



# EEG signal classification via pinball universum twin support vector machine

M. A. Ganaie<sup>1</sup> · M. Tanveer<sup>1</sup>  · Jatin Jangir<sup>2</sup>

Accepted: 20 July 2022 / Published online: 19 August 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

Electroencephalogram (EEG) have been widely used for the diagnosis of neurological diseases like epilepsy and sleep disorders. Support vector machines (SVMs) are widely used classifiers for the classification of EEG signals due to their better generalization performance. However, SVM suffers due to high computational complexity. To reduce the computations, twin support vector machines (TWSVM) solved smaller size quadratic optimization problems. To enhance the performance of the SVM and TWSVM models, prior information known as universum data has been incorporated in the universum SVM (USVM) and universum twin (UTSVM) models. Both SVM and UTSVM employ hinge loss which results in sensitivity to noise and instability. To overcome these issues and incorporate the prior information of the EEG signals, we propose a novel universum twin support vector machine with pinball loss function (Pin-UTSVM) for the classification of EEG signals. The proposed Pin-UTSVM is more stable for resampling and is noise insensitive. Furthermore, the computational complexity of proposed Pin-UTSVM model is similar to the standard UTSVM model. In the proposed approach, we used the interictal EEG signal as the universum data. Numerical experiments at varying level of noise show that the proposed Pin-UTSVM is more robust to noise compared to standard models. To show the efficiency of the proposed Pin-UTSVM model, we used multiple feature extraction techniques for the classification of the EEG signal. Experimental results reveal that the proposed Pin-UTSVM model is performing better compared to the existing models. Moreover, statistical tests show that the proposed Pin-UTSVM model is significantly better in comparison with the existing baseline models.

**Keywords** Universum · Interictal · Support vector machine · Twin support vector machine · EEG signal classification · Pinball loss

---

✉ M. Tanveer  
mtanveer@iiti.ac.in

M. A. Ganaie  
phd1901141006@iiti.ac.in

Jatin Jangir  
ee180002022@iiti.ac.in

<sup>1</sup> Department of Mathematics, Indian Institute of Technology Indore, Simrol, Indore 453552, India

<sup>2</sup> Department of Electrical Engineering, Indian Institute of Technology Indore, Simrol, Indore 453552, India

## 1 Introduction

In recent times, Machine Learning and Deep Learning have seen increased usage in healthcare optimisation (Khalilpourazari & Doulabi, 2021a, b). Electroencephalogram signals, more commonly referred to as EEG, are the principal components of several techniques used for non-invasive diagnostic tests. Such techniques are routinely used to detect epilepsy, brain tumours, stroke, sleep disorders, etc. Many feature extraction techniques have been developed to process the EEG signals, allowing classification algorithms to predict the nature of the data. One of the most used feature extraction methods for EEG data is the wavelet transform which produces frequency domain features localised in time. Several families of wavelet transformation specialise in different types of signals. The Daubechies wavelet with db4 has been used as a forefront feature extraction method for epilepsy detection (Adeli et al., 2003). Other families of wavelet transform such as Daubechies wavelet with db2 have also been used (Güler & Übeyli, 2005). Different feature extraction techniques such as principal component analysis (PCA) and independent component analysis (ICA) have also been applied for EEG signals (Subasi & Gursoy, 2010). Once features are appropriately extracted, a classification algorithm is used to categorize the EEG signals.

The support vector machines (SVMs) (Cortes & Vapnik, 1995) and SVM-based algorithms, such as twin support vector machine (TWSVM) (Jayadeva & Khemchandani, 2007), are excellent supervised learning algorithms employed in diverse domains of classification and regression. They showcase reliable performance and have been used in several fields such as face recognition (Zhou et al., 2010), text categorisation (Wang & Chiang, 2007), electroencephalogram (EEG) signal classification (Richhariya & Tanveer, 2018), data mining (Bollegala et al., 2010), diagnosis of various diseases like epilepsy (Richhariya & Tanveer, 2018; Zhang et al., 2019) and Alzheimer's disease (Tanveer et al., 2020). SVM has gained this recognition because it reduces the generalisation error by maximising the margin between different classes. This is achieved by formulating a convex quadratic programming problem (QPP) to obtain exhaustive or global solution instead of settling for a local optimum, which many other methods such as artificial neural networks cannot overcome. Also, SVM implements the structural risk minimisation principle, which reduces the upper bound on the Vapnik–Chervonenkis (VC) dimension. This measures the capacity of functions that a statistical binary classification algorithm can learn.

Although SVM has been successful, however, SVM is not without disadvantages. These advantages come at the cost of solving a QPP with very high complexity,  $O(N^3)$ , where  $N$  is the total number of training data points. An attempt to solve this issue of complexity was carried out by Jayadeva and Khemchandani (2007) by introducing a novel twin support vector machine (TWSVM). TWSVM attempts to break up the large QPP into two sub-problems by seeking two non-parallel proximal hyperplanes. Further improvements were made by the introduction of twin bounded SVM (TBSVM) (Shao et al., 2011). TBSVM uses an additional regularization variable to carry out the structural risk minimisation principle. Kumar and Gopal made further improvements by introducing the least-squares twin SVM (LSTSV) (Kumar & Gopal, 2009). The LSTSV has a meagre computation time, however, it suffers in presence of noise and outliers as it uses a quadratic loss function. The angle-based twin support vector machine (ATWSVM) proposed by Khemchandani et al. (2018) is an iteration on TWSVM with the advantage of quick computation compared to TWSVM. This advantage arises in ATWSVM due to solving the second problem as an unconstrained minimization problem instead of a QPP. Tanveer et al. introduced a novel robust energy-based least squares twin SVM (RELSTSV) (Tanveer et al., 2016), which performed the best in

a 2019 evaluation (Tanveer et al., 2019) of SVM classifiers on UCI datasets (Dua & Graff, 2017). For further specifics of the TWSVM, we refer the interested readers to (Tanveer et al., 2021). Another approach for improving TWSVM is to utilize L1-norm distance than the traditional L2-norm distance. This approach was implemented in L1-TWSVM (Yan et al., 2019). In order to improve the sparsity and robustness in the original twin SVM, linear programming twin SVM algorithms (Tanveer, 2015a, b) are proposed. Fuzzy membership is another widely used method for improving generalization performance. Entropy-based fuzzy twin support vector machine (Gupta et al., 2019), Entropy-based fuzzy least-squares TWSVM (Chen et al., 2020), intuitionistic fuzzy TWSVM (Rezvani et al., 2019) are some recent examples of fuzzy implementation in TWSVM.

The aforementioned algorithms either improve SVM or TWSVM based models, however, the framework of universum data in SVM-based algorithms has also been proposed. The idea of using universum data to improve the generalisation performance of SVM was proposed by Weston et al. (2006). The idea of universum data stems from the fact that SVM does not have any information about the data distribution in the training set. That is to say, SVM does not have any prior information about the data in the training set. This prior information, which is a form of Bayesian prior, is incorporated in the optimisation problem by introducing universum data. The concept of universum data can be understood as follows: take an example of classifying images of handwritten numbers into their respective classes (i.e. corresponding numbers). The training data consists of the pixel intensities from the images in a vector form. Here, when SVM uses this training data, it does not have any context about the data space, i.e. handwritten characters. Thus, universum data set containing handwritten letters (not numbers) is used to provide prior information about the data space.

As mentioned above, the universum data need not be the same as the training set, it is an unlabelled set used to provide relevant prior information. Thus, universum based SVM (USVM) (Weston et al., 2006) algorithm implements this concept by introducing universum data points in an  $\epsilon$ -insensitive area between the two binary groups. Similar to SVM, USVM is further improved by universum based twin SVM (UTSVM) (Qi et al., 2012) and reduced UTSVM models (Richhariya & Tanveer, 2020). The computation cost of UTSVM is improved by the universum least squares twin SVM (ULSTSVM) (Xu et al., 2016) which uses the least squares technique, introducing the quadratic loss. The fuzzy-based universum least squares twin SVM (FULSTSVM) (Richhariya & Tanveer, 2021) utilises the fuzzy concept to further improve ULSTSVM. The Universum based Lagrangian twin bounded support vector machine (ULTBSVM) (Kumar & Gupta, 2021) uses the square of the 2-norm of the slack variables to formulate a strongly convex objective function, enabling the method to produce unique solutions. The ULTBSVM has additional regularization terms which implement the structural risk minimization (SRM) principle. The Regularized Universum twin support vector machine (Gupta et al., 2019) introduces regularization terms to the formulation of UTWSVM such that it is well-posed, removing the necessity to add another term to ensure that the matrix is positive definite. The iterative universum support vector machine (IUTWSVM) (Richhariya & Gupta, 2019) is different from other methods discussed above. It attempts to be less computationally expensive by utilizing an iterative based approach using the Newton method. This low computation cost enables IUTWSVM to be a viable multi-class classifier.

All the above algorithms are efficient classifiers based on several kinds of loss functions, such as hinge loss, square loss, etc. However, they lack a critical property, i.e. noise-insensitivity. Real-life data sets are inherently noisy due to being recorded in real world conditions especially EEG signals, where entropy and errors are unavoidable. This noise can arise from errors in the measuring equipment or even the user recording the observations. Thus, noise insensitivity is of critical importance for a classifier used for real-life applica-

tions. The noise performance (i.e. insensitivity to noise) for classifiers can be improved by utilising a specific loss function. The pinball loss introduced by Huang et al. (2013) is an excellent candidate for such a function. It boosts insensitivity towards noise and outliers, and it enables stability for re-sampling data. This insensitivity of pinball loss is a result of utilising the quantile distance (Koenker & Hallock, 2001; Christmann & Steinwart, 2007) rather than the shortest distance. SVM with pinball loss (Pin-SVM) (Huang et al., 2013) has shown promising performance in presence of noise. However, Pin-SVM has high computational complexity as it solves a large QPP. Thus, Xu et al. proposed the twin parametric margin support vector machine with pinball loss (Pin-TSVM) (Xu et al., 2016) to overcome the noise sensitivity of TWSVM and reduce the complexity of Pin-SVM. Some examples of the latest algorithms utilizing the pinball loss in 2021 are: Smooth twin bounded support vector machine with pinball loss (Li & Lv, 2021), the Bound estimation-based safe acceleration for maximum margin of twin spheres machine with pinball loss (Yuan & Xu, 2021) and the Robust general twin support vector machine with pinball loss function (Ganaie & Tanveer, 2021).

The aforementioned methods consider that the noise distribution is identical everywhere in the dataset, but that may not be true. Tanveer et al. proposed a novel general twin SVM with pinball loss (Pin-GTSVM) (Tanveer et al., 2019) to deal with this problem. The Pin-GTSVM computes two non-parallel hyperplanes where each hyperplane is proximal either to negative or positive class. To improve the sparsity of the models, sparse support vector machine with pinball loss (Pin-SSVM) (Tanveer et al., 2021) and sparse pinball twin support vector machines (SPTWSVM) (Tanveer et al., 2019) are formulated. To upgrade the performance of models in presence of large datasets, large-scale pinball TWSVM (Tanveer et al., 2021), large-scale twin parametric SVM with pinball loss function (Sharma et al., 2019) have been proposed.

Thus, motivated by the concept of universum data and the excellent performance of pinball loss in the various classifiers, in this work, we propose a novel universum twin support vector machine with pinball loss (Pin-UTSVM) for the EEG signal classification. The main highlights of this work are as follows:

1. The novel universum twin support vector machine with pinball loss (Pin-UTSVM) is proposed.
2. The proposed Pin-UTSVM model is robust to noise and more stable for resampling.
3. The computational complexity of the proposed Pin-UTSVM model is similar to standard UTSVM model. Hence, the proposed Pin-UTSVM brings noise insensitivity and stability without incurring the additional computational cost.
4. For handling the non-linear cases, we use kernel based Pin-UTSVM model for better generalization.
5. The evaluation of the classification models on real world EEG signal classification show that the proposed Pin-UTSVM has better generalization compared to the baseline models.

## 2 Related work

Suppose  $A$  and  $B$  be the sets of positive (+1) and negative (−1) class, respectively. Let  $c_i$  be the positive parameters and  $e_j$  be the vector of ones with the relevant dimensions, for  $i = 1, 2, 3, 4$  and  $j = \{+, -, u\}$ . Also,  $U$  denotes the universum data points and  $D = [A; B]$ .  $\|x\|$  is used to represent 2-norm of any vector  $x$ .

### 2.1 Twin SVM Jayadeva and Khemchandani (2007)

The twin support vector machine (TWSVM) (Jayadeva & Khemchandani, 2007) significantly improve the conventional SVM by improving upon its high computational complexity. SVM requires all data points as constraints while as in TWSVM the patterns of one class gives constraints to the other class. Thus, the TWSVM solves two smaller QPPs instead of a large QPP.

The non-linear TWSVM seeks to find the following two kernel generated surfaces:

$$K(x^t, D^t)u_+ + b_+ = 0 \text{ and } K(x^t, D^t)u_- + b_- = 0, \tag{1}$$

where  $K$  is a kernel function. The optimization problem for TWSVM can be denoted as follows:

$$\begin{aligned} \min_{u_+, b_+, \xi_+} \quad & \frac{1}{2} \|K(A, D^t)u_+ + e_+ b_+\|^2 + c_1 e_+^t \xi_+ \\ \text{s.t.} \quad & -(K(B, D^t)u_+ + e_- b_+) + \xi_+ \geq e_-, \xi_+ \geq 0 \end{aligned} \tag{2}$$

and

$$\begin{aligned} \min_{u_-, b_-, \xi_-} \quad & \frac{1}{2} \|K(B, D^t)u_- + e_- b_-\|^2 + c_2 e_-^t \xi_- \\ \text{s.t.} \quad & (K(A, D^t)u_- + e_+ b_-) + \xi_- \geq e_+, \xi_- \geq 0, \end{aligned} \tag{3}$$

where  $\xi_+, \xi_-$  are slack variables and  $e_+, e_-$  are vector of ones with appropriate dimensions. By using Lagrange multipliers  $\alpha \geq 0, \beta \geq 0$  and using the Karush-Kuhn-Tucker (K.K.T) conditions, the Wolfe dual of (2) and (3) come out to be:

$$\begin{aligned} \max_{\alpha} \quad & e_+^t \alpha - \frac{1}{2} \alpha^t Q(P^t P)^{-1} Q^t \alpha \\ \text{s.t.} \quad & 0 \leq \alpha \leq c_1 \end{aligned} \tag{4}$$

and

$$\begin{aligned} \max_{\beta} \quad & e_-^t \beta - \frac{1}{2} \beta^t P(Q^t Q)^{-1} P^t \beta \\ \text{s.t.} \quad & 0 \leq \beta \leq c_2, \end{aligned} \tag{5}$$

where  $P = [K(A, D^t) \ e_+]$  and  $Q = [K(B, D^t) \ e_-]$ .

After solving (4) and (5), the optimal separating hyperplanes are given by:

$$\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -(P^t P + \delta I)^{-1} Q^t \alpha, \tag{6}$$

$$\begin{bmatrix} u_- \\ b_- \end{bmatrix} = (Q^t Q + \delta I)^{-1} P^t \beta, \tag{7}$$

where  $\delta(\delta > 0)$  is the regularization variable used to circumvent the ill-conditioning of the matrices  $P^t P$  and  $Q^t Q$ . Further new data point  $x$  is assigned the classes using the following equation:

$$\text{class}(x) = \arg \min_{i=\{+,-\}} \frac{|K(x^t, D^t)u_i + b_i|}{\|u_i\|}. \tag{8}$$

### 2.2 Universum twin SVM Qi et al. (2012)

The universum twin support vector machine (UTSVM) (Qi et al., 2012) is an extended version of universum support vector machine (Weston et al., 2006), which improves upon the computational cost.

The UTSVM seeks the following non-linear hyperplanes:

$$K(x^t, D^t)u_+ + b_+ = 0 \text{ and } K(x^t, D^t)u_- + b_- = 0. \tag{9}$$

The optimisation problem for UTSVM can be expressed as follows:

$$\begin{aligned} \min_{u_+, b_+, \xi_+, \psi} \quad & \frac{1}{2} \|K(A, D^t)u_+ + e_+ b_+\|^2 + c_1 e_-^t \xi_+ + c_u e_u^t \psi \\ \text{s.t.} \quad & -(K(B, D^t)u_+ + e_- b_+) + \xi_+ \geq e_-, \quad \xi_+ \geq 0, \\ & (K(U, D^t)u_+ + e_u b_+) + \psi \geq (-1 + \varepsilon)e_u, \quad \psi \geq 0 \end{aligned} \tag{10}$$

and

$$\begin{aligned} \min_{u_-, b_-, \xi_-, \psi} \quad & \frac{1}{2} \|K(B, D^t)u_- + e_- b_-\|^2 + c_2 e_+^t \xi_- + c_u e_u^t \psi \\ \text{s.t.} \quad & (K(A, D^t)u_- + e_+ b_-) + \xi_- \geq e_+, \quad \xi_- \geq 0, \\ & -(K(U, D^t)u_- + e_u b_-) + \psi \geq (-1 + \varepsilon)e_u, \quad \psi \geq 0, \end{aligned} \tag{11}$$

where  $\xi_+, \xi_-, \psi$  are the slack variables,  $\varepsilon$  is the tolerance variable for the universum. Using the Lagrangian multipliers  $\alpha_1, \alpha_2, \mu_1, \mu_2$  and appropriate K.K.T. conditions, the dual of (10) and (11) can be expressed as follows:

$$\begin{aligned} \max_{\alpha_1, \mu_1} \quad & e_-^t \alpha_1 - \frac{1}{2} (\alpha_1^t T - \mu_1^t O) (S^t S)^{-1} (T \alpha_1^t - O \mu_1^t) + (\varepsilon - 1) e_u^t \mu_1 \\ \text{s.t.} \quad & 0 \leq \alpha_1 \leq c_1, 0 \leq \mu_1 \leq c_u \end{aligned} \tag{12}$$

and

$$\begin{aligned} \max_{\alpha_2, \mu_2} \quad & e_+^t \alpha_2 - \frac{1}{2} (\alpha_2^t S - \mu_2^t O) (T^t T)^{-1} (S^t \alpha_2 - O^t \mu_2) + (\varepsilon - 1) e_u^t \mu_2 \\ \text{s.t.} \quad & 0 \leq \alpha_2 \leq c_2, 0 \leq \mu_2 \leq c_u, \end{aligned} \tag{13}$$

where  $O = [K(U, D^t) e_u], S = [K(A, D^t) e_+]$  and  $T = [K(B, D^t) e_-]$ .

