Aus. (c) Yeast-0-5-6-7-9_vs_4. (d) Ripley.

VI. Conclusion

In this article, we proposed intuitionistic fuzzy weighted least squares TWSVMs (IFW-LSTSVMs) for the classification problems. TWSVM, RFLSTSVM-CIL, and RELSTSVM models consider that all the data samples are equally important and ignore the neighborhood of the samples. However, the proposed IFW-LSTSVM model incorporates prior information about the neighborhood of a sample and gives appropriate weight to each sample which improves the performance in presence of noise and outliers. Both TWSVM and IFTWSVM models require the external toolbox for obtaining the optimal separating hyperplanes while as the proposed IFW-LSTSVM model solves system of linear equations which does not require any external toolbox. Although the IFTWSVM model uses fuzzy weight to assign the appropriate weights to the samples based on their sample distance from the centroid, however, it ignores the heterogeneity of the samples. The proposed IFW-LSTSVM model assigns a fuzzy score based on the sample distance from the centroid as well as the heterogeneity of the samples. As both the membership and nonmembership contribute to the fuzzy score this results in improved performance. The evaluation of the proposed IFW-LSTSVM model on several benchmark datasets shows the efficiency of the model. One can see that the proposed IFW-LSTSVM model shows better generalization on UCI and KEEL benchmark datasets. Also, the performance of the proposed IFW-LSTSVM model is better in the diagnosis of the Schizophrenia disease. In future, one can explore the use of interval-valued fuzzy sets, extensions of IFNs, such as triangular IFNs, trapezoidal IFNs, and interval-valued trapezoidal IFNs. Also, one can extend this algorithm for multiclass classification problems.

Acknowledgment

The authors gratefully acknowledge the Indian Institute of Technology Indore for providing facilities and support.

REFERENCES

[1] V. Vapnik, *Statistical Learning Theory*, vol. 1. New York, NY, USA: Wiley, 1998, p. 624.

[2] H. Wang, Y. Shi, L. Niu, and Y. Tian, "Nonparallel support vector ordinal regression," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3306–3317, Oct. 2017.

[3] H. Yu, J. Lu, and G. Zhang, "Continuous support vector regression for nonstationary streaming data," *IEEE Trans. Cybern.*, early access, Sep. 10, 2020, doi: 10.1109/TCYB.2020.3015266.

[4] S. Zhang, Y. Wang, M. Liu, and Z. Bao, "Data-based line trip fault prediction in power systems using LSTM networks and SVM," *IEEE Access*, vol. 6, pp. 7675–7686, 2017.

[5] H. Zhu, X. Liu, R. Lu, and H. Li, "Efficient and privacy-preserving online medical prediagnosis framework using nonlinear SVM," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 3, pp. 838–850, May 2017.

[6] X. Miao, Y. Liu, H. Zhao, and C. Li, "Distributed online one-class support vector machine for anomaly detection over networks," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1475–1488, Apr. 2019.

[7] S. Sun, X. Xie, and C. Dong, "Multiview learning with generalized Eigenvalue proximal support vector machines," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 688–697, Feb. 2019.

[8] M. Mohammadi, S. H. Mousavi, and S. Effati, "Generalized variant support vector machine," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 5, pp. 2798–2809, May 2021.

[9] J. Xu, C. Xu, B. Zou, Y. Y. Tang, J. Peng, and X. You, "New incremental learning algorithm with support vector machines," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 11, pp. 2230–2241, Nov. 2019.

[10] J. Wang, D. Yang, W. Jiang, and J. Zhou, "Semisupervised incremental support vector machine learning based on neighborhood kernel estimation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 10, pp. 2677–2687, Oct. 2017.

[11] X. Pang, Y. Xu, and X. Xiao, "A doubly sparse multiclass support vector machine with simultaneous feature and sample screening," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 11, pp. 6911–6925, Nov. 2021.

[12] O. L. Mangasarian and E. W. Wild, "Multisurface proximal support vector machine classification via generalized Eigenvalues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 69–74, Jan. 2006.

[13] Jayadeva, R. Khemchandani, and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 905–910, May 2007.

[14] Y.-H. Shao, C.-H. Zhang, X.-B. Wang, and N.-Y. Deng, "Improvements on twin support vector machines," *IEEE Trans. Neural Netw.*, vol. 22, no. 6, pp. 962–968, Jun. 2011.

[15] C. Zhang, Y. Tian, and N. Deng, "The new interpretation of support vector machines on statistical learning theory," *Sci. China Ser. A, Math.*, vol. 53, no. 1, pp. 151–164, 2010.

[16] Y. Wu and Y. Liu, "Robust truncated hinge loss support vector machines," *J. Amer. Stat. Assoc.*, vol. 102, no. 479, pp. 974–983, 2007.

