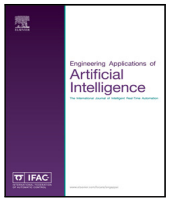
Engineering Applications of Artificial Intelligence 123 (2023) 106427

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Universum twin support vector machine with truncated pinball loss

Anuradha Kumari, M. Tanveer *, for the Alzheimer's Disease Neuroimaging Initiative[1]

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

ARTICLE INFO

Keywords:
twin support vector machine
Electroencephalogram signal
Alzheimer's disease
Truncated loss
Pinball loss
Convex–concave procedure

ABSTRACT

For classification problems, twin support vector machine with pinball loss (Pin-GTSVM) is noise insensitive and has better performance than twin support vector machine (TWSVM). However, it lacks sparsity in comparison to TWSVM. In this article, to maintain a trade-off between the noise insensitivity and sparsity of the model along with preserving the theoretical properties of pinball loss, we propose universum twin support vector machine with truncated pinball loss (Tpin-UTWSVM). The proposed Tpin-UTWSVM considers universum data which gives prior information about the distribution of the data, thus improves the generalization performance of the proposed model. Further, the proposed optimization problem is non-convex and non-differentiable which is solved by concave–convex procedure. We employed the SOR approach to train the proposed model effectively with minimum training time. We conducted numerical experiments on 19 UCI binary datasets with different noise levels to validate the noise insensitivity of the proposed Tpin-UTWSVM model. We also conducted numerical experiments for electroencephalogram (EEG) signal classification and Alzheimer's disease (AD) detection. The overall experimental outcomes and statistical tests demonstrate the superiority of the proposed Tpin-UTWSVM model in comparison to the baseline models. The source code for the proposed Tpin-UTWSVM is available at https://github.com/mtanveer1/Universum-twin-SVM-with-truncated-pinball-loss.

1. Introduction

The introduction of support vector machine (SVM) (Vapnik, 1999) is a major step to solve classification, regression and clustering problems. It is one of the most powerful tools of machine learning with the goal of margin maximization between different classes. It leads to the formulation of a convex quadratic programming problem (QPP) which has a unique global solution. Along with that, structural risk minimization (SRM) is also considered in SVM. Its performance on high dimensional data is good as a consequence of having low Vapnik–Chervonenkis dimension. With the passing years, SVM plays an important role in face detection (Osuna et al., 1997), EEG signal classification (Subasi and Gursoy, 2010; Richhariya and Tanveer, 2018), web mining (Bollegala et al., 2010), feature extraction (Li and Guan, 2008), remote sensing (Pal and Mather, 2005), text categorization (Joachims, 1998) and so on. The major drawback of SVM is its high computational complexity i.e., $O(m^3)$, m being the number of training data points.

In healthcare field, the use of machine learning and deep learning is increasing since last decade. To detect epilepsy, brain tumour, sleep disorders and so on, electroencephalogram (EEG) signals are most used non-invasive technique. EEG signals are processed by several feature extraction techniques. The most commonly used technique is wavelet transform which retrieves the time-localized frequency domain features from time domain. The wavelet analysis utilizes approximation and decomposition coefficients as features. Diverse signal types are the focus of various wavelet transformation families. Orthogonal decimated discrete wavelet technique (Rosso et al., 2005) was developed to identify maturational changes linked to infantile absence epilepsy. Subasi and Gursoy (2010) applied different feature extraction techniques such as independent component analysis (ICA), principal component analysis (PCA), linear discriminant analysis (LDA) to EEG signals followed by SVM. Alzheimer's disease (AD) is an untreatable neurodegenerative disease which affects the older population and it is the most prevalent cause of dementia. As per the world Alzheimer report 2021 (Gauthier et al., 2021), 55 million people live with dementia which has been estimated to reach 78 million by 2030. Normal control (NC) subjects exhibit moderate cognitive decline, which includes judgement, language, memory and thinking issues, before reaching the full-fledged AD stage. The stage of cognitive decline is named as mild cognitive impairment (MCI). The machine learning models have wide application in the classification and prediction of the disease (Richhariya et al., 2020; Pellegrini et al., 2018) and it is a fast growing area of research. The detailed analysis of recent shallow and deep learning models for

* Corresponding author.
E-mail addresses: phd2101141007@iiti.ac.in (A. Kumari), mtanveer@iiti.ac.in (M. Tanveer).

[1] The Alzheimer's Disease Neuroimaging Initiative (ADNI) (adni.loni.usc.edu) database was utilized to gather the information for this article. As such, the ADNI investigators provided data and/or contributed to the design and implementation of the ADNI but did not take part in the analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

https://doi.org/10.1016/j.engappai.2023.106427
Received 8 October 2022; Received in revised form 27 March 2023; Accepted 4 May 2023
Available online 6 June 2023

the diagnosis of AD can be found in Tanveer et al. (2020) and Sharma et al. (2023).

Iterative learning control (ILC), a control methodology improves the performance of systems executing repetitive tasks which is assumed to have fixed constant values. However, ILC for repetitive tasks with randomly varying trial lengths using successive projection (Zhuang et al., 2022) introduces a modified problem to incorporate a randomly changing trial length. In the article, online reinforcement learning multiplayer non-zero sum games of continuous-time Markov jump linear systems (MJLS) (Xin et al., 2022), researchers begin by collecting and learning the subsystem information of inputs and states, then calculate the N coupled algebraic Riccati equations. Modern data-driven fault diagnosis (MIFD) requires enough training samples to get the desired diagnosis. The article (Tao et al., 2022) introduces a method to overcome the aforementioned problems of the model.

Generalized eigenvalue proximal support vector machine (GEPSVM) (Mangasarian and Wild, 2005) introduced the concept of two non parallel hyperplanes to minimize the distance between data points and the corresponding hyperplane. Following the concept of non parallel hyperplanes, twin support vector machine (TWSVM) (Jayadeva et al., 2007) is introduced such that each hyperplane is proximal to its associated class and at least one unit away from the other class. TWSVM solves two QPPs, where the data points of the second class give the constraints for the first class and vice-versa. The complexity for each QPP is approximately $O((m/2)^3)$. Thus, TWSVM has approximately four times reduced complexity in comparison to classical SVM and has applications such as fault detection (Ding et al., 2013), electroencephalogram (EEG) signal classification (Ganaie et al., 2022a) and so on. Incorporation of universum data to classification models gives prior information about the distribution of the data to be classified. However, such information comes at the expense of increased complexity. Also, universum data are obtained from the training samples and their selection depends on the nature of the problem (Chapelle et al., 2007). The concept of universum SVM (USVM) (Weston et al., 2006) improves the generalization performance of SVM. Many variants of TWSVM and its incorporation with universum data have been proposed such as twin bounded SVM (TBSVM) (Shao et al., 2011), least square TWSVM (LSTSVM) (Kumar and Gopal, 2009), a novel twin parametric-margin SVM for pattern recognition (TPMSVM) (Peng, 2011), nonparallel hyperplane SVM for binary classification problems (Shao et al., 2014), robust and sparse linear programming TWSVM (Tanveer, 2015b,a), intuitionistic fuzzy LSTSVM (Laxmi et al., 2022), robust projection TWSVM via DC programming (Li et al., 2022) and so forth. A comprehensive review on TWSVM (Tanveer et al., 2022) presents the recent models along with their advantages and limitations. TWSVM with universum data (UTSVM) (Qi et al., 2012) has less time complexity as compared to USVM. Different variants of UTSVM have been introduced in the literature such as improved universum TWSVM (Richhariya et al., 2018), LSTSVM with universum data for classification (ULSTSVM) (Xu et al., 2016), universum least squares twin parametric-margin SVM (Richhariya and Tanveer, 2020b) angle-based universum LSTSVM (Richhariya et al., 2021a) and so on. The universum variant model has applications such as EEG signal classification using USVM (Richhariya and Tanveer, 2018), facial expression recognition using iterative UTSVM (Richhariya and Gupta, 2019) and so forth.

The classical SVM and TWSVM get effected in the presence of noise and outliers which leads to poor classification performance. To overcome this issue, two approaches have been proposed in literature. First is the assignment of fuzzy membership and non membership value to the input data points as shown in articles intuitionistic fuzzy TWSVM (IFTWSVM) (Rezvani et al., 2019), fuzzy universum LSTVM (Richhariya et al., 2021b) and improved IFTWSVM used for EEG signal classification (Ganaie et al., 2022a). The other approach to deal with noise and outliers of classical SVM and TWSVM models are introduction of the pinball loss. Huang et al. (2013) proposed SVM classifier with pinball loss (pin SVM) which considers quantile distance and less sensitive to noise. General TWSVM with pinball loss function (Pin-GTSVM) (Tanveer et al., 2019b) incorporates pinball loss to classical TWSVM. Universum TWSVM with pinball loss introduced by Ganaie et al. (2022b) is used to classify EEG signals. Pin-GTSVM has the same time complexity as that of TWSVM (Tanveer et al., 2019b). Thus, the advantages of pinball loss are obtained without an increase in the computational complexity (Huang et al., 2013). However, Pin-GTSVM is not effective for large scale problems as inversion of a matrix is computationally expensive. For large scale classification problems, Tanveer et al. (2021) introduced large scale pinball TWSVM.

The benefits of pinball loss are acquired at the expense of losing sparsity of the models. Huang et al. (2013) introduced pinball loss with ϵ insensitive zone which improves sparseness of pin SVM. Tanveer et al. (2019c) introduced sparse pinball twin support vector machine (SPTWSVM) which retains the sparseness of TWSVM. SPTWSVM considers only empirical risk minimization, thus improved sparse pinball TWSVM is presented by Tanveer et al. (2019a) which implements SRM. However, SPTWSVM loses other theoretical properties of pinball loss such as minimization leading to Bayes classifier, bounded misclassification error by Zhang's inequality (Zhang, 2004). Another approach to retain the sparsity of model along with conserving the theoretical properties of pinball loss is SVM classifier with truncated pinball loss (Tpin-SVM) (Shen et al., 2017). Truncated pinball loss acts as a bridge between pinball loss and hinge loss by maintaining an agreement between noise insensitivity and sparsity of the model.

Motivated by the Tpin-SVM, in this article we propose universum twin support vector machine with truncated pinball loss (Tpin-UTWSVM). The proposed Tpin-UTWSVM model incorporates the universum data points in the classical TWSVM and considers the truncated pinball loss for imposing penalty to the data points. The contributions of the proposed Tpin-UTWSVM model are listed as:

- The proposed Tpin-UTWSVM considers quantile distance and punishes both correctly and incorrectly classified samples. Penalizing correctly classified samples lead to noise insensitivity and imposition of equal penalty to the correctly classified data points lead to sparsity of the model. Thus, the proposed model maintains a trade-off between noise insensitivity and sparsity of the model.
- The proposed Tpin-UTWSVM includes universum data points which give prior information about the distribution of data which helps in improving the generalization performance of the model.
- The objective function obtained for the proposed Tpin-UTWSVM is non-differentiable and non-convex. Thus, concave convex procedure (CCCP) is applied to solve the problem. To train the proposed Tpin-UTWSVM, efficient successive overrelaxation (SOR) method is used.
- We performed numerical experiments on 19 UCI binary datasets. Numerical experiments and statistical analysis show the superiority of the proposed Tpin-UTWSVM over the baseline methods. Through numerical experiments, we also analyze the effect of loss parameters on the sparsity and noise insensitivity of the proposed Tpin-UTWSVM model.
- We implemented the proposed Tpin-UTWSVM on the biomedical domain for EEG signal classification and Alzheimer's disease (AD) detection. Numerical experiments performed on EEG and AD datasets show the superiority of the proposed model over baseline models.

The remaining paper is organized as follows: Section 2 gives a brief idea about the literature of the existing models and loss functions. Section 3 discusses the proposed Tpin-UTWSVM in detail along with the construction of the dual problems using CCCP method. Section 4 discusses the properties of the proposed model. Details of numerical experiments on UCI binary datasets and Biomedical data *i.e.,* EEG signal classification and AD detection are provided in Section 5 and Section 6, respectively. Finally, the conclusion and the future related work are given in Section 7.

2. Preliminaries

Let $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ be the training data points for classification, where $x_j \in \mathbb{R}^n$ and $y_j \in \{\pm 1\}$. The set D has m_1 data points of positive label and m_2 data points of negative label. Assume M_1 and M_2 represent matrices of positive and negative class data points with dimensions $m_1 \times n$ and $m_2 \times n$, respectively.

2.1. Loss functions

Loss functions in the models signify the cost imposed on the data points. In the classical SVM and TWSVM, hinge loss is used which has drawbacks such as sensitive to noise and unstable to resampling.

2.1.1. Pinball loss function

Pinball loss (Huang et al., 2013) overcomes the drawbacks of hinge loss. It considers quantile distance between data points and decision boundary. It is defined as:

$$L_\tau(u) = \begin{cases} -\tau u, & u < 0, \\ u, & u \geq 0, \end{cases} \tag{1}$$

where $u \in \mathbb{R}$, $\tau \in [0, 1]$. Pinball loss imposes cost on both correctly and misclassified data points which leads to noise insensitivity near the decision boundary. It is also stable to resampling and robust to outliers. However, it reduces the sparsity of the model as compared to hinge loss.

2.1.2. Truncated pinball loss function

Truncated pinball loss is a modified version of pinball loss which preserves the advantages of pinball loss along with leading to sparsity. It is defined by using an auxiliary function $H_\tau(u) = \max\{\tau u, 0\}$ ($u \in \mathbb{R}$, $\tau \in [0, 1]$ is a constant) as:

$$\mathbb{P}_{\tau,s}(u) = H_{1+\tau}(u) - H_\tau(u+s) + \tau s = \begin{cases} \tau s, & \text{if } u \leq -s, \\ -\tau u, & \text{if } -s < u < 0, \\ u, & \text{if } u \geq 0. \end{cases} \tag{2}$$

2.2. SVM with truncated pinball loss

Tpin-SVM (Shen et al., 2017) incorporates truncated pinball loss in classical SVM which retains the properties of pinball loss along with leading to a sparse model. The goal of SVM is to maximize the margin between the bounding hyperplanes. The mathematical formulation of Tpin-SVM is presented as:

$$\min_{w,b} \ \frac{1}{2}\|w\|^2 + C_1 \sum_{j=1}^{m} \mathbb{P}_{\tau,s}(1 - y_j f(x_j)), \tag{3}$$

where decision hyperplane is given as: $f(x) = x^T w + b = 0$ with $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$; C_1 is a positive regularization parameter. QPP (3) can be solved by using $CCCP$ method. After obtaining the unknowns, the decision function for a test data (say x_t) is given by sign of $(x_t^T w + b)$.

2.3. Twin SVM

TWSVM (Jayadeva et al., 2007) constructs two non parallel hyperplanes such that each hyperplane is proximal to the corresponding class and at least one unit away from other class. The two non parallel hyperplanes are given by:

$$x^T w_1 + b_1 = 0 \text{ and } x^T w_2 + b_2 = 0, \tag{4}$$

where $w_1,\ w_2 \in \mathbb{R}^n$ and $b_1,\ b_2 \in \mathbb{R}$. The formulation of TWSVM can be expressed as:

$$\min_{w_1,b_1,\zeta_2} \ \frac{1}{2}\left\|M_1 w_1 + e_1 b_1\right\|^2 + C_1 e_2^T \zeta_2$$
$$\text{subject to } -(M_2 w_1 + e_2 b_1) + \zeta_2 \geq e_2, \quad \zeta_2 \geq 0 \tag{5}$$

and

$$\min_{w_2,b_2,\zeta_1} \ \frac{1}{2}\left\|M_2 w_2 + e_2 b_2\right\|^2 + C_2 e_1^T \zeta_1$$
$$\text{subject to } (M_1 w_2 + e_1 b_2) + \zeta_1 \geq e_1, \quad \zeta_1 \geq 0. \tag{6}$$

Here, $e_1,\ e_2$ are the vectors of ones having appropriate dimensions. $\zeta_1,\ \zeta_2$ represent slack variables and $C_1,\ C_2$ are the positive regularization parameters. After the determination of the planes (4), an arbitrary test data (say x) is assigned to a class depending on

$$f(x) = \underset{k=1,2}{argmin} \ \frac{|x^T w_k + b_k|}{\|w_k\|}. \tag{7}$$

2.4. Universum TSVM

UTSVM (Qi et al., 2012) introduced the concept of universum data in TWSVM (Jayadeva et al., 2007) which has prior information about the distribution of the data leading to better classification performance, however the improvement in the performance comes at the expense of increased complexity.

Let $U \in \mathbb{R}^{u \times n}$ denotes the universum data set. The non parallel hyperplanes are given by equations in (4) and the loss function considered for universum data is ϵ insensitive loss ($\lambda[t]$) which is a combination of two hinge losses ($\lambda_{-\epsilon}[t]$) and ($\lambda_{-\epsilon}[-t]$) *i.e.,*

$$\lambda[t] = \lambda_{-\epsilon}[t] + \lambda_{-\epsilon}[-t], \tag{8}$$

where $\lambda_{-\epsilon}[t] = \max\{0, t - \epsilon\}$.

The formulation for UTSVM is given by:

$$\min_{w_1,b_1,\zeta_2,\eta_2} \ \frac{1}{2}\left\|M_1 w_1 + e_1 b_1\right\|^2 + C_1 e_2^T \zeta_2 + C_u e_u^T \eta_2$$
$$\text{subject to } -(M_2 w_1 + e_2 b_1) + \zeta_2 \geq e_2, \quad \zeta_2 \geq 0, \tag{9}$$
$$(U w_1 + e_u b_1) + \eta_2 \geq (-1 + \epsilon) e_u, \quad \eta_2 \geq 0$$

and

$$\min_{w_2,b_2,\zeta_1,\eta_1} \ \frac{1}{2}\left\|M_2 w_2 + e_2 b_2\right\|^2 + C_2 e_1^T \zeta_1 + C_u e_u^T \eta_1$$
$$\text{subject to } (M_1 w_2 + e_1 b_2) + \zeta_1 \geq e_1, \quad \zeta_1 \geq 0, \tag{10}$$
$$-(U w_2 + e_u b_2) + \eta_1 \geq (-1 + \epsilon) e_u, \quad \eta_1 \geq 0,$$

where e_u represents the vector of ones having appropriate dimension. $\eta_1,\ \eta_2$ are slack variables; C_u is the positive penalty parameter; ϵ signifies the tolerance value for the universum data.

After determining unknowns of the planes in Eq. (4), for an arbitrary test data point, the class is determined by Eq. (7).

2.5. Pin-GTSVM

The basic idea behind Pin-GTSVM (Tanveer et al., 2019b) is incorporating pinball loss in the classical TWSVM. It overcomes the issues of noise sensitivity and instability to resampling. The proximal non parallel planes are given by (4). The QPPs obtained are as follows:

$$\min_{w_1,b_1,\zeta_2} \ \frac{1}{2}\left\|M_1 w_1 + e_1 b_1\right\|^2 + C_1 e_2^T \zeta_2$$
$$\text{subject to } -(M_2 w_1 + e_2 b_1) + \zeta_2 \geq e_2, \tag{11}$$
$$-(M_2 w_1 + e_2 b_1) - \frac{\zeta_2}{\tau_2} \leq e_2$$

and

$$\min_{w_2,b_2,\zeta_1} \ \frac{1}{2}\left\|M_2 w_2 + e_2 b_2\right\|^2 + C_2 e_1^T \zeta_1$$
$$\text{subject to } (M_1 w_2 + e_1 b_2) + \zeta_1 \geq e_1, \tag{12}$$
$$(M_1 w_2 + e_1 b_2) - \frac{\zeta_1}{\tau_1} \leq e_1.$$

Here $\tau_1,\ \tau_2 \in [0, 1]$ are the parameters of the pinball loss. Similar to TWSVM and UTSVM, for a test data point class is determined from Eq. (7).

3. Proposed work

Pin-GTSVM (Tanveer et al., 2019b) leads to a model which is noise insensitive and stable to resampling, however the model lacks sparsity as compared to TWSVM (Jayadeva et al., 2007). In this section, we introduce a novel classification model by incorporating truncated pinball loss function to the UTSVM (Qi et al., 2012) model and named it as universum twin support vector machine with truncated pinball loss (Tpin-UTWSVM). The truncated pinball loss function acts as a flexible framework of trade-off between sparsity and feature noise insensitivity. Also, the presence of universum data (denoted by $U = \{x_1^*, x_2^*, \dots, x_{|U|}^*\}$) improves the generalization performance of the model.

3.1. Linear case

The two non parallel hyperplanes are:

$$f_j(x) = 0 \text{ with } f_j(x) = x^T w_j + b_j, \tag{13}$$

where $j = 1, 2$ denote positive and negative hyperplanes, respectively. Assume M_1 and M_2 be the matrices of positive and negative class samples with dimension $m_1 \times n$ and $m_2 \times n$, respectively. The QPPs for the linear case are as follows:

$$\min_{w_1,b_1} \frac{1}{2}\left\|M_1 w_1 + e_1 b_1\right\|^2 + C_u \sum_{l=1}^{|U|} \lambda[f_1(x_l^*)] + C_1 \sum_{i \in M_2} \mathbb{P}_{\tau_2,s_2}(1 - y_i f_1(x_i)) \tag{14}$$

and

$$\min_{w_2,b_2} \frac{1}{2}\left\|M_2 w_2 + e_2 b_2\right\|^2 + C_u \sum_{l=1}^{|U|} \lambda[f_2(x_l^*)] + C_2 \sum_{i \in M_1} \mathbb{P}_{\tau_1,s_1}(1 - y_i f_2(x_i)), \tag{15}$$

where C_1, C_2, C_u are the positive regularization parameters. τ_1, $\tau_2 \in [0, 1]$ and s_1, s_2 are positive parameters; $\mathbb{P}_{\tau_1,s_1}$ and $\mathbb{P}_{\tau_2,s_2}$ signify the truncated pinball loss as shown in Eq. (2). Also, $\lambda[f_j(x_l^*)]$, $(j = 1, 2)$ is the ϵ insensitive loss shown in Eq. (8), explicitly written as:

$$\lambda[w_j^T x_l^* + b_j] = \begin{cases} -\epsilon - (w_j^T x_l^* + b_j), & \text{if } w_j^T x_l^* + b_j \le -\epsilon, \\ 0, & \text{if } -\epsilon < w_j^T x_l^* + b_j \le \epsilon, \\ -\epsilon + w_j^T x_l^* + b_j, & \text{if } w_j^T x_l^* + b_j > \epsilon. \end{cases} \tag{16}$$

for $l = 1, 2, \dots, |U|$. The first term in problems (14) and (15) minimize the distance of data points from the respective class. The second and third terms impose a penalty on the data points. Using Eq. (2) of the truncated pinball loss, the QPPs (14) and (15) can be written as:

$$\min_{w_1,b_1} \underbrace{\frac{1}{2}\left\|M_1 w_1 + e_1 b_1\right\|^2 + C_u \sum_{l=1}^{|U|} \lambda[f_1(x_l^*)] + C_1 \sum_{i \in M_2} [H_{1+\tau_2}(1 - y_i f_1(x_i))]}_{J_{convex}} + \underbrace{\left(-C_1 \sum_{i \in M_2} [H_{\tau_2}(1 - y_i f_1(x_i) + s_2)]\right)}_{J_{concave}} \tag{17}$$

and

$$\min_{w_2,b_2} \underbrace{\frac{1}{2}\left\|M_2 w_2 + e_2 b_2\right\|^2 + C_u \sum_{l=1}^{|U|} \lambda[f_2(x_l^*)] + C_2 \sum_{i \in M_1} [H_{1+\tau_1}(1 - y_i f_2(x_i))]}_{J_{convex}} + \underbrace{\left(-C_2 \sum_{i \in M_1} [H_{\tau_1}(1 - y_i f_2(x_i) + s_1)]\right)}_{J_{concave}} \tag{18}$$

The above two optimization problems are non-convex and can be solved by CCCP. It decomposes as the sum of convex (J_{convex}) and concave ($J_{concave}$) part. Convex part is represented by the first three terms and concave part is represented by the fourth term of Eqs. (17) and (18). Now onwards, we assume

$$v_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} \text{ and } v_2 = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix}. \tag{19}$$

3.1.1. Primal problems

CCCP (Yen et al., 2012) is a popular method which is widely used to solve a non-differentiable optimization problem which has the difference of convex functions. It iteratively solves a sequence of convex subproblems. The CCCP algorithm used here converges to the stationary point (Yen et al., 2012) and has a linear convergence rate.