After solving (12) and (13), the optimal separating hyperplanes are given by:

$$\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -(S^t S + \delta I)^{-1} (T^t \alpha_1 - O^t \mu_1), \tag{14}$$

$$\begin{bmatrix} u_- \\ b_- \end{bmatrix} = (T^t T + \delta I)^{-1} (S^t \alpha_2 - O^t \mu_2), \tag{15}$$

where  $\delta I$  is the regularization variable added to circumvent ill-conditioning of  $S^t S$  and  $T^t T$  matrices.

The new data point  $x$  is assigned its class using the following function:

$$\text{class}(x) = \arg \min_{i=\{+,-\}} \frac{|K(x^t, D^t)u_i + b_i|}{\|u_i\|}. \tag{16}$$

### 2.3 Pin-GTSVM Tanveer et al. (2019)

The general twin support vector machine with pinball loss (Pin-GTSVM) (Tanveer et al., 2019) seeks the following hyperplanes:

$$K(x^t, D^t)u_+ + b_+ = 0 \text{ and } K(x^t, D^t)u_- + b_- = 0, \tag{17}$$

where  $K$  is the kernel function.

The optimisation problem for Pin-GTSVM can be expressed as follows:

$$\begin{aligned} \min_{u_+, b_+, \xi_+} \quad & \frac{1}{2} \|K(A, D^t)u_+ + e_+ b_+\|^2 + c_1 e^t \xi_+ \\ \text{s.t.} \quad & -(K(B, D^t)u_+ + e_- b_+) + \xi_+ \geq e_-, \\ & -(K(B, D^t)u_+ + e_- b_+) - \frac{\xi_+}{\tau_2} \leq e_- \end{aligned} \tag{18}$$

and

$$\begin{aligned} \min_{u_-, b_-, \xi_-} \quad & \frac{1}{2} \|K(B, D^t)u_- + e_- b_-\|^2 + c_2 e^t \xi_- \\ \text{s.t.} \quad & K(A, D^t)u_- + e_+ b_- + \xi_- \geq e_+, \\ & K(A, D^t)u_- + e_+ b_- - \frac{\xi_-}{\tau_1} \leq e_+, \end{aligned} \tag{19}$$

where  $\xi_+, \xi_-$  are slack variables. Using the Lagrangian multipliers  $\alpha, \beta, \gamma, \sigma \geq 0$  and employing the appropriate K.K.T. conditions, the dual of (18) and (19) can be formulated as follows:

$$\begin{aligned} \max_{\alpha-\beta} \quad & e^t_-(\alpha - \beta) - \frac{1}{2}(\alpha - \beta)^t Q(P^t P)^{-1} Q^t(\alpha - \beta) \\ \text{s.t.} \quad & -\tau_2 c_1 e_- \leq (\alpha - \beta) \end{aligned} \tag{20}$$

and

$$\begin{aligned} \max_{(\gamma-\sigma)} \quad & e^t_+(\gamma - \sigma) - \frac{1}{2}(\gamma - \sigma)^t P(Q^t Q)^{-1} P^t(\gamma - \sigma) \\ \text{s.t.} \quad & -\tau_1 c_2 e_+ \leq (\gamma - \sigma), \end{aligned} \tag{21}$$

where  $P = [K(A, D^t) \ e_+]$  and  $Q = [K(B, D^t) \ e_-]$ .

After solving (20) and (21), the optimal separating hyperplanes are given by:

$$\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -(P^t P + \delta I)^{-1} Q^t(\alpha - \beta), \tag{22}$$

$$\begin{bmatrix} u_- \\ b_- \end{bmatrix} = (Q^t Q + \delta I)^{-1} P^t(\gamma - \sigma), \tag{23}$$

where  $\delta I (\delta > 0)$  is the regularization variable used to circumvent this ill-conditioning of  $P^t P$  and  $Q^t Q$  matrices.

New data point  $x$  is assigned the class using the following equation:

$$\text{class}(x) = \arg \min_{i=\{+,-\}} \frac{|K(x^t, D^t)u_i + b_i|}{\|u_i\|}. \tag{24}$$

### 3 Proposed universum twin SVM with pinball loss

Universum TWSVM (Qi et al., 2012) is based on the hinge loss, hence, suffers due to issues of noise and is unstable under resampling. Taking inspiration from Pin-TSVM (Tanveer et al., 2019), we include pinball loss in universum twin SVM and propose a novel universum twin SVM with pinball loss function (Pin-UTSVM). The proposed Pin-UTSVM overcome the issues of noise and make it more stable for resampling compared to the standard universum TWSVM model.

#### 3.1 Linear universum twin SVM with pinball loss

The optimization problems for the linear variant of the proposed universum twin SVM with pinball loss (Pin-UTSVM) are given as:

$$\begin{aligned}
 \min_{u_+, b_+, \xi_+, \psi} \quad & \frac{1}{2} \|Au_+ + e_+b_+\|^2 + c_1e'_-\xi_+ + c_2e'_u\psi \\
 \text{s.t.} \quad & -(Bu_+ + e_-b_+) \geq e_- - \xi_+, \\
 & -(Bu_+ + e_-b_+) \leq e_- + \frac{1}{\tau_1}\xi_+, \\
 & Uu_+ + e_ub_+ \geq (-1 + \epsilon)e_u - \psi, \\
 & Uu_+ + e_ub_+ \leq (-1 + \epsilon)e_u + \frac{1}{\tau_2}\psi,
 \end{aligned} \tag{25}$$

and

$$\begin{aligned}
 \min_{u_-, b_-, \xi_-, \psi} \quad & \frac{1}{2} \|Bu_- + e_-b_-\|^2 + c_3e'_+\xi_- + c_4e'_u\psi \\
 \text{s.t.} \quad & Au_- + e_+b_- \geq e_+ - \xi_-, \\
 & Au_- + e_+b_- \leq e_+ + \frac{1}{\tau_3}\xi_-, \\
 & -(Uu_- + e_ub_-) \geq (-1 + \epsilon)e_u - \psi, \\
 & -(Uu_- + e_ub_-) \leq (-1 + \epsilon)e_u + \frac{1}{\tau_4}\psi,
 \end{aligned} \tag{26}$$

where  $\xi_i, \psi$  are the slack variables,  $c_j$  and  $\tau_j$  are positive parameters and  $\epsilon$  is the hyperparameter, here  $i = +, -$  and  $j = 1, 2, 3, 4$ .

Note that when  $\tau_1, \tau_2$  and  $\tau_3, \tau_4$  tend to zero, then the second and fourth constraint of the (25) reduce to  $\xi_+ \geq 0$  and  $\psi \geq 0$  and similarly for the (26). Under these conditions, the proposed Pin-UTSVM reduces to the standard UTSVM. Hence, standard UTSVM is a special case of the proposed Pin-UTSVM model.

To obtain the solution of (25) and (26), we derive their Wolfe dual. The Lagrangian of the optimization problem (25) is given as

$$\begin{aligned}
 L = & \frac{1}{2} \|Au_+ + e_+b_+\|^2 + c_1e'_-\xi_+ + c_2e'_u\psi + \alpha'_1(Bu_+ + e_-b_+ + e_- - \xi_+) \\
 & - \alpha'_2(Bu_+ + e_-b_+ + e_- + \frac{1}{\tau_1}\xi_+) - \beta'_1(Uu_+ + e_ub_+ - (-1 + \epsilon)e_u + \psi) \\
 & + \beta'_2(Uu_+ + e_ub_+ - (-1 + \epsilon)e_u - \frac{1}{\tau_2}\psi).
 \end{aligned} \tag{27}$$



Applying K.K.T. conditions on (27), we have

$$\frac{\partial L}{\partial u_+} = A^t(Au_+ + e_+b_+) + B^t(\alpha_1 - \alpha_2) - U^t(\beta_1 - \beta_2) = 0, \tag{28}$$

$$\frac{\partial L}{\partial b_+} = e_+^t(Au_+ + e_+b_+) + e_-^t(\alpha_1 - \alpha_2) - e_u^t(\beta_1 - \beta_2) = 0, \tag{29}$$

$$\frac{\partial L}{\partial \xi_+} = c_1e_- - \alpha_1 - \frac{1}{\tau_1}\alpha_2 = 0, \tag{30}$$

$$\frac{\partial L}{\partial \psi} = c_2e_u - \beta_1 - \frac{1}{\tau_2}\beta_2 = 0, \tag{31}$$

$$\alpha_1^t(Bu_+ + e_-b_+ + e_- - \xi_+) = 0, \tag{32}$$

$$\alpha_2^t(Bu_+ + e_-b_+ + e_- + \frac{1}{\tau_1}\xi_+) = 0, \tag{33}$$

$$\beta_1^t(Uu_+ + e_ub_+ - (-1 + \epsilon)e_u + \psi) = 0, \tag{34}$$

$$\beta_2^t(Uu_+ + e_ub_+ - (-1 + \epsilon)e_u - \frac{1}{\tau_2}\psi) = 0. \tag{35}$$

Using (30) and  $\alpha_1 \geq 0, \alpha_2 \geq 0$ , we get

$$-\tau_1c_1e_- \leq (\alpha_1 - \alpha_2) \leq c_1e_-. \tag{36}$$

Similarly, using (31) and  $\beta_1 \geq 0, \beta_2 \geq 0$ , we get

$$-\tau_2c_2e_u \leq (\beta_1 - \beta_2) \leq c_2e_u. \tag{37}$$

Let  $X_1 = [A, e_+], X_2 = [B, e_-], X_3 = [U, e_u]$ .

Rewriting (28) and (29), we have

$$\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -(X_1^t X_1)^{-1} \left( X_2^t(\alpha_1 - \alpha_2) - X_3^t(\beta_1 - \beta_2) \right). \tag{38}$$

Suppose  $\alpha = (\alpha_1 - \alpha_2), \beta = (\beta_1 - \beta_2), \gamma = [\alpha; \beta], N = [X_2; -X_3]$ , and  $e_4 = [e_-; (-1 + \epsilon)e_u]$ , then using K.K.T. conditions the Wolfe dual of (25) is given as

$$\begin{aligned} \max_{\gamma} \quad & -\frac{1}{2}\gamma^t N(X_1^t X_1)^{-1} N^t \gamma + e_4^t \gamma \\ \text{s.t.} \quad & \begin{bmatrix} -\tau_1c_1e_- \\ -\tau_2c_2e_u \end{bmatrix} \leq \gamma \leq \begin{bmatrix} c_1e_- \\ c_2e_u \end{bmatrix}. \end{aligned} \tag{39}$$

In the similar manner, the Wolfe dual of (26) is given as follows

$$\begin{aligned} \max_{\theta} \quad & -\frac{1}{2}\theta^t P(X_2^t X_2)^{-1} P^t \theta + e_5^t \theta \\ \text{s.t.} \quad & \begin{bmatrix} -\tau_3c_3e_+ \\ -\tau_4c_4e_u \end{bmatrix} \leq \theta \leq \begin{bmatrix} c_3e_+ \\ c_4e_u \end{bmatrix}, \end{aligned} \tag{40}$$

where  $\theta = [\eta; \zeta], \eta = (\eta_1 - \eta_2), \zeta = (\zeta_1 - \zeta_2), P = [X_1; -X_3], e_5 = [e_+; (-1 + \epsilon)e_u]$ .

The optimal hyperplane corresponding to another class is given as

$$\begin{bmatrix} u_- \\ b_- \end{bmatrix} = (X_2^t X_2)^{-1} (X_1^t \eta - X_3^t \zeta). \tag{41}$$

The testing data sample  $x \in \mathbb{R}^n$  is assigned the class as follows

$$Class(x) = arg \min_{i=\{+,-\}} \frac{|u_i^t x + b_i|}{\|u_i\|}. \tag{42}$$

### 3.2 Non-linear universum twin SVM with pinball loss

The optimization problems of the proposed universum universum twin SVM with pinball loss (Pin-UTSVM) for non-linear case is given as

$$\begin{aligned} \min_{u_+, b_+, \xi_+, \psi} \quad & \frac{1}{2} \|K(A, D^t)u_+ + e_+ b_+\|^2 + c_1 e_-^t \xi_+ + c_2 e_u^t \psi \\ \text{s.t.} \quad & -(K(B, D^t)u_+ + e_- b_+) \geq e_- - \xi_+, \\ & -(K(B, D^t)u_+ + e_- b_+) \leq e_- + \frac{1}{\tau_1} \xi_+, \\ & K(U, D^t)u_+ + e_u b_+ \geq (-1 + \epsilon)e_u - \psi, \\ & K(U, D^t)u_+ + e_u b_+ \leq (-1 + \epsilon)e_u + \frac{1}{\tau_2} \psi \end{aligned} \tag{43}$$

and

$$\begin{aligned} \min_{u_-, b_-, \xi_-, \psi} \quad & \frac{1}{2} \|K(B, D^t)u_- + e_- b_-\|^2 + c_3 e_+^t \xi_- + c_4 e_u^t \psi \\ \text{s.t.} \quad & K(A, D^t)u_- + e_+ b_- \geq e_+ - \xi_-, \\ & K(A, D^t)u_- + e_+ b_- \leq e_+ + \frac{1}{\tau_3} \xi_-, \\ & -(K(U, D^t)u_- + e_u b_-) \geq (-1 + \epsilon)e_u - \psi, \\ & -(K(U, D^t)u_- + e_u b_-) \leq (-1 + \epsilon)e_u + \frac{1}{\tau_4} \psi. \end{aligned} \tag{44}$$

To obtain the solution of (43) and (44), we derive their Wolfe dual. The Lagrangian of the optimization problem (43) is given as

$$\begin{aligned} L = & \frac{1}{2} \|K(A, D^t)u_+ + e_+ b_+\|^2 + c_1 e_-^t \xi_+ + c_2 e_u^t \psi + \alpha_1^t (K(B, D^t)u_+ + e_- b_+ + e_- - \xi_+) \\ & - \alpha_2^t (K(B, D^t)u_+ + e_- b_+ + e_- + \frac{1}{\tau_1} \xi_+) - \beta_1^t (K(U, D^t)u_+ \\ & + e_u b_+ - (-1 + \epsilon)e_u + \psi) + \beta_2^t (K(U, D^t)u_+ + e_u b_+ - (-1 + \epsilon)e_u - \frac{1}{\tau_2} \psi). \end{aligned} \tag{45}$$

Applying K.K.T. conditions on (45), we have

$$\begin{aligned} \frac{\partial L}{\partial u_+} = & (K(A, D^t))^t (K(A, D^t)u_+ + e_+ b_+) \\ & + (K(B, D^t))^t (\alpha_1 - \alpha_2) - (K(U, D^t))^t (\beta_1 - \beta_2) = 0, \end{aligned} \tag{46}$$

$$\frac{\partial L}{\partial b_+} = e_+^t (K(A, D^t)u_+ + e_+ b_+) + e_-^t (\alpha_1 - \alpha_2) - e_u^t (\beta_1 - \beta_2) = 0, \tag{47}$$

$$\frac{\partial L}{\partial \xi_+} = c_1 e_- - \alpha_1 - \frac{1}{\tau_1} \alpha_2 = 0, \tag{48}$$

$$\frac{\partial L}{\partial \psi} = c_2 e_u - \beta_1 - \frac{1}{\tau_2} \beta_2 = 0, \tag{49}$$

$$\alpha_1^t (K(B, D^t) u_+ + e_- b_+ + e_- - \xi_+) = 0, \tag{50}$$

$$\alpha_2^t (K(B, D^t) u_+ + e_- b_+ + e_- + \frac{1}{\tau_1} \xi_+) = 0, \tag{51}$$

$$\beta_1^t (K(U, D^t) u_+ + e_u b_+ - (-1 + \epsilon) e_u + \psi) = 0, \tag{52}$$

$$\beta_2^t (K(U, D^t) u_+ + e_u b_+ - (-1 + \epsilon) e_u - \frac{1}{\tau_2} \psi) = 0. \tag{53}$$

Using (48) and  $\alpha_1 \geq 0, \alpha_2 \geq 0$ , we get

$$-\tau_1 c_1 e_- \leq (\alpha_1 - \alpha_2) \leq c_1 e_-. \tag{54}$$

Similarly, using (49) and  $\beta_1 \geq 0, \beta_2 \geq 0$ , we get

$$-\tau_2 c_2 e_u \leq (\beta_1 - \beta_2) \leq c_2 e_u. \tag{55}$$

Let  $X_1 = [K(A, D^t), e_+], X_2 = [K(B, D^t), e_-], X_3 = [K(U, D^t), e_u]$ .  
 Rewriting (46) and (47), we have

$$\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -(X_1^t X_1)^{-1} (X_2^t (\alpha_1 - \alpha_2) - X_3^t (\beta_1 - \beta_2)). \tag{56}$$

Suppose  $\alpha = (\alpha_1 - \alpha_2), \beta = (\beta_1 - \beta_2), \gamma = [\alpha; \beta], N = [X_2; -X_3]$ , and  $e_4 = [e_-; (-1 + \epsilon) e_u]$ , then using K.K.T. conditions the Wolfe dual of (43) is given as

$$\begin{aligned} \max_{\gamma} \quad & -\frac{1}{2} \gamma^t N (X_1^t X_1)^{-1} N^t \gamma + e_4^t \gamma \\ \text{s.t.} \quad & \begin{bmatrix} -\tau_1 c_1 e_- \\ -\tau_2 c_2 e_u \end{bmatrix} \leq \gamma \leq \begin{bmatrix} c_1 e_- \\ c_2 e_u \end{bmatrix}. \end{aligned} \tag{57}$$

In the similar manner, the Wolfe dual of (44) is given as follows

$$\begin{aligned} \max_{\theta} \quad & -\frac{1}{2} \theta^t P (X_2^t X_2)^{-1} P^t \theta + e_5^t \theta \\ \text{s.t.} \quad & \begin{bmatrix} -\tau_3 c_3 e_+ \\ -\tau_4 c_4 e_u \end{bmatrix} \leq \theta \leq \begin{bmatrix} c_3 e_+ \\ c_4 e_u \end{bmatrix}, \end{aligned} \tag{58}$$

where  $\theta = [\eta; \zeta], \eta = (\eta_1 - \eta_2), \zeta = (\zeta_1 - \zeta_2), P = [X_1; -X_3], e_5 = [e_+; (-1 + \epsilon) e_u]$ .