[17] X. Huang, L. Shi, and J. A. K. Suykens, "Support vector machine classifier with pinball loss," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 984–997, May 2014.

[18] R. Rastogi, A. Pal, and S. Chandra, "Generalized pinball loss SVMs," *Neurocomputing*, vol. 322, pp. 151–165, Dec. 2018.

[19] M. Tanveer, A. Sharma, and P. N. Suganthan, "General twin support vector machine with pinball loss function," *Inf. Sci.*, vol. 494, pp. 311–327, Aug. 2019.

[20] S. Gao, Q. Ye, and N. Ye, "1-norm least squares twin support vector machines," *Neurocomputing*, vol. 74, no. 17, pp. 3590–3597, 2011.

[21] X. Peng, "TPMSVM: A novel twin parametric-margin support vector machine for pattern recognition," *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2678–2692, 2011.

[22] Y. Xu, Z. Yang, and X. Pan, "A novel twin support-vector machine with pinball loss," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 2, pp. 359–370, Feb. 2017.

[23] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 464–471, Mar. 2002.

[24] R. Batuwita and V. Palade, "FSVM-CIL: Fuzzy support vector machines for class imbalance learning," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 3, pp. 558–571, Jun. 2010.

[25] X. Yang, G. Zhang, J. Lu, and J. Ma, "A kernel fuzzy c-means clustering-based fuzzy support vector machine algorithm for classification problems with outliers or noises," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 1, pp. 105–115, Feb. 2011.

[26] R. K. Sevakula and N. K. Verma, "Compounding general purpose membership functions for fuzzy support vector machine under noisy environment," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1446–1459, Dec. 2017.

[27] S.-G. Chen and X.-J. Wu, "A new fuzzy twin support vector machine for pattern classification," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 9, pp. 1553–1564, 2018.

[28] B. Richhariya and M. Tanveer, "A robust fuzzy least squares twin support vector machine for class imbalance learning," *Appl. Soft Comput.*, vol. 71, pp. 418–432, Oct. 2018.

[29] Y. Xu, "Maximum margin of twin spheres support vector machine for imbalanced data classification," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1540–1550, Jun. 2017.

[30] M. Tanveer, C. Gautam, and P. N. Suganthan, "Comprehensive evaluation of twin SVM based classifiers on UCI datasets," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105617.

[31] M. Tanveer, M. A. Khan, and S.-S. Ho, "Robust energy-based least squares twin support vector machines," *Appl. Intell.*, vol. 45, no. 1, pp. 174–186, 2016.

[32] M. Tanveer, T. Rajani, R. Rastogi, Y. H. Shao, and M. A. Ganaie, "Comprehensive review on twin support vector machines," *Ann. Oper. Res.*, 2022, doi: 10.1007/s10479-022-04575-w.

[33] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear SVM," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 408–415.

[34] B.-B. Gao, J.-J. Wang, Y. Wang, and C.-Y. Yang, "Coordinate descent fuzzy twin support vector machine for classification," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, 2015, pp. 7–12.

[35] M.-M. Zhou, L. Li, and Y.-L. Lu, "Fuzzy support vector machine based on density with dual membership," in *Proc. Int. Conf. Mach. Learn. Cybern.*, vol. 2, 2009, pp. 674–678.

[36] Z. Xu and R. R. Yager, "Some geometric aggregation operators based on intuitionistic fuzzy sets," *Int. J. Gen. Syst.*, vol. 35, no. 4, pp. 417–433, 2006.

[37] M. Ha, C. Wang, and J. Chen, "The support vector machine based on intuitionistic fuzzy number and kernel function," *Soft Comput.*, vol. 17, no. 4, pp. 635–641, 2013.

[38] S. Rezvani, X. Wang, and F. Pourpanah, "Intuitionistic fuzzy twin support vector machines," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 11, pp. 2140–2151, Nov. 2019.

[39] M. A. Kumar and M. Gopal, "Least squares twin support vector machines for pattern classification," *Expert Syst. Appl.*, vol. 36, no. 4, pp. 7535–7543, 2009.

[40] X. Hua and S. Ding, "Weighted least squares projection twin support vector machines with local information," *Neurocomputing*, vol. 160, pp. 228–237, Jul. 2015.

[41] E. González-Rufino, P. Carrión, E. Cernadas, M. Fernández-Delgado, and R. Domínguez-Petit, "Exhaustive comparison of colour texture features and classification methods to discriminate cells categories in histological images of fish ovary," *Pattern Recognit.*, vol. 46, no. 9, pp. 2391–2407, 2013.

[42] D. Dua and C. Graff, 2017, "UCI Machine Learning Repository," UCI. [Online]. Available: http://archive.ics.uci.edu/ml

[43] J. Alcalá-Fdez *et al.*, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Multiple-Valued Logic Soft Comput.*, vol. 17, nos. 2–3, pp. 255–287, 2011.