This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2021.3059910, IEEE Journal of Biomedical and Health Informatics



# Sparse Twin Support Vector Clustering using Pinball Loss

M. Tanveer, *Senior Member, IEEE,* Tarun Gupta, Miten Shah, and Bharat Richhariya

***Abstract*—Clustering is a widely used machine learning technique for unlabelled data. One of the recently proposed techniques is the twin support vector clustering (TWSVC) algorithm. The idea of TWSVC is to generate hyperplanes for each cluster. TWSVC utilizes the hinge loss function to penalize the misclassification. However, the hinge loss relies on shortest distance between different clusters, and is unstable for noise-corrupted datasets, and for re-sampling. In this paper, we propose a novel Sparse Pinball loss Twin Support Vector Clustering (SPTSVC). The proposed SPTSVC involves the $\epsilon$-insensitive pinball loss function to formulate a sparse solution. Pinball loss function provides noise-insensitivity and re-sampling stability. The $\epsilon$-insensitive zone provides sparsity to the model and improves testing time. Numerical experiments on synthetic as well as real world benchmark datasets are performed to show the efficacy of the proposed model. An analysis on the sparsity of various clustering algorithms is presented in this work. In order to show the feasibility and applicability of the proposed SPTSVC on biomedical data, experiments have been performed on epilepsy and breast cancer datasets.**

***Index Terms*—TWSVC, SVM, pinball loss, quantile distance, sparsity, noise insensitivity, noisy data.**

## I. INTRODUCTION

SUPPORT vector machines (SVMs) have proven to be one of the most accurate classification techniques in the past few decades [1]. The solution of SVM involves a quadratic programming problem (QPP) to generate a classifying hyperplane. SVM and its variants have been applied in various applications such as Alzheimer's disease [2], [3], image classification [4], investor sentiment classification [5], EEG classification [6], and breast cancer [7]. The solution of SVM requires the solution of a QPP of large size. This incurs high computation time for the SVM algorithm. In order to reduce the computational complexity of SVM, Jayadeva et al. [8] proposed a novel technique termed as twin support vector machine (TWSVM). In TWSVM, instead of one large QPP, two small QPPs are solved to generate twin hyperplanes. Shao et al. [9] proposed a twin bounded support vector machine (TBSVM) to embody the structural risk minimization principle (SRM) of statistical theory in TWSVM. The SRM principle is introduced in TBSVM by including a regularization term in the objective function of TWSVM. Some twin SVM based models are proposed for multiclass classification [10] as well. Tanveer [11] used a smoothing technique to propose a smooth linear programming TWSVM (SLPTSVM). It leads to unconstrained optimization problems, which in turn reduces the computation cost. The smoothing function is used to approximate the loss, and makes it differentiable. Moreover, Newton method is used to propose an iterative implicit Lagrangian TWSVM algorithm [12]. Another improvement over SVM is proposed as least squares twin support vector machine (LSTSVM) [13], which only involves linear equations in the solution. An energy-based LSTSVM (RELS-TSVM) [14] performed well on benchmark datasets in a recent survey [15].

To reduce the effect of noise in SVM, various techniques have been proposed, such as fuzzy based techniques [16], [17]. To improve the noise insensitivity, Huang et al. [18] proposed a pinball loss function for SVM. More efficient models based on pinball loss are also proposed in the past. A twin parametric-margin SVM using pinball loss (Pin-TPMSVM) is also proposed [19], based on the twin parametric-margin approach. This leads to decrease in computational time of pin-SVM algorithm. Recently, a general pinball twin SVM (Pin-GTSVM) was proposed for pattern classification. The Pin-GTSVM model is in the spirit of the original TWSVM model of classification, rather than the parametric version as in [19]. In order to include sparsity in the Pin-GTSVM, Tanveer et al. [20] proposed a sparse pinball twin support vector machine (SPTWSVM). The solution of SPTWSVM is sparser than Pin-GTSVM. This leads to better generalization performance in SPTWSVM with lesser testing time. By including the SRM principle in Pin-GTSVM, an improved sparse pinball twin support vector machine (ISPTSVM) [21] is proposed to improve the generalization performance. However, most of these pinball loss based models have been used for classification problems.

Clustering algorithms are very useful in real world applications, since a large amount of data is unlabelled in nature. Many algorithms are proposed for clustering in the past [22]. Support vector clustering (SVC) is proposed [23] for unlabelled data. An efficient formulation termed as twin support vector clustering (TWSVC) [24] is proposed using the concave-convex programming (CCCP). To improve the computation time, a fuzzy based least squares TWSVC (FLSTWSVC) is formulated [25]. However, the additional fuzzy function in FLSTWSVC incurs additional computational cost. To include the SRM principle in clustering algorithms, a twin bounded support vector clustering (TBSVC) [26] was proposed, using additional regularization term in the formulation of TWSVC. Further, a ramp loss based twin support vector

Manuscript received –, 2020.