The optimal hyperplane corresponding to another class is given as

$$\begin{bmatrix} u_- \\ b_- \end{bmatrix} = (X_2^t X_2)^{-1} (X_1^t \eta - X_3^t \zeta). \tag{59}$$

The testing data sample  $x \in \mathbb{R}^n$  is assigned the class as follows

$$\text{class}(x) = \arg \min_{i=\{+,-\}} \frac{|K(x^t, D^t) u_i + b_i|}{\|u_i\|}. \tag{60}$$

The algorithm for Pin-UTSVM is briefly described in Algorithm 1.

**Algorithm 1:** Algorithm for Pin-UTSVM.**Input** :  $A, B, U$  and hyperparameters  $c_i, \tau_i, \epsilon | i = \{1, 2, 3, 4\}$ **Output:** The weight vectors and bias i.e.,  $u_i, b_i | i = \{+, -\}$  for positive and negative class, respectively.

1. Construct matrices  $X_1, X_2$  and  $X_3$  from matrices  $A, B$  and  $C$  for linear/non-linear case.
2. Solve for the Lagrangian Multipliers by formulating the two QPPs as described in Sect. 3 using matrices  $X_1, X_2$  and  $X_3$ , and the hyperparameters.
3. Calculate  $u_i, b_i | i = \{+, -\}$  using the above determined Lagrangian Multipliers.
4. Return  $u_i, b_i | i = \{+, -\}$  for the construction of hyperplanes of the two classes.

### 3.3 Computational complexity

Let  $m_1, m_2$  and  $m_u$  be the samples of positive, negative and universum data, with  $N = m_1 + m_2$ . Then, the computational complexity of SVM is  $T = O(N^3)$ , where  $N$  is the dataset size. Since, TWSVM solves two smaller size QPPs and hence, its complexity is  $T = O(m_1^3) + O(m_2^3)$ . The comparison of the Wolfe dual of the proposed Pin-UTSVM and standard UTSVM model reveals that both have same number of constraints and the order of the matrices where inversion is involved is same. Thus, the complexity of proposed Pin-UTSVM is  $T = O((m_1 + m_u)^3) + O((m_2 + m_u)^3)$  which is same as that of the standard UTSVM model.

### 3.4 Comparison of proposed Pin-UTSVM with the baseline models

The proposed Pin-UTSVM is different from the TWSVM, Pin-GTSVM and UTSVM as follows:

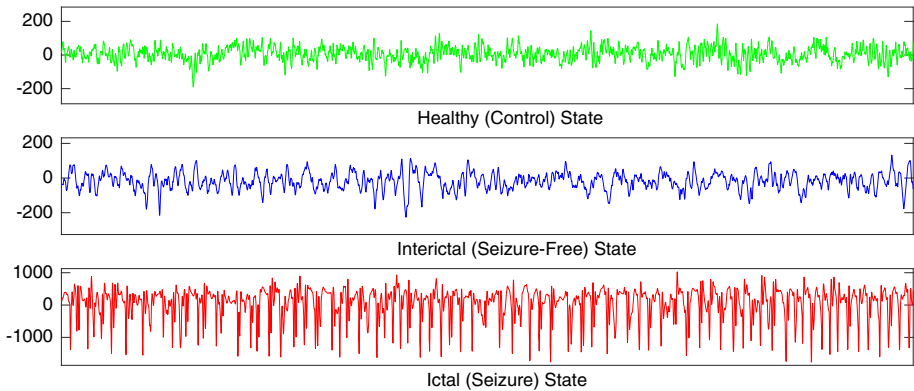
- TWSVM model uses hinge loss function while as the proposed Pin-UTSVM models uses pinball loss function to penalize the errors. Moreover, the proposed Pin-UTSVM uses universum data to improve the performance which is ignored by the standard TWSVM model.
- Pin-UTSVM model is different from the Pin-GTSVM model as the former uses universum concept while as the later ignores it.
- UTSVM model uses hinge loss function to penalize the errors while as the proposed Pin-UTSVM model uses pinball loss function to penalize the errors.

## 4 Experiments

This section discusses the experimental setup followed for evaluating the performance of the classification models. Moreover, we also discuss the data acquisition and the preprocessing followed for the extraction of the features. We analyse the performance of the models statistically, and evaluated the effect of different hyperparameters on the performance of classification models.

### 4.1 Dataset acquisition, pre-processing and experimental setup

This section details the experiments performed and the results obtained for classifying EEG Signals. The dataset used is taken from a journal article by Andrzejak et al. (2001). This dataset consisted of five collections (denoted S, F, N, O and Z), each consisting of hundred EEG



**Fig. 1** Sample EEG signals for **a** Healthy (control), **b** Interictal (seizure-free) and **c** Ictal (seizure) state

signals (single-channel) which were sampled at 173.61 Hz for 23.6 seconds. The collections O and Z represent healthy (control) subjects with eyes close and open, respectively. The collections F and N represent subjects in the interictal (seizure-free) state, and the collection S represents subjects in the ictal (seizure) state. The mode of EEG recording is intra-cranial for sets S, F and N, and all the EEG signals were processed using the same 128-channel amplifier system with an average common reference.

In the numerical experiments, ten-fold cross-validation is used. The tests were designed for binary classification, with the two classes being the interictal and the ictal state. Thus, Z vs S and O vs S, as both Z and O represent healthy subjects. The universum data is used to provide prior information about the training dataset. Thus, following the paper by Richhariya and Tanveer (2018), the set N, which consists of interictal state subjects, is used as universum data. Various methods of feature extraction, i.e. independent component analysis (ICA), principal component analysis (PCA) and wavelet transform were used. Several families of wavelet transform with different levels of decomposition were used. The Daubechies wavelet- db2, db4 and haar wavelet were set at level three, and db2 and db6 were set at level two. In the case of wavelet transform and ICA, PCA is applied first to reduce the number of dimensions. The class discriminatory ratio (CDR) was used to organise the PCA components and choose the most applicable components. The implementation of ICA used is the ICA architecture (Bartlett et al., 2002). The proposed Pin-UTSVM effectiveness was tested against Pin-GTSVM (Tanveer et al., 2019), TWSVM (Jayadeva & Khemchandani, 2007) and UTSVM (Qi et al., 2012). The experiments were performed on both the linear and non-linear variants of all the methods. For the non-linear variants, the Gaussian kernel, given by equation (61), was used. The Gaussian kernel parameter  $\mu$  is calculated using equation (62) (Tsang et al., 2006). The values and range of the various hyper-parameters used in different methods have been catalogued in Table 1. Additionally, to check the noise performance of the algorithms in presence of noise, we added Gaussian noise with zero-mean and standard deviation  $\sigma = 0$  (i.e. no noise), 0.05, 0.075, 0.1.

$$K(a, b) = e^{-\frac{1}{2\mu^2} \|a-b\|^2}, \quad (61)$$

$$\mu = \frac{1}{N^2} \sum_{i,j=1}^N \|x_i - x_j\|^2, \quad (62)$$

**Table 1** Parameter ranges for various methods

Parameter name	Symbol	Range/Value
Penalty parameters	$c_1, c_2, c_3, c_4$	$\{10^i   i = -5, -3, -1, 1, 3, 5\}$
Pinball loss parameter	$\tau_1, \tau_2, \tau_3, \tau_4$	$\{0.5, 0.6, 0.7, 0.8\}$
	$\epsilon$	$\{0.1, 0.2, 0.3, 0.5, 0.6, 0.7\}$
	$\delta$	$10^{-4}$

where  $N$  is the total number of data points and  $x_i$  represents  $i^{th}$  data point. The hyperparameters for UTSVM were set as  $c = c_1 = c_2, c_u$ . The hyperparameters for Pin-UTSVM were set as  $c_1 = c_3, c_2 = c_4$  and  $\tau_1 = \tau_3, \tau_2 = \tau_4$ .

All the computations were performed on a High-Performance PC running Windows 7 OS with Intel® Xeon® CPU E5-2697 v4 @ 2.30 GHz and 128 GB RAM. The program used for coding was MATLAB® R2017a.

## 4.2 Results

Here, we discuss the performance of the standard models and the proposed Pin-UTSVM model for classification of EEG signal with different feature extraction techniques. The performance of the models with linear and non-linear cases are given as follows:

### 4.2.1 Evaluation of the models with linear kernel

The results of the experiments corresponding to the linear kernel are presented in Table 2. The optimal parameters obtained for linear kernel are available in Table 3. The Table 2 reports the average accuracy obtained after the cross-validation at the optimal parameters. The optimal parameters are obtained by using the grid search method. The table reports the average accuracy and average rank of the methods. One can note that the proposed Pin-UTSVM model performs better as it achieves lowest average rank of 1. 1 is the lowest rank possible, thus establishing that the proposed method has the best performance in the linear case. The best accuracy in Table 2 for a no-noise dataset is 0.89, obtained by the proposed Pin-UTSVM using the wavelet (Haar) feature extraction for the O&S dataset. Additionally, our proposed method obtains a substantially better average accuracy attaining around 5% better than the preceding method, UTSVM. TWSVM and UTSVM both perform similarly, achieving 2.625 and 2.375 average rank, respectively. The Pin-GTSVM is the worst performing method.

Specifically evaluating the Z&S dataset for the linear case, we observe that the wavelet (Haar) and wavelet (db2) feature extraction methods perform the best for the proposed Pin-UTSVM method. In contrast, ICA performs the worst. As for the O&S dataset, the same observations as Z&S are made with regards to the proposed Pin-UTSVM. The ICA feature extraction performs the worst for all of the methods in the linear case.

### 4.2.2 Evaluation of the models with Gaussian kernel

The results of the experiments corresponding to the Gaussian kernel are presented in Table 4. The optimal parameters obtained for gaussian kernel are available in Table 5. The Table 4 is arranged similar to the linear case, i.e., the table reports the average accuracy obtained

**Table 2** Accuracies obtained for different methods using linear kernel on EEG data with four noise levels

Dataset	Feature extraction method	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007) Acc $\pm$ STD Time (s)	Pin-GTSVM (Tanveer et al., 2019) Acc $\pm$ STD Time (s)	UTSVM (Qi et al., 2012) Acc $\pm$ STD Time (s)	Pin-UTSVM Acc $\pm$ STD Time (s)
O & S	Wavelet (db1)	0	0.815 $\pm$ 0.075 0.0067703	0.685 $\pm$ 0.094 0.0074847	0.815 $\pm$ 0.075 0.0166475	0.865 $\pm$ 0.058 0.0185722
		0.05	0.81 $\pm$ 0.077 0.0069548	0.69 $\pm$ 0.099 0.0073136	0.81 $\pm$ 0.077 0.0157978	0.87 $\pm$ 0.048 0.0177221
		0.075	0.825 $\pm$ 0.072 0.0067478	0.69 $\pm$ 0.088 0.0073899	0.825 $\pm$ 0.072 0.0165016	0.87 $\pm$ 0.048 0.0187166
		0.1	0.825 $\pm$ 0.072 0.0071503	0.695 $\pm$ 0.119 0.0073974	0.825 $\pm$ 0.072 0.0170084	0.88 $\pm$ 0.048 0.0185133
		0	0.82 $\pm$ 0.071 0.0067815	0.71 $\pm$ 0.107 0.0074163	0.82 $\pm$ 0.071 0.0198211	0.865 $\pm$ 0.082 0.0176953
		0.05	0.815 $\pm$ 0.075 0.0071352	0.71 $\pm$ 0.097 0.0080099	0.82 $\pm$ 0.116 0.0165155	0.865 $\pm$ 0.067 0.0177791
Z & S	Wavelet (db1)	0.075	0.825 $\pm$ 0.086 0.0072259	0.715 $\pm$ 0.106 0.0074337	0.825 $\pm$ 0.086 0.0167706	0.87 $\pm$ 0.063 0.0179893
		0.1	0.825 $\pm$ 0.063 0.0068943	0.71 $\pm$ 0.107 0.0073888	0.825 $\pm$ 0.111 0.0160683	0.86 $\pm$ 0.07 0.018068
		0	0.815 $\pm$ 0.078 0.0070884	0.705 $\pm$ 0.096 0.0074291	0.815 $\pm$ 0.078 0.0165969	0.885 $\pm$ 0.058 0.0184315
		0.05	0.815 $\pm$ 0.082 0.0067943	0.705 $\pm$ 0.096 0.0073718	0.815 $\pm$ 0.082 0.016343	0.885 $\pm$ 0.058 0.018418
		0.075	0.815 $\pm$ 0.071 0.0069006	0.7 $\pm$ 0.094 0.0074294	0.815 $\pm$ 0.071 0.0178668	0.89 $\pm$ 0.057 0.0191544
		0.1	0.815 $\pm$ 0.078 0.0067256	0.7 $\pm$ 0.103 0.0075071	0.815 $\pm$ 0.071 0.017623	0.875 $\pm$ 0.072 0.0189841
O & S	Wavelet (db2)	0	0.815 $\pm$ 0.075 0.0067703	0.685 $\pm$ 0.094 0.0074847	0.815 $\pm$ 0.075 0.0166475	0.865 $\pm$ 0.058 0.0185722
		0.05	0.81 $\pm$ 0.077 0.0069548	0.69 $\pm$ 0.099 0.0073136	0.81 $\pm$ 0.077 0.0157978	0.87 $\pm$ 0.048 0.0177221
		0.075	0.825 $\pm$ 0.072 0.0067478	0.69 $\pm$ 0.088 0.0073899	0.825 $\pm$ 0.072 0.0165016	0.87 $\pm$ 0.048 0.0187166
		0.1	0.825 $\pm$ 0.072 0.0071503	0.695 $\pm$ 0.119 0.0073974	0.825 $\pm$ 0.072 0.0170084	0.88 $\pm$ 0.048 0.0185133
		0	0.82 $\pm$ 0.071 0.0067815	0.71 $\pm$ 0.107 0.0074163	0.82 $\pm$ 0.071 0.0198211	0.865 $\pm$ 0.082 0.0176953
		0.05	0.815 $\pm$ 0.075 0.0071352	0.71 $\pm$ 0.097 0.0080099	0.82 $\pm$ 0.116 0.0165155	0.865 $\pm$ 0.067 0.0177791

**Table 2** continued

Dataset	Feature extraction method	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007) Acc $\pm$ STD Time (s)	Pin-GTSVM (Tanveer et al., 2019) Acc $\pm$ STD Time (s)	UTSVM (Qi et al., 2012) Acc $\pm$ STD Time (s)	Pin-UTSVM Acc $\pm$ STD Time (s)
Z & S	Wavelet (db2)	0	0.855 $\pm$ 0.06 0.0070488	0.705 $\pm$ 0.08 0.0076242	0.86 $\pm$ 0.061 0.017376	0.875 $\pm$ 0.063 0.0164364
		0.05	0.855 $\pm$ 0.06 0.0067716	0.7 $\pm$ 0.088 0.007404	0.855 $\pm$ 0.069 0.018307	0.875 $\pm$ 0.063 0.0163427
		0.075	0.85 $\pm$ 0.058 0.0068471	0.71 $\pm$ 0.081 0.0075633	0.855 $\pm$ 0.06 0.0171498	0.875 $\pm$ 0.042 0.0159136
		0.1	0.85 $\pm$ 0.058 0.0068984	0.71 $\pm$ 0.077 0.0074707	0.85 $\pm$ 0.058 0.0178303	0.88 $\pm$ 0.042 0.0161949
		0	0.835 $\pm$ 0.082 0.0067375	0.68 $\pm$ 0.089 0.0074533	0.855 $\pm$ 0.064 0.0175906	0.88 $\pm$ 0.095 0.0199954
		0.05	0.835 $\pm$ 0.067 0.0067624	0.68 $\pm$ 0.071 0.0074604	0.855 $\pm$ 0.072 0.0184745	0.875 $\pm$ 0.082 0.0205307
O & S	Wavelet (db6)	0.075	0.84 $\pm$ 0.094 0.007303	0.675 $\pm$ 0.075 0.007466	0.845 $\pm$ 0.055 0.0173767	0.875 $\pm$ 0.082 0.0207107
		0.1	0.84 $\pm$ 0.094 0.0065982	0.68 $\pm$ 0.075 0.0074359	0.845 $\pm$ 0.086 0.0180879	0.875 $\pm$ 0.086 0.0187949
		0	0.85 $\pm$ 0.053 0.0075253	0.705 $\pm$ 0.086 0.0074273	0.85 $\pm$ 0.053 0.0234509	0.87 $\pm$ 0.048 0.0240283
		0.05	0.85 $\pm$ 0.047 0.007203	0.705 $\pm$ 0.101 0.0079486	0.85 $\pm$ 0.047 0.0170526	0.87 $\pm$ 0.048 0.0230243
		0.075	0.84 $\pm$ 0.061 0.0066553	0.705 $\pm$ 0.09 0.0079346	0.84 $\pm$ 0.061 0.0160262	0.865 $\pm$ 0.071 0.0233146
		0.1	0.855 $\pm$ 0.05 0.0081158	0.705 $\pm$ 0.086 0.0079532	0.855 $\pm$ 0.05 0.0171934	0.87 $\pm$ 0.071 0.0204829
Z & S	Wavelet (db6)	0	0.85 $\pm$ 0.053 0.0075253	0.705 $\pm$ 0.086 0.0074273	0.85 $\pm$ 0.053 0.0234509	0.87 $\pm$ 0.048 0.0240283
		0.05	0.85 $\pm$ 0.047 0.007203	0.705 $\pm$ 0.101 0.0079486	0.85 $\pm$ 0.047 0.0170526	0.87 $\pm$ 0.048 0.0230243
		0.075	0.84 $\pm$ 0.061 0.0066553	0.705 $\pm$ 0.09 0.0079346	0.84 $\pm$ 0.061 0.0160262	0.865 $\pm$ 0.071 0.0233146
		0.1	0.855 $\pm$ 0.05 0.0081158	0.705 $\pm$ 0.086 0.0079532	0.855 $\pm$ 0.05 0.0171934	0.87 $\pm$ 0.071 0.0204829