By applying CCCP method, the optimization problem (17) and (18) takes the form:

$$\min_{v_j} J_{convex} + \langle J'_{concave}(v_j^r), v_j \rangle \tag{20}$$

for $j = 1, 2$, respectively. Here, superscript r denotes the iterations which we will drop in further analysis. Representing the derivative $J'_{concave}(v_j)$ by ϱ_j, where $\varrho_j = (\varrho_{j1}, \varrho_{j2}, \dots, \varrho_{j\tilde{m}})^T$ having $\tilde{m}$ as m_1, m_2 for j being 1, 2, respectively. We have $\forall\ i = 1, 2, \dots, m_2$,

$$\varrho_{2i} = \begin{cases} C_1 \tau_2, & \text{if } 1 - y_i f_1(x_i) + s_2 > 0, \\ 0, & \text{if } 1 - y_i f_1(x_i) + s_2 \le 0. \end{cases} \tag{21}$$

Applying CCCP method to QPP (17), we get:

$$\min_{w_1,b_1} \frac{1}{2}\left\|M_1 w_1 + e_1 b_1\right\|^2 + C_u \sum_{l=1}^{|U|} \lambda[f_1(x_l^*)] + C_1 \sum_{i \in M_2} [H_{1+\tau_2}(1 - y_i f_1(x_i))] - \varrho_2^T (e_2 + M_2 w_1 + e_2 b_1), \tag{22}$$

where each component of ϱ_2 is defined in Eq. (21). Similarly,

$$\varrho_{1i} = \begin{cases} C_2 \tau_1, & \text{if } 1 - y_i f_2(x_i) + s_1 > 0, \\ 0, & \text{if } 1 - y_i f_2(x_i) + s_1 \le 0, \end{cases} \tag{23}$$

for $i = 1, 2, \dots, m_1$. Applying CCCP to QPP (18), we get

$$\min_{w_2,b_2} \frac{1}{2}\left\|M_2 w_2 + e_2 b_2\right\|^2 + C_u \sum_{l=1}^{|U|} \lambda[f_2(x_l^*)] + C_2 \sum_{i \in M_1} [H_{1+\tau_1}(1 - y_i f_2(x_i))] - \varrho_1^T (e_1 - (M_1 w_2 + e_1 b_2)) \tag{24}$$

with each component of ϱ_1 as defined in Eq. (23). Introducing slack variables ζ_2, η_2 in problem (22) and ζ_1, η_1 in problem (24), the primal problems are:

$$\begin{aligned} &\min_{w_1,b_1} \frac{1}{2}\left\|M_1 w_1 + e_1 b_1\right\|^2 + C_u e_u^T \eta_2 + C_1 e_2^T \zeta_2 - \varrho_2^T (e_2 + M_2 w_1 + e_2 b_1) \\ &\text{subject to } -(M_2 w_1 + e_2 b_1) + \frac{\zeta_2}{1 + \tau_2} \ge e_2,\ \zeta_2 \ge 0, \\ &\qquad (U w_1 + e_u b_1) + \eta_2 \ge (-1 + \epsilon) e_u,\ \eta_2 \ge 0 \end{aligned} \tag{25}$$

and

$$\begin{aligned} &\min_{w_2,b_2} \frac{1}{2}\left\|M_2 w_2 + e_2 b_2\right\|^2 + C_u e_u^T \eta_1 + C_2 e_1^T \zeta_1 - \varrho_1^T (e_1 - (M_1 w_2 + e_1 b_2)) \\ &\text{subject to } (M_1 w_2 + e_1 b_2) + \frac{\zeta_1}{1 + \tau_1} \ge e_1,\ \zeta_1 \ge 0 \\ &\qquad -(U w_2 + e_u b_2) + \eta_1 \ge (-1 + \epsilon) e_u,\ \eta_1 \ge 0. \end{aligned} \tag{26}$$

Thus, QPPs (25) and (26) are the primal problems of the proposed Tpin-UTWSVM model.

3.1.2. Dual problems

We will demonstrate the detailed calculation to obtain the dual of QPP (25) and similarly the dual of QPP (26) can be obtained. The Lagrangian obtained for the problem (25) is:

$$\begin{aligned} L(w_1, b_1, \zeta_2, \eta_2, \beta_1, \gamma_1, \delta_1, \mu_1) &= \frac{1}{2}\left\|M_1 w_1 + e_1 b_1\right\|^2 + C_u e_u^T \eta_2 + C_1 e_2^T \zeta_2 \\ &- \varrho_2^T (e_2 + M_2 w_1 + e_2 b_1) + \beta_1^T \left(e_2 - \frac{\zeta_2}{1 + \tau_2} + (M_2 w_1 + e_2 b_1)\right) \\ &+ \gamma_1^T ((-1 + \epsilon) e_u - (U w_1 + e_u b_1) - \eta_2) - \delta_1^T \eta_2 - \mu_1^T \zeta_2, \end{aligned} \tag{27}$$

where β_1, γ_1, δ_1, μ_1 are vectors of non negative Lagrange multipliers. Use the necessary and sufficient optimality condition *i.e.,* Karush Kuhn

Tucker (K.K.T.) conditions. Differentiating the Lagrangian (27), we get the following K.K.T. conditions:

$$\frac{\partial L}{\partial w_1} = M_1^T(M_1 w_1 + e_1 b_1) + M_2^T(\beta_1 - \varrho_2) - U^T\gamma_1 = 0, \tag{28}$$

$$\frac{\partial L}{\partial b_1} = e_1^T(M_1 w_1 + e_1 b_1) + e_2^T(\beta_1 - \varrho_2) - e_u^T\gamma_1 = 0, \tag{29}$$

$$\frac{\partial L}{\partial \zeta_2} = C_1 e_2 - \frac{\beta_1}{1+\tau_2} - \mu_1 = 0, \tag{30}$$

$$\frac{\partial L}{\partial \eta_2} = C_u e_u - \gamma_1 - \delta_1 = 0, \tag{31}$$

$$\beta_1^T\left(e_2 - \frac{\zeta_2}{1+\tau_2} + (M_2 w_1 + e_2 b_1)\right) = 0, \tag{32}$$

$$\gamma_1^T((-1+\epsilon)e_u - (U w_1 + e_u b_1) - \eta_2) = 0, \tag{33}$$

$$\delta_1^T \eta_2 = 0, \tag{34}$$

$$\mu_1^T \zeta_2 = 0, \tag{35}$$

$$\beta_1,\ \gamma_1,\ \delta_1,\ \mu_1 \geq 0. \tag{36}$$

On solving Eqs. (28) and (29), we get

$$v_1 = -(E^T E)^{-1}[F^T(\beta_1 - \varrho_2) - O^T\gamma_1], \tag{37}$$

where

$$E = \begin{bmatrix} M_1 & e_1 \end{bmatrix},\ F = \begin{bmatrix} M_2 & e_2 \end{bmatrix} \text{ and } O = \begin{bmatrix} U & e_u \end{bmatrix}. \tag{38}$$

From (30) and (36), we obtain

$$0 \leq \beta_1 \leq C_1 e_2(1+\tau_2). \tag{39}$$

From (31) and (36), we get

$$0 \leq \gamma_1 \leq C_u e_u. \tag{40}$$

From Lagrangian (27) and Eqs. (28)–(31), Wolfe dual obtained for primal problem (25) is:

$$\begin{aligned} \max_{\beta_1,\gamma_1}\ & -\frac{1}{2}((\beta_1 - \varrho_2)^T F - \gamma_1^T O)(E^T E)^{-1}(F^T(\beta_1 - \varrho_2) - O^T\gamma_1) \\ & + e_2^T(\beta_1 - \varrho_2) + (\epsilon - 1)e_u^T\gamma_1 \\ \text{subject to}\ & 0 \leq \beta_1 \leq C_1 e_2(1+\tau_2), \\ & 0 \leq \gamma_1 \leq C_u e_u. \end{aligned} \tag{41}$$

Setting $\alpha_1 = \beta_1 - \varrho_2$, the dual problem changes to

$$\begin{aligned} \max_{\alpha_1,\gamma_1}\ & -\frac{1}{2}(\alpha_1^T F - \gamma_1^T O)(E^T E)^{-1}(F^T\alpha_1 - O^T\gamma_1) + e_2^T\alpha_1 + (\epsilon - 1)e_u^T\gamma_1 \\ \text{subject to}\ & -\varrho_2 \leq \alpha_1 \leq C_1 e_2(1+\tau_2) - \varrho_2, \\ & 0 \leq \gamma_1 \leq C_u e_u. \end{aligned} \tag{42}$$

Similarly, writing the Lagrangian for problem (26) and solving using K.K.T. conditions, we obtain:

$$v_2 = (F^T F)^{-1}[E^T(\beta_2 - \varrho_1) - O^T\gamma_2]. \tag{43}$$

The Wolfe dual problem of (26) can be derived as

$$\begin{aligned} \max_{\alpha_2,\gamma_2}\ & -\frac{1}{2}(\alpha_2^T E - \gamma_2^T O)(F^T F)^{-1}(E^T\alpha_2 - O^T\gamma_2) + e_1^T\alpha_2 + (\epsilon - 1)e_u^T\gamma_2 \\ \text{subject to}\ & -\varrho_1 \leq \alpha_2 \leq C_2 e_1(1+\tau_1) - \varrho_1, \\ & 0 \leq \gamma_2 \leq C_u e_u, \end{aligned} \tag{44}$$

where $\alpha_2 = \beta_2 - \varrho_1$ and $\beta_2,\ \gamma_2$ are Lagrange multipliers. After determining $\alpha_k,\ \gamma_k$, calculate $w_k,\ b_k$ for $k = 1, 2$. For a test point say x, the label can be determined by $f(x)$, where

$$f(x) = \underset{k=1,2}{argmin}\ \frac{|x^T w_k + b_k|}{\|w_k\|}. \tag{45}$$

The block diagram in Fig. 1 represents the method to obtain the optimal plane. Algorithm 1 consists of framework to get the optimal hyperplane.

Algorithm 1 CCCP algorithm for proposed Tpin-UTWSVM

Input: Training data $\{(x_1, y_1), \ldots, (x_m, y_m)\}$, testing data set and tolerance value 'tol'.

Output: The labels of testing data.

- Choose appropriate $\tau_1, \tau_2 \in [0, 1]$, $s_1, s_2 \geq 0$ and the positive regularization parameters $C_1,\ C_2,\ C_u$.
- Obtain the optimal solution (w_1, b_1) and (w_2, b_2).
  1. For iteration variable $r = 0$, initialize $w_1^0, b_1^0, w_2^0, b_2^0$ and calculate $\varrho_2^0,\ \varrho_1^0$ using equations (21) and (23).
  2. Solve dual problem (42) and (44) in r^{th} iteration to get $\alpha_j^r,\ \gamma_j^r$, thus (w_j^r, b_j^r) using equations (37) and (43) for $j = 1, 2$, respectively. Compute $f_j(x_i)$ for $i = 1, 2, \ldots, m$.
  3. Update ϱ_2 and ϱ_1 by (21) and (23), respectively.
  4. If $\left\|\varrho_i^r - \varrho_i^{r-1}\right\| <$ tol, stop the iteration and obtain optimal (w_i, b_i), $i = 1,\ 2$, else set $r = r + 1$ and repeat step 2.
- For a test point x, determine the label using equation (45).

3.2. Non linear case

For the non linear case, we map the data points to some higher dimensional space. Let the map be θ, so x is mapped to $\theta(x)$. Also, introduce the kernel function $K(x_i, x_j) = (\theta(x_i), \theta(x_j))$. The positive and negative non parallel kernel generated surfaces are given by

$$K(x^T, B^T)u_1 + b_1 = 0 \text{ and } K(x^T, B^T)u_2 + b_2 = 0, \tag{46}$$

respectively, where $B = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix}$ and $u_1,\ u_2 \in \mathbb{R}^m$; $b_1,\ b_2 \in \mathbb{R}$. Following the similar algorithm as that of linear case, we get the primal problem:

$$\begin{aligned} \underset{u_1,b_1}{min}\ & \frac{1}{2}\left\|K(M_1, B^T)u_1 + e_1 b_1\right\|^2 + C_u\sum_{l=1}^{|U|}\lambda[f_1(\theta(x_l^*))] \\ & + C_1\sum_{i\in M_2}\mathbb{P}_{\tau_2,s_2}(1 - y_i f_1(x_i)) \end{aligned} \tag{47}$$

and

$$\begin{aligned} \underset{u_2,b_2}{min}\ & \frac{1}{2}\left\|K(M_2, B^T)u_2 + e_2 b_2\right\|^2 + C_u\sum_{l=1}^{|U|}\lambda[f_2(\theta(x_l^*))] \\ & + C_2\sum_{i\in M_1}\mathbb{P}_{\tau_1,s_1}(1 - y_i f_2(x_i)), \end{aligned} \tag{48}$$

where C_1, C_2, C_u are the positive regularization parameters. M_1, M_2 denote the positive and negative class data points respectively, $\tau_1,\ \tau_2 \in [0, 1]$ and $s_1,\ s_2$ are positive parameters. $\mathbb{P}_{\tau_1,s_1}$ and $\mathbb{P}_{\tau_2,s_2}$ signify the truncated pinball loss as shown in Eq. (2). Also, $\lambda[f_j(\theta(x_l^*))]$ $(j = 1, 2)$ is the ϵ insensitive loss shown in Eq. (8). Introducing the slack variables and applying the CCCP, the primal forms obtained are given by:

$$\begin{aligned} \underset{u_1,b_1}{min}\ & \frac{1}{2}\left\|K(M_1, B^T)u_1 + e_1 b_1\right\|^2 + C_u e_u^T\eta_2 + C_1 e_2^T\zeta_2 \\ & - \varrho_2^T(e_2 + K(M_2, B^T)u_1 + e_2 b_1) \\ \text{subject to}\ & (K(U, B^T)u_1 + e_u b_1) + \eta_2 \geq (-1+\epsilon)e_u,\ \eta_2 \geq 0, \\ & -(K(M_2, B^T)u_1 + e_2 b_1) + \frac{\zeta_2}{1+\tau_2} \geq e_2,\ \zeta_2 \geq 0 \end{aligned} \tag{49}$$

and

$$\begin{aligned} \underset{u_2,b_2}{min}\ & \frac{1}{2}\left\|K(M_2, B^T)u_2 + e_2 b_2\right\|^2 + C_u e_u^T\eta_1 + C_2 e_1^T\zeta_1 \\ & - \varrho_1^T(e_1 - (K(M_1, B^T)u_2 + e_1 b_2)) \\ \text{subject to}\ & -(K(U, B^T)u_2 + e_u b_2) + \eta_1 \geq (-1+\epsilon)e_u,\ \eta_1 \geq 0, \\ & (K(M_1, B^T)u_2 + e_1 b_2) + \frac{\zeta_1}{1+\tau_1} \geq e_1,\ \zeta_1 \geq 0, \end{aligned} \tag{50}$$

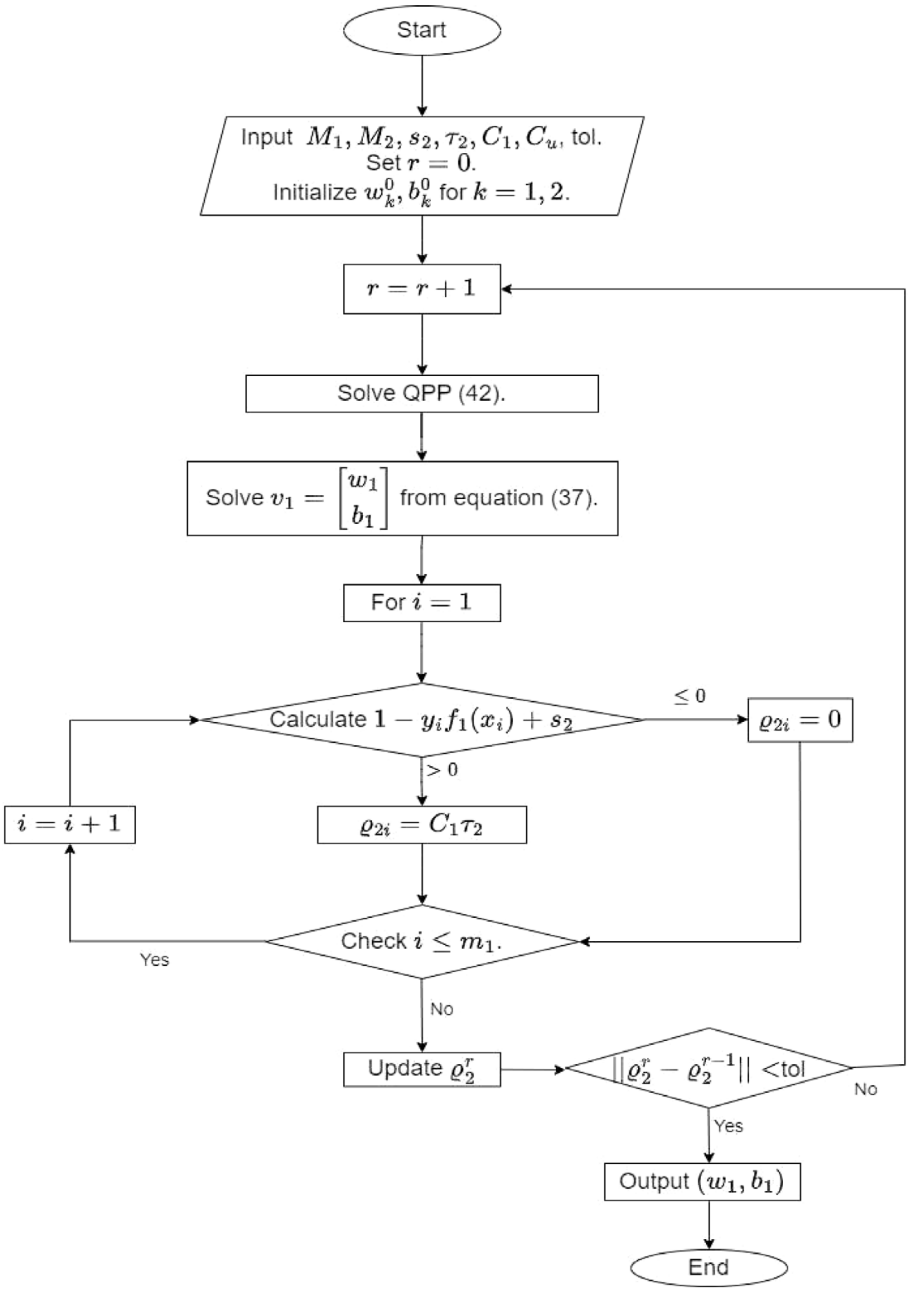


Fig. 1. Block diagram to obtain the unknowns (w_1, b_1) of the first hyperplane in Eq. (13).

where ζ_1, ζ_2 and η_1, η_2 are the slack variables. Also, ϱ_2, ϱ_1 are defined similar to Eqs. (21) and (23), respectively. Proceeding in the similar way as done for the linear case, the dual obtained for QPP (49) is given by:

$$\begin{aligned} \max_{\alpha_1,\gamma_1} \quad & -\frac{1}{2}(\alpha_1^T F - \gamma_1^T O)(E^T E)^{-1}(F^T \alpha_1 - O^T \gamma_1) + e_2^T \alpha_1 + (\epsilon - 1)e_u^T \gamma_1 \\ \text{subject to} \quad & -\varrho_2 \leq \alpha_1 \leq C_1 e_2(1 + \tau_2) - \varrho_2, \\ & 0 \leq \gamma_1 \leq C_u e_u. \end{aligned} \tag{51}$$

Similarly, the dual problem of QPP (50), can be derived as

$$\begin{aligned} \max_{\alpha_2,\gamma_2} \quad & -\frac{1}{2}(\alpha_2^T E - \gamma_2^T O)(F^T F)^{-1}(E^T \alpha_2 - O^T \gamma_2) + e_1^T \alpha_2 + (\epsilon - 1)e_u^T \gamma_2 \\ \text{subject to} \quad & -\varrho_1 \leq \alpha_2 \leq C_2 e_1(1 + \tau_1) - \varrho_1, \\ & 0 \leq \gamma_2 \leq C_u e_u, \end{aligned} \tag{52}$$

where $E = [K(M_1, B^T)\ e_1]$, $F = [K(M_2, B^T)\ e_2]$ and $O = [K(U, B^T)\ e_u]$ and $\alpha_1 = \beta_1 - \varrho_2$, $\alpha_2 = \beta_2 - \varrho_1$. β_1, β_2, γ_1, and γ_2 are the vectors of non-negative Lagrangian multipliers. After getting the unknowns, label of a test point (say) x can be determined by:

$$f(x) = argmin\left(\frac{|K(x^T, B^T)u_1 + b_1|}{\sqrt{u_1^T K(B, B^T)u_1}}, \frac{|K(x^T, B^T)u_2 + b_2|}{\sqrt{u_2^T K(B, B^T)u_2}}\right). \tag{53}$$

4. Properties of Tpin-UTWSVM

In this section, we analyze Tpin-UTWSVM with respect to noise insensitivity and sparsity. Also, we will show the bounds on the number of negative and positive boundary errors.

4.1. Noise insensitivity and sparsity

The main advantage of considering truncated pinball loss is insensitivity to noise around the decision boundary and sparsity of the model. Subgradient of truncated pinball loss function (2) is defined as:

$$\partial \mathbb{P}_{\tau,s}(x,y,f(x)) = \begin{cases} 0, & \text{if } 1-yf(x) < -s, \\ [-\tau,0], & \text{if } 1-yf(x) = -s, \\ -\tau, & \text{if } -s < 1-yf(x) < 0, \\ [-\tau,1], & \text{if } 1-yf(x) = 0, \\ 1, & \text{if } 1-yf(x) > 0. \end{cases} \tag{54}$$

For convenience, we will illustrate the property of noise insensitivity and sparsity for linear positive classifier, others can be analyzed similarly. Using the K.K.T. condition of optimality in problem (14), we get:

$$0 \in M_1^T(M_1 w_1 + e_1 b_1) + C_u \sum_{l=1}^{|U|} \partial\lambda[f_1(x_l^*)] + C_1 \sum_{i \in M_2} \partial \mathbb{P}_{\tau_2,s_2}(1 - y_i f_1(x_i)). \tag{55}$$

Here, 0 denotes a vector having each component zero. For a fixed value of w_1 and b_1, we split the whole index set ($i = 1, 2, \ldots, m_2$) into the following five sets:

$$\begin{aligned} S_0^{w_1,b_1} &= \{i \mid 1 - y_i(w_1^T x_i + b_1) < -s_2\}, \\ S_1^{w_1,b_1} &= \{i \mid 1 - y_i(w_1^T x_i + b_1) = -s_2\}, \\ S_2^{w_1,b_1} &= \{i \mid -s_2 < 1 - y_i(w_1^T x_i + b_1) < 0\}, \\ S_3^{w_1,b_1} &= \{i \mid 1 - y_i(w_1^T x_i + b_1) = 0\}, \\ S_4^{w_1,b_1} &= \{i \mid 1 - y_i(w_1^T x_i + b_1) > 0\}. \end{aligned} \tag{56}$$

The data points in $S_0^{w_1,b_1}$ have no support vectors so it does not contribute to w_1. $S_0^{w_1,b_1}$ is related to the sparsity of the proposed Tpin-UTWSVM. The number of negative class data points in $S_0^{w_1,b_1}$ is determined by the parameter s_2. If s_2 is small, $S_0^{w_1,b_1}$ has more number of points and the model will become sparser. The truncated pinball loss approaches hinge loss as $s_2 \to 0$. Contrarily, larger s_2 means less negative samples $S_0^{w_1,b_1}$ has and the model is insensitive to noise *i.e.,* as $s_2 \to \infty$, the sparsity of the model will be lost entirely.

For the ϵ insensitive loss function represented by Eq. (16), the subgradient can be written as:

$$\partial\lambda[w_1^T x_l^* + b_1] = \begin{cases} -1, & \text{if } w_1^T x_l^* + b_1 < -\epsilon, \\ [-1,0], & \text{if } w_1^T x_l^* + b_1 = -\epsilon, \\ 0, & \text{if } -\epsilon < w_1^T x_l^* + b_1 < \epsilon, \\ [0,1], & \text{if } w_1^T x_l^* + b_1 = \epsilon, \\ 1, & \text{if } w_1^T x_l^* + b_1 > \epsilon. \end{cases} \tag{57}$$

Again splitting the index set of the universum data, we have the following five sets:

$$\begin{aligned} P_0 &= \{j \in U \mid w_1^T x_l^* + b_1 < -\epsilon\}, \\ P_1 &= \{j \in U \mid w_1^T x_l^* + b_1 = -\epsilon\}, \\ P_2 &= \{j \in U \mid -\epsilon < w_1^T x_l^* + b_1 < \epsilon\}, \\ P_3 &= \{j \in U \mid w_1^T x_l^* + b_1 = \epsilon\}, \\ P_4 &= \{j \in U \mid w_1^T x_l^* + b_1 > \epsilon\}. \end{aligned} \tag{58}$$

From Eq. (55), and using the index sets $S_k^{w_1,b_1}$ and P_k, where $k = 0, 1, 2, 3, 4$, we obtain

$$\begin{aligned} &\frac{M_1}{C_1}^T (M_1 w_1 + e_1 b_1) + \frac{C_u}{C_1}\left(\sum_{j \in P_0} x_j^* + \sum_{j \in P_1} a_j x_j^* + \sum_{j \in P_3} b_j x_j^* + \sum_{j \in P_4} x_j^*\right) \\ &+ \left(\sum_{i \in S_1^{w_1,b_1}} \psi_i x_i + \sum_{i \in S_2^{w_1,b_1}} -\tau_1 x_i + \sum_{i \in S_3^{w_1,b_1}} \xi_i x_i + \sum_{i \in S_4^{w_1,b_1}} x_i\right) = 0 \end{aligned} \tag{59}$$

where $a_j \in [-1,0]$, $b_j \in [0,1]$ and $\psi_i \in [-\tau, 0]$, $\xi_i \in [-\tau_1, 1]$. In comparison to the index set $S_0^{w_1,b_1}$, $S_2^{w_1,b_1}$ and $S_4^{w_1,b_1}$, the number of samples in $S_1^{w_1,b_1}$ and $S_3^{w_1,b_1}$ is very less, so we will not consider them in the analysis. Fixing the value of s_2, number of samples in $S_0^{w_1,b_1}$, $S_2^{w_1,b_1}$ and $S_4^{w_1,b_1}$ depend on τ_2. If the number of samples in $S_0^{w_1,b_1}$ is getting effected, it also effects the sparsity of the model. The number of samples in $S_4^{w_1,b_1}$ is small when τ_2 is small, which results in a model sensitive to feature noise around the decision hyperplane. Conversely, if τ_2 is large, $S_k^{w_1,b_1}$ ($k \in \{0,2,4\}$) will have large number of samples, resulting in less sensitive model. Changing τ_2 changes number of samples in $S_0^{w_1,b_1}$, hence also effecting the sparsity of model. Thus, both s_2, τ_2 determine the noise sensitivity and sparsity of the model. The bar graph shown in Fig. 2 shows the effect of changing τ on support vectors, hence sparsity of binary UCI datasets with a fixed s. Similarly, the bar graph in Fig. 2(b) shows the consequence of changing s with fixed τ on the support vectors, thus sparsity of the binary UCI datasets. Table 1 shows the comparison among the proposed Tpin-UTWSVM model and the baseline models.

The following proposition which is also followed by Pin-GTSVM (Tanveer et al., 2019b) gives the upper bound on the negative and positive boundary errors.

Proposition 1. *Assume r_1 represents the number of negative boundary errors. Then, there exists an upper bound given by $\frac{e_2^T \beta_1}{C_1(1+\tau_2)}$ on number of negative boundary errors.*

Proof. For negative data points having error *i.e.,* $\zeta_2 \neq 0$. We have from Eq. (35), $\mu_1 = 0$ which implies $\beta_1 = (1+\tau_2)C_1 e_2$.

$\beta_{1i} = (1+\tau_2)C_1 \quad \forall\ i = 1, 2, \ldots, m_2.$

As per assumption, r_1 points has $\zeta_2 \neq 0$.

$$\begin{aligned} &\implies r_1 \beta_{1i} \leq e_2^T \beta_1 \\ &\implies r_1(1+\tau_2)C_1 \leq e_2^T \beta_1 \\ &\implies r_1 \leq \frac{e_2^T \beta_1}{C_1(1+\tau_2)}. \end{aligned}$$

Proposition 2. *Assume r_2 represents the number of positive boundary errors. Then, there exists an upper bound given by $\frac{e_1^T \beta_2}{C_2(1+\tau_1)}$ on number of positive boundary errors.*

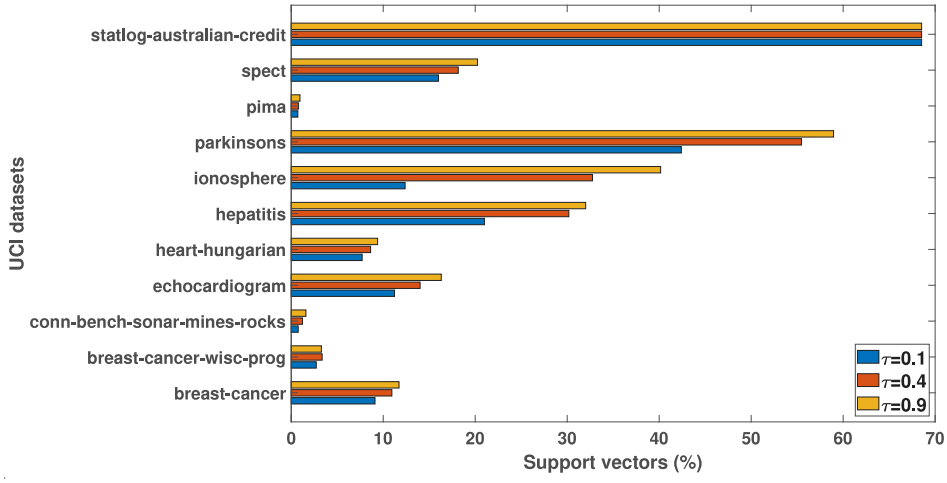
Proof. Proceed similarly as above.

4.2. Complexity

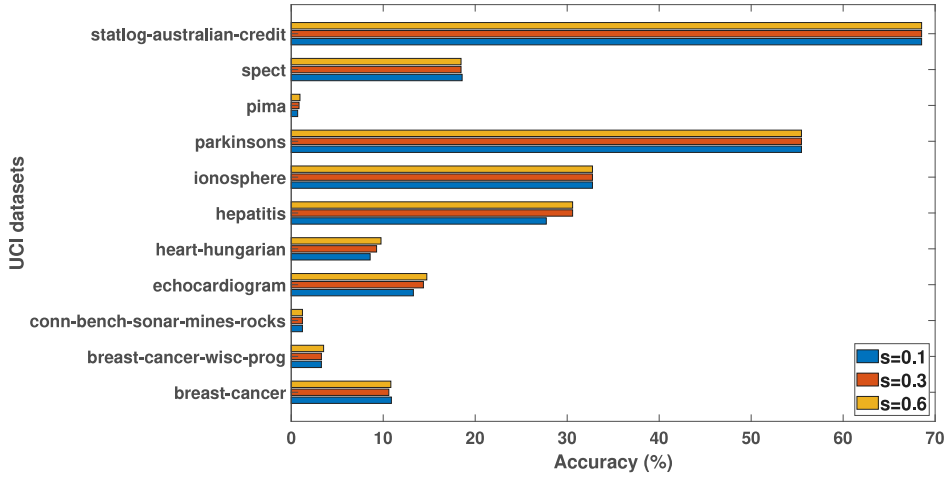
The computational complexity of SVM is $O(m^3)$ (Cortes and Vapnik, 1995), m being the number of samples. In TWSVM, we divide the samples into two parts, so its computational complexity is approximately $O(2 * (m/2)^3)$. The computational complexity of Pin-GTSVM and TWSVM are similar (Tanveer et al., 2019b). Due to the non convexity of the proposed model, the complexity may increase. However, CCCP acts as a speedup technique and compensates for the complexity. The computational cost of taking inverse of a matrix is $O(n^3)$ (Zhang et al., 2015), where n is the order of the matrix. Also, in the proposed Tpin-UTWSVM, we considered universum data leading to complexity $O((m+2|U|)^3/4)$ (Richhariya et al., 2021b), where $|U|$ denotes the number of universum data. Thus, universum data is responsible for the increased computational time as compared to the baseline models. Hence, the approximate complexity of the proposed Tpin-UTWSVM is

$O((m+2|U|)^3/4)$.

To make the training of the proposed Tpin-UTWSVM efficient, we used successive overrelaxation (SOR) technique (Mangasarian and Musicant, 1999). It has the advantage of processing datasets without storing them in memory (Shao et al., 2011). We have also applied SOR technique to the TWSVM, UTSVM and Pin-GTSVM, thus boosting their performance.



(a) Bar graph representing the effect of changing τ with fixed $s = 0.2$ on the support vectors of noise free data ($r = 0$).



(b) Bar graph representing the effect of changing s with fixed $\tau = 0.4$ on the support vectors of noise free data ($r = 0$).

Fig. 2. Bar graph illustrating.

5. Experiments

In this section, we performed numerical experiments to compare the proposed Tpin-UTWSVM model with the baseline models *i.e.,* pin SVM (Huang et al., 2013), TWSVM (Jayadeva et al., 2007), UTSVM (Qi et al., 2012), Pin-GTSVM (Tanveer et al., 2019b) and carried out statistical tests to demonstrate the superiority among the compared models. We also evaluated the aforementioned models for the classification of EEG signals and detection of Alzheimer's disease.

5.1. Experimental setup

All the experiments have been performed on a system with MATLAB R2022a version and Windows 10 operating system, with Intel(R) Xenon(R) W-2223 CPU @ 3.6 GHz processor and 64 GB RAM. To obtain the optimal parameters, we used grid search with ten fold cross validation (CV) which considers each possible combination of the parameter values and split the training datasets into 10 parts. Taking one part as the validation data and remaining as the training data, we obtain classification accuracy and training time. Repeating it 9 more times with each different part as the validation data, we get 10 different classification accuracies with training times. Averaging the 10 accuracies and training times, we get the accuracy and time with particular set of parameter values. The combination of parameter values corresponding to the highest classification accuracy are called optimal hyperparameters. We have split training and testing data randomly in the ratio 70 : 30. Using the optimal hyperparameters, obtain the accuracy on testing data and refer it as testing accuracy. In further article, we will term testing accuracy as accuracy. For the non-linear case, we have considered Gaussian kernel given by

$$K(x_p, x_q) = exp\left(-\frac{\left\|x_p - x_q\right\|^2}{2\sigma^2}\right),$$

where σ signifies the kernel parameter. For the comparison of the proposed Tpin-UTWSVM model, we have evaluated the models on 19 benchmark binary datasets from the UCI repository (Newman, 1998), Alzheimer's disease data publicly available on Alzheimer's Disease Neuroimaging Initiative (ADNI), and EEG data publicly available in the article (Andrzejak et al., 2001). The training and testing data are corrupted by the same zero mean Gaussian noise. The ratio of variance of noise to that of each feature (denoted by r) is at $r = 0,\ 0.05$ and 0.1 (Huang et al., 2013). We have employed z-score normalization to the datasets.

Table 1
Comparison of the models under study.

pin SVM (Huang et al., 2013)	TWSVM (Jayadeva et al., 2007)	UTSVM (Qi et al., 2012)	Pin-GTSVM (Tanveer et al., 2019b)	Proposed Tpin-UTWSVM
Uses pinball loss for imposing penalty to both correctly and wrongly classified data points.	Uses hinge loss for imposing penalty to misclassified data points.	Uses hinge loss as penalty for misclassified data points and ϵ insensitive loss for universum data.	Uses pinball loss for imposing penalty on both correctly and incorrectly classified data points.	Uses truncated pinball loss for imposing penalty on both correctly and incorrectly classified data points.
Insensitive to noise.	Sensitive to noise.	Sensitive to noise.	Insensitive to noise.	Insensitive to noise.
The model is not sparse.	The model is sparse.	The model is sparse.	The model is not sparse.	The model is sparse.
Robust to outliers.	Effected by outliers.	Effected by outliers.	Robust to outliers.	Robust to outliers.
Leads to convex optimization problems.	Leads to convex optimization problems.	Leads to convex optimization problems.	Leads to convex optimization problems.	Leads to non-convex optimization problems.
It has very high complexity of $O(m^3)$.	It is efficient than pin SVM having complexity $\approx O((m/2)^3)$ (Jayadeva et al., 2007).	The complexity of the model is $O((m+2\|U\|)^3)$ (Richhariya et al., 2021b)	Its complexity is similar to TWSVM (Tanveer et al., 2019b).	It has complexity similar to UTSVM, however the sparsity of the model leads to reduced complexity as compared to UTSVM.
The pinball loss function has theoretical properties of minimization leading to Bayes's classifier and bounded misclassification error by Zhang's inequality.	Hinge loss is not a Bayesian classifier and does not have a bounded misclassification error.	Hinge loss is not a Bayesian classifier and does not have a bounded misclassification error.	The loss function of the model has the theoretical properties of being a Bayesian classifier and having a bounded misclassification error.	The truncated pinball loss function has the theoretical properties of pinball loss (Shen et al., 2017).

5.2. Parameter selection

For the baseline models and the proposed Tpin-UTWSVM model, we set the penalty parameters $C_1 = C_2$, to ease the computational complexity. The parameters C_k, ($k = 1, 2$ and u) of the proposed Tpin-UTWSVM model and baseline models i.e., pin SVM (Huang et al., 2013), TWSVM (Jayadeva et al., 2007), UTSVM (Qi et al., 2012), Pin-GTSVM (Tanveer et al., 2019b) have been chosen from the range: $[10^{-5}, \ldots, 10^{5}]$, σ is chosen from $[2^{-5}, \ldots, 2^{5}]$ (Richhariya and Tanveer, 2020a). The truncated pinball loss parameter for the proposed Tpin-UTWSVM are chosen as $s_1 = s_2 = 0.25$, $\tau_1 = \tau_2 = 0.5$ following the article (Wang et al., 2021). For Pin-GTSVM, τ is considered as 0.5. The parameter ϵ in the proposed Tpin-UTWSVM model and UTSVM is chosen from the range $[0.1, 0.3, 0.5, 0.6]$ (Richhariya et al., 2018). τ for pin SVM is taken from the range $[0.1, 0.2, 0.5, 1]$ from Huang et al. (2013). To initialize the variables for the CCCP algorithm, we have followed initialization method in Wang et al. (2015). Following the article (Tanveer et al., 2019b), for evaluation of the models, we have calculated the metric accuracy. To evaluate the compared models in a better way, two additional metrics precision and recall are being considered. These metrics are defined as follows:

$$\text{Accuracy} = \frac{tp + tn}{tp + fp + tn + fn},$$

$$\text{Recall} = \frac{tp}{tp + fn},$$

$$\text{Precision} = \frac{tp}{tp + fp},$$

where tp, tn and fp, fn signify the correctly classified positive, negative samples and misclassified positive, negative samples, respectively.

5.3. Experimental analysis on the binary datasets

The classification performance of the proposed Tpin-UTWSVM along with models pin SVM (Huang et al., 2013), TWSVM (Jayadeva et al., 2007), UTSVM (Qi et al., 2012) and Pin-GTSVM (Tanveer et al., 2019b) on the binary datasets is shown in Table 2, where sd denotes the standard deviation of the model on each dataset. Accuracy with plus minus standard deviation, recall and precision for the three noise levels $r = 0, 0.05, 0.1$ have been shown in Table 2 for each model on the randomly chosen 19 UCI datasets. For $r = 0$, the average accuracies of pin SVM, TWSVM, UTSVM, Pin-GTSVM and the proposed Tpin-UTWSVM are 79.76, 79.88, 79.79, 76.91 and 80.93, respectively. For the same order of models, the average accuracies are 79.15, 80.18, 77.63, 77.6 and 80.74 for $r = 0.05$ and 78.9, 80.33, 77.75, 77.23 and 80.68 for $r = 0.1$. Thus, though the level of noise increases, the performance of the proposed Tpin-UTWSVM remains the best. There may be a possibility that higher accuracy of the model over one dataset compensates for lower accuracy over other datasets, hence, influencing the average accuracy. In order to overcome such biasness, we assigned ranks to each dataset for every model. The best performing model over a dataset is given rank 1 and vice-versa. For models having the same performance on a dataset, consider same ranks and take the average of the considered ranks and assign it to the models having same performance. The lowest average rank signifies the superior model. Among the models into consideration, for all the three values of r, the proposed Tpin-UTWSVM has the best average rank *i.e.,* 2.55, 2.32 and 2.18 for $r = 0,\ 0.05$ and 0.1, respectively. The training time and optimal hyperparameters for the models under comparison over the 19 UCI datasets are shown in Table 3. The Table 3 also includes size $(f \times s)$ of each UCI dataset, where f and s denote the number of features and number of samples of the datasets, respectively. Fig. 3 represents the comparison of accuracies using bar graph of different models under study with different noise levels ($r = 0, 0.05, 0.1$) over 19 UCI datasets. Fig. 4 depicts the comparison of training time of TWSVM, UTSVM, Pin-GTSVM and proposed Tpin-UTWSVM for all the noise levels and datasets under consideration. Comparison with pin SVM is shown in the Table 3 only, for the ease of representation it is not included in Fig. 4.

5.4. Sensitivity of hyperparameters

The hyperparameters C_1, C_u and σ effect the generalization performance of the proposed Tpin-UTWSVM model which is depicted in Fig. 5. It clearly represents the evaluation of sensitivity of the proposed Tpin-UTWSVM model with user specified hyperparameters having Gaussian kernel on UCI data. Figs. 5(a) and 5(b) show the effect of C_1 and C_u on the accuracy of blood, and heart-hungarian data, respectively. Fig. 5(a) shows the accuracy decreases with increased value of C_u for blood data and for the heart-hungarian data (Fig. 5(b)) middle range of values of C_u have good performance. Figs. 5(c) and 5(d) depict the effect of varying the pair (σ, C_u) and (C_1, σ) on the performance of breast-cancer and breast-cancer-wisc data, respectively. Fig. 5(c) represents increase in accuracy of the model at the higher and lower range values of C_u. Fig. 5(d) shows increase in accuracy of the proposed Tpin-UTWSVM model for breast-cancer-wisc data at the middle range values of C_1 which remains nearly constant at higher values of C_1.

Table 2
Performance of the proposed Tpin-UTWSVM model with baseline models on UCI binary datasets with Gaussian kernel.

Dataset	r	pin SVM (Huang et al., 2013)	TWSVM (Jayadeva et al., 2007)	UTSVM (Qi et al., 2012)	Pin-GTSVM (Tanveer et al., 2019b)	Tpin-UTWSVM
		(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)
blood	0	(79.91 ± 5.88) (0.377, 0.7667)	(76.34 ± 6.28) (0.2951, 0.6429)	(78.57 ± 5.12) (0.3279, 0.7407)	(63.84 ± 7.71) (0.3934, 0.3529)	(76.79 ± 6.48) (0.2951, 0.6667)
	0.05	(76.34 ± 5.26) (0.3443, 0.6176)	(77.68 ± 5.91) (0.3115, 0.7037)	(73.66 ± 5.8) (0.0328, 1)	(72.32 ± 6.45) (0, 0)	(78.13 ± 5.09) (0.2951, 0.75)
	0.1	(72.77 ± 6.84) (0.3607, 0.5)	(77.23 ± 5.18) (0.3279, 0.6667)	(74.11 ± 6.04) (0.1148, 0.6364)	(72.32 ± 6.63) (0, 0)	(77.23 ± 6.26) (0.3279, 0.6667)
breast-cancer	0	(74.42 ± 8.23) (0.4348, 0.5263)	(74.42 ± 11.35) (0.5652, 0.52)	(80.23 ± 10.87) (0.3043, 0.875)	(72.09 ± 10.27) (0, 0)	(76.74 ± 10.12) (0.4348, 0.5882)
	0.05	(74.42 ± 7.17) (0.4348, 0.5263)	(74.42 ± 11.56) (0.5652, 0.52)	(76.74 ± 10.87) (0.1739, 0.8)	(70.93 ± 8.43) (0.3913, 0.45)	(75.58 ± 11.17) (0.4783, 0.55)
	0.1	(73.26 ± 9.44) (0.2174, 0.5)	(74.42 ± 11.83) (0.5652, 0.52)	(79.07 ± 8.88) (0.3478, 0.7273)	(69.77 ± 11.11) (0.5652, 0.4483)	(75.58 ± 11.89) (0.5217, 0.5455)
breast-cancer-wisc	0	(95.71 ± 2.57) (0.961, 0.925)	(96.67 ± 2.44) (0.961, 0.9487)	(95.24 ± 2.31) (0.9351, 0.9351)	(95.71 ± 1.73) (0.9221, 0.9595)	(96.67 ± 2.44) (0.961, 0.9487)
	0.05	(95.71 ± 2.52) (0.961, 0.925)	(93.81 ± 2.2) (0.8961, 0.9324)	(94.76 ± 3.05) (0.8961, 0.9583)	(94.76 ± 1.73) (0.8961, 0.9583)	(94.29 ± 1.38) (0.8831, 0.9577)
	0.1	(96.19 ± 2.03) (0.974, 0.9259)	(94.76 ± 1.99) (0.8961, 0.9583)	(96.19 ± 2.51) (0.974, 0.9259)	(95.24 ± 1.73) (0.9221, 0.9467)	(94.29 ± 1.44) (0.8831, 0.9577)
breast-cancer-wisc-prog	0	(74.58 ± 5.96) (0, 0)	(74.58 ± 4.87) (0, 0)	(77.97 ± 10.06) (0, 0)	(74.58 ± 5.91) (0, 0)	(72.88 ± 12.58) (0.3077, 0.3636)
	0.05	(74.58 ± 5.96) (0, 0)	(74.58 ± 4.92) (0, 0)	(69.49 ± 8.96) (0, 0)	(74.58 ± 4.92) (0, 0)	(72.88 ± 11.22) (0.3077, 0.3636)
	0.1	(74.58 ± 5.96) (0, 0)	(76.27 ± 9.61) (0.0769, 0.3333)	(72.88 ± 9.48) (0, 0)	(74.58 ± 4.92) (0, 0)	(76.27 ± 5.96) (0, 0)
conn-bench-sonar-mines-rocks	0	(82.26 ± 11.99) (0.8788, 0.8056)	(79.03 ± 5.61) (0.6364, 0.9545)	(82.26 ± 9.48) (0.8485, 0.8235)	(85.48 ± 9.75) (0.7273, 1)	(82.26 ± 7.28) (0.7576, 0.8929)
	0.05	(80.65 ± 11.99) (0.8485, 0.8)	(85.48 ± 10.21) (0.7273, 1)	(80.65 ± 6.31) (0.8485, 0.8)	(85.48 ± 10.21) (0.7273, 1)	(85.48 ± 7.93) (0.8182, 0.9)
	0.1	(82.26 ± 10.64) (0.8485, 0.8235)	(85.48 ± 9.62) (0.7273, 1)	(79.03 ± 7.42) (0.8182, 0.7941)	(85.48 ± 9.53) (0.7273, 1)	(85.48 ± 6.36) (0.7879, 0.9286)
echocardiogram	0	(84.62 ± 11.16) (0.4545, 1)	(87.18 ± 10.33) (0.5455, 1)	(89.74 ± 9.28) (0.6364, 1)	(74.36 ± 15.57) (0.6364, 0.5385)	(89.74 ± 10.76) (0.6364, 1)
	0.05	(84.62 ± 11.16) (0.4545, 1)	(89.74 ± 10.32) (0.6364, 1)	(87.18 ± 10.76) (0.5455, 1)	(71.79 ± 14.06) (0.6364, 0.5)	(89.74 ± 10.76) (0.6364, 1)
	0.1	(82.05 ± 12.68) (0.3636, 1)	(89.74 ± 9.4) (0.6364, 1)	(87.18 ± 10.76) (0.5455, 1)	(71.79 ± 16.52) (0.6364, 0.5)	(89.74 ± 10.33) (0.6364, 1)
haberman-survival	0	(69.57 ± 11.78) (0.0741, 0.4)	(69.57 ± 12.39) (0.0741, 0.4)	(73.91 ± 11.34) (0.2222, 0.6667)	(67.39 ± 13.48) (0, 0)	(69.57 ± 11.59) (0.037, 0.3333)
	0.05	(69.57 ± 12.29) (0.0741, 0.4)	(69.57 ± 12.39) (0.0741, 0.4)	(71.74 ± 12.61) (0.1111, 0.6)	(70.65 ± 12.43) (0, −)	(70.65 ± 12.25) (0.037, 0.5)
	0.1	(69.57 ± 12.1) (0.0741, 0.4)	(70.65 ± 12.23) (0.1111, 0.5)	(73.91 ± 12.11) (0.2222, 0.6667)	(70.65 ± 12.8) (0, −)	(71.74 ± 12.01) (0.037, 1)
heart-hungarian	0	(86.36 ± 8.03) (0.6774, 0.913)	(85.23 ± 7.77) (0.7419, 0.8214)	(78.41 ± 11.66) (0.7419, 0.6765)	(84.09 ± 11.9) (0.7742, 0.7742)	(86.36 ± 6.59) (0.8065, 0.8065)
	0.05	(86.36 ± 7.48) (0.6774, 0.913)	(78.41 ± 9.7) (0.7097, 0.6875)	(81.82 ± 5.25) (0.8387, 0.7027)	(79.55 ± 12.03) (0.7419, 0.697)	(84.09 ± 10.08) (0.7419, 0.7931)
	0.1	(86.36 ± 7.48) (0.6774, 0.913)	(80.68 ± 8.2) (0.6774, 0.75)	(79.55 ± 12.41) (0.7097, 0.7097)	(78.41 ± 11.04) (0.7419, 0.6765)	(84.09 ± 8.87) (0.7742, 0.7742)
heart-switzerland	0	(54.05 ± 20.39) (0, 0)	(56.76 ± 14.34) (0.2353, 0.5714)	(54.05 ± 14.73) (0.3529, 0.5)	(54.05 ± 25.06) (0.4118, 0.5)	(56.76 ± 14.34) (0.2353, 0.5714)
	0.05	(51.35 ± 16.81) (0, 0)	(56.76 ± 16.97) (0.2353, 0.5714)	(51.35 ± 10.74) (0.2941, 0.4545)	(54.05 ± 18.35) (0, 0)	(56.76 ± 16.31) (0.2353, 0.5714)
	0.1	(48.65 ± 14.38) (0.0588, 0.25)	(56.76 ± 16.97) (0.2353, 0.5714)	(48.65 ± 16.48) (0.1765, 0.375)	(54.05 ± 18.35) (0, 0)	(56.76 ± 16.97) (0.2353, 0.5714)
hepatitis	0	(86.96 ± 9.6) (0.875, 0.9722)	(91.3 ± 12.7) (0.925, 0.9737)	(91.3 ± 9.6) (0.95, 0.95)	(86.96 ± 8.72) (0.9, 0.9474)	(89.13 ± 11.16) (0.925, 0.9487)
	0.05	(86.96 ± 11.35) (0.875, 0.9722)	(91.3 ± 10.88) (0.925, 0.9737)	(91.3 ± 9.6) (1, 0.9091)	(89.13 ± 9.74) (0.925, 0.9487)	(89.13 ± 10.88) (0.925, 0.9487)
	0.1	(86.96 ± 11.35) (0.875, 0.9722)	(91.3 ± 12.7) (0.925, 0.9737)	(91.3 ± 9.6) (0.95, 0.95)	(91.3 ± 12.23) (0.95, 0.95)	(91.3 ± 12.38) (0.925, 0.9737)
ionosphere	0	(90.48 ± 6.64) (0.8592, 1)	(97.14 ± 3.43) (0.9859, 0.9722)	(97.14 ± 7.43) (0.9859, 0.9722)	(96.19 ± 3.14) (0.9718, 0.9718)	(96.19 ± 3.47) (0.9577, 0.9855)
	0.05	(89.52 ± 7.61) (0.8451, 1)	(97.14 ± 2.86) (0.9859, 0.9722)	(93.33 ± 4.36) (0.9155, 0.9848)	(95.24 ± 5.56) (0.9577, 0.9714)	(95.24 ± 3.95) (0.9437, 0.9853)
	0.1	(91.43 ± 8.55) (0.8732, 1)	(97.14 ± 3.91) (0.9859, 0.9722)	(94.29 ± 4.13) (0.9296, 0.9851)	(94.29 ± 6.31) (0.9296, 0.9851)	(95.24 ± 5.49) (0.9577, 0.9714)
molec-biol-promoter	0	(75 ± 9) (0.7692, 0.6667)	(71.88 ± 9.93) (0.7692, 0.625)	(65.63 ± 12.21) (0.7692, 0.5556)	(71.88 ± 9.93) (0.7692, 0.625)	(68.75 ± 6.58) (0.6923, 0.6)

(*continued on next page*)

Table 2 (*continued*).