**Table 2** continued

Dataset	Feature extraction method	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007) Acc $\pm$ STD Time (s)	Pin-GTSVM (Tanveer et al., 2019) Acc $\pm$ STD Time (s)	UTSVM (Qi et al., 2012) Acc $\pm$ STD Time (s)	Pin-UTSVM Acc $\pm$ STD Time (s)
O & S	Wavelet (db4)	0	0.84 $\pm$ 0.066 0.0083041	0.705 $\pm$ 0.09 0.0075658	0.84 $\pm$ 0.115 0.0213752	0.875 $\pm$ 0.063 0.0207815
		0.05	0.845 $\pm$ 0.064 0.0098547	0.71 $\pm$ 0.081 0.0075533	0.845 $\pm$ 0.064 0.0172382	0.88 $\pm$ 0.059 0.0214544
		0.075	0.84 $\pm$ 0.074 0.0068799	0.705 $\pm$ 0.09 0.0074954	0.84 $\pm$ 0.074 0.0181628	0.88 $\pm$ 0.089 0.0239684
		0.1	0.83 $\pm$ 0.071 0.0069321	0.705 $\pm$ 0.09 0.0080444	0.83 $\pm$ 0.071 0.019594	0.875 $\pm$ 0.063 0.023126
		0	0.805 $\pm$ 0.069 0.0073992	0.665 $\pm$ 0.106 0.0082542	0.805 $\pm$ 0.069 0.022903	0.855 $\pm$ 0.055 0.0206868
		0.05	0.805 $\pm$ 0.086 0.0087061	0.665 $\pm$ 0.071 0.0083415	0.81 $\pm$ 0.066 0.0219452	0.855 $\pm$ 0.055 0.0214962
Z & S	Wavelet (db4)	0.075	0.8 $\pm$ 0.067 0.0076478	0.665 $\pm$ 0.106 0.0078303	0.805 $\pm$ 0.069 0.0231559	0.86 $\pm$ 0.052 0.0216787
		0.1	0.805 $\pm$ 0.09 0.0076455	0.665 $\pm$ 0.106 0.0080056	0.805 $\pm$ 0.09 0.0194493	0.86 $\pm$ 0.057 0.0212222
		0	0.835 $\pm$ 0.058 0.0092206	0.685 $\pm$ 0.091 0.0075617	0.835 $\pm$ 0.058 0.0177684	0.89 $\pm$ 0.057 0.0214958
		0.05	0.835 $\pm$ 0.067 0.0078578	0.68 $\pm$ 0.082 0.0075886	0.835 $\pm$ 0.067 0.0171026	0.885 $\pm$ 0.053 0.0236019
		0.075	0.83 $\pm$ 0.071 0.0073749	0.68 $\pm$ 0.106 0.0079714	0.83 $\pm$ 0.071 0.0175865	0.89 $\pm$ 0.052 0.0219737
		0.1	0.835 $\pm$ 0.067 0.0074854	0.69 $\pm$ 0.097 0.0080074	0.835 $\pm$ 0.067 0.0196373	0.885 $\pm$ 0.063 0.0236645
O & S	Wavelet (Haar)	0	0.835 $\pm$ 0.058 0.0092206	0.685 $\pm$ 0.091 0.0075617	0.835 $\pm$ 0.058 0.0177684	0.89 $\pm$ 0.057 0.0214958
		0.05	0.835 $\pm$ 0.067 0.0078578	0.68 $\pm$ 0.082 0.0075886	0.835 $\pm$ 0.067 0.0171026	0.885 $\pm$ 0.053 0.0236019
		0.075	0.83 $\pm$ 0.071 0.0073749	0.68 $\pm$ 0.106 0.0079714	0.83 $\pm$ 0.071 0.0175865	0.89 $\pm$ 0.052 0.0219737
		0.1	0.835 $\pm$ 0.067 0.0074854	0.69 $\pm$ 0.097 0.0080074	0.835 $\pm$ 0.067 0.0196373	0.885 $\pm$ 0.063 0.0236645

Table 2 continued

Dataset	Feature extraction method	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007) Acc $\pm$ STD Time (s)	Pin-GTSVM (Tanveer et al., 2019) Acc $\pm$ STD Time (s)	UTSVM (Qi et al., 2012) Acc $\pm$ STD Time (s)	Pin-UTSVM Acc $\pm$ STD Time (s)
Z & S	Wavelet (Haar)	0	0.825 $\pm$ 0.079 0.0069795	0.715 $\pm$ 0.103 0.0083141	0.83 $\pm$ 0.059 0.0286963	0.87 $\pm$ 0.063 0.0239676
		0.05	0.845 $\pm$ 0.069 0.0074206	0.72 $\pm$ 0.101 0.007559	0.845 $\pm$ 0.069 0.0214907	0.87 $\pm$ 0.063 0.0213416
		0.075	0.83 $\pm$ 0.098 0.0087142	0.715 $\pm$ 0.103 0.0091684	0.83 $\pm$ 0.098 0.0197149	0.87 $\pm$ 0.054 0.0200001
		0.1	0.83 $\pm$ 0.075 0.0087073	0.715 $\pm$ 0.1 0.0080402	0.83 $\pm$ 0.075 0.0170522	0.87 $\pm$ 0.054 0.0197997
		0	0.65 $\pm$ 0.105 0.0063539	0.55 $\pm$ 0.078 0.0070141	0.65 $\pm$ 0.105 0.0149436	0.725 $\pm$ 0.079 0.0271362
		0.05	0.645 $\pm$ 0.093 0.0080839	0.54 $\pm$ 0.084 0.0070269	0.645 $\pm$ 0.093 0.0159007	0.73 $\pm$ 0.048 0.0241891
Z & S	ICA	0.075	0.645 $\pm$ 0.104 0.0072039	0.54 $\pm$ 0.084 0.0071844	0.645 $\pm$ 0.104 0.0172135	0.72 $\pm$ 0.063 0.0204093
		0.1	0.645 $\pm$ 0.104 0.0074145	0.55 $\pm$ 0.078 0.0073809	0.645 $\pm$ 0.104 0.017203	0.71 $\pm$ 0.107 0.0270421
		0	0.69 $\pm$ 0.088 0.0068629	0.58 $\pm$ 0.042 0.0070291	0.69 $\pm$ 0.107 0.0168279	0.725 $\pm$ 0.075 0.0272151
		0.05	0.69 $\pm$ 0.088 0.0068708	0.56 $\pm$ 0.07 0.0073105	0.69 $\pm$ 0.099 0.0165634	0.72 $\pm$ 0.092 0.0252823
		0.075	0.69 $\pm$ 0.099 0.0069741	0.55 $\pm$ 0.078 0.0076307	0.69 $\pm$ 0.099 0.0148552	0.71 $\pm$ 0.115 0.0243982
		0.1	0.7 $\pm$ 0.103 0.0070148	0.55 $\pm$ 0.078 0.0071902	0.7 $\pm$ 0.103 0.0162905	0.72 $\pm$ 0.103 0.0172879
O & S	ICA	0	0.69 $\pm$ 0.088 0.0068629	0.58 $\pm$ 0.042 0.0070291	0.69 $\pm$ 0.107 0.0168279	0.725 $\pm$ 0.075 0.0272151
		0.05	0.69 $\pm$ 0.088 0.0068708	0.56 $\pm$ 0.07 0.0073105	0.69 $\pm$ 0.099 0.0165634	0.72 $\pm$ 0.092 0.0252823
		0.075	0.69 $\pm$ 0.099 0.0069741	0.55 $\pm$ 0.078 0.0076307	0.69 $\pm$ 0.099 0.0148552	0.71 $\pm$ 0.115 0.0243982
		0.1	0.7 $\pm$ 0.103 0.0070148	0.55 $\pm$ 0.078 0.0071902	0.7 $\pm$ 0.103 0.0162905	0.72 $\pm$ 0.103 0.0172879

Table 2 continued

Dataset	Feature extraction method $\sigma$	TWSVM (Jayadeva & Khemchandani, 2007) Acc $\pm$ STD Time (s)	Pin-GTSVM (Tanveer et al., 2019) Acc $\pm$ STD Time (s)	UTSVM (Qi et al., 2012) Acc $\pm$ STD Time (s)	Pin-UTSVM Acc $\pm$ STD Time (s)	
O & S	PCA	0	0.81 $\pm$ 0.088	0.68 $\pm$ 0.092	0.825 $\pm$ 0.079	0.86 $\pm$ 0.039
			0.0077521	0.0074658	0.0185152	0.0204777
		0.05	0.815 $\pm$ 0.1	0.68 $\pm$ 0.092	0.815 $\pm$ 0.1	0.86 $\pm$ 0.052
			0.006741	0.0073176	0.0170763	0.0164226
		0.075	0.815 $\pm$ 0.085	0.68 $\pm$ 0.092	0.815 $\pm$ 0.085	0.855 $\pm$ 0.055
			0.0068522	0.0074029	0.0175409	0.0166932
Z & S	PCA	0.1	0.8 $\pm$ 0.078	0.68 $\pm$ 0.092	0.81 $\pm$ 0.084	0.865 $\pm$ 0.075
			0.0069073	0.0074775	0.0182551	0.0190718
		0	0.815 $\pm$ 0.082	0.695 $\pm$ 0.06	0.82 $\pm$ 0.059	0.865 $\pm$ 0.053
			0.0071196	0.0074377	0.0169641	0.0193064
		0.05	0.825 $\pm$ 0.086	0.69 $\pm$ 0.057	0.825 $\pm$ 0.086	0.865 $\pm$ 0.053
			0.0069548	0.0074509	0.0172147	0.0201486
Average Accuracy		0.075	0.815 $\pm$ 0.082	0.7 $\pm$ 0.075	0.815 $\pm$ 0.082	0.87 $\pm$ 0.071
			0.0070628	0.007431	0.0178496	0.0166967
		0.1	0.81 $\pm$ 0.088	0.685 $\pm$ 0.082	0.82 $\pm$ 0.059	0.87 $\pm$ 0.054
			0.0071095	0.0074574	0.0175212	0.019966
		0.804821429	0.804821429	0.674464286	0.806964286	0.850089286
		2.625	2.625	4	2.375	1

**Table 3** Optimal parameters obtained for different methods using linear kernel on EEG data with four noise levels

Dataset	Feature extraction method	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007)	Pin-GTSVM (Tanveer et al., 2019)	UTSVM (Qi et al., 2012)	Pin-UTSVM
			$C_1$	$C_1, C_2, \tau_1$	$C_1, C_3, \epsilon$	$C_1, C_2, \tau_1, \tau_2, \epsilon$
O & S	Wavelet (db1)	0	0.01	$10^{-5}, 0.01, 0.8$	$0.01, 10^{-5}, 0.1$	$10^{-3}, 0.1, 0.7, 0.6, 0.2$
		0.05	0.01	$0.1, 0.1, 0.5$	$0.01, 10^{-5}, 0.1$	$10^{-3}, 0.1, 0.8, 0.6, 0.3$
		0.075	0.01	$0.1, 0.1, 0.5$	$0.01, 10^{-5}, 0.1$	$10^{-3}, 0.1, 0.8, 0.7, 0.2$
		0.1	0.01	$0.01, 0.1, 0.5$	$0.01, 10^{-5}, 0.1$	$10^{-3}, 0.1, 0.7, 0.7, 0.1$
Z & S	Wavelet (db1)	0	0.1	$10^{-5}, 0.1, 0.6$	$0.1, 10^{-5}, 0.1$	$0.01, 0.1, 0.8, 0.5, 0.6$
		0.05	0.1	$10^{-5}, 0.1, 0.7$	$0.01, 0.01, 0.7$	$10^{-4}, 0.01, 0.6, 0.6, 0.1$
		0.075	$10^{-4}$	$0.1, 0.1, 0.7$	$10^{-4}, 10^{-5}, 0.1$	$0.01, 0.1, 0.5, 0.8, 0.5$
		0.1	0.1	$0.1, 0.1, 0.7$	$0.01, 10^{-5}, 0.1$	$0.01, 0.1, 0.7, 0.8, 0.5$
O & S	Wavelet (db2)	0	0.01	$0.01, 0.1, 0.5$	$0.01, 10^{-5}, 0.1$	$10^{-3}, 0.1, 0.7, 0.7, 0.3$
		0.05	0.01	$10^{-5}, 0.1, 0.5$	$0.01, 10^{-5}, 0.1$	$10^{-3}, 0.1, 0.5, 0.5, 0.5$
		0.075	$10^{-3}$	$0.1, 0.1, 0.5$	$10^{-3}, 10^{-5}, 0.1$	$10^{-3}, 0.1, 0.8, 0.8, 0.3$
		0.1	0.01	$10^{-5}, 0.1, 0.5$	$10^{-3}, 10^{-5}, 0.1$	$10^{-3}, 0.1, 0.5, 0.7, 0.1$
Z & S	Wavelet (db2)	0	0.1	$10^{-5}, 0.1, 0.6$	$0.1, 10^{-3}, 0.6$	$0.1, 10^{-5}, 0.8, 0.5, 0.1$
		0.05	0.1	$10^{-5}, 0.1, 0.5$	$0.1, 0.1, 0.5$	$0.1, 10^{-5}, 0.8, 0.5, 0.1$
		0.075	0.1	$10^{-5}, 0.1, 0.6$	$0.1, 0.01, 0.3$	$0.01, 0.01, 0.6, 0.5, 0.1$
		0.1	0.1	$0.1, 0.1, 0.6$	$0.1, 10^{-5}, 0.1$	$0.01, 0.01, 0.5, 0.5, 0.1$
O & S	Wavelet (db6)	0	0.1	$0.1, 0.1, 0.5$	$0.1, 0.1, 0.5$	$10^{-4}, 0.1, 0.5, 0.7, 0.7$
		0.05	0.1	$10^{-5}, 0.1, 0.6$	$0.1, 0.1, 0.6$	$10^{-5}, 0.1, 0.5, 0.5, 0.7$
		0.075	0.01	$0.01, 0.1, 0.5$	$0.1, 0.1, 0.5$	$10^{-5}, 0.1, 0.5, 0.6, 0.7$
		0.1	0.01	$10^{-5}, 0.1, 0.5$	$0.1, 0.1, 0.5$	$10^{-4}, 0.1, 0.5, 0.5, 0.7$

**Table 3** continued

Dataset	Feature extraction method	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007)	Pin-GTSVM (Tanveer et al., 2019)	UTSVM (Qi et al., 2012)	Pin-UTSVM
		$C_1$	$C_1$	$C_1, C_2, \tau_1$	$C_1, C_3, \epsilon$	$C_1, C_2, \tau_1, \tau_2, \epsilon$
Z & S	Wavelet (db6)	0	0.01	$10^{-5}, 0.1, 0.6$	$0.01, 10^{-5}, 0.1$	$10^{-4}, 0.01, 0.5, 0.8, 0.1$
		0.05	0.01	$10^{-5}, 0.1, 0.6$	$0.01, 10^{-5}, 0.1$	$10^{-4}, 0.01, 0.5, 0.8, 0.7$
		0.075	0.01	$10^{-5}, 0.1, 0.6$	$0.01, 10^{-5}, 0.1$	$0.01, 0.1, 0.5, 0.5, 0.7$
		0.1	0.01	$10^{-5}, 0.1, 0.7$	$0.01, 10^{-5}, 0.1$	$0.01, 0.1, 0.5, 0.5, 0.7$
O & S	Wavelet (db4)	0	$10^{-3}$	$10^{-5}, 0.1, 0.6$	$10^{-5}, 10^{-5}, 0.1$	$10^{-3}, 0.01, 0.8, 0.5, 0.1$
		0.05	$10^{-3}$	$0.01, 0.1, 0.7$	$10^{-3}, 10^{-5}, 0.1$	$10^{-3}, 0.01, 0.8, 0.5, 0.1$
		0.075	$10^{-3}$	$10^{-5}, 0.1, 0.6$	$10^{-3}, 10^{-5}, 0.1$	$10^{-5}, 0.1, 0.5, 0.6, 0.1$
		0.1	$10^{-3}$	$0.01, 0.1, 0.6$	$10^{-3}, 10^{-5}, 0.1$	$10^{-4}, 0.01, 0.5, 0.7, 0.5$
Z & S	Wavelet (db4)	0	$10^{-3}$	$10^{-5}, 0.1, 0.5$	$10^{-3}, 10^{-5}, 0.1$	$0.01, 10^{-3}, 0.6, 0.5, 0.1$
		0.05	$10^{-4}$	$10^{-5}, 10^{-3}, 0.8$	$0.1, 0.1, 0.2$	$0.01, 10^{-3}, 0.6, 0.5, 0.1$
		0.075	$10^{-3}$	$10^{-5}, 0.1, 0.5$	$0.1, 0.01, 0.7$	$0.01, 10^{-5}, 0.6, 0.5, 0.1$
		0.1	$10^{-4}$	$10^{-5}, 0.1, 0.5$	$10^{-4}, 10^{-5}, 0.1$	$0.01, 10^{-4}, 0.7, 0.5, 0.1$
O & S	Wavelet (Haar)	0	$10^{-3}$	$0.1, 0.1, 0.7$	$10^{-3}, 10^{-5}, 0.1$	$10^{-4}, 0.01, 0.7, 0.6, 0.7$
		0.05	$10^{-3}$	$0.1, 0.1, 0.5$	$10^{-3}, 10^{-5}, 0.1$	$10^{-4}, 0.01, 0.8, 0.8, 0.6$
		0.075	0.01	$10^{-5}, 0.01, 0.5$	$10^{-3}, 10^{-5}, 0.1$	$10^{-4}, 0.01, 0.5, 0.6, 0.7$
		0.1	$10^{-3}$	$10^{-5}, 0.01, 0.6$	$10^{-3}, 10^{-5}, 0.1$	$10^{-3}, 0.1, 0.5, 0.6, 0.3$
Z & S	Wavelet (Haar)	0	0.01	$10^{-5}, 0.1, 0.5$	$10^{-4}, 1, 0.5$	$0.01, 0.1, 0.7, 0.8, 0.5$
		0.05	$10^{-3}$	$0.1, 0.1, 0.6$	$10^{-3}, 10^{-5}, 0.1$	$0.01, 0.1, 0.7, 0.8, 0.5$
		0.075	0.01	$10^{-5}, 0.1, 0.5$	$0.01, 10^{-5}, 0.1$	$0.01, 0.1, 0.8, 0.8, 0.5$
		0.1	0.01	$0.1, 0.1, 0.6$	$0.01, 10^{-5}, 0.1$	$0.01, 0.1, 0.6, 0.8, 0.5$