Dataset	r	pin SVM (Huang et al., 2013)	TWSVM (Jayadeva et al., 2007)	UTSVM (Qi et al., 2012)	Pin-GTSVM (Tanveer et al., 2019b)	Tpin-UTWSVM
		(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)
	0.05	(75 ± 9) (0.7692, 0.6667)	(71.88 ± 9.93) (0.7692, 0.625)	(65.63 ± 15.5) (0.6923, 0.5625)	(71.88 ± 9.93) (0.7692, 0.625)	(71.88 ± 8.89) (0.7692, 0.625)
	0.1	(75 ± 10.61) (0.7692, 0.6667)	(71.88 ± 9.93) (0.7692, 0.625)	(65.63 ± 13.31) (0.8462, 0.55)	(71.88 ± 9.93) (0.7692, 0.625)	(71.88 ± 10.61) (0.7692, 0.625)
monks-1	0	(88.62 ± 4.39) (0.8621, 0.9146)	(98.8 ± 2.96) (0.977, 1)	(92.81 ± 4.01) (0.8851, 0.9747)	(91.62 ± 4.51) (0.954, 0.8925)	(100 ± 1.81) (1, 1)
	0.05	(86.83 ± 4.72) (0.8736, 0.8736)	(95.81 ± 3.02) (0.9195, 1)	(87.43 ± 4.33) (0.7701, 0.9853)	(94.01 ± 3.58) (0.9425, 0.9425)	(95.21 ± 3.59) (0.9425, 0.9647)
	0.1	(85.63 ± 4.19) (0.8621, 0.8621)	(93.41 ± 2.96) (0.931, 0.9419)	(84.43 ± 5.81) (0.8046, 0.8861)	(89.22 ± 2.48) (0.8736, 0.9157)	(90.42 ± 4.52) (0.908, 0.908)
parkinsons	0	(89.66 ± 8.18) (0.9778, 0.898)	(93.1 ± 4.32) (0.9778, 0.9362)	(91.38 ± 6.34) (0.9556, 0.9348)	(93.1 ± 6.29) (0.9778, 0.9362)	(96.55 ± 5.74) (1, 0.9574)
	0.05	(86.21 ± 8.37) (0.9778, 0.8627)	(93.1 ± 5.09) (0.9778, 0.9362)	(93.1 ± 7.75) (0.9778, 0.9362)	(94.83 ± 5.46) (0.9778, 0.9565)	(94.83 ± 5.28) (0.9778, 0.9565)
	0.1	(87.93 ± 7.96) (1, 0.8654)	(94.83 ± 5.09) (1, 0.9375)	(89.66 ± 9.62) (0.9778, 0.898)	(93.1 ± 5.83) (0.9778, 0.9362)	(94.83 ± 4.54) (0.9778, 0.9565)
pima	0	(74.35 ± 4.38) (0.4839, 0.8036)	(76.52 ± 4.15) (0.5054, 0.8545)	(71.3 ± 7.89) (0.4946, 0.7077)	(65.22 ± 5.85) (0.5484, 0.573)	(76.09 ± 4.65) (0.4839, 0.8654)
	0.05	(74.35 ± 3.75) (0.4839, 0.8036)	(76.52 ± 4.28) (0.5054, 0.8545)	(71.3 ± 6.65) (0.4839, 0.7143)	(67.39 ± 5.63) (0.5591, 0.6047)	(74.78 ± 5.22) (0.4194, 0.907)
	0.1	(73.91 ± 4.17) (0.4946, 0.7797)	(75.65 ± 3.65) (0.4946, 0.8364)	(73.48 ± 4.3) (0.5054, 0.7581)	(67.39 ± 6) (0.5484, 0.6071)	(75.65 ± 4.12) (0.4731, 0.8627)
pittsburg-bridges-T-OR-D	0	(90.32 ± 17.82) (0.6667, 0.5)	(87.1 ± 11.27) (0, 0)	(87.1 ± 12.05) (0, 0)	(83.87 ± 12.51) (0, 0)	(87.1 ± 9.99) (0, 0)
	0.05	(93.55 ± 17.1) (0.6667, 0.6667)	(87.1 ± 11.27) (0, 0)	(80.65 ± 11.76) (0, 0)	(87.1 ± 12.51) (0, 0)	(83.87 ± 9.99) (0, 0)
	0.1	(93.55 ± 14.75) (0.6667, 0.6667)	(87.1 ± 11.27) (0, 0)	(87.1 ± 11.76) (0, 0)	(87.1 ± 9.52) (0, 0)	(87.1 ± 10.1) (0, 0)
planning	0	(78.18 ± 15.75) (0.0769, 1)	(78.18 ± 15.75) (0.0769, 1)	(78.18 ± 15.75) (0.0769, 1)	(78.18 ± 15.75) (0.0769, 1)	(78.18 ± 15.75) (0.0769, 1)
	0.05	(78.18 ± 15.75) (0.0769, 1)	(78.18 ± 15.75) (0.0769, 1)	(76.36 ± 15.84) (0, 0)	(76.36 ± 15.75) (0, −)	(78.18 ± 15.75) (0.0769, 1)
	0.1	(78.18 ± 15.75) (0.0769, 1)	(78.18 ± 15.75) (0.0769, 1)	(74.55 ± 15.84) (0, 0)	(76.36 ± 15.75) (0, −)	(78.18 ± 15.75) (0.0769, 1)
spect	0	(74.68 ± 13.93) (0.5625, 0.75)	(58.23 ± 16.48) (0.4375, 0.4828)	(64.56 ± 15.49) (0.2188, 0.7)	(56.96 ± 14.89) (0.5938, 0.475)	(72.15 ± 13.85) (0.625, 0.6667)
	0.05	(73.42 ± 16.26) (0.5938, 0.7037)	(65.82 ± 15.18) (0.4688, 0.6)	(63.29 ± 14.82) (0.2813, 0.6)	(58.23 ± 9.89) (0.2813, 0.4737)	(77.22 ± 14.67) (0.625, 0.7692)
	0.1	(74.68 ± 16.06) (0.625, 0.7143)	(64.56 ± 16.61) (0.4375, 0.5833)	(62.03 ± 12.31) (0.5, 0.5333)	(58.23 ± 12.68) (0.2813, 0.4737)	(70.89 ± 15.2) (0.5938, 0.6552)
statlog-australian-credit	0	(65.7 ± 6.36) (0.9927, 0.6602)	(65.7 ± 5.72) (0.9927, 0.6602)	(66.18 ± 5.92) (1, 0.6618)	(65.7 ± 5.72) (0.9927, 0.6602)	(65.7 ± 5.72) (0.9927, 0.6602)
	0.05	(66.18 ± 5.98) (1, 0.6618)	(66.18 ± 5.98) (1, 0.6618)	(65.22 ± 5.84) (0.9854, 0.6585)	(66.18 ± 5.98) (1, 0.6618)	(66.18 ± 5.72) (1, 0.6618)
	0.1	(66.18 ± 7.16) (0.9562, 0.6718)	(66.18 ± 5.98) (1, 0.6618)	(64.25 ± 5.94) (0.9635, 0.6567)	(66.18 ± 5.98) (1, 0.6618)	(66.18 ± 5.98) (1, 0.6618)
average accuracy	0	79.76	79.88	79.79	76.91	**80.93**
	0.05	79.15	80.18	77.63	77.6	**80.74**
	0.1	78.9	80.33	77.75	77.23	**80.68**
average rank	0	3.08	2.74	2.76	3.87	**2.55**
	0.05	3.13	2.5	3.63	3.42	**2.32**
	0.1	3.21	2.26	3.66	3.68	**2.18**

5.5. Statistical analysis on the binary datasets

We will conduct statistical test to show the difference among pin SVM (Huang et al., 2013), TWSVM (Jayadeva et al., 2007), UTSVM (Qi et al., 2012), Pin-GTSVM (Tanveer et al., 2019b) and the proposed Tpin-UTWSVM models. The statistical test conducted is Friedman test (Demšar, 2006).

In the Friedman test, we have the null hypothesis assumption that each model has the same average rank. Friedman statistic has χ^2 distribution and it is calculated as:

$$\chi_F^2 = \frac{12L}{d(d+1)}\left[\sum_i R_i^2 - \frac{d(d+1)^2}{4}\right], \tag{60}$$

where d and L signify the number of classifiers and datasets, respectively. R_i denotes the average rank of i^{th} dataset. The F_F statistics is obtained by:

$$F_F = \frac{(L-1)\chi_F^2}{L(d-1)-\chi_F^2} \tag{61}$$

and has the degree of freedom $((d-1),(d-1)(L-1))$. Here, we have $L = 19$ and $d = 5$. On simple calculation, we get $\chi_F^2 = 8.2916$ and $F_F = 2.2043$. F_F has degree of freedom $(4, 72)$. The critical value $F_{(4,72)} = 2.4989$ with 5% level of significance. As the obtained F_F is greater than $F_{(4,72)}$, so the null hypothesis fails, and the considered models are statistically different.

Also, Table 4 gives the detail about number of wins, ties and loses of the proposed Tpin-UTWSVM in comparison to baseline models at different noise levels ($r = 0, 0.05$ and 0.1). The table clearly shows the proposed model has more number of wins for all the three cases.

Table 3
Training time and optimal hyperparameters of models on binary datasets with Gaussian kernel.

Dataset	r	pin SVM (Huang et al., 2013)	TWSVM (Jayadeva et al., 2007)	UTSVM (Qi et al., 2012)	Pin-GTSVM (Tanveer et al., 2019b)	Tpin-UTWSVM
Size ($f \times s$)		(Time (s), τ) (C_1, σ)	Time (s) ($C_1 = C_2, \sigma$)	(Time (s), C_u) ($C_1 = C_2, \sigma, \epsilon$)	Time (s) ($C_1 = C_2, \sigma$)	(Time (s), C_u) ($C_1 = C_2, \sigma, \epsilon$)
blood	0	(5.9297, 0.2)	0.0426	(0.0989, 1)	0.0742	(0.1736, 10^{-5})
(748 × 4)		(10, 1)	(0.1, 4)	(0.01, 16, 0.1)	(10^{-5}, 0.5)	(0.1, 4, 0.1)
	0.05	(5.59, 0.5)	0.0457	(0.0479, 0.01)	0.0874	(0.3374, 1)
		(100, 1)	(0.1, 2)	(0.1, 16, 0.1)	(10^{-5}, 0.03125)	(0.1, 2, 0.5)
	0.1	(5.2235, 1)	0.0474	(0.0318, 0.001)	0.1307	(0.2518, 0.1)
		(1000, 1)	(0.1, 4)	(0.01, 0.5, 0.6)	(10^{-5}, 0.03125)	(0.1, 4, 0.6)
breast-cancer	0	(0.3723, 0.1)	0.021	(0.005, 10^{-5})	0.0052	(0.0674, 10^{-5})
(286 × 9)		(1, 0.25)	(1, 8)	(10^{-5}, 2, 0.1)	(10^{-5}, 0.03125)	(1, 8, 0.1)
	0.05	(0.3888, 0.1)	0.0097	(0.0043, 0.001)	0.0068	(0.0663, 10^{-5})
		(1, 0.25)	(1, 8)	(0.01, 4, 0.1)	(0.01, 32)	(1, 8, 0.1)
	0.1	(0.2869, 0.1)	0.0104	(0.0055, 0.001)	0.0072	(0.1203, 0.1)
		(1, 0.0625)	(1, 8)	(0.1, 2, 0.5)	(0.01, 32)	(1, 8, 0.6)
breast-cancer-wisc	0	(6.3268, 0.1)	0.0223	(0.0345, 10^{-5})	0.0697	(0.0619, 10^{-5})
(699 × 9)		(0.01, 1)	(10^{-5}, 8)	(10^{-5}, 4, 0.1)	(0.1, 16)	(10^{-5}, 8, 0.1)
	0.05	(5.673, 0.1)	0.026	(0.0336, 0.001)	0.0672	(0.2601, 10^{-5})
		(0.01, 1)	(10^{-4}, 4)	(0.001, 8, 0.1)	(0.1, 16)	(10, 16, 0.1)
	0.1	(2.3398, 0.1)	0.0243	(0.0436, 0.01)	0.0648	(0.2994, 0.1)
		(10, 1)	(0.01, 8)	(0.1, 4, 0.5)	(10^{-5}, 32)	(10, 16, 0.1)
breast-cancer-wisc-prog	0	(0.1086, 0.1)	0.0023	(0.0023, 10^{-5})	0.0016	(0.0394, 10)
(198 × 33)		(1, 0.5)	(0.001, 2)	(10^{-5}, 4, 0.1)	(10^{-5}, 2)	(1, 32, 0.3)
	0.05	(0.1479, 0.1)	0.0024	(0.0021, 10^{-4})	0.0021	(0.0369, 10)
		(1, 0.5)	(10^{-4}, 2)	(10^{-5}, 4, 0.1)	(10^{-5}, 2)	(1, 32, 0.3)
	0.1	(0.1237, 0.1)	0.0026	(0.0023, 10^{-4})	0.002	(0.0063, 10^{-5})
		(1, 0.5)	(10^{-4}, 4)	(10^{-5}, 4, 0.1)	(10^{-5}, 2)	(10^{-4}, 2, 0.5)
conn-bench-sonar-mines-rocks	0	(0.1233, 0.1)	0.0031	(0.003, 0.01)	0.0025	(0.0084, 10^{-4})
(208 × 60)		(1, 0.25)	(10^{-5}, 8)	(0.001, 4, 0.1)	(10^{-5}, 4)	(10^{-4}, 8, 0.5)
	0.05	(0.1226, 0.1)	0.0031	(0.0039, 0.01)	0.0027	(0.0068, 10^{-4})
		(10, 0.25)	(10^{-4}, 4)	(0.001, 4, 0.1)	(10^{-5}, 4)	(10^{-4}, 8, 0.6)
	0.1	(0.1338, 0.1)	0.0029	(0.004, 0.01)	0.0027	(0.0083, 10^{-5})
		(1, 0.25)	(10^{-4}, 4)	(10^{-4}, 4, 0.1)	(10^{-4}, 4)	(10^{-4}, 8, 0.5)
echocardiogram	0	(0.0507, 0.1)	0.0016	(0.0015, 0.01)	0.0037	(0.0078, 10^{-5})
(131 × 10)		(1, 0.5)	(0.001, 4)	(0.1, 16, 0.1)	(0.1, 8)	(1, 32, 0.1)
	0.05	(0.0458, 0.1)	0.0023	(0.0042, 0.01)	0.0041	(0.0173, 10^{-5})
		(1, 0.5)	(0.1, 8)	(1, 8, 0.1)	(10^{-5}, 8)	(1, 32, 0.1)
	0.1	(0.0305, 0.1)	0.0017	(0.0039, 0.01)	0.0036	(0.0138, 0.01)
		(10, 0.25)	(0.1, 8)	(1, 8, 0.1)	(10^{-5}, 8)	(1, 8, 0.1)
haberman-survival	0	(0.5338, 0.1)	0.0041	(0.0205, 1)	0.0092	(0.0425, 10)
(306 × 3)		(10, 0.125)	(10^{-5}, 32)	(0.01, 16, 0.1)	(10^{-5}, 0.03125)	(1, 16, 0.5)
	0.05	(0.3771, 0.1)	0.0054	(0.014, 1)	0.009	(0.044, 10)
		(100, 0.125)	(10^{-5}, 32)	(0.1, 16, 0.5)	(10^{-5}, 0.03125)	(1, 16, 0.5)
	0.1	(0.338, 0.1)	0.0042	(0.0105, 1)	0.011	(0.036, 100)
		(10, 0.125)	(10^{-5}, 32)	(0.1, 16, 0.1)	(10^{-5}, 0.03125)	(1, 16, 0.5)
heart-hungarian	0	(0.3926, 0.1)	0.005	(0.0321, 100)	0.016	(0.032, 10^{-5})
(294 × 12)		(1, 0.125)	(10^{-5}, 2)	(0.01, 32, 0.3)	(100, 16)	(0.1, 8, 0.1)
	0.05	(0.2904, 0.1)	0.0049	(0.0154, 1)	0.0149	(0.0476, 10^{-5})
		(10, 0.125)	(0.001, 2)	(10^{-5}, 32, 0.1)	(10, 8)	(10, 16, 0.1)
	0.1	(0.3048, 0.1)	0.0046	(0.03, 10)	0.0151	(0.0412, 10^{-5})
		(1, 0.25)	(10^{-4}, 2)	(10^{-5}, 32, 0.1)	(10, 8)	(10, 16, 0.1)
heart-switzerland	0	(0.0615, 0.1)	0.0016	(0.0095, 100)	0.0027	(0.0056, 10^{-5})
(123 × 12)		(10, 2)	(0.1, 32)	(10, 8, 0.3)	(100, 32)	(0.1, 32, 0.1)
	0.05	(0.0431, 0.1)	0.0014	(0.0107, 100)	0.0012	(0.0047, 1)
		(1000, 2)	(0.1, 32)	(10, 8, 0.5)	(10^{-5}, 0.03125)	(0.1, 32, 0.3)
	0.1	(0.058, 0.1)	0.0015	(0.002, 0.01)	0.001	(0.0108, 10^{-5})
		(1, 1)	(0.1, 32)	(0.01, 16, 0.5)	(10^{-5}, 0.03125)	(0.1, 32, 0.1)
hepatitis	0	(0.0603, 0.1)	0.0034	(0.0017, 0.01)	0.0068	(0.0288, 0.1)
(155 × 19)		(1, 0.25)	(1, 32)	(10^{-4}, 2, 0.1)	(10, 16)	(0.1, 16, 0.6)
	0.05	(0.0522, 0.1)	0.0035	(0.0041, 0.1)	0.0067	(0.0095, 0.1)
		(1, 0.25)	(1, 32)	(1, 32, 0.1)	(10, 16)	(0.1, 16, 0.5)
	0.1	(0.0634, 0.1)	0.0035	(0.0019, 0.01)	0.0048	(0.0306, 1)
		(1, 0.25)	(1, 32)	(10^{-4}, 2, 0.1)	(0.01, 16)	(1, 32, 0.5)
ionosphere	0	(0.6903, 0.1)	0.0056	(0.0075, 0.1)	0.0099	(0.0158, 10^{-4})
(351 × 33)		(10, 1)	(10^{-5}, 2)	(10^{-5}, 32, 0.1)	(10^{-5}, 2)	(10^{-5}, 2, 0.3)
	0.05	(1.0821, 0.1)	0.008	(0.0069, 10^{-5})	0.0052	(0.0135, 10^{-4})
		(1, 1)	(10^{-5}, 2)	(10^{-4}, 2, 0.1)	(10^{-5}, 2)	(10^{-5}, 2, 0.3)
	0.1	(0.4487, 0.2)	0.007	(0.006, 10^{-5})	0.0036	(0.0097, 10^{-5})
		(1, 1)	(10^{-5}, 2)	(10^{-4}, 2, 0.1)	(10^{-5}, 2)	(0.001, 4, 0.1)
molec-biol-promoter	0	(0.0315, 0.1)	0.0009	(0.0013, 0.01)	0.0008	(0.0044, 10^{-4})
(106 × 57)		(1, 0.125)	(10^{-4}, 4)	(0.001, 32, 0.1)	(10^{-5}, 4)	(0.001, 8, 0.5)

(continued on next page)

Table 3 (*continued*).

Dataset	r	pin SVM (Huang et al., 2013)	TWSVM (Jayadeva et al., 2007)	UTSVM (Qi et al., 2012)	Pin-GTSVM (Tanveer et al., 2019b)	Tpin-UTWSVM
Size ($f \times s$)		(Time (s), τ) (C_1, σ)	Time (s) ($C_1 = C_2, \sigma$)	(Time (s), C_u) ($C_1 = C_2, \sigma, \epsilon$)	Time (s) ($C_1 = C_2, \sigma$)	(Time (s), C_u) ($C_1 = C_2, \sigma, \epsilon$)
	0.05	(0.0326, 0.1)	0.0011	(0.0017, 1)	0.0009	(0.0037, 10^{-4})
		(1, 0.125)	(10^{-4}, 4)	(10^{-5}, 32, 0.1)	(10^{-5}, 4)	(0.001, 8, 0.6)
	0.1	(0.024, 0.1)	0.0015	(0.0012, 0.01)	0.0009	(0.006, 10^{-4})
		(10, 0.25)	(10^{-4}, 4)	(0.001, 32, 0.1)	(10^{-5}, 4)	(10^{-4}, 4, 0.6)
monks-1	0	(1.7698, 0.2)	0.0106	(0.0138, 0.001)	0.0138	(0.0312, 0.001)
(124 × 6)		(10, 1)	(10^{-4}, 2)	(0.001, 2, 0.1)	(10^{-5}, 2)	(10^{-4}, 2, 0.5)
	0.05	(2.1398, 0.1)	0.0139	(0.0133, 0.001)	0.0155	(0.048, 10^{-4})
		(10, 1)	(0.01, 2)	(0.1, 2, 0.1)	(10^{-5}, 2)	(0.001, 2, 0.5)
	0.1	(2.3103, 0.1)	0.0156	(0.0149, 0.01)	0.0146	(0.0422, 0.001)
		(10, 1)	(0.1, 2)	(0.001, 2, 0.3)	(10^{-5}, 2)	(0.001, 2, 0.1)
parkinsons	0	(0.1928, 0.1)	0.0026	(0.0021, 0.001)	0.0019	(0.0088, 10^{-5})
(195 × 22)		(10, 1)	(0.001, 2)	(10^{-5}, 2, 0.1)	(10^{-5}, 2)	(10^{-4}, 2, 0.1)
	0.05	(0.1051, 0.1)	0.0025	(0.0027, 0.001)	0.0022	(0.0063, 0.001)
		(10, 1)	(0.001, 2)	(10^{-5}, 2, 0.1)	(0.001, 2)	(0.01, 2, 0.1)
	0.1	(0.2886, 0.1)	0.0027	(0.0025, 0.001)	0.0022	(0.0159, 10^{-4})
		(1, 1)	(0.001, 2)	(10^{-5}, 2, 0.1)	(0.001, 2)	(0.001, 2, 0.1)
pima	0	(3.7004, 0.1)	0.0718	(0.2115, 10)	0.091	(0.5041, 10)
(768 × 8)		(1, 0.5)	(1, 32)	(10^{-4}, 32, 0.1)	(0.1, 4)	(1, 32, 0.1)
	0.05	(3.8614, 0.1)	0.067	(0.2134, 10)	0.0798	(0.4701, 0.01)
		(10, 0.5)	(1, 32)	(10^{-5}, 32, 0.1)	(0.01, 4)	(1, 32, 0.3)
	0.1	(2.5047, 0.1)	0.0696	(0.1113, 1)	0.0867	(0.1279, 1)
		(10, 0.5)	(1, 32)	(0.01, 32, 0.3)	(0.01, 4)	(0.1, 32, 0.6)
pittsburg-bridges-T-OR-D	0	(0.026, 0.5)	0.0013	(0.0008, 0.001)	0.0016	(0.012, 10)
(102 × 4)		(1, 0.5)	(0.01, 4)	(0.01, 2, 0.1)	(10^{-5}, 4)	(1, 16, 0.3)
	0.05	(0.0361, 0.1)	0.0016	(0.0045, 1)	0.0015	(0.0254, 10)
		(1, 0.25)	(0.01, 2)	(1, 32, 0.1)	(10^{-5}, 4)	(1, 16, 0.3)
	0.1	(0.0275, 0.1)	0.0012	(0.0036, 1)	0.0011	(0.012, 10)
		(1, 0.25)	(0.001, 2)	(1, 32, 0.3)	(10^{-5}, 4)	(1, 16, 0.6)
planning	0	(0.1833, 0.1)	0.0022	(0.0022, 10^{-4})	0.0019	(0.004, 10^{-5})
(182 × 12)		(1, 2)	(10^{-5}, 0.03125)	(10^{-5}, 0.03125, 0.1)	(10^{-5}, 0.03125)	(10^{-5}, 0.03125, 0.1)
	0.05	(0.179, 0.1)	0.0022	(0.0024, 10^{-5})	0.005	(0.0068, 10^{-5})
		(1, 2)	(10^{-5}, 0.25)	(0.01, 16, 0.1)	(10^{-5}, 0.125)	(10^{-5}, 0.25, 0.1)
	0.1	(0.1245, 0.1)	0.0017	(0.0025, 10^{-5})	0.0018	(0.0079, 10^{-5})
		(1, 2)	(10^{-5}, 0.5)	(10^{-4}, 2, 0.1)	(10^{-5}, 0.25)	(10^{-5}, 0.5, 0.1)
spect	0	(0.4291, 0.1)	0.0043	(0.007, 0.1)	0.0133	(0.0819, 0.1)
(80 × 22)		(1, 0.125)	(0.01, 4)	(1, 32, 0.6)	(10, 8)	(1, 8, 0.5)
	0.05	(0.2821, 0.1)	0.0038	(0.0049, 0.001)	0.0045	(0.0618, 0.1)
		(10, 0.25)	(10^{-5}, 32)	(0.01, 4, 0.6)	(10^{-5}, 1)	(1, 16, 0.3)
	0.1	(0.2327, 0.1)	0.0048	(0.0136, 1)	0.0031	(0.0788, 1)
		(1, 0.25)	(0.1, 32)	(0.1, 32, 0.1)	(10^{-5}, 1)	(1, 8, 0.1)
statlog-australian-credit	0	(4.0553, 0.2)	0.1338	(0.0264, 10^{-5})	0.1413	(0.2216, 10^{-5})
(690 × 14)		(1, 8)	(10^{-4}, 0.125)	(0.001, 16, 0.1)	(10^{-5}, 0.125)	(10^{-4}, 0.125, 0.1)
	0.05	(1.4346, 0.1)	0.0206	(0.1634, 0.01)	0.0274	(0.2537, 10^{-4})
		(10^{-5}, 0.03125)	(10^{-5}, 0.03125)	(10, 8, 0.3)	(10^{-5}, 0.03125)	(10^{-4}, 0.125, 0.6)
	0.1	(3.526, 0.1)	0.0261	(0.1577, 1)	0.0218	(0.0437, 10^{-5})
		(0.1, 0.5)	(10^{-5}, 0.03125)	(100, 16, 0.6)	(10^{-5}, 0.03125)	(10^{-5}, 0.03125, 0.1)
average time	0	1.3178	0.0179	0.0253	0.0246	0.028
	0.05	1.1518	0.0119	0.0291	0.0181	0.0328
	0.1	0.9679	0.0123	0.0236	0.0199	0.0177

Table 4
Pairwise win-tie-loss of the proposed Tpin-UTWSVM model with respect to (wrt) baseline models over UCI datasets. With noise level r in the column, comparison of row method with the proposed Tpin-UTWSVM. Here, [a,b,c] indicates that the proposed Tpin-UTWSVM technique wins 'a' number of times, ties 'b' number of times, and loses 'c' number of times wrt the row method.

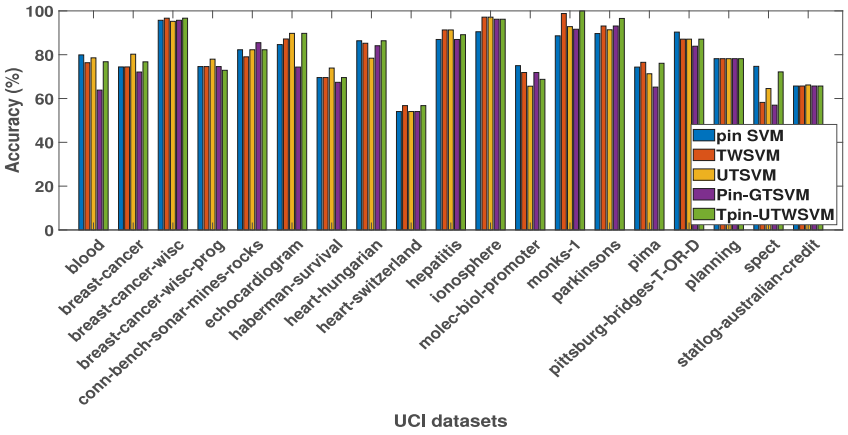
Model	Tpin-UTWSVM			
	$r = 0$	$r = 0.05$	$r = 0.1$	Total
pin SVM (Huang et al., 2013)	[9, 5, 5]	[12, 2, 5]	[12, 2, 5]	[33, 9, 15]
TWSVM (Jayadeva et al., 2007)	[8, 6, 5]	[7, 6, 6]	[4, 12, 3]	[19, 24, 14]
UTSVM (Qi et al., 2012)	[8, 4, 7]	[15, 0, 4]	[14, 2, 3]	[37, 6, 14]
Pin-GTSVM (Tanveer et al., 2019b)	[13, 3, 3]	[9, 7, 3]	[13, 5, 1]	[35, 15, 7]

5.6. Confusion matrix of proposed tpin-UTWSVM on UCI datasets

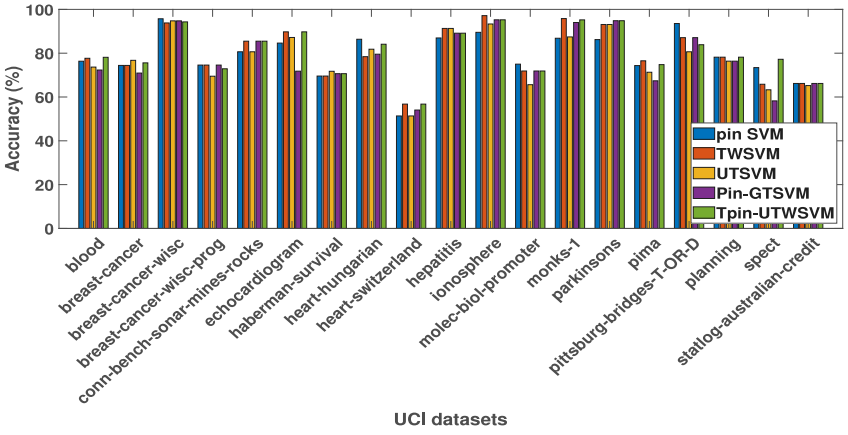
The confusion matrices of the proposed Tpin-UTWSVM with the optimal hyperparameters over 19 UCI datasets having $r = 0$ are shown here from Tables 5 to 23. It is defined as

confusion matrix =

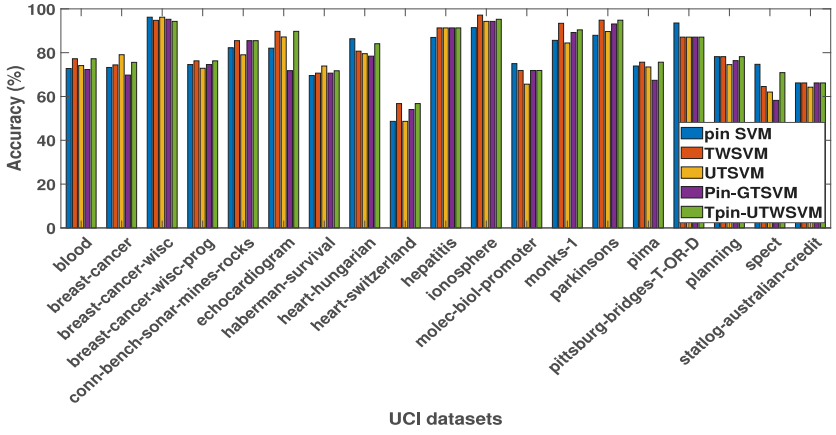
		Actual value	
		positive label	negative label
Predicted value	positive label	tp	fp
	negative label	fn	tn



(a) Bar graph to compare the accuracies of models on UCI datasets with $r = 0$.



(b) Bar graph to compare the accuracies of models on UCI datasets with $r = 0.05$.



(c) Bar graph to compare the accuracies of models on UCI datasets with $r = 0.1$.

Fig. 3. Bar graph representation to compare the performance of models over UCI datasets with different noise levels.

where tp, tn and fp, fn signify the correctly classified positive, negative samples and misclassified positive, negative samples, respectively. The Tables 5 to 23 can be described as: left upper entry denotes tp, right upper entry denotes fp, left lower entry denotes fn and right lower entry denotes tn of the respective datasets.

Table 5
Blood.

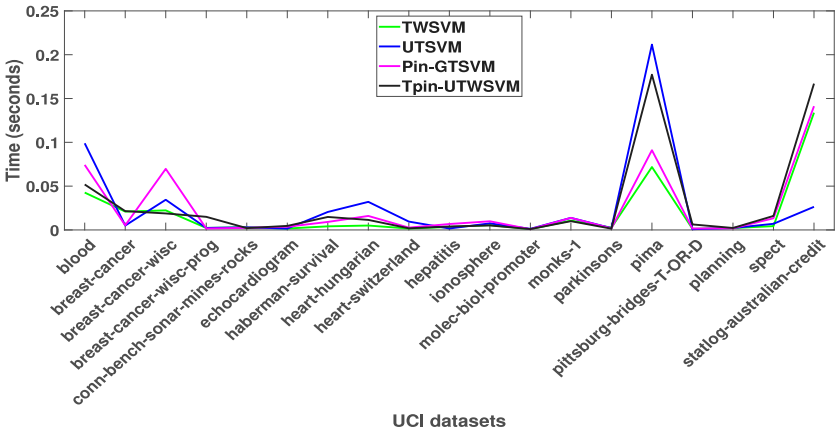
18	9
43	154

6. Applications

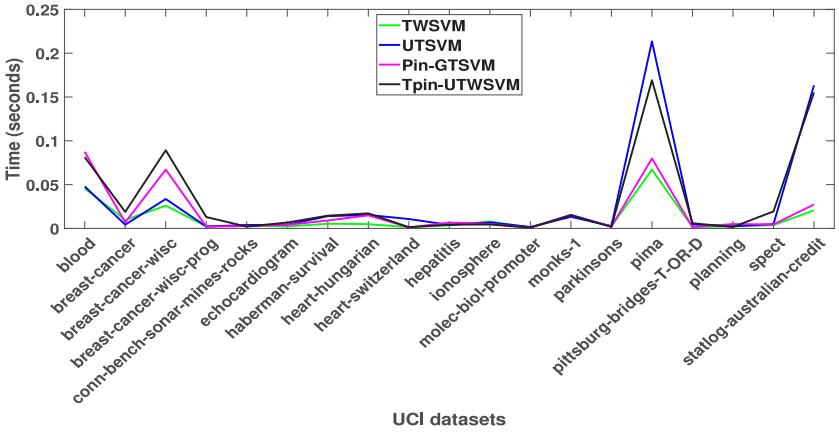
In this section, we will demonstrate the applications of the proposed Tpin-UTWSVM model on EEG signal classification and Alzheimer disease detection.

6.1. EEG dataset and preprocessing

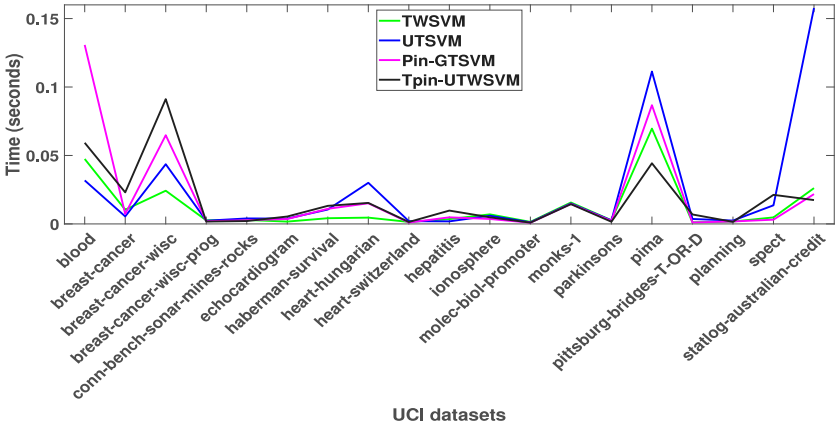
The proposed Tpin-UTWSVM model is used for the classification of the EEG signals. The EEG dataset to carry out the numerical experiments is taken from the publicly available data (Andrzejak et al.,



(a) Comparison of training time of models on UCI datasets with $r = 0$.



(b) Comparison of training time of models on UCI datasets with $r = 0.05$.



(c) Comparison of training time of models on UCI datasets with $r = 0.1$.

Fig. 4. Bar graph representation to compare the training of models over UCI datasets with different noise levels.

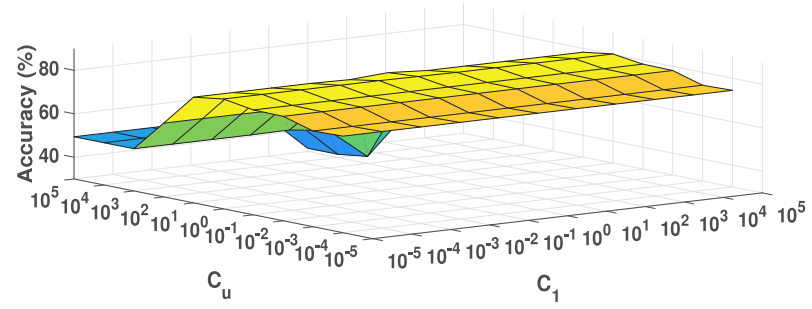
Table 6
Breast-cancer.

10	7
13	56

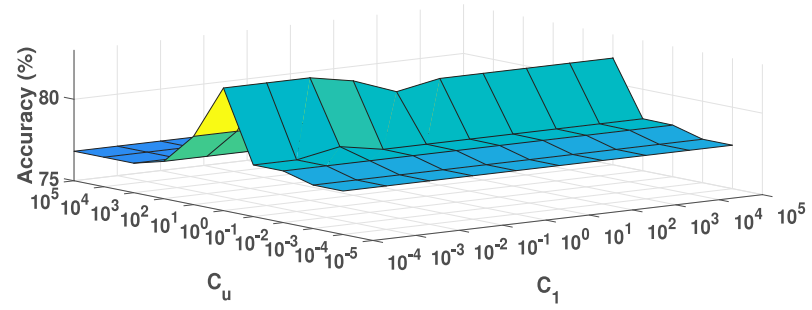
Table 7
Breast-cancer-wisc.

74	4
3	129

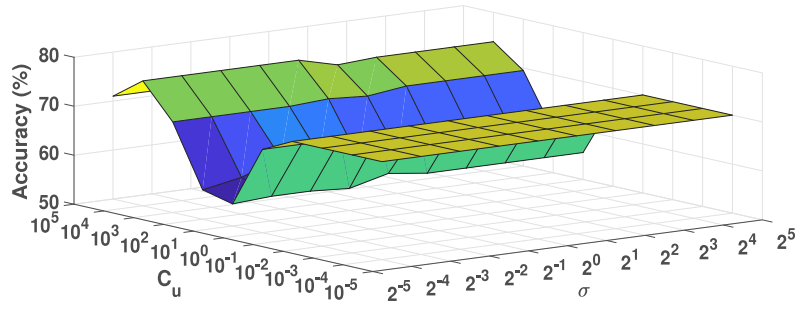
2001). The dataset is having 5 collections namely S, N, F, Z, O, where the signals in each of the collections were sampled at 173.61 Hz for 23.6 s having hundred single channel signals. The five collections represent the five states: S signifies ictal (seizure) state, N and F denote interictal state (seizure-free), Z and O correspond to healthy (control) subjects with open and close eyes, respectively. The processing of all the EEG signals was done by 128 channel amplifier system. The collections S, F and N has intra-cranial recording mode. The



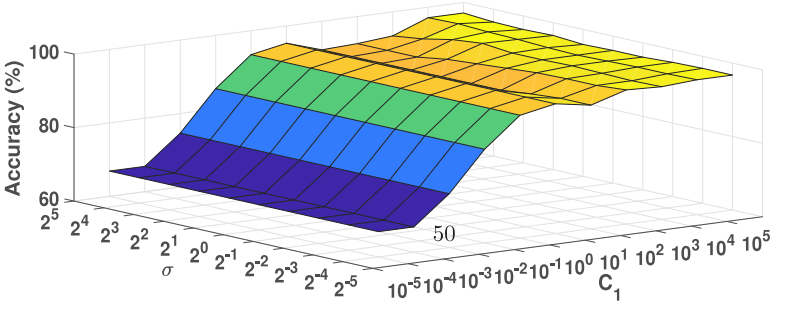
(a) blood



(b) heart-hungarian



(c) breast-cancer



(d) breast-cancer-wisc

Fig. 5. Sensitivity evaluation of the proposed Tpin-UTWSVM classifier on UCI binary datasets using user-specified parameters and Gaussian kernel ($C_1 = C_2$).

Table 8
Breast-cancer-wisc-prog.

4	7
9	39

Table 9
Conn-bench-sonar-mines-rocks.

25	3
8	26

Table 10
Echocardiogram.

7	0
4	28

Table 11
Haberman-survival.

1	2
26	63

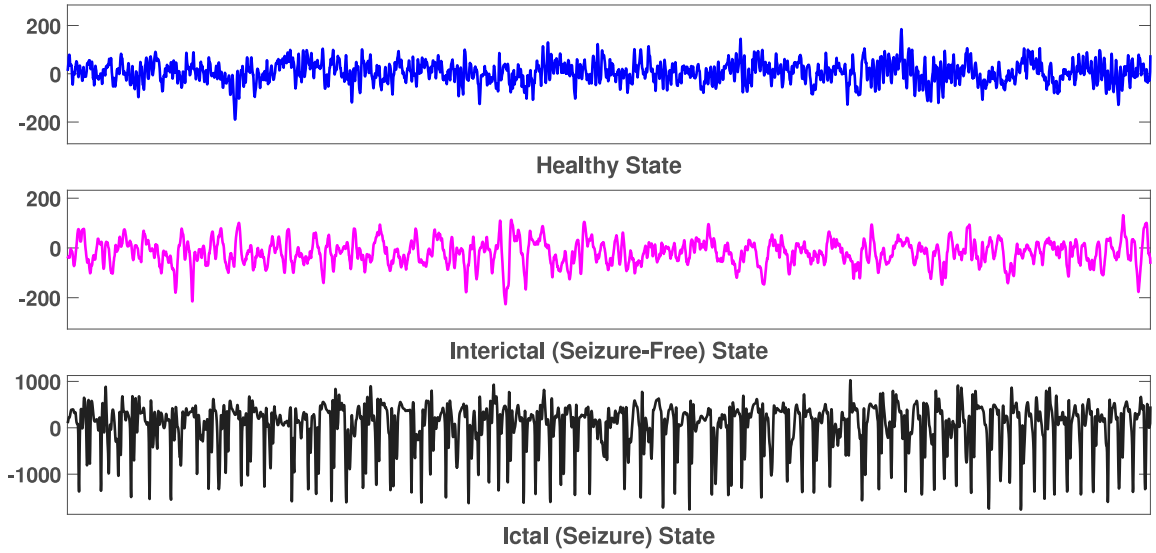


Fig. 6. Samples of EEG signal for different states.

Table 12
Heart-hungarian.

25	6
6	51

Table 13
Heart-switzerland.

4	3
13	17

Table 14
Hepatitis.

37	2
3	4

Table 15
Ionosphere.

68	1
3	33

Table 16
Molec-biol-promoter.

9	6
4	13

Table 17
Monks-1.

87	0
0	80

Table 18
Parkinsons.

45	2
0	11

Table 19
Pima.

45	7
48	130

Table 20
Pittsburg-bridges-T-OR-D.

0	1
3	27

Table 21
Planning.

1	0
12	42

Table 22
Spect.

20	10
12	37

Table 23
Statlog-australian-credit.

136	70
1	0

rankfeatures() function and its various criterion's of MATLAB is used to assess the significance of each features. Different criterions are T-Test (Theodoridis and Koutroumbas, 2009), Entropy (Kullback, 1997), Bhattacharyya (Theodoridis and Koutroumbas, 2009), ROC (Theodoridis and Koutroumbas, 2009) and Wilcoxon (Wilcoxon, 1992). Fig. 6 depicts samples of EEG signals for different states.

6.2. Experimental analysis over EEG data

We compared the proposed Tpin-UTWSVM model with the baseline models, pin SVM (Huang et al., 2013), TWSVM (Jayadeva et al.,

Table 24
Performance of proposed Tpin-UTWSVM model with baseline models on EEG datasets with Gaussian kernel.

Dataset	Features	r	pin SVM (Huang et al., 2013)	TWSVM (Jayadeva et al., 2007)	UTSVM (Qi et al., 2012)	Pin-GTSVM (Tanveer et al., 2019b)	Tpin-UTWSVM
			(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)
F_vs_S	Bhattacharyya	0	(80 ± 11.761) (0.5862, 1)	(98.33 ± 4.8211) (0.9655, 1)	(98.33 ± 4.8211) (0.9655, 1)	(95 ± 3.0117) (0.9655, 0.9333)	(96.67 ± 2.2588) (0.9655, 0.9655)
		0.05	(80 ± 11.6885) (0.5862, 1)	(95 ± 4.9943) (1, 0.9063)	(98.33 ± 4.8211) (0.9655, 1)	(96.67 ± 3.4503) (0.9655, 0.9655)	(96.67 ± 2.2588) (0.9655, 0.9655)
		0.1	(78.33 ± 12.7997) (0.5517, 1)	(95 ± 4.9943) (1, 0.9063)	(95 ± 3.0117) (1, 0.9063)	(96.67 ± 3.4503) (0.9655, 0.9655)	(98.33 ± 3.0117) (1, 0.9667)
F_vs_S	Entropy	0	(80 ± 11.761) (0.5862, 1)	(98.33 ± 4.8211) (0.9655, 1)	(98.33 ± 4.8211) (0.9655, 1)	(95 ± 3.0117) (0.9655, 0.9333)	(96.67 ± 2.2588) (0.9655, 0.9655)
		0.05	(80 ± 11.6885) (0.5862, 1)	(95 ± 4.9943) (1, 0.9063)	(98.33 ± 4.8211) (0.9655, 1)	(96.67 ± 3.4503) (0.9655, 0.9655)	(98.33 ± 2.2588) (0.9655, 1)
		0.1	(78.33 ± 12.7997) (0.5517, 1)	(95 ± 4.9943) (1, 0.9063)	(95 ± 3.0117) (1, 0.9063)	(96.67 ± 3.4503) (0.9655, 0.9655)	(98.33 ± 3.0117) (1, 0.9667)
F_vs_S	ROC	0	(78.33 ± 11.8809) (0.5517, 1)	(90 ± 5.8805) (0.9655, 0.8485)	(91.67 ± 7.1031) (0.8276, 1)	(91.67 ± 5.8805) (1, 0.8529)	(88.33 ± 5.8805) (0.9655, 0.8235)
		0.05	(75 ± 12.9538) (0.4828, 1)	(91.67 ± 4.9943) (0.9655, 0.875)	(88.33 ± 7.1031) (0.7586, 1)	(93.33 ± 6.5638) (1, 0.8788)	(91.67 ± 4.9943) (1, 0.8529)
		0.1	(76.67 ± 19.4744) (0.5172, 1)	(91.67 ± 4.8211) (1, 0.8529)	(86.67 ± 5.6344) (0.931, 0.8182)	(91.67 ± 7.3771) (0.8966, 0.9286)	(91.67 ± 4.8211) (1, 0.8529)
F_vs_S	T-Test	0	(80 ± 9.9887) (0.5862, 1)	(91.67 ± 4.9943) (1, 0.8529)	(91.67 ± 5.8805) (1, 0.8529)	(93.33 ± 7.3771) (0.931, 0.931)	(91.67 ± 4.9943) (1, 0.8529)
		0.05	(80 ± 11.3938) (0.5862, 1)	(90 ± 4.9943) (0.931, 0.871)	(90 ± 4.9943) (1, 0.8286)	(91.67 ± 5.6344) (1, 0.8529)	(93.33 ± 4.8211) (0.931, 0.931)
		0.1	(80 ± 20.7703) (0.5862, 1)	(91.67 ± 5.6344) (0.931, 0.9)	(90 ± 3.7646) (1, 0.8286)	(91.67 ± 5.6344) (1, 0.8529)	(93.33 ± 5.8805) (0.931, 0.931)
F_vs_S	Wilcoxon	0	(76.67 ± 11.8809) (0.5172, 1)	(90 ± 7.3771) (0.9655, 0.8485)	(90 ± 4.0546) (0.9655, 0.8485)	(88.33 ± 6.5638) (1, 0.8056)	(91.67 ± 5.8805) (1, 0.8529)
		0.05	(76.67 ± 11.8809) (0.5172, 1)	(91.67 ± 5.6344) (0.931, 0.9)	(88.33 ± 4.8211) (0.9655, 0.8235)	(91.67 ± 7.5668) (0.8966, 0.9286)	(91.67 ± 5.8805) (1, 0.8529)
		0.1	(75 ± 16.2185) (0.4828, 1)	(90 ± 5.6344) (1, 0.8286)	(91.67 ± 4.5175) (1, 0.8529)	(90 ± 5.8805) (1, 0.8286)	(93.33 ± 5.8805) (1, 0.8788)
N_vs_S	ROC	0	(65 ± 10.1296) (0.2759, 1)	(95 ± 2.2588) (0.9655, 0.9333)	(96.67 ± 3.0117) (0.931, 1)	(96.67 ± 0) (0.931, 1)	(98.33 ± 0) (0.9655, 1)
		0.05	(66.67 ± 13.9036) (0.3103, 1)	(96.67 ± 2.2588) (0.9655, 0.9655)	(96.67 ± 3.0117) (0.931, 1)	(98.33 ± 0) (0.9655, 1)	(95 ± 0) (0.9655, 0.9333)
		0.1	(61.67 ± 15.5036) (0.2069, 1)	(96.67 ± 0) (0.9655, 0.9655)	(95 ± 3.6886) (0.8966, 1)	(98.33 ± 0) (0.9655, 1)	(98.33 ± 0) (0.9655, 1)
N_vs_S	T-Test	0	(63.33 ± 10.2132) (0.2414, 1)	(96.67 ± 0) (0.9655, 0.9655)	(96.67 ± 3.4503) (0.931, 1)	(98.33 ± 0) (0.9655, 1)	(95 ± 0) (0.9655, 0.9333)
		0.05	(63.33 ± 12.0468) (0.2414, 1)	(96.67 ± 0) (0.9655, 0.9655)	(93.33 ± 3.0117) (1, 0.8788)	(98.33 ± 0) (0.9655, 1)	(98.33 ± 0) (0.9655, 1)
		0.1	(63.33 ± 17.2352) (0.2414, 1)	(98.33 ± 0) (0.9655, 1)	(93.33 ± 0) (0.9655, 0.9032)	(98.33 ± 0) (0.9655, 1)	(98.33 ± 0) (0.9655, 1)
N_vs_S	Wilcoxon	0	(66.67 ± 10.1296) (0.3103, 1)	(95 ± 3.0117) (0.9655, 0.9333)	(96.67 ± 3.0117) (0.931, 1)	(98.33 ± 0) (0.9655, 1)	(98.33 ± 0) (0.9655, 1)
		0.05	(63.33 ± 11.0657) (0.2414, 1)	(96.67 ± 0) (0.9655, 0.9655)	(96.67 ± 3.0117) (0.931, 1)	(95 ± 0) (0.8966, 1)	(96.67 ± 0) (0.9655, 0.9655)
		0.1	(61.67 ± 15.8114) (0.2069, 1)	(96.67 ± 0) (0.9655, 0.9655)	(95 ± 2.2588) (0.9655, 0.9333)	(95 ± 2.2588) (0.8966, 1)	(96.67 ± 0) (0.931, 1)
O_vs_F	Bhattacharyya	0	(60 ± 20.3289) (0.6897, 0.5714)	(61.67 ± 11.193) (0.8966, 0.5652)	(68.33 ± 11.2687) (0.6207, 0.6923)	(78.33 ± 9.6715) (0.8621, 0.7353)	(76.67 ± 11.3938) (0.931, 0.6923)
		0.05	(60 ± 20.1469) (0.6897, 0.5714)	(66.67 ± 10.6745) (0.6207, 0.6667)	(68.33 ± 6.0234) (0.6207, 0.6923)	(75 ± 11.2938) (0.8276, 0.7059)	(63.33 ± 10.6745) (1, 0.5686)
		0.1	(66.67 ± 19.7633) (0.8276, 0.6154)	(75 ± 8.5516) (0.7586, 0.7333)	(61.67 ± 13.5735) (0.5172, 0.625)	(76.67 ± 11.761) (0.8276, 0.7273)	(60 ± 12.6213) (1, 0.5472)
O_vs_F	Entropy	0	(60 ± 20.3289) (0.6897, 0.5714)	(61.67 ± 11.193) (0.8966, 0.5652)	(68.33 ± 11.2687) (0.6207, 0.6923)	(78.33 ± 9.6715) (0.8621, 0.7353)	(76.67 ± 11.3938) (0.931, 0.6923)
		0.05	(60 ± 20.3289) (0.6897, 0.5714)	(61.67 ± 11.193) (0.8966, 0.5652)	(68.33 ± 8.7805) (0.6207, 0.6923)	(78.33 ± 11.2938) (0.8621, 0.7353)	(63.33 ± 10.6745) (1, 0.5686)
		0.1	(66.67 ± 19.7633) (0.8276, 0.6154)	(71.67 ± 6.5638) (0.7586, 0.6875)	(63.33 ± 11.0657) (0.6897, 0.6061)	(78.33 ± 11.2938) (0.8621, 0.7353)	(78.33 ± 11.6885) (0.8621, 0.7353)
O_vs_F	ROC	0	(75 ± 15.8651) (0.8966, 0.6842)	(73.33 ± 12.5988) (0.9655, 0.6512)	(75 ± 10.5409) (0.8966, 0.6842)	(83.33 ± 12.1638) (0.8276, 0.8276)	(73.33 ± 7.5668) (0.6897, 0.7407)
		0.05	(75 ± 15.8651) (0.8966, 0.6842)	(81.67 ± 8.5516) (0.7931, 0.8214)	(76.67 ± 9.788) (0.7931, 0.7419)	(83.33 ± 11.2938) (0.8276, 0.8276)	(86.67 ± 11.193) (0.931, 0.8182)
		0.1	(73.33 ± 15.8651) (0.8966, 0.6667)	(83.33 ± 7.5292) (0.7931, 0.8519)	(75 ± 9.788) (0.7931, 0.7188)	(83.33 ± 10.2131) (0.8276, 0.8276)	(86.67 ± 11.193) (0.931, 0.8182)

(*continued on next page*)

Table 24 (*continued*).