**Table 3** continued

Dataset	Feature extraction method	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007)	Pin-GTSVM (Tanveer et al., 2019)	UTSVM (Qi et al., 2012)	Pin-UTSVM
		$C_1$	$C_1, C_2, \tau_1$	$C_1, C_2, \tau_1$	$C_1, C_3, \epsilon$	$C_1, C_2, \tau_1, \tau_2, \epsilon$
Z & S	ICA	0	0.01	$10^3, 10, 0.7$	$0.01, 10^{-5}, 0.1$	$0.1, 1, 0.6, 0.6, 0.3$
		0.05	0.01	$10^3, 1, 0.8$	$0.01, 10^{-5}, 0.1$	$0.1, 1, 0.5, 0.6, 0.3$
		0.075	0.01	$10^3, 10, 0.6$	$0.01, 10^{-5}, 0.1$	$1, 0.1, 0.8, 0.5, 0.5$
		0.1	0.01	$100, 10^3, 0.6$	$0.01, 10^{-5}, 0.1$	$0.1, 1, 0.6, 0.6, 0.3$
O & S	ICA	0	0.01	$10^3, 100, 0.7$	$10^{-5}, 10^{-5}, 0.1$	$1, 10, 0.5, 0.6, 0.3$
		0.05	0.01	$10^3, 10, 0.8$	$10^{-5}, 10^{-5}, 0.1$	$1, 10, 0.6, 0.7, 0.3$
		0.075	$10^{-3}$	$10^3, 10^3, 0.8$	$10^{-3}, 10^{-5}, 0.1$	$1, 10, 0.6, 0.6, 0.3$
		0.1	$10^{-3}$	$10^3, 1, 0.5$	$10^{-5}, 10^{-5}, 0.1$	$10^{-5}, 0.1, 0.5, 0.7, 0.1$
O & S	PCA	0	$10^{-3}$	$0.1, 10^{-5}, 0.8$	$0.1, 0.1, 0.6$	$10^{-5}, 0.1, 0.5, 0.8, 0.3$
		0.05	0.01	$0.1, 10^{-5}, 0.7$	$0.01, 10^{-5}, 0.1$	$10^{-5}, 10^{-3}, 0.5, 0.5, 0.1$
		0.075	$10^{-3}$	$0.1, 10^{-5}, 0.8$	$10^{-3}, 10^{-5}, 0.1$	$0.01, 0.01, 0.5, 0.5, 0.1$
		0.1	$10^{-3}$	$0.1, 10^{-5}, 0.8$	$0.1, 0.1, 0.6$	$10^{-3}, 0.1, 0.7, 0.8, 0.5$
Z & S	PCA	0	$10^{-3}$	$0.01, 0.1, 0.8$	$0.1, 0.1, 0.3$	$10^{-4}, 0.1, 0.5, 0.7, 0.5$
		0.05	$10^{-3}$	$10^{-5}, 0.1, 0.8$	$10^{-3}, 10^{-5}, 0.1$	$10^{-5}, 0.1, 0.5, 0.8, 0.3$
		0.075	$10^{-3}$	$10^{-5}, 0.1, 0.8$	$10^{-3}, 10^{-5}, 0.1$	$0.01, 10^{-4}, 0.8, 0.5, 0.1$
		0.1	$10^{-3}$	$10^{-5}, 0.1, 0.5$	$0.1, 0.1, 0.3$	$10^{-4}, 0.1, 0.6, 0.7, 0.5$

**Table 4** Accuracies obtained for different methods using Gaussian kernel on EEG data with four noise levels

Dataset	Feature extraction method	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007) Acc $\pm$ STD Time (s)	Pin-GTSVM (Tanveer et al., 2019) Acc $\pm$ STD Time (s)	UTSVM (Qi et al., 2012) Acc $\pm$ STD Time (s)	Pin-UTSVM Acc $\pm$ STD Time (s)
O & S	Wavelet (db1)	0	0.91 $\pm$ 0.052 0.0086729	0.88 $\pm$ 0.035 0.0578769	0.91 $\pm$ 0.052 0.0535213	0.95 $\pm$ 0.123 0.0835986
		0.05	0.905 $\pm$ 0.037 0.0092753	0.885 $\pm$ 0.041 0.0574524	0.905 $\pm$ 0.05 0.0535328	0.93 $\pm$ 0.123 0.0845558
		0.075	0.91 $\pm$ 0.052 0.0085282	0.88 $\pm$ 0.035 0.0576991	0.91 $\pm$ 0.052 0.053426	0.93 $\pm$ 0.123 0.0837327
		0.1	0.91 $\pm$ 0.066 0.0092779	0.885 $\pm$ 0.041 0.0579467	0.91 $\pm$ 0.066 0.0532457	0.915 $\pm$ 0.058 0.0813939
		0	0.94 $\pm$ 0.057 0.0084774	0.895 $\pm$ 0.05 0.0580878	0.94 $\pm$ 0.057 0.0533821	0.975 $\pm$ 0.042 0.0817996
		0.05	0.935 $\pm$ 0.058 0.0089629	0.895 $\pm$ 0.05 0.0574088	0.94 $\pm$ 0.046 0.0538935	0.975 $\pm$ 0.042 0.0810221
Z & S	Wavelet (db1)	0.075	0.945 $\pm$ 0.044 0.0086215	0.895 $\pm$ 0.05 0.0578031	0.945 $\pm$ 0.044 0.0532888	0.98 $\pm$ 0.035 0.0818672
		0.1	0.945 $\pm$ 0.044 0.0085822	0.895 $\pm$ 0.05 0.058212	0.945 $\pm$ 0.044 0.0527493	0.98 $\pm$ 0.035 0.0813745
		0	0.885 $\pm$ 0.075 0.0092099	0.89 $\pm$ 0.057 0.0577297	0.905 $\pm$ 0.05 0.057288	0.94 $\pm$ 0.052 0.0820042
		0.05	0.885 $\pm$ 0.075 0.0086335	0.895 $\pm$ 0.05 0.0576182	0.905 $\pm$ 0.05 0.056905	0.935 $\pm$ 0.053 0.0819432
		0.075	0.895 $\pm$ 0.064 0.0085143	0.89 $\pm$ 0.057 0.0574378	0.905 $\pm$ 0.05 0.056753	0.93 $\pm$ 0.075 0.0816901
		0.1	0.89 $\pm$ 0.07 0.0085926	0.89 $\pm$ 0.057 0.0573045	0.905 $\pm$ 0.05 0.0563936	0.925 $\pm$ 0.075 0.0995019
O & S	Wavelet (db2)	0	0.885 $\pm$ 0.075 0.0092099	0.89 $\pm$ 0.057 0.0577297	0.905 $\pm$ 0.05 0.057288	0.94 $\pm$ 0.052 0.0820042
		0.05	0.885 $\pm$ 0.075 0.0086335	0.895 $\pm$ 0.05 0.0576182	0.905 $\pm$ 0.05 0.056905	0.935 $\pm$ 0.053 0.0819432
		0.075	0.895 $\pm$ 0.064 0.0085143	0.89 $\pm$ 0.057 0.0574378	0.905 $\pm$ 0.05 0.056753	0.93 $\pm$ 0.075 0.0816901
		0.1	0.89 $\pm$ 0.07 0.0085926	0.89 $\pm$ 0.057 0.0573045	0.905 $\pm$ 0.05 0.0563936	0.925 $\pm$ 0.075 0.0995019

Table 4 continued

Dataset	Feature extraction method	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007) Acc $\pm$ STD Time (s)	Pin-GTSVM (Tanveer et al., 2019) Acc $\pm$ STD Time (s)	UTSVM (Qi et al., 2012) Acc $\pm$ STD Time (s)	Pin-UTSVM Acc $\pm$ STD Time (s)
Z & S	Wavelet (db2)	0	0.935 $\pm$ 0.047 0.0085623	0.905 $\pm$ 0.06 0.0600382	0.95 $\pm$ 0.067 0.0599432	0.96 $\pm$ 0.039 0.0829832
		0.05	0.94 $\pm$ 0.046 0.008373	0.905 $\pm$ 0.06 0.0602979	0.95 $\pm$ 0.058 0.0604642	0.965 $\pm$ 0.041 0.0842925
		0.075	0.935 $\pm$ 0.058 0.0084747	0.895 $\pm$ 0.05 0.0588745	0.945 $\pm$ 0.083 0.0608203	0.965 $\pm$ 0.041 0.0830553
		0.1	0.935 $\pm$ 0.058 0.0084927	0.905 $\pm$ 0.06 0.0585898	0.94 $\pm$ 0.057 0.060484	0.965 $\pm$ 0.041 0.0831525
O & S	Wavelet (db6)	0	0.9 $\pm$ 0.071 0.0085916	0.885 $\pm$ 0.058 0.0588824	0.915 $\pm$ 0.078 0.0572666	0.92 $\pm$ 0.054 0.0833423
		0.05	0.91 $\pm$ 0.07 0.0085342	0.885 $\pm$ 0.058 0.0582375	0.91 $\pm$ 0.07 0.0545413	0.915 $\pm$ 0.067 0.0816987
		0.075	0.91 $\pm$ 0.077 0.0086546	0.885 $\pm$ 0.058 0.0581589	0.92 $\pm$ 0.067 0.0560809	0.915 $\pm$ 0.047 0.0848108
		0.1	0.905 $\pm$ 0.072 0.0085359	0.885 $\pm$ 0.058 0.0581987	0.91 $\pm$ 0.07 0.0562161	0.92 $\pm$ 0.042 0.0841472
Z & S	Wavelet (db6)	0	0.95 $\pm$ 0.047 0.0085895	0.9 $\pm$ 0.053 0.0686746	0.95 $\pm$ 0.047 0.0635375	0.965 $\pm$ 0.047 0.0932359
		0.05	0.955 $\pm$ 0.037 0.0084184	0.9 $\pm$ 0.053 0.063331	0.955 $\pm$ 0.037 0.0629456	0.97 $\pm$ 0.042 0.0928587
		0.075	0.95 $\pm$ 0.047 0.0094305	0.9 $\pm$ 0.053 0.068864	0.95 $\pm$ 0.047 0.0595354	0.965 $\pm$ 0.041 0.0922739
		0.1	0.945 $\pm$ 0.05 0.008574	0.9 $\pm$ 0.053 0.065278	0.95 $\pm$ 0.047 0.0594476	0.96 $\pm$ 0.039 0.094555



Table 4 continued

Dataset	Feature extraction method	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007) Acc $\pm$ STD Time (s)	Pin-GTSVM (Tanveer et al., 2019) Acc $\pm$ STD Time (s)	UTSVM (Qi et al., 2012) Acc $\pm$ STD Time (s)	Pin-UTSVM Acc $\pm$ STD Time (s)
O & S	Wavelet (db4)	0	0.89 $\pm$ 0.032 0.0115564	0.91 $\pm$ 0.039 0.0643275	0.92 $\pm$ 0.054 0.0672824	0.945 $\pm$ 0.028 0.100723
		0.05	0.89 $\pm$ 0.032 0.0093384	0.91 $\pm$ 0.039 0.0634427	0.925 $\pm$ 0.049 0.0710768	0.95 $\pm$ 0.024 0.0960298
		0.075	0.895 $\pm$ 0.037 0.0107236	0.91 $\pm$ 0.039 0.0660262	0.925 $\pm$ 0.042 0.0793309	0.95 $\pm$ 0.033 0.0960476
Z & S	Wavelet (db4)	0.1	0.89 $\pm$ 0.032 0.0101156	0.91 $\pm$ 0.039 0.0688774	0.92 $\pm$ 0.059 0.0832469	0.935 $\pm$ 0.041 0.0959586
		0	0.935 $\pm$ 0.053 0.0109854	0.89 $\pm$ 0.077 0.0638101	0.945 $\pm$ 0.037 0.0901288	0.95 $\pm$ 0.041 0.0840803
		0.05	0.935 $\pm$ 0.053 0.0102901	0.89 $\pm$ 0.077 0.0782333	0.945 $\pm$ 0.037 0.0660453	0.95 $\pm$ 0.041 0.0896145
O & S	Wavelet (Haar)	0.075	0.935 $\pm$ 0.053 0.0110756	0.895 $\pm$ 0.069 0.0601623	0.95 $\pm$ 0.033 0.0600703	0.955 $\pm$ 0.037 0.089858
		0.1	0.935 $\pm$ 0.053 0.010819	0.89 $\pm$ 0.077 0.0607709	0.95 $\pm$ 0.033 0.0614134	0.95 $\pm$ 0.041 0.091823
		0	0.895 $\pm$ 0.037 0.0088981	0.875 $\pm$ 0.042 0.0653404	0.895 $\pm$ 0.037 0.0722004	0.935 $\pm$ 0.058 0.0870436
O & S	Wavelet (Haar)	0.05	0.895 $\pm$ 0.037 0.0088981	0.87 $\pm$ 0.042 0.0653855	0.895 $\pm$ 0.037 0.0661554	0.93 $\pm$ 0.063 0.0937993
		0.075	0.895 $\pm$ 0.037 0.0093007	0.875 $\pm$ 0.042 0.0695904	0.895 $\pm$ 0.037 0.0638544	0.925 $\pm$ 0.063 0.100696
		0.1	0.895 $\pm$ 0.037 0.0099377	0.875 $\pm$ 0.042 0.068734	0.895 $\pm$ 0.037 0.0645842	0.93 $\pm$ 0.059 0.0976353

Table 4 continued

Dataset	Feature extraction method	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007) Acc $\pm$ STD Time (s)	Pin-GTSVM (Tanveer et al., 2019) Acc $\pm$ STD Time (s)	UTSVM (Qi et al., 2012) Acc $\pm$ STD Time (s)	Pin-UTSVM Acc $\pm$ STD Time (s)
Z & S	Wavelet (Haar)	0	0.93 $\pm$ 0.059 0.0089648	0.9 $\pm$ 0.058 0.0789973	0.935 $\pm$ 0.075 0.0645657	0.96 $\pm$ 0.039 0.0905899
		0.05	0.93 $\pm$ 0.059 0.0090619	0.9 $\pm$ 0.058 0.0778687	0.94 $\pm$ 0.061 0.0808941	0.965 $\pm$ 0.041 0.093998
		0.075	0.935 $\pm$ 0.053 0.0093104	0.9 $\pm$ 0.062 0.0745811	0.935 $\pm$ 0.053 0.0670411	0.96 $\pm$ 0.039 0.0900987
		0.1	0.93 $\pm$ 0.059 0.009075	0.9 $\pm$ 0.062 0.0647041	0.935 $\pm$ 0.075 0.0656169	0.96 $\pm$ 0.039 0.0905212
		0	0.935 $\pm$ 0.041 0.0090276	0.94 $\pm$ 0.046 0.0602605	0.975 $\pm$ 0.026 0.0620424	0.93 $\pm$ 0.035 0.0867308
Z & S	ICA	0.05	0.935 $\pm$ 0.041 0.0091266	0.94 $\pm$ 0.046 0.0630796	0.975 $\pm$ 0.035 0.068311	0.925 $\pm$ 0.042 0.0936528
		0.075	0.945 $\pm$ 0.037 0.0090444	0.935 $\pm$ 0.053 0.064828	0.975 $\pm$ 0.026 0.0653539	0.945 $\pm$ 0.037 0.0876098
		0.1	0.935 $\pm$ 0.047 0.0093315	0.945 $\pm$ 0.044 0.0655867	0.97 $\pm$ 0.035 0.0607341	0.93 $\pm$ 0.048 0.0869871
		0	0.885 $\pm$ 0.063 0.0095646	0.905 $\pm$ 0.072 0.0770429	0.905 $\pm$ 0.06 0.0669057	0.895 $\pm$ 0.055 0.098613
		0.05	0.885 $\pm$ 0.047 0.0089679	0.905 $\pm$ 0.055 0.0642936	0.905 $\pm$ 0.06 0.0618375	0.905 $\pm$ 0.055 0.108434
O & S	ICA	0.075	0.885 $\pm$ 0.075 0.0088019	0.885 $\pm$ 0.071 0.0621271	0.895 $\pm$ 0.069 0.0624017	0.89 $\pm$ 0.057 0.0996449
		0.1	0.89 $\pm$ 0.081 0.0092394	0.895 $\pm$ 0.064 0.0646773	0.9 $\pm$ 0.062 0.0647453	0.91 $\pm$ 0.066 0.0912846

Table 4 continued

Dataset	Feature extraction method	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007) Acc $\pm$ STD Time (s)	Pin-GTSVM (Tanveer et al., 2019) Acc $\pm$ STD Time (s)	UTSVM (Qi et al., 2012) Acc $\pm$ STD Time (s)	Pin-UTSVM Acc $\pm$ STD Time (s)
O & S	PCA	0	0.905 $\pm$ 0.037 0.0091058	0.885 $\pm$ 0.041 0.0588881	0.905 $\pm$ 0.037 0.0546391	0.93 $\pm$ 0.125 0.0842369
		0.05	0.905 $\pm$ 0.037 0.0088523	0.885 $\pm$ 0.041 0.0576457	0.905 $\pm$ 0.037 0.0556292	0.93 $\pm$ 0.125 0.0842962
		0.075	0.905 $\pm$ 0.037 0.0087033	0.885 $\pm$ 0.041 0.0579486	0.905 $\pm$ 0.037 0.0548321	0.925 $\pm$ 0.123 0.0841391
		0.1	0.905 $\pm$ 0.037 0.0087709	0.885 $\pm$ 0.041 0.0581411	0.905 $\pm$ 0.037 0.0545417	0.925 $\pm$ 0.123 0.0840632
		0	0.935 $\pm$ 0.071 0.0085591	0.905 $\pm$ 0.06 0.0581038	0.94 $\pm$ 0.057 0.0544801	0.96 $\pm$ 0.046 0.0808274
		0.05	0.935 $\pm$ 0.071 0.008566	0.905 $\pm$ 0.06 0.0728284	0.94 $\pm$ 0.057 0.0538913	0.965 $\pm$ 0.047 0.0818374
Z & S	PCA	0.075	0.94 $\pm$ 0.039 0.0084128	0.905 $\pm$ 0.06 0.0573404	0.94 $\pm$ 0.039 0.0530759	0.96 $\pm$ 0.046 0.0814021
		0.1	0.945 $\pm$ 0.044 0.0084534	0.905 $\pm$ 0.06 0.0573136	0.945 $\pm$ 0.044 0.0527196	0.965 $\pm$ 0.047 0.0810528
			0.917946429	0.897053571	0.927857143	0.942410714
			2.973214286	3.669642857	2.071428571	1.285714286
		Average accuracy				
		Average rank				