Dataset	Features	r	pin SVM (Huang et al., 2013)	TWSVM (Jayadeva et al., 2007)	UTSVM (Qi et al., 2012)	Pin-GTSVM (Tanveer et al., 2019b)	Tpin-UTWSVM
			(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)
O_vs_F	T-Test	0	(73.33 ± 7.8608) (0.6552, 0.76)	(73.33 ± 7.8608) (0.6552, 0.76)	(78.33 ± 13.8013) (0.8966, 0.7222)	(66.67 ± 12.5085) (0.9655, 0.5957)	(75 ± 11.2939) (0.931, 0.675)
		0.05	(73.33 ± 7.8608) (0.6552, 0.76)	(73.33 ± 6.2543) (0.6552, 0.76)	(78.33 ± 12.6213) (0.8966, 0.7222)	(66.67 ± 11.8809) (0.9655, 0.5957)	(76.67 ± 11.2939) (0.9655, 0.6829)
		0.1	(73.33 ± 7.3771) (0.6552, 0.76)	(66.67 ± 10.5409) (0.9655, 0.5957)	(80 ± 12.5085) (0.931, 0.7297)	(83.33 ± 9.1906) (0.7931, 0.8519)	(85 ± 9.0664) (0.931, 0.7941)
O_vs_N	Bhattacharyya	0	(76.67 ± 22.08) (0.5172, 1)	(80 ± 13.0627) (0.8276, 0.7742)	(80 ± 13.8013) (0.7586, 0.8148)	(80 ± 11.6885) (0.8276, 0.7742)	(81.67 ± 12.2336) (0.8621, 0.7813)
		0.05	(76.67 ± 22.08) (0.5172, 1)	(80 ± 13.3843) (0.8276, 0.7742)	(78.33 ± 13.8013) (0.7586, 0.7857)	(80 ± 12.1638) (0.8276, 0.7742)	(81.67 ± 12.2336) (0.8621, 0.7813)
		0.1	(76.67 ± 21.5604) (0.5172, 1)	(80 ± 13.4897) (0.8276, 0.7742)	(76.67 ± 13.8013) (0.7586, 0.7586)	(81.67 ± 12.0468) (0.8621, 0.7813)	(85 ± 11.5666) (0.931, 0.7941)
O_vs_N	Entropy	0	(76.67 ± 21.9384) (0.5172, 1)	(80 ± 13.1277) (0.8276, 0.7742)	(78.33 ± 13.8013) (0.7586, 0.7857)	(80 ± 11.6885) (0.8276, 0.7742)	(81.67 ± 12.2336) (0.8621, 0.7813)
		0.05	(76.67 ± 21.5604) (0.5172, 1)	(80 ± 14.7542) (0.8276, 0.7742)	(78.33 ± 13.8013) (0.7586, 0.7857)	(80 ± 11.6885) (0.8276, 0.7742)	(81.67 ± 10.6745) (0.8621, 0.7813)
		0.1	(76.67 ± 21.9384) (0.5172, 1)	(80 ± 12.1405) (0.8276, 0.7742)	(78.33 ± 12.1639) (0.7586, 0.7857)	(80 ± 12.0468) (0.8276, 0.7742)	(80 ± 11.5666) (0.8621, 0.7576)
O_vs_N	ROC	0	(63.33 ± 19.6483) (0.8276, 0.5854)	(81.67 ± 10.6479) (0.8966, 0.7647)	(61.67 ± 16.166) (0.5517, 0.6154)	(80 ± 12.0468) (0.931, 0.7297)	(80 ± 12.1405) (0.931, 0.7297)
		0.05	(63.33 ± 19.6483) (0.8276, 0.5854)	(78.33 ± 11.5666) (0.8966, 0.7222)	(61.67 ± 14.7542) (0.5862, 0.6071)	(76.67 ± 10.6745) (0.8621, 0.7143)	(80 ± 10.8849) (0.931, 0.7297)
		0.1	(66.67 ± 19.1071) (0.3448, 0.9091)	(78.33 ± 11.3938) (0.8966, 0.7222)	(63.33 ± 14.7542) (0.6207, 0.6207)	(78.33 ± 10.6479) (0.8966, 0.7222)	(76.67 ± 10.8849) (0.8621, 0.7143)
O_vs_N	T-Test	0	(71.67 ± 19.7059) (0.4828, 0.875)	(76.67 ± 12.5085) (0.8276, 0.7273)	(75 ± 15.8829) (0.7586, 0.7333)	(76.67 ± 8.9087) (0.8276, 0.7273)	(75 ± 7.8608) (0.8621, 0.6944)
		0.05	(66.67 ± 17.561) (0.3448, 0.9091)	(76.67 ± 11.0657) (0.8276, 0.7273)	(75 ± 16.2185) (0.7586, 0.7333)	(76.67 ± 8.9087) (0.8276, 0.7273)	(73.33 ± 7.8608) (0.8276, 0.6857)
		0.1	(66.67 ± 21.0953) (0.5172, 0.7143)	(76.67 ± 10.1015) (0.8276, 0.7273)	(75 ± 18.2108) (0.7586, 0.7333)	(76.67 ± 9.1906) (0.8276, 0.7273)	(80 ± 9.788) (0.8966, 0.7429)
O_vs_N	Wilcoxon	0	(66.67 ± 20.3289) (0.4483, 0.7647)	(80 ± 10.2132) (0.7931, 0.7931)	(71.67 ± 7.5292) (0.8276, 0.6667)	(80 ± 6.9007) (0.931, 0.7297)	(83.33 ± 7.1031) (0.8276, 0.8276)
		0.05	(68.33 ± 20.3289) (0.4828, 0.7778)	(83.33 ± 7.6783) (0.7931, 0.8519)	(76.67 ± 7.8608) (0.8276, 0.7273)	(80 ± 9.0351) (0.8276, 0.8)	(83.33 ± 8.1092) (0.7931, 0.8519)
		0.1	(66.67 ± 20.1469) (0.4828, 0.7368)	(81.67 ± 9.0351) (0.8276, 0.8)	(76.67 ± 6.7344) (0.8276, 0.7273)	(81.67 ± 10.8849) (0.6897, 0.7692)	(83.33 ± 5.8805) (0.7931, 0.8519)
O_vs_S	Bhattacharyya	0	(86.67 ± 15.1336) (0.7241, 1)	(95 ± 3.0117) (1, 0.9063)	(100 ± 2.2588) (1, 1)	(96.67 ± 0) (1, 0.9355)	(95 ± 3.0117) (1, 0.9063)
		0.05	(88.33 ± 16.2185) (0.7586, 1)	(95 ± 3.0117) (1, 0.9063)	(100 ± 3.0117) (1, 1)	(96.67 ± 0) (1, 0.9355)	(96.67 ± 2.2588) (1, 0.9355)
		0.1	(51.67 ± 19.343) (1, 0.5)	(95 ± 3.0117) (1, 0.9063)	(95 ± 2.2588) (1, 0.9063)	(96.67 ± 2.2588) (1, 0.9355)	(96.67 ± 2.2588) (1, 0.9355)
O_vs_S	Entropy	0	(86.67 ± 15.1336) (0.7241, 1)	(95 ± 3.0117) (1, 0.9063)	(100 ± 2.2588) (1, 1)	(96.67 ± 2.2588) (1, 0.9355)	(95 ± 3.0117) (1, 0.9063)
		0.05	(86.67 ± 16.2185) (0.7241, 1)	(95 ± 3.0117) (1, 0.9063)	(100 ± 2.2588) (1, 1)	(96.67 ± 2.2588) (1, 0.9355)	(95 ± 2.2588) (1, 0.9063)
		0.1	(50 ± 14.85) (1, 0.4915)	(95 ± 3.0117) (1, 0.9063)	(95 ± 3.0117) (1, 0.9063)	(96.67 ± 2.2588) (1, 0.9355)	(96.67 ± 2.2588) (1, 0.9355)
O_vs_S	T-Test	0	(51.67 ± 19.343) (1, 0.5)	(93.33 ± 2.2588) (0.9655, 0.9032)	(93.33 ± 4.9943) (0.9655, 0.9032)	(93.33 ± 3.0117) (0.931, 0.931)	(95 ± 3.0117) (0.9655, 0.9333)
		0.05	(51.67 ± 19.343) (1, 0.5)	(93.33 ± 3.0117) (0.9655, 0.9032)	(93.33 ± 4.8211) (0.9655, 0.9032)	(95 ± 3.0117) (0.931, 0.9643)	(93.33 ± 3.0117) (0.931, 0.931)
		0.1	(51.67 ± 14.0658) (1, 0.5)	(93.33 ± 3.4503) (0.9655, 0.9032)	(95 ± 3.4503) (0.9655, 0.9333)	(95 ± 3.0117) (0.931, 0.9643)	(93.33 ± 3.0117) (0.931, 0.931)
O_vs_S	Wilcoxon	0	(51.67 ± 14.6772) (1, 0.5)	(96.67 ± 2.2588) (1, 0.9355)	(93.33 ± 3.0117) (1, 0.8788)	(95 ± 3.4503) (0.931, 0.9643)	(96.67 ± 3.0117) (0.9655, 0.9655)
		0.05	(51.67 ± 14.6772) (1, 0.5)	(91.67 ± 2.2588) (0.9655, 0.875)	(93.33 ± 3.0117) (1, 0.8788)	(93.33 ± 4.9943) (1, 0.8788)	(96.67 ± 3.0117) (0.9655, 0.9655)
		0.1	(51.67 ± 14.6772) (1, 0.5)	(91.67 ± 2.2588) (0.9655, 0.875)	(93.33 ± 2.2588) (1, 0.8788)	(93.33 ± 4.9943) (1, 0.8788)	(96.67 ± 3.0117) (0.9655, 0.9655)
Z_vs_F	Bhattacharyya	0	(55 ± 12.2567) (0.7586, 0.5238)	(68.33 ± 15.2269) (0.7241, 0.6563)	(78.33 ± 13.4897) (0.7241, 0.8077)	(73.33 ± 12.2336) (0.7931, 0.697)	(70 ± 12.1639) (0.7931, 0.6571)
		0.05	(55 ± 12.2336) (0.7931, 0.5227)	(65 ± 13.2352) (0.6897, 0.625)	(78.33 ± 12.2336) (0.7241, 0.8077)	(63.33 ± 19.1071) (0.6207, 0.6207)	(66.67 ± 13.041) (0.6207, 0.6667)
		0.1	(53.33 ± 10.6745) (0.7931, 0.5111)	(65 ± 9.404) (0.6552, 0.6333)	(70 ± 13.5526) (0.6207, 0.72)	(60 ± 13.3843) (0.4828, 0.6087)	(66.67 ± 11.6885) (0.6207, 0.6667)
Z_vs_F	Entropy	0	(55 ± 12.2567) (0.7586, 0.5238)	(68.33 ± 15.2269) (0.7241, 0.6563)	(78.33 ± 13.4897) (0.7241, 0.8077)	(73.33 ± 12.2336) (0.7931, 0.697)	(70 ± 12.1639) (0.7931, 0.6571)
		0.05	(58.33 ± 12.2336) (0.7931, 0.5476)	(63.33 ± 13.2352) (0.6552, 0.6129)	(76.67 ± 13.0627) (0.6897, 0.8)	(61.67 ± 19.3576) (0.5862, 0.6071)	(75 ± 11.8809) (0.7931, 0.7188)
		0.1	(55 ± 10.6745) (0.7931, 0.5227)	(60 ± 13.4897) (0.5862, 0.5862)	(73.33 ± 11.761) (0.6897, 0.7407)	(60 ± 11.761) (0.4483, 0.619)	(58.33 ± 12.7108) (0.4483, 0.5909)

(*continued on next page*)

Table 24 (*continued*).

Dataset	Features	r	pin SVM (Huang et al., 2013)	TWSVM (Jayadeva et al., 2007)	UTSVM (Qi et al., 2012)	Pin-GTSVM (Tanveer et al., 2019b)	Tpin-UTWSVM
			(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)	(Accuracy ± sd) (Recall, Precision)
Z_vs_F	T-Test	0	(63.33 ± 13.1277) (0.7241, 0.6)	(76.67 ± 8.2822) (0.7241, 0.7778)	(75 ± 8.2479) (0.6897, 0.7692)	(75 ± 6.9007) (0.6897, 0.7692)	(80 ± 7.8608) (0.8276, 0.7742)
		0.05	(63.33 ± 14.3846) (0.7241, 0.6)	(80 ± 9.6421) (0.7931, 0.7931)	(71.67 ± 8.1092) (0.6552, 0.7308)	(75 ± 6.7763) (0.6897, 0.7692)	(76.67 ± 6.7763) (0.8276, 0.7273)
		0.1	(65 ± 13.9849) (0.7586, 0.6111)	(81.67 ± 9.6421) (0.8276, 0.8)	(71.67 ± 10.1015) (0.6552, 0.7308)	(75 ± 9.788) (0.6897, 0.7692)	(78.33 ± 7.6783) (0.7241, 0.8077)
Z_vs_F	Wilcoxon	0	(80 ± 12.9538) (0.8966, 0.7429)	(81.67 ± 6.9007) (0.8276, 0.8)	(76.67 ± 7.5292) (0.6552, 0.8261)	(86.67 ± 15.9542) (0.8276, 0.8889)	(76.67 ± 9.9887) (0.7586, 0.7586)
		0.05	(80 ± 12.9538) (0.8966, 0.7429)	(80 ± 6.9007) (0.7931, 0.7931)	(76.67 ± 5.8321) (0.6552, 0.8261)	(86.67 ± 16.078) (0.8276, 0.8889)	(81.67 ± 5.6344) (0.8276, 0.8)
		0.1	(80 ± 12.9538) (0.8966, 0.7429)	(80 ± 5.8805) (0.7931, 0.7931)	(76.67 ± 7.1031) (0.7241, 0.7778)	(78.33 ± 5.8805) (0.7931, 0.7667)	(83.33 ± 6.2543) (0.8276, 0.8276)
Z_vs_N	Bhattacharyya	0	(75 ± 8.3842) (1, 0.6591)	(86.67 ± 14.3055) (0.7931, 0.92)	(75 ± 13.9647) (0.7241, 0.75)	(80 ± 20.2031) (0.6552, 0.9048)	(85 ± 15.1336) (0.7931, 0.8846)
		0.05	(75 ± 8.4179) (1, 0.6591)	(86.67 ± 14.6965) (0.7931, 0.92)	(75 ± 13.1277) (0.7241, 0.75)	(80 ± 21.4153) (0.6207, 0.9474)	(85 ± 15.0773) (0.7931, 0.8846)
		0.1	(75 ± 8.4179) (1, 0.6591)	(91.67 ± 14.6965) (0.8966, 0.9286)	(73.33 ± 13.9647) (0.8276, 0.6857)	(83.33 ± 14.3055) (0.7241, 0.913)	(81.67 ± 16.7684) (0.7241, 0.875)
Z_vs_N	Entropy	0	(75 ± 9.0351) (1, 0.6591)	(90 ± 15.1336) (0.8966, 0.8966)	(75 ± 14.3649) (0.7586, 0.7333)	(80 ± 20.3289) (0.6552, 0.9048)	(83.33 ± 14.6965) (0.7586, 0.88)
		0.05	(75 ± 8.4179) (1, 0.6591)	(90 ± 14.3846) (0.8966, 0.8966)	(75 ± 14.3846) (0.7586, 0.7333)	(85 ± 14.5997) (0.7931, 0.8846)	(85 ± 14.6965) (0.7931, 0.8846)
		0.1	(75 ± 9.6421) (1, 0.6591)	(88.33 ± 15.5036) (0.8621, 0.8929)	(75 ± 12.6885) (0.8276, 0.7059)	(85 ± 15.5948) (0.7931, 0.8846)	(86.67 ± 14.3649) (0.8276, 0.8889)
Z_vs_N	ROC	0	(68.33 ± 14.7542) (0.931, 0.6136)	(78.33 ± 9.788) (0.7586, 0.7857)	(76.67 ± 8.7805) (0.8276, 0.7273)	(81.67 ± 10.2131) (0.8621, 0.7813)	(78.33 ± 9.0351) (0.7586, 0.7857)
		0.05	(68.33 ± 14.7542) (0.931, 0.6136)	(78.33 ± 9.788) (0.7586, 0.7857)	(76.67 ± 9.0351) (0.8276, 0.7273)	(80 ± 9.0664) (0.8621, 0.7576)	(78.33 ± 9.0351) (0.7586, 0.7857)
		0.1	(68.33 ± 14.7542) (0.931, 0.6136)	(78.33 ± 9.404) (0.7586, 0.7857)	(78.33 ± 7.5668) (0.8276, 0.75)	(78.33 ± 9.0664) (0.8276, 0.75)	(78.33 ± 9.0351) (0.7586, 0.7857)
Z_vs_N	T-Test	0	(70 ± 13.8013) (0.931, 0.6279)	(76.67 ± 14.2857) (0.8276, 0.7273)	(68.33 ± 10.351) (0.7241, 0.6563)	(63.33 ± 10.1295) (0.7241, 0.6)	(78.33 ± 12.5085) (0.7931, 0.7667)
		0.05	(70 ± 13.6568) (0.931, 0.6279)	(80 ± 13.3843) (0.7931, 0.7931)	(68.33 ± 10.351) (0.7241, 0.6563)	(63.33 ± 9.9887) (0.7241, 0.6)	(80 ± 12.3489) (0.8276, 0.7742)
		0.1	(70 ± 14.983) (0.931, 0.6279)	(78.33 ± 12.6213) (0.8276, 0.75)	(68.33 ± 11.2687) (0.7241, 0.6563)	(63.33 ± 9.5535) (0.7241, 0.6)	(83.33 ± 14.0658) (0.8276, 0.8276)
Z_vs_N	Wilcoxon	0	(73.33 ± 12.1639) (1, 0.6444)	(86.67 ± 8.1092) (0.8966, 0.8387)	(75 ± 8.7805) (0.7931, 0.7188)	(81.67 ± 5.6344) (0.8276, 0.8)	(86.67 ± 8.1092) (0.8966, 0.8387)
		0.05	(70 ± 12.1639) (1, 0.617)	(81.67 ± 11.8809) (0.7586, 0.8462)	(76.67 ± 9.9887) (0.7241, 0.7778)	(85 ± 9.9887) (0.8621, 0.8333)	(83.33 ± 9.788) (0.7931, 0.8519)
		0.1	(70 ± 12.1639) (1, 0.617)	(80 ± 12.1405) (0.7241, 0.84)	(78.33 ± 9.404) (0.7241, 0.8077)	(83.33 ± 10.7539) (0.8276, 0.8276)	(81.67 ± 9.788) (0.7586, 0.8462)
Z_vs_S	Bhattacharyya	0	(95 ± 8.9087) (0.8966, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)
		0.05	(95 ± 8.7805) (0.8966, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)
		0.1	(88.33 ± 7.5668) (0.7586, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)
Z_vs_S	Entropy	0	(95 ± 8.9087) (0.8966, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)
		0.05	(93.33 ± 11.1677) (0.8621, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)
		0.1	(93.33 ± 8.1092) (0.8621, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)
Z_vs_S	T-Test	0	(78.33 ± 12.7997) (0.5517, 1)	(96.67 ± 0) (1, 0.9355)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)	(98.33 ± 0) (1, 0.9667)
		0.05	(80 ± 14.5997) (0.5862, 1)	(96.67 ± 0) (1, 0.9355)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)
		0.1	(91.67 ± 25.5218) (0.8276, 1)	(96.67 ± 0) (1, 0.9355)	(95 ± 0) (1, 0.9063)	(100 ± 0) (1, 1)	(100 ± 0) (1, 1)
Average accuracy		0	71.92	85.25	84.29	85.96	**86.31**
		0.05	71.72	85.2	84.04	85.45	**86.21**
		0.1	69.65	85.56	82.73	85.66	**86.82**
Average rank		0	4.7	2.7	2.82	2.41	**2.38**
		0.05	4.79	2.7	2.98	2.41	**2.12**
		0.1	4.65	2.62	3.58	2.29	**1.86**

2007), UTSVM (Qi et al., 2012) and Pin-GTSVM (Tanveer et al., 2019b) on EEG dataset. Each of the dataset considered has 200 samples and 200 features. Table 24 depicts the accuracy, standard deviation (sd), recall and precision of each model with different noise levels (r = 0, 0.05, 0.1) on different EEG datasets. The average accuracies of pin SVM, TWSVM, UTSVM, Pin-GTSVM, and the proposed Tpin-UTWSVM for noise-free data (r = 0) are 71.92%, 85.25%, 84.29%, 85.96% and 86.31%, respectively, which clearly depicts the proposed

Table 25
Training time and optimal hyperparameters of models on EEG datasets with Gaussian kernel.

Dataset	Features	r	pin SVM (Huang et al., 2013)	TWSVM (Jayadeva et al., 2007)	UTSVM (Qi et al., 2012)	Pin-GTSVM (Tanveer et al., 2019b)	Tpin-UTWSVM
Size ($f \times s$) (200×200)			(Time (s), τ) (C_1, σ)	Time (s) ($C_1 = C_2, \sigma$)	(Time (s), C_u) ($C_1 = C_2, \sigma, \epsilon$)	Time (s) ($C_1 = C_2, \sigma$)	(Time (s), C_u) ($C_1 = C_2, \sigma, \epsilon$)
F_vs_S	Bhattacharyya	0	(0.4238, 0.1)	0.0023	(0.0024, 0.00001)	0.0033	(0.0108, 0.00001)
			(1, 1)	(0.00001, 2)	(0.00001, 2, 0.1)	(0.0001, 4)	(0.0001, 4, 0.6)
		0.05	(0.3733, 0.1)	0.0022	(0.0029, 0.00001)	0.0037	(0.0326, 0.00001)
			(1, 1)	(0.0001, 4)	(0.00001, 2, 0.1)	(0.0001, 4)	(0.0001, 4, 0.6)
		0.1	(0.4949, 0.1)	0.0059	(0.0092, 1)	0.0036	(0.0112, 0.00001)
			(1, 1)	(0.0001, 4)	(0.00001, 32, 0.1)	(0.00001, 4)	(0.0001, 4, 0.3)
F_vs_S	Entropy	0	(0.4273, 0.1)	0.0129	(0.0061, 0.00001)	0.0038	(0.0035, 0.00001)
			(1, 1)	(0.00001, 2)	(0.00001, 2, 0.1)	(0.0001, 4)	(0.0001, 4, 0.6)
		0.05	(0.4098, 0.1)	0.0044	(0.0036, 0.00001)	0.0029	(0.0034, 0.00001)
			(1, 1)	(0.0001, 4)	(0.00001, 2, 0.1)	(0.0001, 4)	(0.0001, 4, 0.6)
		0.1	(0.3089, 0.1)	0.008	(0.0098, 10)	0.0029	(0.0035, 0.00001)
			(1, 1)	(0.0001, 4)	(0.00001, 32, 0.1)	(0.00001, 4)	(0.0001, 4, 0.3)
F_vs_S	ROC	0	(0.4179, 0.1)	0.0046	(0.0031, 0.0001)	0.0041	(0.0035, 0.00001)
			(1, 1)	(0.0001, 16)	(0.00001, 2, 0.1)	(0.001, 8)	(0.0001, 16, 0.1)
		0.05	(0.3715, 0.1)	0.0032	(0.0034, 0.0001)	0.0033	(0.0034, 0.00001)
			(1, 1)	(0.0001, 16)	(0.00001, 2, 0.1)	(0.0001, 8)	(0.0001, 16, 0.1)
		0.1	(0.2846, 0.1)	0.0031	(0.0031, 0.00001)	0.0031	(0.0036, 0.00001)
			(1, 1)	(0.0001, 16)	(0.00001, 8, 0.1)	(0.00001, 4)	(0.0001, 16, 0.1)
F_vs_S	T-Test	0	(0.173, 0.1)	0.005	(0.0082, 1)	0.004	(0.0036, 0.00001)
			(1, 1)	(0.00001, 16)	(0.001, 16, 0.1)	(0.00001, 4)	(0.00001, 16, 0.1)
		0.05	(0.5232, 0.1)	0.007	(0.0034, 0.1)	0.0029	(0.0365, 0.00001)
			(1, 1)	(0.0001, 4)	(0.001, 16, 0.1)	(0.00001, 8)	(0.00001, 4, 0.1)
		0.1	(0.309, 0.1)	0.0022	(0.0047, 1)	0.0034	(0.0201, 0.0001)
			(1, 1)	(0.0001, 4)	(0.001, 16, 0.5)	(0.00001, 8)	(0.0001, 4, 0.5)
F_vs_S	Wilcoxon	0	(0.4574, 0.1)	0.0023	(0.0043, 1)	0.0027	(0.0038, 0.0001)
			(1, 1)	(0.0001, 16)	(0.00001, 16, 0.1)	(0.00001, 16)	(0.001, 8, 0.1)
		0.05	(0.8652, 0.1)	0.0044	(0.0035, 1)	0.003	(0.011, 0.00001)
			(1, 1)	(0.00001, 16)	(0.00001, 16, 0.1)	(0.00001, 4)	(0.001, 8, 0.1)
		0.1	(0.6062, 0.1)	0.0061	(0.012, 0.1)	0.0036	(0.0139, 0.00001)
			(1, 1)	(0.001, 8)	(0.00001, 16, 0.1)	(0.00001, 8)	(0.001, 8, 0.1)
N_vs_S	ROC	0	(0.3573, 0.1)	0.0043	(0.003, 0.0001)	0.0022	(0.0036, 0.00001)
			(10, 1)	(0.0001, 4)	(0.00001, 2, 0.1)	(0.00001, 4)	(0.0001, 4, 0.1)
		0.05	(0.3466, 0.1)	0.0041	(0.0029, 0.0001)	0.0017	(0.0128, 0.00001)
			(10, 1)	(0.0001, 4)	(0.00001, 2, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
		0.1	(0.2449, 0.1)	0.0162	(0.0025, 0.0001)	0.0023	(0.0035, 0.00001)
			(1, 1)	(0.00001, 4)	(0.00001, 2, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
N_vs_S	T-Test	0	(0.4479, 0.1)	0.0056	(0.0167, 0.0001)	0.0058	(0.0061, 0.00001)
			(10, 1)	(0.0001, 4)	(0.00001, 2, 0.1)	(0.00001, 4)	(0.00001, 4, 0.3)
		0.05	(0.1901, 0.1)	0.0053	(0.011, 0.1)	0.0019	(0.0099, 0.00001)
			(1, 1)	(0.00001, 4)	(0.0001, 16, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
		0.1	(2.0203, 0.1)	0.006	(0.0045, 0.00001)	0.0108	(0.0108, 0.00001)
			(1, 1)	(0.00001, 4)	(0.0001, 4, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
N_vs_S	Wilcoxon	0	(0.0704, 0.1)	0.0045	(0.0051, 0.0001)	0.0021	(0.0035, 0.00001)
			(10, 1)	(0.0001, 4)	(0.00001, 2, 0.1)	(0.00001, 4)	(0.0001, 4, 0.1)
		0.05	(0.8785, 0.1)	0.004	(0.0034, 0.0001)	0.0013	(0.0038, 0.00001)
			(1, 1)	(0.0001, 4)	(0.00001, 2, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
		0.1	(0.3614, 0.1)	0.0045	(0.0029, 0.00001)	0.0039	(0.0038, 0.00001)
			(10, 1)	(0.0001, 4)	(0.0001, 4, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
O_vs_F	Bhattacharyya	0	(0.196, 0.1)	0.008	(0.0233, 1)	0.0034	(0.0039, 0.00001)
			(0.1, 0.5)	(0.0001, 2)	(1, 16, 0.1)	(0.001, 4)	(0.001, 4, 0.6)
		0.05	(0.3557, 0.1)	0.0032	(0.0228, 1)	0.0068	(0.0037, 0.0001)
			(0.1, 0.5)	(0.001, 8)	(1, 16, 0.1)	(0.001, 4)	(0.0001, 2, 0.6)
		0.1	(0.384, 0.1)	0.0044	(0.0318, 1)	0.0021	(0.0039, 0.001)
			(0.1, 0.5)	(0.0001, 4)	(10, 16, 0.1)	(0.0001, 4)	(0.0001, 2, 0.6)
O_vs_F	Entropy	0	(0.5333, 0.1)	0.0046	(0.0264, 1)	0.0015	(0.0038, 0.00001)
			(0.1, 0.5)	(0.0001, 2)	(1, 16, 0.1)	(0.001, 4)	(0.001, 4, 0.6)
		0.05	(0.4898, 0.1)	0.003	(0.0286, 1)	0.0365	(0.0035, 0.0001)
			(0.1, 0.5)	(0.0001, 2)	(1, 16, 0.1)	(0.001, 4)	(0.0001, 2, 0.6)
		0.1	(0.1146, 0.1)	0.0039	(0.0106, 0.1)	0.0063	(0.0153, 0.0001)
			(0.1, 0.5)	(0.01, 32)	(0.01, 8, 0.1)	(0.0001, 4)	(0.001, 4, 0.1)
O_vs_F	ROC	0	(0.1597, 0.1)	0.0043	(0.0072, 0.0001)	0.0025	(0.0043, 0.0001)
			(1, 0.03125)	(0.0001, 4)	(0.00001, 4, 0.1)	(0.0001, 8)	(0.001, 16, 0.6)
		0.05	(0.0941, 0.1)	0.0042	(0.0021, 0.001)	0.0015	(0.0037, 0.00001)
			(1, 0.03125)	(0.0001, 8)	(0.00001, 8, 0.1)	(0.0001, 8)	(0.0001, 8, 0.5)
		0.1	(0.1474, 0.1)	0.0077	(0.0039, 0.001)	0.0016	(0.0033, 0.00001)
			(1, 0.03125)	(0.0001, 8)	(0.00001, 8, 0.1)	(0.0001, 8)	(0.0001, 8, 0.3)
O_vs_F	T-Test	0	(0.0727, 0.1)	0.0243	(0.0023, 0.01)	0.0015	(0.0034, 0.0001)
			(1, 0.0625)	(0.001, 16)	(0.00001, 8, 0.1)	(0.00001, 4)	(0.001, 8, 0.5)

(*continued on next page*)

Table 25 (*continued*).

Dataset	Features	r	pin SVM (Huang et al., 2013)	TWSVM (Jayadeva et al., 2007)	UTSVM (Qi et al., 2012)	Pin-GTSVM (Tanveer et al., 2019b)	Tpin-UTWSVM
Size ($f \times s$) (200×200)			(Time (s), τ) (C_1, σ)	Time (s) ($C_1 = C_2, \sigma$)	(Time (s), C_u) ($C_1 = C_2, \sigma, \epsilon$)	Time (s) ($C_1 = C_2, \sigma$)	(Time (s), C_u) ($C_1 = C_2, \sigma, \epsilon$)
		0.05	(0.074, 0.1)	0.0087	(0.0033, 0.01)	0.0071	(0.0034, 0.0001)
			(1, 0.0625)	(0.001, 16)	(0.00001, 8, 0.1)	(0.00001, 4)	(0.001, 8, 0.5)
		0.1	(0.0542, 0.1)	0.0036	(0.0017, 0.01)	0.0031	(0.0035, 0.00001)
			(1, 0.0625)	(0.0001, 4)	(0.00001, 8, 0.1)	(0.00001, 8)	(0.001, 8, 0.5)
O_vs_N	Bhattacharyya	0	(0.1024, 0.1)	0.005	(0.0708, 0.0001)	0.0038	(0.0036, 0.00001)
			(1, 0.25)	(0.0001, 8)	(0.0001, 8, 0.1)	(0.00001, 8)	(0.00001, 8, 0.6)
		0.05	(0.0584, 0.1)	0.0048	(0.0042, 0.0001)	0.0029	(0.0101, 0.00001)
			(1, 0.25)	(0.0001, 8)	(0.0001, 8, 0.1)	(0.00001, 8)	(0.00001, 8, 0.6)
		0.1	(0.1596, 0.1)	0.0025	(0.0035, 0.0001)	0.0028	(0.0034, 0.0001)
			(1, 0.25)	(0.0001, 8)	(0.0001, 8, 0.1)	(0.00001, 8)	(0.00001, 8, 0.5)
O_vs_N	Entropy	0	(0.1593, 0.1)	0.0046	(0.0039, 0.0001)	0.0019	(0.0035, 0.00001)
			(1, 0.25)	(0.0001, 8)	(0.00001, 8, 0.1)	(0.00001, 8)	(0.00001, 8, 0.6)
		0.05	(0.0971, 0.1)	0.0057	(0.0037, 0.0001)	0.0015	(0.0112, 0.00001)
			(1, 0.25)	(0.0001, 8)	(0.0001, 8, 0.1)	(0.00001, 8)	(0.00001, 8, 0.6)
		0.1	(0.0714, 0.1)	0.0041	(0.0042, 0.0001)	0.0033	(0.0035, 0.00001)
			(1, 0.25)	(0.0001, 8)	(0.00001, 8, 0.1)	(0.00001, 8)	(0.00001, 8, 0.6)
O_vs_N	ROC	0	(0.1802, 0.1)	0.0061	(0.0067, 0.001)	0.0012	(0.0139, 0.00001)
			(0.1, 0.25)	(0.0001, 8)	(0.00001, 8, 0.1)	(0.001, 8)	(0.0001, 8, 0.1)
		0.05	(0.2474, 0.1)	0.0047	(0.0039, 0.001)	0.0013	(0.0036, 0.00001)
			(0.1, 0.25)	(0.0001, 8)	(0.00001, 8, 0.1)	(0.0001, 8)	(0.0001, 8, 0.1)
		0.1	(0.1067, 0.1)	0.0039	(0.0031, 0.001)	0.0013	(0.0035, 0.00001)
			(0.01, 0.0625)	(0.0001, 8)	(0.00001, 8, 0.1)	(0.00001, 8)	(0.001, 8, 0.1)
O_vs_N	T-Test	0	(0.1693, 0.1)	0.0097	(0.0083, 0.0001)	0.0014	(0.0034, 0.001)
			(0.01, 0.125)	(0.001, 8)	(0.00001, 8, 0.1)	(0.00001, 8)	(0.0001, 8, 0.5)
		0.05	(0.1438, 0.1)	0.0052	(0.0031, 0.0001)	0.0053	(0.0036, 0.001)
			(0.01, 0.0625)	(0.001, 8)	(0.0001, 8, 0.1)	(0.00001, 8)	(0.0001, 8, 0.5)
		0.1	(0.0616, 0.1)	0.0044	(0.0029, 0.0001)	0.0012	(0.0035, 0.00001)
			(1, 0.25)	(0.001, 8)	(0.0001, 8, 0.1)	(0.00001, 8)	(0.00001, 8, 0.6)
O_vs_N	Wilcoxon	0	(0.4515, 0.1)	0.007	(0.0024, 0.001)	0.0027	(0.0031, 0.001)
			(1, 0.25)	(0.001, 16)	(0.00001, 8, 0.1)	(0.00001, 8)	(0.0001, 16, 0.5)
		0.05	(0.276, 0.1)	0.0037	(0.0173, 0.0001)	0.0048	(0.0034, 0.001)
			(1, 0.25)	(0.001, 32)	(0.00001, 8, 0.1)	(0.00001, 8)	(0.001, 32, 0.5)
		0.1	(0.3436, 0.1)	0.0109	(0.0116, 0.0001)	0.0015	(0.0034, 0.00001)
			(1, 0.25)	(0.001, 16)	(0.0001, 8, 0.1)	(0.00001, 16)	(0.0001, 16, 0.1)
O_vs_S	Bhattacharyya	0	(0.2371, 0.1)	0.0033	(0.004, 0.0001)	0.0013	(0.0041, 0.00001)
			(1, 1)	(0.0001, 4)	(0.00001, 2, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
		0.05	(0.1745, 0.1)	0.0047	(0.0031, 0.0001)	0.0223	(0.0103, 0.00001)
			(1, 1)	(0.00001, 4)	(0.00001, 2, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
		0.1	(0.0994, 0.1)	0.0048	(0.0066, 0.00001)	0.0032	(0.0035, 0.00001)
			(1, 0.5)	(0.00001, 4)	(0.00001, 8, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
O_vs_S	Entropy	0	(0.2877, 0.1)	0.0036	(0.0036, 0.0001)	0.0016	(0.0087, 0.00001)
			(1, 1)	(0.0001, 4)	(0.00001, 2, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
		0.05	(0.3541, 0.1)	0.0043	(0.0019, 0.0001)	0.0041	(0.0042, 0.00001)
			(1, 1)	(0.00001, 4)	(0.00001, 2, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
		0.1	(0.2072, 0.1)	0.0067	(0.0024, 0.00001)	0.0022	(0.0034, 0.00001)
			(1, 0.25)	(0.00001, 4)	(0.00001, 8, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
O_vs_S	T-Test	0	(0.1848, 0.1)	0.0057	(0.0018, 0.001)	0.0034	(0.0194, 0.00001)
			(1, 0.5)	(0.0001, 4)	(0.00001, 4, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
		0.05	(0.2979, 0.1)	0.004	(0.0053, 0.001)	0.0023	(0.0052, 0.00001)
			(1, 0.5)	(0.00001, 4)	(0.00001, 4, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
		0.1	(0.2239, 0.1)	0.0158	(0.0018, 0.001)	0.004	(0.011, 0.00001)
			(1, 0.25)	(0.00001, 4)	(0.00001, 4, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
O_vs_S	Wilcoxon	0	(0.0999, 0.1)	0.0052	(0.002, 0.001)	0.0024	(0.0034, 0.00001)
			(1, 0.25)	(0.0001, 4)	(0.00001, 4, 0.1)	(0.00001, 4)	(0.00001, 4, 0.3)
		0.05	(0.1028, 0.1)	0.0172	(0.0022, 0.001)	0.0107	(0.0034, 0.00001)
			(1, 0.25)	(0.00001, 4)	(0.00001, 4, 0.1)	(0.00001, 8)	(0.00001, 4, 0.3)
		0.1	(0.0773, 0.1)	0.0071	(0.0018, 0.001)	0.0031	(0.0036, 0.00001)
			(1, 0.25)	(0.00001, 4)	(0.00001, 4, 0.1)	(0.00001, 8)	(0.00001, 4, 0.5)
Z_vs_F	Bhattacharyya	0	(0.3403, 0.1)	0.0063	(0.0026, 0.00001)	0.003	(0.0035, 0.01)
			(1, 0.5)	(0.0001, 4)	(0.00001, 4, 0.1)	(0.0001, 2)	(0.001, 8, 0.1)
		0.05	(0.1516, 0.1)	0.0044	(0.0023, 0.0001)	0.0017	(0.0043, 0.00001)
			(1, 0.5)	(0.0001, 4)	(0.0001, 4, 0.1)	(0.0001, 4)	(0.0001, 4, 0.1)
		0.1	(0.1541, 0.1)	0.0048	(0.0023, 0.00001)	0.0043	(0.004, 0.00001)
			(1, 0.5)	(0.0001, 4)	(0.00001, 4, 0.1)	(0.0001, 4)	(0.0001, 4, 0.6)
Z_vs_F	Entropy	0	(0.2304, 0.1)	0.0157	(0.0145, 0.00001)	0.0013	(0.0033, 0.01)
			(1, 0.5)	(0.0001, 4)	(0.00001, 4, 0.1)	(0.0001, 2)	(0.001, 8, 0.1)
		0.05	(0.1035, 0.1)	0.019	(0.0027, 0.0001)	0.0032	(0.0034, 0.001)
			(1, 0.5)	(0.0001, 4)	(0.00001, 4, 0.1)	(0.001, 4)	(0.0001, 2, 0.3)
		0.1	(0.0817, 0.1)	0.0081	(0.0025, 0.00001)	0.0017	(0.0041, 0.00001)
			(1, 0.5)	(0.001, 4)	(0.00001, 4, 0.1)	(0.001, 4)	(0.001, 4, 0.1)

(*continued on next page*)

Table 25 (*continued*).

Dataset	Features	r	pin SVM (Huang et al., 2013)	TWSVM (Jayadeva et al., 2007)	UTSVM (Qi et al., 2012)	Pin-GTSVM (Tanveer et al., 2019b)	Tpin-UTWSVM
Size ($f \times s$) (200×200)			(Time (s), τ) (C_1, σ)	Time (s) ($C_1 = C_2, \sigma$)	(Time (s), C_u) ($C_1 = C_2, \sigma, \epsilon$)	Time (s) ($C_1 = C_2, \sigma$)	(Time (s), C_u) ($C_1 = C_2, \sigma, \epsilon$)
Z_vs_F	T-Test	0	(0.138, 0.1)	0.004	(0.0019, 0.0001)	0.0052	(0.0045, 0.1)
			(1, 0.25)	(0.01, 32)	(0.0001, 8, 0.1)	(0.00001, 32)	(0.01, 32, 0.6)
		0.05	(0.2069, 0.1)	0.0091	(0.0022, 0.0001)	0.004	(0.0042, 0.1)
			(10, 0.25)	(0.001, 16)	(0.0001, 8, 0.1)	(0.00001, 32)	(0.01, 32, 0.6)
		0.1	(0.0728, 0.1)	0.0026	(0.0023, 0.0001)	0.0023	(0.0036, 0.001)
			(1, 0.25)	(0.001, 16)	(0.001, 8, 0.1)	(0.00001, 32)	(0.0001, 16, 0.3)
Z_vs_F	Wilcoxon	0	(0.19, 0.1)	0.0048	(0.0022, 0.0001)	0.0024	(0.0039, 0.0001)
			(1, 0.25)	(0.0001, 8)	(0.00001, 8, 0.1)	(0.00001, 4)	(0.00001, 8, 0.6)
		0.05	(0.1707, 0.1)	0.004	(0.0018, 0.0001)	0.0013	(0.0034, 0.00001)
			(1, 0.25)	(0.0001, 8)	(0.00001, 8, 0.1)	(0.00001, 4)	(0.0001, 8, 0.1)
		0.1	(0.0775, 0.1)	0.0043	(0.0019, 0.0001)	0.0095	(0.0034, 0.00001)
			(1, 0.25)	(0.0001, 8)	(0.00001, 8, 0.1)	(0.00001, 8)	(0.0001, 8, 0.1)
Z_vs_N	Bhattacharyya	0	(0.0596, 0.1)	0.0052	(0.0024, 0.0001)	0.0038	(0.0036, 0.00001)
			(1, 0.25)	(0.0001, 8)	(0.00001, 8, 0.1)	(0.0001, 4)	(0.0001, 8, 0.6)
		0.05	(0.1309, 0.1)	0.0036	(0.0019, 0.0001)	0.0043	(0.0031, 0.00001)
			(1, 0.25)	(0.0001, 8)	(0.001, 8, 0.1)	(0.00001, 4)	(0.0001, 8, 0.6)
		0.1	(0.1867, 0.1)	0.0052	(0.0053, 0.01)	0.0021	(0.0127, 0.001)
			(1, 0.25)	(0.0001, 8)	(0.00001, 8, 0.1)	(0.00001, 8)	(0.0001, 8, 0.1)
Z_vs_N	Entropy	0	(0.071, 0.1)	0.0049	(0.0018, 0.0001)	0.0012	(0.0035, 0.0001)
			(1, 0.25)	(0.0001, 8)	(0.0001, 8, 0.1)	(0.00001, 4)	(0.0001, 8, 0.3)
		0.05	(0.1027, 0.1)	0.0078	(0.0021, 0.0001)	0.007	(0.0037, 0.001)
			(1, 0.25)	(0.0001, 8)	(0.001, 8, 0.1)	(0.00001, 8)	(0.0001, 8, 0.3)
		0.1	(0.1255, 0.1)	0.0043	(0.002, 0.01)	0.0014	(0.0036, 0.00001)
			(10, 0.25)	(0.0001, 8)	(0.00001, 8, 0.1)	(0.00001, 8)	(0.0001, 8, 0.6)
Z_vs_N	ROC	0	(0.4702, 0.1)	0.0069	(0.002, 0.0001)	0.0069	(0.0054, 0.00001)
			(1, 0.125)	(0.00001, 8)	(0.00001, 8, 0.1)	(0.00001, 16)	(0.00001, 8, 0.6)
		0.05	(0.1132, 0.1)	0.0041	(0.002, 0.0001)	0.0031	(0.0035, 0.00001)
			(1, 0.125)	(0.00001, 8)	(0.00001, 8, 0.1)	(0.00001, 16)	(0.00001, 8, 0.6)
		0.1	(0.0882, 0.1)	0.004	(0.0021, 0.0001)	0.0014	(0.0037, 0.00001)
			(1, 0.125)	(0.00001, 8)	(0.00001, 8, 0.1)	(0.00001, 16)	(0.00001, 8, 0.6)
Z_vs_N	T-Test	0	(0.2917, 0.1)	0.0094	(0.002, 0.001)	0.0013	(0.0077, 0.01)
			(1, 0.125)	(0.01, 32)	(0.00001, 8, 0.1)	(0.0001, 8)	(0.01, 32, 0.1)
		0.05	(0.0862, 0.1)	0.0025	(0.0021, 0.001)	0.0052	(0.0083, 0.00001)
			(1, 0.125)	(0.001, 32)	(0.00001, 8, 0.1)	(0.0001, 8)	(0.01, 32, 0.1)
		0.1	(0.1394, 0.1)	0.0055	(0.0021, 0.001)	0.0012	(0.0034, 0.01)
			(1, 0.125)	(0.01, 32)	(0.0001, 8, 0.1)	(0.0001, 8)	(0.001, 32, 0.6)
Z_vs_N	Wilcoxon	0	(0.2177, 0.1)	0.0068	(0.0019, 0.0001)	0.0023	(0.0034, 0.00001)
			(1, 0.125)	(0.00001, 16)	(0.00001, 8, 0.1)	(0.001, 32)	(0.00001, 16, 0.1)
		0.05	(0.1137, 0.1)	0.0072	(0.0019, 0.001)	0.0012	(0.0033, 0.00001)
			(1, 0.125)	(0.00001, 8)	(0.00001, 8, 0.1)	(0.00001, 16)	(0.00001, 8, 0.6)
		0.1	(0.1421, 0.1)	0.004	(0.002, 0.001)	0.0045	(0.0034, 0.0001)
			(1, 0.125)	(0.00001, 8)	(0.00001, 8, 0.1)	(0.00001, 16)	(0.00001, 8, 0.6)
Z_vs_S	Bhattacharyya	0	(0.1579, 0.1)	0.0046	(0.002, 0.00001)	0.0023	(0.0042, 0.00001)
			(0.1, 1)	(0.00001, 2)	(0.00001, 2, 0.1)	(0.00001, 2)	(0.00001, 2, 0.1)
		0.05	(0.1351, 0.1)	0.0047	(0.0024, 0.00001)	0.0041	(0.0099, 0.00001)
			(0.1, 1)	(0.00001, 2)	(0.00001, 2, 0.1)	(0.00001, 2)	(0.00001, 2, 0.1)
		0.1	(0.3219, 0.1)	0.0068	(0.0052, 0.00001)	0.0015	(0.0037, 0.00001)
			(1, 1)	(0.00001, 2)	(0.00001, 2, 0.1)	(0.00001, 2)	(0.00001, 2, 0.1)
Z_vs_S	Entropy	0	(0.2352, 0.1)	0.0057	(0.0045, 0.00001)	0.0031	(0.0142, 0.00001)
			(0.1, 1)	(0.00001, 2)	(0.00001, 2, 0.1)	(0.00001, 2)	(0.00001, 2, 0.1)
		0.05	(0.1345, 0.1)	0.0026	(0.0032, 0.00001)	0.004	(0.0034, 0.00001)
			(0.1, 1)	(0.00001, 2)	(0.00001, 2, 0.1)	(0.00001, 2)	(0.00001, 2, 0.1)
		0.1	(0.0905, 0.1)	0.007	(0.0052, 0.00001)	0.0053	(0.0034, 0.00001)
			(1, 1)	(0.00001, 2)	(0.00001, 2, 0.1)	(0.00001, 2)	(0.00001, 2, 0.1)
Z_vs_S	T-Test	0	(0.0986, 0.1)	0.0042	(0.0017, 0.0001)	0.0037	(0.0035, 0.00001)
			(1, 1)	(0.00001, 4)	(0.00001, 2, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
		0.05	(0.2026, 0.1)	0.0043	(0.0022, 0.0001)	0.0013	(0.0034, 0.00001)
			(10, 1)	(0.00001, 4)	(0.00001, 2, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
		0.1	(0.1042, 0.1)	0.0039	(0.0092, 0.00001)	0.0044	(0.0036, 0.00001)
			(1, 1)	(0.00001, 4)	(0.00001, 4, 0.1)	(0.00001, 4)	(0.00001, 4, 0.1)
Average time		0	0.2457	0.0064	0.0076	0.0028	0.0054
		0.05	0.2538	0.0055	0.0049	0.0051	0.0072
		0.1	0.2505	0.0058	0.0054	0.0033	0.0057

Tpin-UTWSVM model has the highest accuracy for $r = 0$ case. For $r = 0.05$ and $r = 0.1$ case, the accuracies for the aforementioned order of models are 71.72%, 85.2%, 84.04%, 85.45%, 86.21% and 69.65%, 85.56%, 82.73%, 85.66%, 86.82%, respectively. Hence, the proposed Tpin-UTWSVM has the best accuracies for different r values. Table 24 also shows the average ranks of all the models under comparison. The proposed Tpin-UTWSVM has the lowest rank 2.38, 2.12, 1.86 for $r = 0, 0.05, 0.1$, respectively. In the Table 24, the best average values are depicted in bold. Table 25 consist of training time and different optimal hyperparameters for the models under study over the 33 EEG datasets and different noise levels. Fig. 7 pictorially represents the comparison of different models on EEG datasets for three noise levels.