**Table 5** Optimal parameters obtained for different methods using Gaussian kernel on EEG data with four noise levels

Dataset	Feature extraction method	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007)	Pin-GTSVM (Tanveer et al., 2019)	UTSVM (Qi et al., 2012)	Pin-UTSVM
			$C_1, \mu$	$C_1, C_2, \tau_1, \mu$	$C_1, C_3, \epsilon, \mu$	$C_1, C_2, \tau_1, \tau_2, \epsilon, \mu$
O & S	Wavelet (db1)	0	0.01, 16919.8	$10^{-5}, 10^{-5}, 0.5, 16919.8$	0.01, $10^{-5}, 0.1, 16919.8$	1, 100, 0.8, 0.6, 0.5, 16919.8
		0.05	0.1, 16920.8	$10^{-5}, 10^{-3}, 0.5, 16920.8$	0.01, $10^{-5}, 0.1, 16920.8$	1, 100, 0.6, 0.5, 0.5, 16920.8
		0.075	0.01, 16924.5	$10^{-5}, 10^{-5}, 0.5, 16924.5$	0.01, $10^{-5}, 0.1, 16924.5$	1, 100, 0.7, 0.6, 0.5, 16924.5
		0.1	0.01, 16919.8	$10^{-5}, 10^{-5}, 0.5, 16919.8$	0.01, $10^{-5}, 0.1, 16919.8$	$10^{-4}, 0.1, 0.7, 0.8, 0.6, 16919.8$
Z & S	Wavelet (db1)	0	0.01, 17416	$10^{-5}, 10^{-5}, 0.5, 17416$	0.01, $10^{-5}, 0.1, 17416$	0.1, 10, 0.8, 0.5, 0.6, 17416
		0.05	0.01, 17412.7	$10^{-5}, 10^{-5}, 0.5, 17412.7$	0.01, 0.1, 0.6, 17412.7	0.1, 10, 0.8, 0.5, 0.6, 17412.7
		0.075	0.01, 17408.9	$10^{-5}, 10^{-5}, 0.5, 17408.9$	0.01, $10^{-5}, 0.1, 17408.9$	0.1, 10, 0.8, 0.5, 0.6, 17408.9
		0.1	0.01, 17412.6	$10^{-5}, 10^{-5}, 0.5, 17412.6$	0.01, $10^{-5}, 0.1, 17412.6$	0.1, 10, 0.8, 0.5, 0.6, 17412.6
O & S	Wavelet (db2)	0	0.01, 16564	$10^{-5}, 10^{-5}, 0.5, 16564$	100, 100, 0.1, 16564	$10^{-5}, 0.1, 0.5, 0.7, 0.7, 16564$
		0.05	0.01, 16559.2	$10^{-5}, 10^{-5}, 0.5, 16559.2$	100, 100, 0.1, 16559.2	$10^{-5}, 0.1, 0.5, 0.7, 0.7, 16559.2$
		0.075	0.01, 16568.4	$10^{-5}, 10^{-5}, 0.5, 16568.4$	100, 100, 0.1, 16568.4	$10^{-5}, 0.1, 0.5, 0.6, 0.7, 16568.4$
		0.1	0.01, 16565.5	$10^{-5}, 10^{-5}, 0.5, 16565.5$	100, 100, 0.1, 16565.5	$10^{-5}, 0.1, 0.5, 0.6, 0.7, 16565.5$
Z & S	Wavelet (db2)	0	$10^{-4}, 17815.2$	$10^{-5}, 10^{-5}, 0.5, 17815.2$	$10^3, 10^3, 0.1, 17815.2$	0.1, 10, 0.7, 0.5, 0.6, 17815.2
		0.05	0.01, 17815.7	$10^{-5}, 10^{-5}, 0.5, 17815.7$	$10^3, 10^3, 0.1, 17815.7$	0.1, 10, 0.8, 0.6, 0.6, 17815.7
		0.075	0.01, 17820.8	$10^{-5}, 10^{-5}, 0.5, 17820.8$	$10^3, 10^3, 0.1, 17820.8$	0.1, 10, 0.7, 0.5, 0.6, 17820.8
		0.1	0.01, 17815.5	$10^{-5}, 10^{-5}, 0.5, 17815.5$	$10^3, 10^3, 0.1, 17815.5$	0.1, 10, 0.7, 0.5, 0.6, 17815.5
O & S	wavelet (db6)	0	0.01, 16527.8	$10^{-5}, 0.01, 0.5, 16527.8$	10, 100, 0.2, 16527.8	$10^{-4}, 0.1, 0.5, 0.7, 0.7, 16527.8$
		0.05	0.01, 16525.7	$10^{-5}, 0.01, 0.5, 16525.7$	0.01, $10^{-5}, 0.1, 16525.7$	$10^{-5}, 0.1, 0.5, 0.6, 0.7, 16525.7$
		0.075	0.01, 16529.2	$10^{-5}, 0.01, 0.5, 16529.2$	10, 100, 0.2, 16529.2	$10^{-4}, 0.1, 0.5, 0.7, 0.7, 16529.2$
		0.1	0.01, 16526.9	$10^{-5}, 0.01, 0.5, 16526.9$	10, 100, 0.2, 16526.9	$10^{-4}, 1, 0.5, 0.5, 0.2, 16526.9$

**Table 5** continued

Dataset	Feature extraction method	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007)	Pin-GTSVM (Tanveer et al., 2019)	UTSVM (Qi et al., 2012)	Pin-UTSVM
		$C_1, \mu$	$C_1, C_2, \tau_1, \mu$	$C_1, C_3, \epsilon, \mu$	$C_1, C_2, \tau_1, \tau_2, \epsilon, \mu$	
Z & S	Wavelet (db6)	0	0.01, 17811.6	$10^{-5}, 10^{-5}, 0.5, 17811.6$	0.01, $10^{-5}, 0.1, 17811.6$	$10^{-3}, 0.01, 0.5, 0.6, 0.7, 17811.6$
		0.05	0.01, 17812.1	$10^{-5}, 10^{-5}, 0.5, 17812.1$	0.01, $10^{-5}, 0.1, 17812.1$	$10^{-3}, 0.01, 0.5, 0.6, 0.7, 17812.1$
		0.075	0.01, 17814	$10^{-5}, 10^{-5}, 0.5, 17814$	0.01, $10^{-5}, 0.1, 17814$	$10^{-3}, 0.01, 0.7, 0.6, 0.1, 17814$
		0.1	0.01, 17814.7	$10^{-5}, 10^{-5}, 0.5, 17814.7$	0.01, 0.01, 0.6, 17814.7	$10^{-3}, 0.01, 0.5, 0.6, 0.7, 17814.7$
O & S	Wavelet (db4)	0	$10^3, 16696.4$	$10^{-5}, 10^{-5}, 0.5, 16696.4$	$100, 10^3, 0.1, 16696.4$	1, 100, 0.8, 0.5, 0.6, 16696.4
		0.05	$10^3, 16697.1$	$10^{-5}, 10^{-5}, 0.5, 16697.1$	$100, 10^3, 0.1, 16697.1$	1, 100, 0.8, 0.5, 0.6, 16697.1
		0.075	$10^3, 16693.7$	$10^{-5}, 10^{-5}, 0.5, 16693.7$	$100, 10^3, 0.1, 16693.7$	1, 100, 0.8, 0.5, 0.6, 16693.7
		0.1	$10^3, 16699.5$	$10^{-5}, 10^{-5}, 0.5, 16699.5$	$100, 10^3, 0.1, 16699.5$	1, 100, 0.8, 0.5, 0.6, 16699.5
Z & S	Wavelet (db4)	0	$10^3, 17706.9$	$10^{-5}, 10^{-5}, 0.5, 17706.9$	$100, 10^3, 0.1, 17706.9$	0.01, 1, 0.5, 0.8, 0.7, 17706.9
		0.05	$10^3, 17704.5$	$10^{-5}, 10^{-5}, 0.5, 17704.5$	$100, 10^3, 0.1, 17704.5$	0.01, 1, 0.5, 0.8, 0.7, 17704.5
		0.075	$10^3, 17707$	$10^{-5}, 10^{-5}, 0.5, 17707$	$100, 10^3, 0.1, 17707$	0.01, 1, 0.7, 0.8, 0.7, 17707
		0.1	1, 17707.7	$10^{-5}, 10^{-5}, 0.5, 17707.7$	$100, 10^3, 0.1, 17707.7$	0.01, 1, 0.5, 0.8, 0.7, 17707.7
O & S	Wavelet (Haar)	0	0.1, 16325.2	$10^{-5}, 0.01, 0.5, 16325.2$	0.1, $10^{-5}, 0.1, 16325.2$	1, 100, 0.7, 0.6, 0.6, 16325.2
		0.05	0.1, 16324.4	$10^{-5}, 0.01, 0.5, 16324.4$	0.1, $10^{-5}, 0.1, 16324.4$	1, 100, 0.5, 0.6, 0.6, 16324.4
		0.075	0.1, 16324.8	$10^{-5}, 0.01, 0.5, 16324.8$	0.1, $10^{-5}, 0.1, 16324.8$	1, 100, 0.8, 0.6, 0.6, 16324.8
		0.1	0.1, 16325.7	$10^{-5}, 0.01, 0.5, 16325.7$	0.1, $10^{-5}, 0.1, 16325.7$	1, 100, 0.5, 0.6, 0.6, 16325.7
Z & S	Wavelet (Haar)	0	$10^{-3}, 17628.8$	$10^{-5}, 10^{-3}, 0.5, 17628.8$	0.01, 0.1, 0.7, 17628.8	$10^{-3}, 0.01, 0.6, 0.8, 0.7, 17628.8$
		0.05	$10^{-3}, 17630.5$	$10^{-5}, 10^{-3}, 0.5, 17630.5$	0.01, 0.1, 0.7, 17630.5	$10^{-3}, 0.01, 0.6, 0.8, 0.7, 17630.5$
		0.075	$10^{-3}, 17620.7$	$10^{-5}, 10^{-4}, 0.7, 17620.7$	$10^{-3}, 10^{-5}, 0.1, 17620.7$	$10^{-3}, 0.01, 0.6, 0.8, 0.7, 17620.7$
		0.1	$10^{-3}, 17627.8$	$10^{-5}, 10^{-5}, 0.5, 17627.8$	0.01, 0.01, 0.7, 17627.8	$10^{-3}, 0.01, 0.5, 0.8, 0.7, 17627.8$

Table 5 continued

Dataset Feature extraction method	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007)	Pin-GTSVM (Tanveer et al., 2019)	UTSVM (Qi et al., 2012)	Pin-UTSVM
	$C_1, \mu$	$C_1, C_2, \tau_1, \mu$	$C_1, C_3, \epsilon, \mu$	$C_1, C_2, \tau_1, \tau_2, \epsilon, \mu$	
Z & S ICA	0	0.01, 59.0404	$10, 10^{-5}, 0.5, 59.0404$	$10, 10, 0.1, 59.0404$	$10^{-4}, 10^{-5}, 0.5, 0.5, 0.1, 59.0404$
	0.05	$10^{-3}, 59.1961$	$10, 10^{-5}, 0.5, 59.1961$	$10, 100, 0.1, 59.1961$	$10^{-4}, 10^{-5}, 0.5, 0.5, 0.1, 59.1961$
	0.075	$0.01, 59.3788$	$10, 10^{-5}, 0.5, 59.3788$	$10, 100, 0.1, 59.3788$	$10^{-4}, 10^{-3}, 0.5, 0.5, 0.1, 59.3788$
	0.1	$0.01, 59.2345$	$10, 10^{-5}, 0.6, 59.2345$	$1, 10, 0.7, 59.2345$	$10^{-4}, 10^{-5}, 0.5, 0.5, 0.1, 59.2345$
O & S ICA	0	0.1, 87.5473	$100, 10^{-5}, 0.5, 87.5473$	$0.1, 1, 0.1, 87.5473$	$10^{-5}, 1, 0.5, 0.6, 0.1, 87.5473$
	0.05	0.1, 87.5459	$0.01, 10^{-5}, 0.5, 87.5459$	$0.1, 1, 0.1, 87.5459$	$10^{-5}, 1, 0.5, 0.5, 0.2, 87.5459$
	0.075	$0.01, 88.0187$	$0.1, 10^{-5}, 0.5, 88.0187$	$10, 100, 0.1, 88.0187$	$10^{-3}, 0.01, 0.5, 0.5, 0.1, 88.0187$
	0.1	$0.01, 87.5325$	$10, 10^{-5}, 0.5, 87.5325$	$1, 100, 0.2, 87.5325$	$10^{-5}, 1, 0.5, 0.6, 0.3, 87.5325$
O & S PCA	0	0.1, 16913.5	$10^{-5}, 10^{-5}, 0.5, 16913.5$	$0.1, 10^{-5}, 0.1, 16913.5$	$1, 100, 0.5, 0.5, 0.5, 16913.5$
	0.05	0.1, 16913	$10^{-5}, 10^{-5}, 0.5, 16913$	$0.1, 10^{-5}, 0.1, 16913$	$1, 100, 0.7, 0.5, 0.5, 16913$
	0.075	$0.1, 16919.3$	$10^{-5}, 10^{-5}, 0.5, 16919.3$	$0.1, 10^{-5}, 0.1, 16919.3$	$1, 100, 0.6, 0.5, 0.5, 16919.3$
	0.1	0.1, 16911	$10^{-5}, 10^{-5}, 0.5, 16911$	$0.1, 10^{-5}, 0.1, 16911$	$1, 100, 0.6, 0.5, 0.5, 16911$
Z & S PCA	0	$0.01, 17724.4$	$10^{-5}, 10^{-5}, 0.5, 17724.4$	$0.01, 0.1, 0.6, 17724.4$	$0.1, 10, 0.7, 0.5, 0.6, 17724.4$
	0.05	$0.01, 17723.1$	$10^{-5}, 10^{-5}, 0.5, 17723.1$	$0.01, 0.1, 0.6, 17723.1$	$0.1, 10, 0.6, 0.5, 0.6, 17723.1$
	0.075	$10^{-3}, 17725.4$	$10^{-5}, 10^{-5}, 0.5, 17725.4$	$10^{-3}, 10^{-5}, 0.1, 17725.4$	$10, 0.6, 0.5, 0.6, 17725.4$
	0.1	$10^{-3}, 17728.1$	$10^{-5}, 10^{-5}, 0.5, 17728.1$	$10^{-3}, 10^{-5}, 0.1, 17728.1$	$10, 0.7, 0.5, 0.6, 17728.1$

after the cross-validation at the optimal parameters, which were obtained by the grid search method. The table reports the average accuracy and average rank of the methods. Analogous to the linear case, the proposed model performs exceptionally well, achieving an average rank of 1.286.

The best accuracy in Table 4 for a no-noise dataset is 0.975, obtained by both, the proposed Pin-UTSVM using the wavelet (db1) feature extraction for the Z&S dataset and UTSVM using the ICA feature extraction for the Z&S dataset. The runner-up method is the UTSVM with an average rank of 2.071. The TWSVM method obtains an average rank of 2.973 and the Pin-GTSVM the worst performing method receives 3.67 average rank. All the methods perform significantly better with Gaussian kernel than the linear counterpart in Table 2. The average accuracy of each method increased by 9–22% when switching to Gaussian kernel. The least growth is observed in the proposed method Pin-UTSVM, suggesting its lower degree of dependence on Gaussian kernel to form separating hyperplanes. This revelation enables us to suggest utilising the fast linear kernel when speed is of priority, with a minimal compromise to accuracy.

Specifically evaluating the Z&S dataset for the Gaussian kernel case, we observe that the wavelet (db1) performs the best for our method. In contrast, ICA performs the worst. As for the O&S dataset, wavelet (db4) performs the best, and ICA performs the worst. A difference from the linear case, can be observed in the case the ICA feature extraction, the accuracy obtained is comparable or higher when comparing with other methods, indicating a necessity of non-linear kernel when using ICA feature extraction.

The performance of the models is also evaluated on the UCI (Dua & Graff, 2017) datasets. Table 16 show that the performance of the proposed Pin-UTSVM model is better compared to the baseline models.

### 4.3 Statistical analysis

Here, we evaluate the performance of the various models statistically via pairwise sign test, the Friedman test and ANOVA & Tukey-Kramer Test.

#### 4.3.1 ANOVA & Tukey-Kramer (TK) test

The evaluation is performed for both linear and Gaussian kernel cases. We first analyse ANOVA ( $\alpha = 0.05$ ) results to determine if there exists a significance difference between the models. If a difference exists, then the TK post hoc analysis is used to determine the specific differences (Montgomery, 2017). The TK test uses the  $q$  statistic as defined below when the sample sizes of the various groups are equal.

$$q = \frac{\bar{Y}_i - \bar{Y}_j}{s.e.} \quad (63)$$

where,  $\bar{Y}_i$  is the mean of samples of Group  $i$ , standard error  $s.e. = MS(\text{within group})/n$  and  $n$  is the number of samples.

For the linear case we can observe from Table 6 that  $F > F_{crit}$  and  $P$  value  $< \alpha$ , indicating a significant difference between groups. Using a studentised  $q$  table we can find closest value of  $q_{crit}$  for  $\alpha = 0.05$ ,  $df = 220$ ,  $k = 4$  as 3.659. The Table 7 calculates the  $q$  values for the proposed method Pin-UTSVM vs TWSVM, Pin-GTSVM and UTSVM. Since all the calculated values of  $q$  are smaller than  $q_{crit}$ , there is a significant difference between Pin-UTSVM and all other methods.

**Table 6** ANOVA results for Linear case

Source of variation	SS	df	MS	<i>F</i>	<i>P</i> value	<i>F</i> <sub>crit</sub>
Between groups	0.970296763	3	0.323432254	102.589472	1.43427E-41	2.645636537
Within groups	0.693590625	220	0.003152685			

**Table 7** Tukey-Kramer analysis for Linear case

Group 1	Group 2	$\bar{Y}_1 - \bar{Y}_2$	<i>n</i>	<i>s.e.</i>	<i>q</i>	<i>q</i> <sub>crit</sub>	Significant difference
Pin-UTSVM	TWSVM	0.0453	56	0.007503195	6.033143891	3.659	Yes
Pin-UTSVM	Pin-GTSVM	0.1756	56	0.007503195	23.40669435	3.659	Yes
Pin-UTSVM	UTSVM	0.0431	56	0.007503195	5.747551281	3.659	Yes

**Table 8** ANOVA results for Non-Linear case

Source of variation	SS	df	MS	<i>F</i>	<i>P</i> value	<i>F</i> <sub>crit</sub>
Between groups	0.060916406	3	0.020305469	46.37872699	2.88248E-23	2.645636537
Within groups	0.096320089	220	0.000437819			

**Table 9** Tukey-Kramer analysis for Non-Linear case

Group 1	Group 2	$\bar{Y}_1 - \bar{Y}_2$	<i>n</i>	<i>s.e.</i>	<i>q</i>	<i>q</i> <sub>crit</sub>	Significant difference
Pin-UTSVM	TWSVM	0.0245	56	0.002796102	8.749423853	3.659	Yes
Pin-UTSVM	Pin-GTSVM	0.0454	56	0.002796102	16.22155955	3.659	Yes
Pin-UTSVM	UTSVM	0.0146	56	0.002796102	5.204949227	3.659	Yes

For the non-linear case we can observe from Table 8 that  $F > F_{crit}$  and  $P$  value  $< \alpha$ , indicating a significant difference between groups. Using a studentised  $q$  table we can find closest value of  $q_{crit}$  for  $\alpha = 0.05$ ,  $df = 220$ ,  $k = 4$  as 3.659. The Table 9 calculates the  $q$  values for the proposed method Pin-UTSVM vs TWSVM, Pin-GTSVM and UTSVM. Since all the calculated values of  $q$  are smaller than  $q_{crit}$ , there is a significant difference between Pin-UTSVM and all other methods.

### 4.3.2 Friedman test

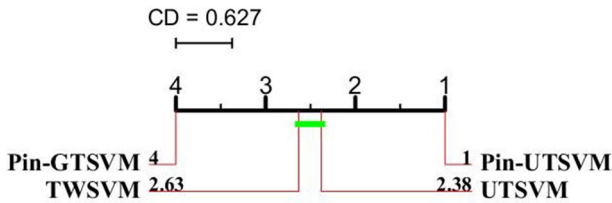
The evaluation is performed for both linear and Gaussian kernel cases. We follow Friedman test to analyse the significance of the models. In Friedman test, the performance of the model is ranked on each dataset with the worse performing models assigned a higher rank and vice versa. Let  $r_j^i$  be the rank of the  $j$ th model on the  $i$ th dataset. Suppose  $K$  classification models are evaluated on  $T$  number of datasets, then the average rank of the  $j^{th}$  classifier  $R_j = \sum_{i=1}^T r_j^i$ . The Friedman statistic follows  $\chi_F^2$  distribution with  $(K - 1)$  degrees of freedom and is given as

$$\chi_F^2 = \frac{12T}{K(K + 1)} \left[ \sum_j R_j^2 - \frac{K(K + 1)^2}{4} \right]. \tag{64}$$



**Table 10** Significant difference of models with linear kernel based on Nemenyi post-hoc test

	TWSVM (2.63)	Pin-GTSVM (4)	UTSVM (2.38)
TWSVM (2.63)		1.37	
Pin-GTSVM (4)			1.62
Proposed Pin-UTSVM (1)	1.63	3	1.38



**Fig. 2** Nemenyi critical difference for linear case

Friedman statistic is undesirably conservative, hence, a better statistic is given as

$$F_F = \frac{(T - 1)\chi_F^2}{T(K - 1) - \chi_F^2}, \tag{65}$$

which follows  $F_F$  distribution with  $(K - 1)$  and  $(K - 1)(T - 1)$  degrees of freedom. Under the null hypothesis, all models are performing identically and their average ranks are equal. If the null hypothesis is void, then Nemenyi test is followed for the pairwise comparison of the models. In Nemenyi posthoc test, two models are equivalent if their average rank difference is less than the critical difference (CD). Mathematically,

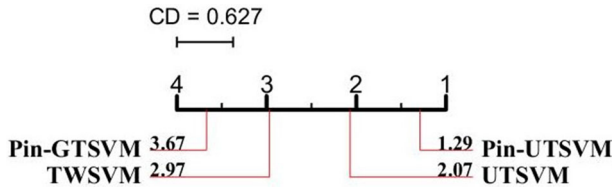
$$CD = q_{\alpha} \sqrt{\frac{K(K + 1)}{6T}}. \tag{66}$$

After simple calculations, the average rank of the TWSVM, Pin-GTSVM, UTSVM, and proposed Pin-UTSVM for linear case are 2.636, 4, 2.38, and 1, respectively. At 5% level of significance,  $\chi_F^2 = 153.9317$ ,  $F_F = 601.7958$ . With  $T = 56$ ,  $K = 4$ . From statistical table,  $F_F(3, 165) = 2.65$ . Since  $601.7958 > 2.65$ , hence, the null hypothesis is void and substantial differences exist among the classification models. To check the significant difference within the models, we follow Nemenyi test. Given  $q_{\alpha=0.05} = 2.5690$ , the  $CD = 0.63$ . Table 10 gives the significant difference among the models. Here, the blank entries denote that methods in the corresponding row and column are equal and no significant difference exists among them. The numbers in the corresponding cells denote the average rank difference among the models. Each entry denotes that significant difference exists among the methods of row and column, with row method performing better than the corresponding column method. One can see that the proposed Pin-UTSVM model is significantly better compared to the existing TWSVM, Pin-GTSVM and UTSVM models. Figure 2 gives the pictorial representation of the significant difference. The models connected via line denote that they are performing equally and no significant difference exists among them. Thus, one can see that proposed Pin-UTSVM model is significantly better compared to the existing models.

The average rank of the TWSVM, Pin-GTSVM, UTSVM, and proposed Pin-UTSVM with Gaussian kernel are 2.97, 3.67, 2.07, and 1.29, respectively. With simple calculation at

**Table 11** Significant difference of models with Gaussian kernel based on Nemenyi post-hoc test

	TWSVM (2.97)	Pin-GTSVM (3.67)	UTSVM (2.07)
TWSVM (2.97)		0.7	
Pin-GTSVM (3.67)			
UTSVM (2.07)	0.9	1.6	
Proposed Pin-UTSVM (1.29)	1.68	2.38	0.78



**Fig. 3** Nemenyi Critical difference for non-linear case

5% level of significance, we get  $\chi_F^2 = 108.8237$ ,  $F_F = 101.1436$ . With  $T = 56$ ,  $K = 4$ ,  $F_F(3, 165) = 2.65$ . As  $101.1436 > 2.65$ , we dismiss the null hypothesis i.e. substantial differences exist in the classification models. To check the differences within the models, we follow pairwise Nemenyi test. With  $q_{\alpha=0.05} = 2.5690$ , we have  $CD = 0.6268$ . Thus, the models are significantly different if the average ranks of the models differ by at least 0.6268. Table 11 gives the pairwise comparison of the models. Here, the entries in each cell have similar meaning as given above. One can see that the proposed Pin-UTSVM models is performing significantly better compared to the baseline models. Figure 3 gives the pictorial comparison of the models.

### 4.3.3 Statistical analysis based on pair wise: sign test

In pairwise sign test, the number of datasets on which a model is winner or loser is counted along with the count of ties (if any). Then sign test is followed for the pairwise comparison. Under the null hypothesis, the two methods perform identically thus each model wins on approximately  $T/2$  datasets of  $T$  datasets. The model is significantly better if it wins on  $T/2 + 1.96\sqrt{T}/2$  datasets with  $p < 0.05$ . If there is tie between the algorithms, then the number of ties is evenly distributed if the number is even. In case of odd number of ties, we ignore one. With simple calculations, the number of win-tie-loss count for each pair of models for linear and non-linear cases are given in Table 12 and Table 14, respectively. Each entry of a table is given as  $[a, b, c]$  which denotes that the row model wins  $a$  times, ties  $b$  times and loses  $c$  times with respect to the column method. With simple calculations, significant difference exists among the models if the number of wins  $\geq 35.3336$ . Table 13 shows that the proposed Pin-UTSVM is significantly better compared to the baseline models. Also, Table 15 gives the significant difference of the models with Gaussian kernel. The methods not reported in Table 13 and Table 15 doesn't show any significant difference. One can see that proposed Pin-UTSVM model is significantly better compared to the baseline models. Also, UTSVM is significantly better compared to the TWSVM and the Pin-GTSVM models.

**Table 12** Sign test with linear case

	TWSVM	Pin-GTSVM	UTSVM
Pin-GTSVM	[0, 0, 56]		
UTSVM	[14, 42, 0]	[56, 0, 0]	
Proposed Pin-UTSVM	[56, 0, 0]	[56, 0, 0]	[56, 0, 0]

**Table 13** Significant difference among the models with linear case via pairwise sign test

	TWSVM	Pin-GTSVM	UTSVM
Proposed Pin-UTSVM	✓	✓	✓

**Table 14** Sign test with Gaussian kernel

	TWSVM	Pin-GTSVM	UTSVM
Pin-GTSVM	[12, 2, 42]		
UTSVM	[34, 22, 0]	[54, 2, 0]	
Proposed Pin-UTSVM	[52, 1, 3]	[51, 1, 4]	[47, 2, 7]

**Table 15** Significant difference among the models with Gaussian kernel via pairwise sign test

	TWSVM	Pin-GTSVM	UTSVM
UTSVM	✓	✓	
Proposed Pin-UTSVM	✓	✓	✓

#### 4.4 Parameter sensitivity analysis

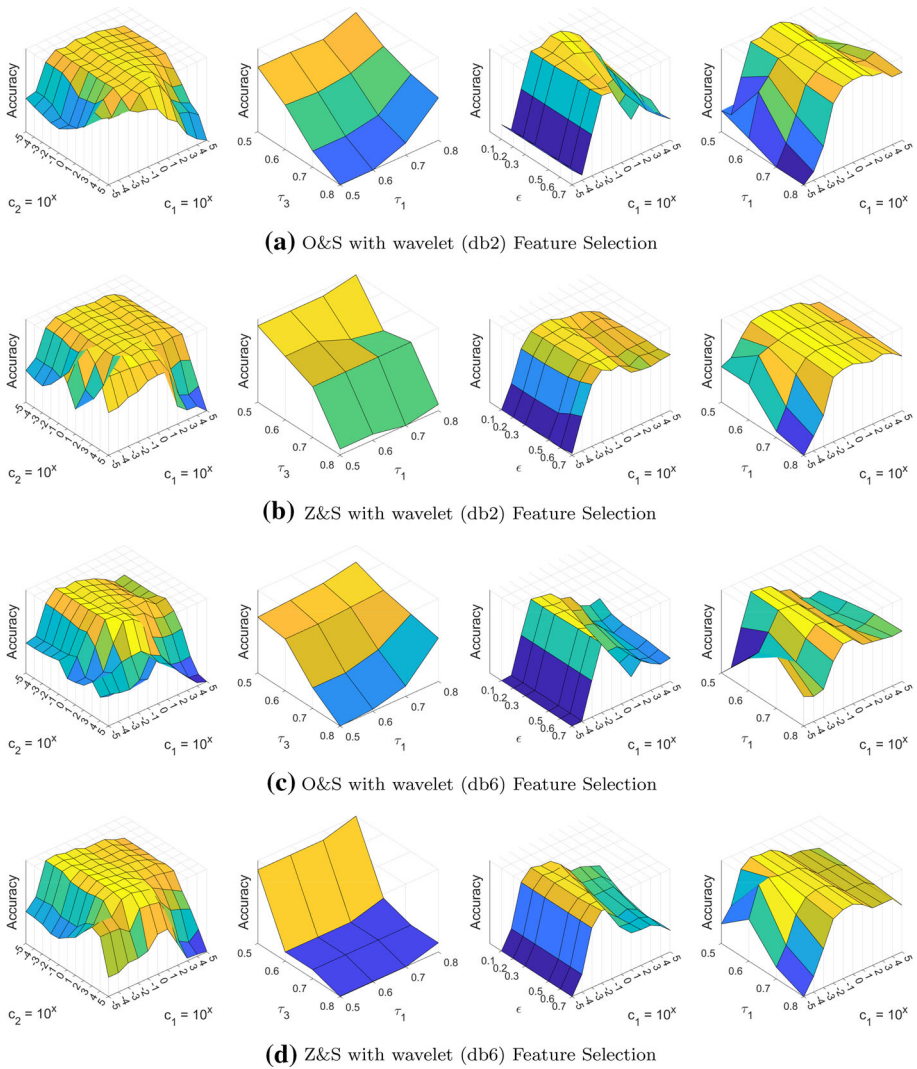
Figure 4 presents the effect of hyperparameters on the accuracy of our proposed algorithm. We present four graphs for each dataset, Accuracy vs (i)  $c_1$  vs  $c_2$ , (ii)  $\tau_1$  vs  $\tau_3$ , (iii)  $c_1$  vs  $\epsilon$  and (iv)  $c_1$  vs  $\tau_1$ . While plotting a graph against two parameters, the rest of the parameters are kept at their optimal values. Here, we have only presented the four most relevant graphs for appropriately selecting the hyperparameters. From all the (i)  $c_1$  vs  $c_2$  graphs, one can conclude that the values of  $c_1$  and  $c_2$  have inverse effects on the accuracy, in general, large  $c_1$  and small  $c_2$  leads to better accuracy. From all the (ii)  $\tau_1$  vs  $\tau_3$  graphs, we can observe that  $\tau_1$  and  $\tau_3$  have a similar relationship as  $c_1$  and  $c_2$ , respectively, i.e., large  $\tau_1$  and small  $\tau_3$  leads to better accuracy. From all the (iii)  $c_1$  vs  $\epsilon$  graphs, we can note that  $\epsilon$  has minimal effect on accuracy, but lower values generally lead to better accuracy. We can again note that lower values of  $c_1$  lead to abysmal accuracies. From all the (iv)  $c_1$  vs  $\tau_1$  graphs, we can note that at higher values of  $c_1$ ,  $\tau_1$  has minimal effect on the accuracy, but in general, we can observe a minor improvement from larger values of  $\tau_1$ .

#### 4.5 Noise sensitivity analysis

The primary purpose of implementing the pinball loss function in our proposed method Pin-UTSVM is to enable noise insensitivity. To test this property, we introduced Gaussian noise to EEG data with varying standard deviations. The results of this experiment are tabulated in Tables 2 and 4. To properly analyse the noise insensitivity performance, we present Fig. 5.

**Table 16** Evaluation of the models on UCI datasets using Gaussian Kernel

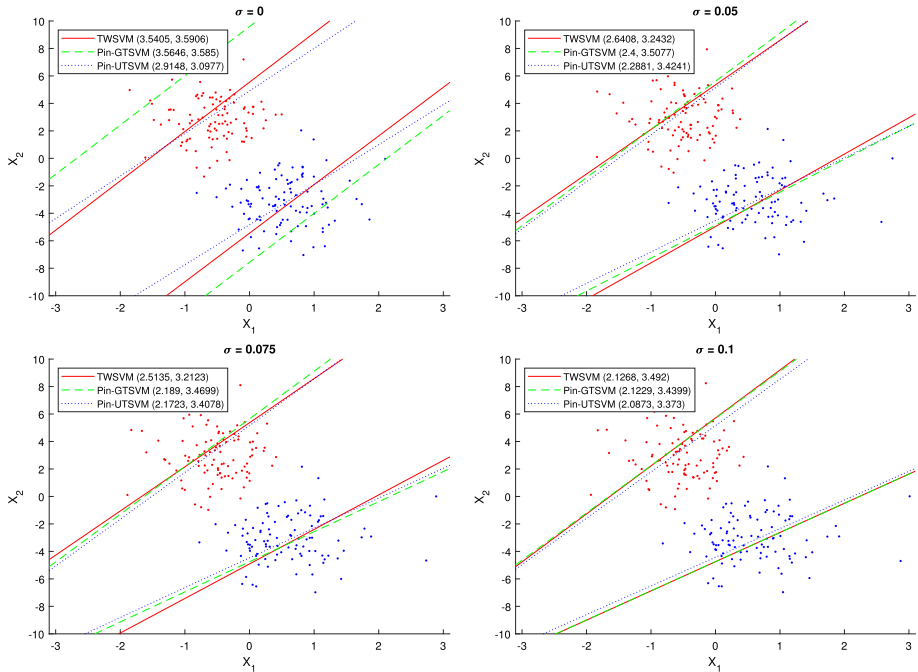
Dataset	$\sigma$	TWSVM (Jayadeva & Khemchandani, 2007)	Pin-GTSVM (Tanveer et al., 2019)	UTSVM (Qi et al., 2012)	Pin-UTSVM
Banknote	0	1	1	1	1
crossplane150	0	1	1	1	1
crossplane500	0	1	1	1	1
wdbc	0	0.978947	0.977193	0.978916	0.977162
monks-1	0	0.996396	0.998214	0.996396	0.998182
breast-cancer-wisconsin	0	0.975107	0.967818	0.975107	0.975085
brwisconsin	0	0.975107	0.967818	0.975107	0.975085
wdbc	0	0.762105	0.778421	0.767368	0.766579
iono	0	0.954444	0.92873	0.943095	0.957302
Banknote	0.05	1	1	1	1
crossplane500	0.05	0.988	0.988	0.99	0.992
wdbc	0.05	0.975439	0.975439	0.98067	0.977162
brwisconsin	0.05	0.976556	0.969288	0.976556	0.978026
monks-1	0.05	0.978377	0.991006	0.978377	0.980162
breast-cancer-wisconsin	0.05	0.976556	0.963448	0.975107	0.976577
ger	0.05	0.739	0.731	0.74	0.754
crossplane150	0.05	0.98	0.973333	0.98	1
heart-stat	0.05	0.840741	0.803704	0.848148	0.851852
cmc	0.05	0.597417	0.655782	0.598097	0.629992
Banknote	0.1	0.997816	0.997816	0.997816	0.998545
crossplane500	0.1	0.99	0.99	0.99	0.992
brwisconsin	0.1	0.975085	0.969309	0.975085	0.975107
wdbc	0.1	0.971898	0.977193	0.977162	0.978916
breast-cancer-wisconsin	0.1	0.973657	0.967818	0.975085	0.975107
cmc	0.1	0.587227	0.618418	0.588587	0.599476
monks-1	0.1	0.981981	0.983799	0.956818	0.956753
Average accuracy		92.97	92.98	92.94	93.33



**Fig. 4** Effect of various hyperparameters on the performance of proposed Pin-UTSVM model

In Fig. 5, we have generated a two-dimensional synthetic dataset with two classes, each from a Gaussian distribution such that  $x_i, i \in \{i : y_i = 1\} \sim \mathcal{N}(\mu_1, \Sigma_1)$  (represented by blue in Fig. 5) and  $x_i, i \in \{i : y_i = -1\} \sim \mathcal{N}(\mu_2, \Sigma_2)$  (represented by red in Fig. 5). For the universum data required for UTSVM and Pin-UTSVM, we used points belonging to  $\mathcal{N}(\mu_3, \Sigma_3)$ , where  $\mu_1 = [0.5, -3]^T, \mu_2 = [-0.5, 3]^T, \mu_3 = [-0.5, -3]^T$  and  $\Sigma_1 = \Sigma_2 = \Sigma_3 = \begin{bmatrix} 0.2 & 0 \\ 0 & 3 \end{bmatrix}$ . The ideal separating hyperplane for the above two Gaussian distribution is given by the Bayes classifier viz.  $f_c(x) = 2.5x_1 - x_2$ , i.e. a slope of 2.5. We have plotted the hyperplanes for different methods and have provided the slopes in the legend.