Table 26
Performance of proposed Tpin-UTWSVM model with baseline models on EEG datasets with Gaussian kernel.

Dataset	r	pin SVM (Huang et al., 2013)	TWSVM (Jayadeva et al., 2007)	UTSVM (Qi et al., 2012)	Pin-GTSVM (Tanveer et al., 2019b)	Tpin-UTWSVM
Size ($f \times s$)		(Accuracy ± sd) (Recall, Precision) (Time (s), τ) (C_1, σ)	(Accuracy ± sd) (Recall, Precision) Time (s) ($C_1 = C_2, \sigma$)	(Accuracy ± sd) (Recall, Precision) (Time (s), C_u) ($C_1 = C_2, \sigma, \epsilon$)	(Accuracy ± sd) (Recall, Precision) Time (s) ($C_1 = C_2, \sigma$)	(Accuracy ± sd) (Recall, Precision) (Time (s), C_u) ($C_1 = C_2, \sigma, \epsilon$)
(415 × 91)						
CN_vs_AD	0	82.26 ± 6.63 (0.6863, 0.8537) (0.5428, 0.1) (1, 0.125)	(86.29 ± 5.72) (0.9412, 0.7742) 0.0114 (0.1, 4)	(76.61 ± 8.98) (0.549, 0.8235) (0.0122, 0.00001) (0.001, 8, 0.1)	(89.52 ± 6.62) (0.8824, 0.8654) 0.0043 (0.00001, 8)	(**90.32 ± 6.64**) (0.8824, 0.8824) (0.008, 0.0001) (0.00001, 8, 0.5)
CN_vs_AD	0.05	82.26 ± 6 (0.6863, 0.8537) (0.5373, 0.1) (1, 0.125)	(88.71 ± 5.5) (0.9216, 0.8246) 0.0237 (0.00001, 0.5)	(86.29 ± 14.41) (0.7647, 0.8864) (0.008, 0.01) (0.00001, 4, 0.1)	(89.52 ± 6.58) (0.8824, 0.8654) 0.0039 (0.00001, 8)	(**93.55 ± 8.24**) (0.902, 0.9388) (0.0065, 0.0001) (0.0001, 8, 0.3)
CN_vs_AD	0.1	83.06 ± 6.42 (0.6863, 0.875) (0.7724, 0.1) (1, 0.125)	(87.1 ± 9.75) (0.902, 0.807) 0.008 (0.001, 1)	(87.1 ± 12.85) (0.7647, 0.907) (0.0078, 0.01) (0.00001, 4, 0.1)	(90.32 ± 7.34) (0.8627, 0.898) 0.0045 (0.01, 16)	(**90.32 ± 6.45**) (0.8824, 0.8824) (0.0072, 0.00001) (0.00001, 8, 0.3)

Table 27
Pairwise win-tie-loss of the proposed Tpin-UTWSVM wrt baseline models over EEG datasets. With noise level r in the column, comparison of row method with the proposed Tpin-UTWSVM model. Here, [a,b,c] indicates that the proposed Tpin-UTWSVM technique wins 'a' number of times, ties 'b' number of times, and loses 'c' number of times wrt the row method.

Model	Tpin-UTWSVM			
	$r = 0$	$r = 0.05$	$r = 0.1$	Total
pin SVM (Huang et al., 2013)	[31, 0, 2]	[33, 0, 0]	[32, 0, 1]	[96, 0, 3]
TWSVM (Jayadeva et al., 2007)	[15, 9, 9]	[17, 10, 6]	[19, 8, 6]	[51, 27, 21]
UTSVM (Qi et al., 2012)	[17, 5, 11]	[17, 6, 10]	[26, 3, 4]	[60, 14, 25]
Pin-GTSVM (Tanveer et al., 2019b)	[15, 3, 15]	[18, 4, 11]	[19, 4, 10]	[52, 11, 36]

We statistically analyze the results of the models under comparison. We employed Friedman test for $r = 0$ case and calculate $\chi_F^2 = 50.2447$ and $F_F = 19.6537$ from Eqs. (60) and (61), respectively, for $L = 33$ and $d = 5$. The degree of freedom of F_F is $(4, 128)$ and have critical value $F_{(4,128)} = 2.4424$ at 5% level of significance. As $F_{(4,128)} < F_F$, thus the models are statistically different. Table 27 shows the number of wins, ties and losses of baseline models wrt the proposed Tpin-UTWSVM over the EEG datasets for different noise levels.

6.3. AD dataset

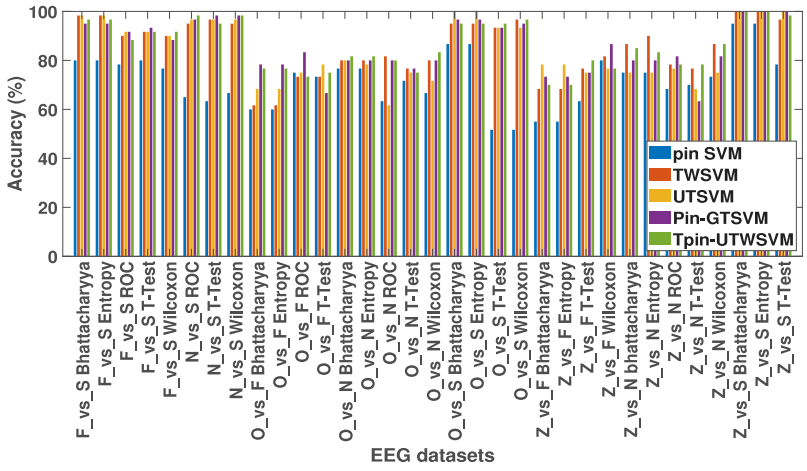
AD data is publicly available on ADNI repository which was launched by Michael W. Weiner as Principal Investigator in 2003 and aims to analyze different approaches for neuroimaging such as magnetic resonance imaging (MRI) to diagnose AD from mild cognitive impairment (MCI) stage. The reader can visit www.adni-info.org. for further details. The MRI images of brain from ADNI data is shown in Fig. 8 from 3 different planes. From ADNI database 150 T1-weighted structural MRI images are downloaded. The three categories control normal (CN), mild cognitive impairment (MCI) and AD consist of 50 subjects each in the year range 60 – 90 having 75.83 and 6.07 as the mean and standard deviation, respectively. For extracting the features, we opted the methods mentioned in Richhariya et al. (2020). The classification performance of the compared models over AD data is obtained in terms of classifying CN versus AD subjects, CN versus MCI subjects and MCI versus AD subjects.

6.4. Experimental analysis over AD data

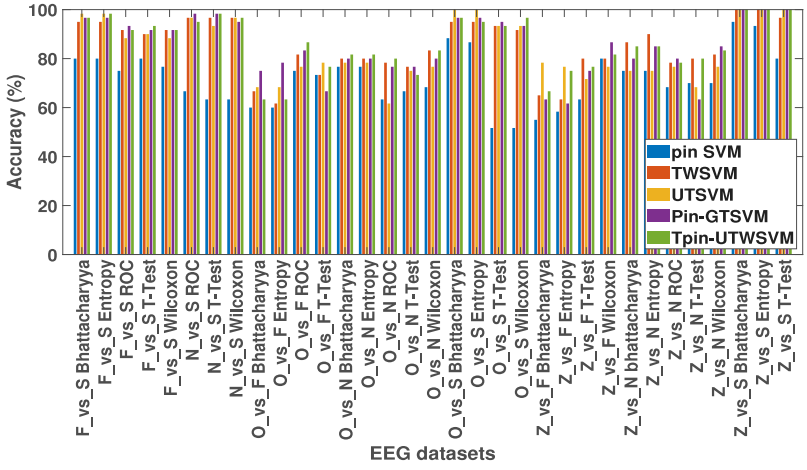
The performance of the proposed Tpin-UTWSVM and baseline models pin SVM (Huang et al., 2013), TWSVM (Jayadeva et al., 2007), UTSVM (Qi et al., 2012), Pin-GTSVM (Tanveer et al., 2019b) have been evaluated on CN versus AD (CN_vs_AD), CN versus MCI (CN_vs_MCI) and MCI versus AD (MCI_vs_AD) cases for $r = 0, 0.05, 0.1$. The proposed Tpin-UTWSVM model performs better on CN_vs_AD case only as compare to the baseline models. Thus, in Table 26, we represent the results with optimal hyperparameters of the proposed Tpin-UTWSVM wrt the baseline models for CN_vs_AD case only. It can be seen from Table 26 that for $r = 0, 0.05$ and 0.1, the proposed Tpin-UTWSVM model has the best performance for the CN_vs_AD case which are 90.32, 93.55 and 90.32, respectively.

7. Conclusion

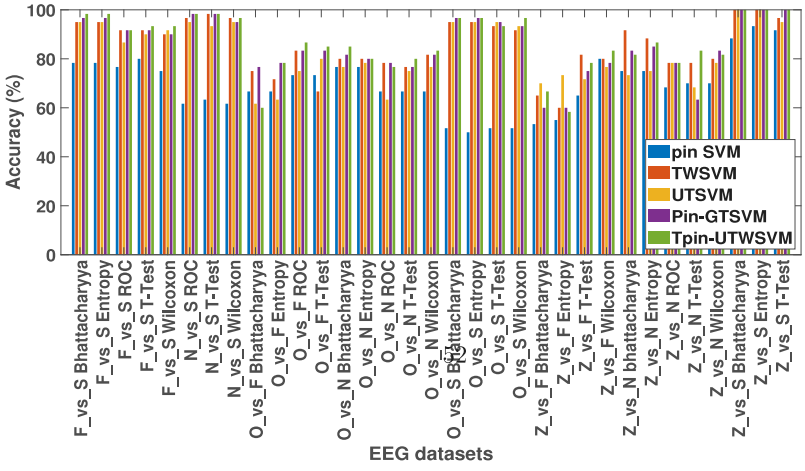
In this article, we proposed universum twin support vector machine with truncated pinball loss (Tpin-UTWSVM) for classification problems. The model with pinball loss has the advantages of noise insensitivity and stability to resampling, however the model loses sparsity in comparison to the model with hinge loss. The proposed Tpin-UTWSVM combines the merits of pinball loss and hinge loss leading to a model which is sparse, insensitive to noise, stable to resampling and robust to outliers without compromising with the theoretical properties of pinball loss. The proposed non-convex and non-differentiable optimization problem is solved by concave convex procedure (CCCP). In order to train the proposed model, SOR technique is used which is efficient and does not occupy memory. To demonstrate the noise insensitivity of the proposed Tpin-UTWSVM model, we performed the numerical experiments of the proposed Tpin-UTWSVM and baseline models, *i.e.,* pin SVM, Pin-GTSVM, UTSVM and TWSVM at different noise levels. The average accuracies of pin SVM, TWSVM, UTSVM, Pin-GTSVM and Tpin-UTWSVM for $r = 0$ case over UCI datasets are 79.76%, 79.88%, 79.79%, 76.91% and 80.93%, respectively. For the same order of models, the accuracies for the case $r = 0.05$ and $r = 0.1$ over UCI datasets are 79.15%, 80.18%, 77.63%, 77.6%, 80.74% and 78.9%, 80.33%, 77.75%, 77.23%, 80.68%, respectively. Thus, the numerical experiments and statistical analysis performed on 19 binary UCI datasets showed that the proposed Tpin-UTWSVM has the best performance with the highest average accuracies than the baseline models. We also illustrated through numerical experiments, the effect of loss function parameters τ and s on the sparsity of the proposed Tpin-UTWSVM model. We implemented the proposed Tpin-UTWSVM and baseline models to the biomedical domain. The numerical experiments



(a) Bar graph to compare the accuracies of models on EEG datasets with $r = 0$.



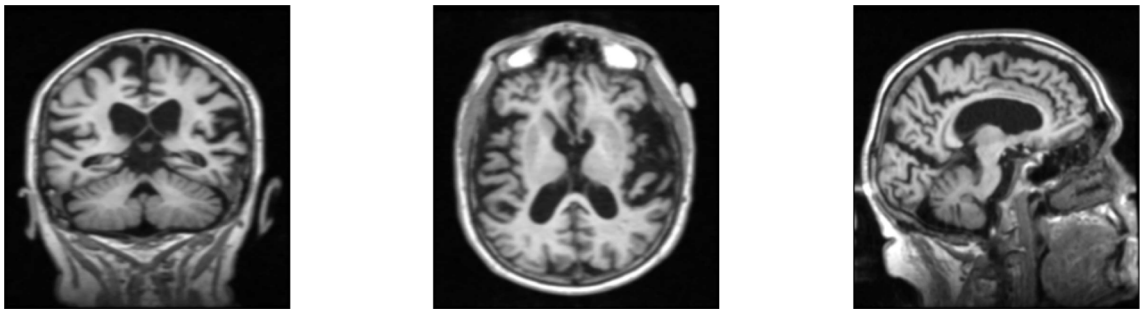
(b) Bar graph to compare the accuracies of models on EEG datasets with $r = 0.05$.



(c) Bar graph to compare the accuracies of models on EEG datasets with $r = 0.1$.

Fig. 7. Bar graph representation to compare the performance of models over EEG datasets with different noise levels.

conducted on EEG dataset shows that the proposed Tpin-UTWSVM model has the highest average accuracies of 86.31%, 86.21%, 86.82% and lowest average rank of 2.38, 2.12, 1.86 for the noise level being 0, 0.05 and 0.1 respectively. The experimental results for AD detection reveals that for classifying AD subjects versus control normal subjects, the proposed Tpin-UTWSVM has the best accuracy of 90.32%, 93.55% and 90.32% for $r = 0,\ 0.05$ and 0.1, respectively. Though the proposed Tpin-UTWSVM has several assets however, the proposed model involves inversion of matrix of order $(n + 1)$, n being number of features. Also, the proposed model uses universum data, which contributes to the complexity. Hence, the proposed model is not suitable for large scale datasets. Truncated pinball loss has flattened curve on one side which leads to model's sparsity. Flattening the curve of pinball loss on both sides will lead to a robust and sparse model along with



(a) coronal plane (b) axial plane (c) sagittal plane

Fig. 8. MRI image of brain in three different planes.

preserving the properties of the pinball loss. In future, researchers can incorporate the idea of both sides truncated pinball loss with twin support vector machine model. One can also extend the proposed idea for multi-class classification.

CRediT authorship contribution statement

Anuradha Kumari: Conceptualization, Methodology, Writing – original draft, Validation, Visualization, Resources. **M. Tanveer:** Conceptualization, Methodology, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work is funded by National Supercomputing Mission under DST and Miety, Govt. of India under Grant No. DST/NSM/R&D HPC Appl/2021/03.29, SERB under Mathematical Research Impact-Centric Support (MATRICS), India scheme grant no. MTR/2021/000787 and DST under FIST scheme, India grant no. SR/FST/MS-I/2018/26. Ms. Anuradha Kumari (File no - 09/1022 (12437)/2021-EMR-I) acknowledge the financial support given as scholarship by Council of Scientific and Industrial Research (CSIR), New Delhi, India.

The Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health, United States Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012) financed the collection of data and its dissemination for this investigation. ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. Support for ADNI clinical locations in Canada is being given by the Canadian Institutes of Health Research. The Foundation for the National Institutes of Health (www.fnih.org) facilitates private sector contributions. The Northern California Institute for Research and Education is the grantee agency, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. The Center for Neuro Imaging at the University of Southern California disseminates ADNI data.

References

Andrzejak, R.G., Lehnertz, K., Mormann, F., Rieke, C., David, P., Elger, C.E., 2001. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. Phys. Rev. E 64 (6), 061907.

Bollegala, D., Matsuo, Y., Ishizuka, M., 2010. A web search engine-based approach to measure semantic similarity between words. IEEE Trans. Knowl. Data Eng. 23 (7), 977–990.

Chapelle, O., Agarwal, A., Sinz, F., Schölkopf, B., 2007. An analysis of inference with the universum. Adv. Neural Inf. Process. Syst. 20.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30.

Ding, M., Yang, D., Li, X., 2013. Fault diagnosis for wireless sensor by twin support vector machine. Math. Probl. Eng. 2013.

Ganaie, M., Kumari, A., Malik, A.K., Tanveer, M., 2022a. EEG signal classification using improved intuitionistic fuzzy twin support vector machines. Neural Comput. Appl. 1–17. http://dx.doi.org/10.1007/s00521-022-07655-x.

Ganaie, M., Tanveer, M., Jangir, J., 2022b. EEG signal classification via pinball universum twin support vector machine. Ann. Oper. Res. 1–42.

Gauthier, S., Rosa-Neto, P., Morais, J., Webster, C., 2021. Journey through the Diagnosis of Dementia. World Alzheimer Report 2021.

Huang, X., Shi, L., Suykens, J.A., 2013. Support vector machine classifier with pinball loss. IEEE Trans. Pattern Anal. Mach. Intell. 36 (5), 984–997.

Jayadeva, Khemchandani, R., Chandra, S., 2007. Twin support vector machines for pattern classification. IEEE Trans. Pattern Anal. Mach. Intell. 29 (5), 905–910.

Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. In: European Conference on Machine Learning. Springer, pp. 137–142.

Kullback, S., 1997. Information Theory and Statistics. Courier Corporation.

Kumar, M.A., Gopal, M., 2009. Least squares twin support vector machines for pattern classification. Expert Syst. Appl. 36 (4), 7535–7543.

Laxmi, S., Gupta, S., Kumar, S., 2022. Intuitionistic fuzzy least square twin support vector machines for pattern classification. Ann. Oper. Res. 1–50. http://dx.doi.org/10.1007/s10479-022-04626-2.

Li, Y., Guan, C., 2008. Joint feature re-extraction and classification using an iterative semi-supervised support vector machine algorithm. Mach. Learn. 71 (1), 33–53.

Li, G., Yin, L., Yang, L., Wu, Z., 2022. Robust projection twin support vector machine via DC programming. Optimization 1–23. http://dx.doi.org/10.1080/02331934.2021.1960331.

Mangasarian, O.L., Musicant, D.R., 1999. Successive overrelaxation for support vector machines. IEEE Trans. Neural Netw. 10 (5), 1032–1037.

Mangasarian, O.L., Wild, E.W., 2005. Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE Trans. Pattern Anal. Mach. Intell. 28 (1), 69–74.

Newman, D.J., 1998. UCI repository of machine learning database. http://www.ics.uci.edu/~mlearn/mlrepository.html.

Osuna, E., Freund, R., Girosit, F., 1997. Training support vector machines: an application to face detection. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, pp. 130–136.

Pal, M., Mather, P.M., 2005. Support vector machines for classification in remote sensing. Int. J. Remote Sens. 26 (5), 1007–1011.

Pellegrini, E., Ballerini, L., Hernandez, M.d.C.V., Chappell, F.M., González-Castro, V., Anblagan, D., Danso, S., Muñoz-Maniega, S., Job, D., Pernet, C., et al., 2018. Machine learning of neuroimaging for assisted diagnosis of cognitive impairment and dementia: a systematic review. Alzheimer's Dement.: Diagn. Assess. Dis. Monit. 10, 519–535.

Peng, X., 2011. TPMSVM: a novel twin parametric-margin support vector machine for pattern recognition. Pattern Recognit. 44 (10–11), 2678–2692. http://dx.doi.org/10.1016/j.patcog.2011.03.031.

Qi, Z., Tian, Y., Shi, Y., 2012. Twin support vector machine with universum data. Neural Netw. 36, 112–119.

Rezvani, S., Wang, X., Pourpanah, F., 2019. Intuitionistic fuzzy twin support vector machines. IEEE Trans. Fuzzy Syst. 27 (11), 2140–2151.

Richhariya, B., Gupta, D., 2019. Facial expression recognition using iterative universum twin support vector machine. Appl. Soft Comput. 76, 53–67.

Richhariya, B., Sharma, A., Tanveer, M., 2018. Improved universum twin support vector machine. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, pp. 2045–2052.

Richhariya, B., Tanveer, M., 2018. EEG signal classification using universum support vector machine. Expert Syst. Appl. 106, 169–182.

Richhariya, B., Tanveer, M., 2020a. A reduced universum twin support vector machine for class imbalance learning. Pattern Recognit. 102, 107150.

Richhariya, B., Tanveer, M., 2020b. Universum least squares twin parametric-margin support vector machine. In: 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–8.

Richhariya, B., Tanveer, M., Initiative, A.D.N., 2021a. An efficient angle-based universum least squares twin support vector machine for classification. ACM Trans. Internet Technol. (TOIT) 21 (3), 1–24.

Richhariya, B., Tanveer, M., Initiative, A.D.N., 2021b. A fuzzy universum least squares twin support vector machine (FULSTSVM). Neural Comput. Appl. 1–12. http://dx.doi.org/10.1007/s00521-021-05721-4.

Richhariya, B., Tanveer, M., Rashid, A.H., Initiative, A.D.N., 2020. Diagnosis of Alzheimer's disease using universum support vector machine based recursive feature elimination (USVM-RFE). Biomed. Signal Process. Control 59, 101903.

Rosso, O., Hyslop, W., Gerlach, R., Smith, R., Rostas, J., Hunter, M., 2005. Quantitative EEG analysis of the maturational changes associated with childhood absence epilepsy. Physica A 356 (1), 184–189.

Shao, Y.-H., Chen, W.-J., Deng, N.-Y., 2014. Nonparallel hyperplane support vector machine for binary classification problems. Inform. Sci. 263, 22–35.

Shao, Y.-H., Zhang, C.-H., Wang, X.-B., Deng, N.-Y., 2011. Improvements on twin support vector machines. IEEE Trans. Neural Netw. 22 (6), 962–968.

Sharma, R., Goel, T., Tanveer, M., Lin, C., Murugan, R., 2023. Deep learning based diagnosis and prognosis of Alzheimer's disease: A comprehensive review. IEEE Trans. Cogn. Dev. Syst. http://dx.doi.org/10.1109/TCDS.2023.3254209.

Shen, X., Niu, L., Qi, Z., Tian, Y., 2017. Support vector machine classifier with truncated pinball loss. Pattern Recognit. 68, 199–210.

Subasi, A., Gursoy, M.I., 2010. EEG signal classification using PCA, ICA, LDA and support vector machines. Expert Syst. Appl. 37 (12), 8659–8666.

Tanveer, M., 2015a. Application of smoothing techniques for linear programming twin support vector machines. Knowl. Inf. Syst. 45 (1), 191–214.

Tanveer, M., 2015b. Robust and sparse linear programming twin support vector machines. Cogn. Comput. 7 (1), 137–149.

Tanveer, M., Rajani, T., Ganaie, M., 2019a. Improved sparse pinball twin SVM. In: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC). IEEE, pp. 3287–3291.

Tanveer, M., Rajani, T., Rastogi, R., Shao, Y.-H., Ganaie, M., 2022. Comprehensive review on twin support vector machines. Ann. Oper. Res. 1–46. http://dx.doi.org/10.1007/s10479-022-04575-w.

Tanveer, M., Richhariya, B., Khan, R.U., Rashid, A.H., Khanna, P., Prasad, M., Lin, C., 2020. Machine learning techniques for the diagnosis of Alzheimer's disease: A review. ACM Trans. Multimedia Comput. Commun. Appl. (TOMM) 16 (1s), 1–35.

Tanveer, M., Sharma, A., Suganthan, P.N., 2019b. General twin support vector machine with pinball loss function. Inform. Sci. 494, 311–327.

Tanveer, M., Tiwari, A., Choudhary, R., Ganaie, M., 2021. Large-scale pinball twin support vector machines. Mach. Learn. 1–24. http://dx.doi.org/10.1007/s10994-021-06061-z.

Tanveer, M., Tiwari, A., Choudhary, R., Jalan, S., 2019c. Sparse pinball twin support vector machines. Appl. Soft Comput. 78, 164–175.

Tao, H., Cheng, L., Qiu, J., Stojanovic, V., 2022. Few shot cross equipment fault diagnosis method based on parameter optimization and feature mertic. Meas. Sci. Technol. 33 (11), 115005.

Theodoridis, S., Koutroumbas, K., 2009. Feature selection. Pattern Recognit.. 261–322.

Vapnik, V., 1999. The Nature of Statistical Learning Theory. Springer Science & Business Media.

Wang, Z., Shao, Y.-H., Bai, L., Deng, N.-Y., 2015. Twin support vector machine for clustering. IEEE Trans. Neural Netw. Learn. Syst. 26 (10), 2583–2588.

Wang, H., Xu, Y., Zhou, Z., 2021. Twin-parametric margin support vector machine with truncated pinball loss. Neural Comput. Appl. 33 (8), 3781–3798.

Weston, J., Collobert, R., Sinz, F., Bottou, L., Vapnik, V., 2006. Inference with the universum. In: Proceedings of the 23rd International Conference on Machine Learning. pp. 1009–1016.

Wilcoxon, F., 1992. Individual Comparisons by Ranking Methods. Springer.

Xin, X., Tu, Y., Stojanovic, V., Wang, H., Shi, K., He, S., Pan, T., 2022. Online reinforcement learning multiplayer non-zero sum games of continuous-time Markov jump linear systems. Appl. Math. Comput. 412, 126537.

Xu, Y., Chen, M., Li, G., 2016. Least squares twin support vector machine with universum data for classification. Internat. J. Systems Sci. 47 (15), 3637–3645.

Yen, I.E., Peng, N., Wang, P.-W., Lin, S.-D., 2012. On convergence rate of concave-convex procedure. In: Proceedings of the NIPS 2012 Optimization Workshop. pp. 31–35.

Zhang, T., 2004. Statistical analysis of some multi-category large margin classification methods. J. Mach. Learn. Res. 5 (Oct), 1225–1251.

Zhang, Y., Duchi, J., Wainwright, M., 2015. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. J. Mach. Learn. Res. 16 (1), 3299–3340.

Zhuang, Z., Tao, H., Chen, Y., Stojanovic, V., Paszke, W., 2022. Iterative learning control for repetitive tasks with randomly varying trial lengths using successive projection. Internat. J. Adapt. Control Signal Process. 36 (5), 1196–1215.