The  $\sigma$  (Std. deviation) = 0 plot represents data without any noise. Next, we introduce Gaussian noise to these data points, with  $\sigma = [0.05, 0.075, 0.1]$ . This noise changes the



**Fig. 5** The above four figures depict the noise insensitivity property of the pinball loss via four noise levels of a synthetic dataset

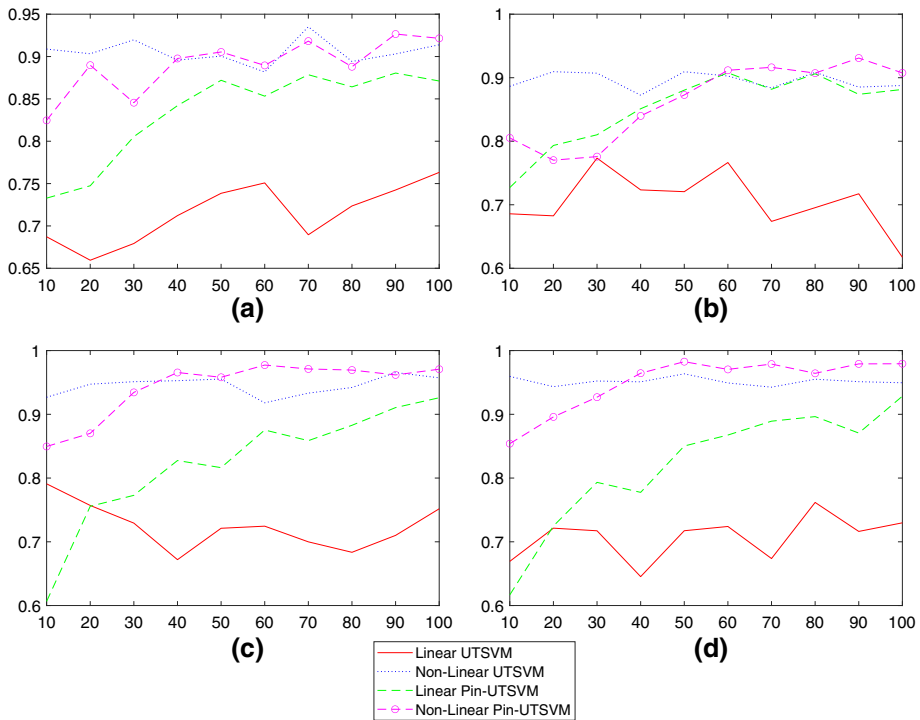
distribution of the data points, thus disrupting the hyperplanes. One can observe that the hyperplanes deviate more with an increase in noise. This deviation is significantly less severe in Pin-UTSVM than other methods, implying better insensitivity to noise.

**4.6 Effect of universum data**

We also present the accuracy obtained for various selections of the number of universum points used for datasets (a) O&S using db1, (b) O&S using db6, (c) Z&S using Haar, (d) Z&S using db6 in Fig. 6. From Fig. 6, one can observe that the proposed method performs better than UTSVM in most cases for both linear and non-linear variations. Linear UTSVM being a minor exception, as a general trend, increasing the number of universum data points leads to better accuracy. We observe plateauing accuracy at around 40 – 60 universum data points in nearly all the cases and techniques, thus suggesting the minimum number of universum points to be used as around 50 % of the training dataset size.

**4.7 Performance of the models on different feature extraction**

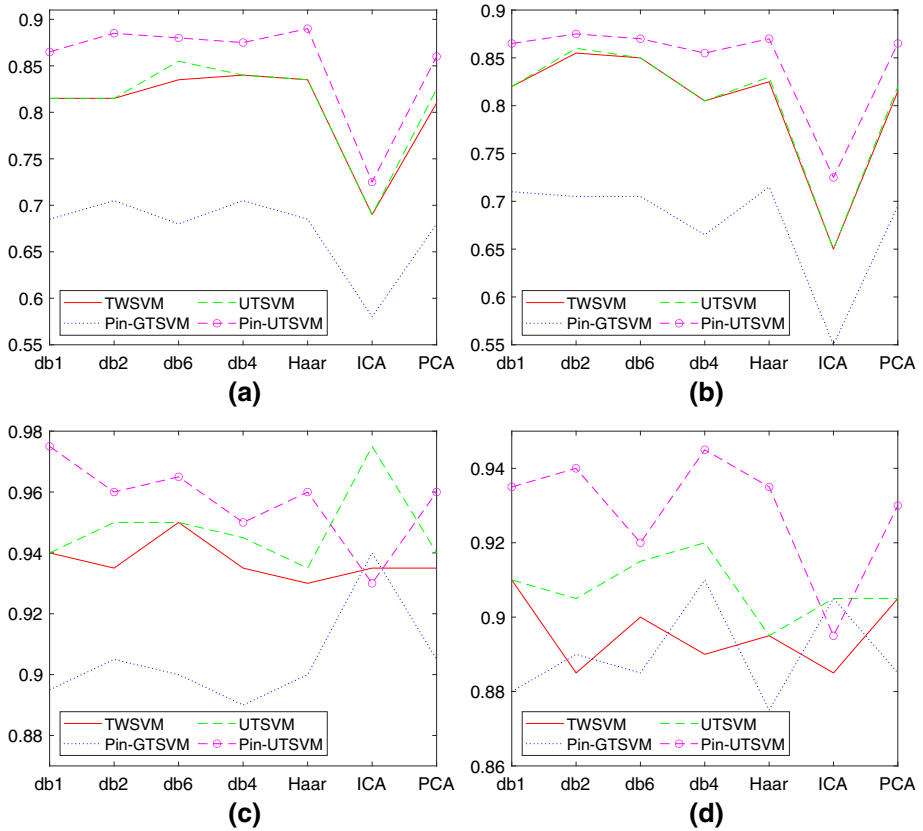
Figure 7 presents the accuracies obtained for different algorithms for various Feature extraction techniques. One can observe that the proposed algorithm performs the best for nearly all the feature selection algorithms for both Z&S and O&S. One can observe that ICA produces unreliable results, nearly all classification algorithms suffer due to it (as seen by a dip in accuracy graph). But ICA also produces the best accuracy in the case of (c).



**Fig. 6** Performance comparison of linear and non-linear UTSVM and Pin-UTSVM w.r.t. number of universum samples used for **a** O&S using wavelet (db1), **b** O&S using wavelet (db6), **c** Z&S using Haar and **d** Z&S using wavelet (db6) feature extraction technique

### 5 Conclusion

In this paper, we proposed a novel universum twin SVM with pinball loss functions (Pin-UTSVM) for the EEG signal classification. Pinball loss function has been widely used in the literature of classification and regression due its association between quantile regression. Here, we incorporated pinball loss function in baseline universum TWSVM model. The proposed Pin-UTSVM model is stable for resampling data and is insensitive to noise. To show the efficacy of our proposed Pin-UTSVM model, we evaluated it for the classification of the EEG signal. Experimental results and the statistical tests prove the competence of the proposed Pin-UTSVM compared to the standard models. To prove the robust of proposed Pin-UTSVM model in presence of noise, we corrupted the EEG signals with different levels of noise and evaluated the classification models. The results show that the proposed Pin-UTSVM model is effective in presence of noise while as the existing baseline models suffer. In future, one can apply this algorithm in the diagnosis of other diseases like Alzheimer’s disease. Moreover, one can also focus on improving the performance of the model via efficient optimization algorithms. The source code will be available at <https://github.com/mtanveer1>.



**Fig. 7** Accuracy comparison against various feature extraction techniques for classification of EEG signals using different algorithms for **a** Z&S using linear kernel, **b** O&S dataset using linear kernel, **c** Z&S using non-linear kernel and **d** O&S dataset using non-linear kernel

**Acknowledgements** The funding for this work is provided by the National Supercomputing Mission under DST and Miety, Govt. of India under Grant No. DST/NSM/R&D\_HPC\_Appl/2021/03.29, as well as the Department of Science and Technology under Interdisciplinary Cyber Physical Systems (ICPS) Scheme grant no. DST/ICPS/CPS-Individual/2018/276 and Science and Engineering Research Board (SERB), Government of India under Ramanujan Fellowship Scheme, Grant No. SB/S2/RJN-001/2016. We also acknowledge Council of Scientific & Industrial Research (CSIR), New Delhi, INDIA for funding under Extra Mural Research (EMR) Scheme grant no. 22(0751)/17/EMR-II. We are grateful to IIT Indore for the facilities and support being provided.

## References

- Adeli, H., Zhou, Z., & Dadmehr, N. (2003). Analysis of EEG records in an epileptic patient using wavelet transform. *Journal of Neuroscience Methods*, *123*(1), 69–87.
- 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. *Physical Review E*, *64*(6), 061907.
- Bartlett, M. S., Movellan, J. R., & Sejnowski, T. J. (2002). Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, *13*(6), 1450–1464.



- Bollegala, D., Matsuo, Y., & Ishizuka, M. (2010). A web search engine-based approach to measure semantic similarity between words. *IEEE Transactions on Knowledge and Data Engineering*, 23(7), 977–990.
- Chen, S., Cao, J., Chen, F., & Liu, B. (2020). Entropy-based fuzzy least squares twin support vector machine for pattern classification. *Neural Processing Letters*, 51(1), 41–66.
- Christmann, A., & Steinwart, I. (2007). How SVMs can estimate quantiles and the median. *Advances in Neural Information Processing Systems*, 20, 305–312.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
- Dua, D., & Graff, C. (2017). UCI machine learning repository.
- Ganaie, M., & Tanveer, M. (2021). Robust general twin support vector machine with pinball loss function. *Machine Learning for Intelligent Multimedia Analytics* (pp. 103–125). Berlin: Springer.
- Güler, I., & Übeyli, E. D. (2005). Adaptive neuro-fuzzy inference system for classification of EEG signals using wavelet coefficients. *Journal of Neuroscience Methods*, 148(2), 113–121.
- Gupta, D., Sarma, H. J., Mishra, K., & Prasad, M. (2019). Regularized universum twin support vector machine for classification of eeg signal. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, pp. 2298–2304.
- Gupta, D., Richhariya, B., & Borah, P. (2019). A fuzzy twin support vector machine based on information entropy for class imbalance learning. *Neural Computing and Applications*, 31(11), 7153–7164.
- Huang, X., Shi, L., & Suykens, J. A. (2013). Support vector machine classifier with pinball loss. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5), 984–997.
- Jayadeva, R., & Khemchandani, S. (2007). Chandra, Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5), 905–910.
- Khalilpourazari, S., & Doulabi, H. H. (2021). Using reinforcement learning to forecast the spread of covid-19 in france. In *2021 IEEE International Conference on Autonomous Systems (ICAS)*. IEEE, pp. 1–8.
- Khalilpourazari, S., & Doulabi, H. Hashemi. (2021). Designing a hybrid reinforcement learning based algorithm with application in prediction of the covid-19 pandemic in quebec. *Annals of Operations Research*, 321(2), 1261–1305.
- Khemchandani, R., Saigal, P., & Chandra, S. (2018). Angle-based twin support vector machine. *Annals of Operations Research*, 269(1), 387–417.
- Koenker, R., & Hallock, K. F. (2001). Quantile regression. *Journal of Economic Perspectives*, 15(4), 143–156.
- Kumar, M. A., & Gopal, M. (2009). Least squares twin support vector machines for pattern classification. *Expert Systems with Applications*, 36(4), 7535–7543.
- Kumar, B., & Gupta, D. (2021). Universum based lagrangian twin bounded support vector machine to classify eeg signals. *Computer Methods and Programs in Biomedicine*, 208, 106244.
- Li, K., & Lv, Z. (2021). Smooth twin bounded support vector machine with pinball loss. *Applied Intelligence*, 51(8), 5489–5505.
- Montgomery, D. C. (2017). *Design and analysis of experiments*. New York: John Wiley & sons.
- Qi, Z., Tian, Y., & Shi, Y. (2012). Twin support vector machine with universum data. *Neural Networks*, 36, 112–119.
- Rezvani, S., Wang, X., & Pourpanah, F. (2019). Intuitionistic fuzzy twin support vector machines. *IEEE Transactions on Fuzzy Systems*, 27(11), 2140–2151.
- Richhariya, B., & Gupta, D. (2019). Facial expression recognition using iterative universum twin support vector machine. *Applied Soft Computing*, 76, 53–67.
- Richhariya, B., & Tanveer, M. (2018). EEG signal classification using universum support vector machine. *Expert Systems with Applications*, 106, 169–182.
- Richhariya, B., & Tanveer, M. (2020). A reduced universum twin support vector machine for class imbalance learning. *Pattern Recognition*, 102, 107150.
- Richhariya, B., & Tanveer, M. (2021). A fuzzy universum least squares twin support vector machine (FUL-STSVM). *Neural Computing and Applications*, 34(14), 11411–11422.
- Shao, Y.-H., Zhang, C.-H., Wang, X.-B., & Deng, N.-Y. (2011). Improvements on twin support vector machines. *IEEE Transactions on Neural Networks*, 22(6), 962–968.
- Sharma, S., Rastogi, R., & Chandra, S. (2019). Large-scale twin parametric support vector machine using pinball loss function. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(2), 987–1003.
- Subasi, A., & Gursoy, M. I. (2010). EEG signal classification using PCA, ICA, LDA and support vector machines. *Expert Systems with Applications*, 37(12), 8659–8666.
- Tanveer, M., Rajani, T., Rastogi, R., Shao, Y., & Ganaie, M. A. (2021). *Comprehensive review on twin support vector machines*. arXiv preprint [arXiv:2105.00336](https://arxiv.org/abs/2105.00336)
- Tanveer, M., Tiwari, A., Choudhary, R., & Ganaie, M. (2021). Large-scale pinball twin support vector machines. *Machine Learning*, 1–24.
- Tanveer, M. (2015). Robust and sparse linear programming twin support vector machines. *Cognitive Computation*, 7(1), 137–149.

- Tanveer, M. (2015). Application of smoothing techniques for linear programming twin support vector machines. *Knowledge and Information Systems*, 45(1), 191–214.
- Tanveer, M., Gautam, C., & Suganthan, P. N. (2019). Comprehensive evaluation of twin SVM based classifiers on UCI datasets. *Applied Soft Computing*, 83, 105617.
- Tanveer, M., Khan, M. A., & Ho, S.-S. (2016). Robust energy-based least squares twin support vector machines. *Applied Intelligence*, 45(1), 174–186.
- Tanveer, M., Richhariya, B., Khan, R., Rashid, A., Khanna, P., Prasad, M., & Lin, C. (2020). Machine learning techniques for the diagnosis of Alzheimer's disease: A review. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(1s), 1–35.
- Tanveer, M., Sharma, S., Rastogi, R., & Anand, P. (2021). Sparse support vector machine with pinball loss. *Transactions on Emerging Telecommunications Technologies*, 32(2), e3820.
- Tanveer, M., Sharma, A., & Suganthan, P. N. (2019). General twin support vector machine with pinball loss function. *Information Sciences*, 494, 311–327.
- Tanveer, M., Tiwari, A., Choudhary, R., & Jalan, S. (2019). Sparse pinball twin support vector machines. *Applied Soft Computing*, 78, 164–175.
- Tsang, I. W., Kocsor, A., & Kwok, J. T. (2006). Efficient kernel feature extraction for massive data sets. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 724–729.
- Wang, T.-Y., & Chiang, H.-M. (2007). Fuzzy support vector machine for multi-class text categorization. *Information Processing Management*, 43(4), 914–929.
- 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.
- Xu, Y., Chen, M., & Li, G. (2016). Least squares twin support vector machine with universum data for classification. *International Journal of Systems Science*, 47(15), 3637–3645.
- Xu, Y., Yang, Z., & Pan, X. (2016). A novel twin support-vector machine with pinball loss. *IEEE Transactions on Neural Networks and Learning Systems*, 28(2), 359–370.
- Yan, H., Ye, Q.-L., & Yu, D.-J. (2019). Efficient and robust twsvm classification via a minimum  $l_1$ -norm distance metric criterion. *Machine Learning*, 108(6), 993–1018.
- Yuan, M., & Xu, Y. (2021). Bound estimation-based safe acceleration for maximum margin of twin spheres machine with pinball loss. *Pattern Recognition*, 114, 107860.
- Zhang, T., Chen, W., & Li, M. (2019). Classification of inter-ictal and ictal EEGs using multi-basis modwpt, dimensionality reduction algorithms and LS-SVM: A comparative study. *Biomedical Signal Processing and Control*, 47, 240–251.
- Zhou, X., Jiang, W., Tian, Y., & Shi, Y. (2010). Kernel subclass convex hull sample selection method for SVM on face recognition. *Neurocomputing*, 73(10–12), 2234–2246.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.