M. Tanveer is with the Department of Mathematics, Indian Institute of Technology Indore, Simrol, Indore 453552, India (e-mail: mtanveer@iiti.ac.in).

Tarun Gupta and Miten Shah are with the Department of Computer Science and Engineering, Indian Institute of Technology Indore, Simrol, Indore 453552, India (e-mail: tarungupta360@gmail.com; mitenshah16@gmail.com).

Bharat Richhariya is with the Department of Mathematics, Indian Institute of Technology Indore, Simrol, Indore 453552, India (e-mail: phd1701241001@iiti.ac.in).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2021.3059910, IEEE Journal of Biomedical and Health Informatics



clustering (RampTWSVC) [27] is proposed to bring robustness to classifier, in presence of points lying far away from their clusters. RampTWSVC utilises the within class and between class scatter information to improve the noise insensitivity. However, the ramp loss function is non-convex in nature. Recently, a least squares projection twin SVC (LSPTSVC) [28] is proposed, where the intra-class scatter is minimized, and inter-class scatter is maximized in the formulation. To include the pinball loss in the optimization problem of twin SVM based clustering algorithms, a pinball loss based twin support vector clustering (pinTSVC) [29] algorithm is proposed. This lead to improvement in the noise insensitivity of the clustering algorithm.

However, the generalization performance of SVM based models is determined by the sparsity of the solution obtained. Classifiers with higher sparsity level often give better generalization performance [20]. In order to include sparsity with noise insensitivity, motivated by the works on pinball twin SVM and TWSVC, we propose a novel sparse twin support vector clustering using pinball loss (SPTSVC). The formulation of proposed SPTSVC involves a $\epsilon$-insensitive pinball loss function to construct the clustering hyperplanes. The proposed SPTSVC solves the optimization problem with sparser solution compared to existing algorithms, while still enjoying properties of noise-insensitivity and re-sampling stability. The main contributions of our paper are as follows:

- A novel sparse pinball loss twin support vector clustering algorithm is proposed for noisy data.
- The proposed SPTSVC is insensitive to noisy data. Further, it is stable for re-sampling which is required for large scale datasets. Thus its applicable to various real world problems.
- The proposed SPTSVC leads to sparse solution to the optimization problem, thereby reducing testing time.
- An analysis on the sparsity of proposed and existing algorithms is presented in this work.
- Biomedical applications are presented for clustering by using image and time-series data.

We have organized the paper as follows: Section II gives brief description of recent works on clustering. In Section III, formulation of the proposed algorithm has been provided. Section IV discusses the advantages of the proposed SPTSVC model, while the experimental results are given in Section V. Section VI shows the applicability of the proposed model on biomedical data. The conclusions with future work are given in Section VII.

## II. Background

In this paper, $m$ denotes the number of samples. Each sample has $n$ number of features. Hence, every data sample of a dataset can be represented as $x_i = (x_i^1, x_i^2, ..., x_i^n)$ for $i = 1, 2, ..., m$.

Suppose there are $k$ clusters. Then for $i = 1, 2, ..., k$, matrix $X_i$ of dimensions $m_i \times n$ contains all points in the $i^{th}$ cluster. Rest of the points are collected in matrix $\hat{X}_i$ of dimensions $(m - m_i) \times n$. All vectors are column vectors, $L_2$ norm is denoted as $\|.\|$, and $e$ represents vector of ones of appropriate dimension.

### A. TWSVC

Based on the principles of TWSVM, TWSVC is proposed by Shao et al. [24]. TWSVC [24] obtains $k$ cluster center planes $\omega_i^T x + b_i = 0, i = 1, ..., k$. These cluster center planes are obtained by solving the following optimization problem:

$$\begin{aligned} \min_{\omega_i, b_i, \xi_i} \ & \frac{1}{2}\|X_i\omega_i + b_i e\|^2 + ce^T\xi_i \\ s.t. \quad & |\hat{X}_i\omega_i + b_i e| \geq e - \xi_i, \ \xi_i \geq 0, \end{aligned} \tag{1}$$

here $c > 0$ is the penalty parameter, and $\xi_i$ is the slack variable for bounding the error term.

From this formulation one can observe that the $i^{th}$ cluster center plane is close to the points $X_i$ and far away from the points $\hat{X}_i$. Shao et al. [24] showed that the above optimization problem can be decomposed into smaller convex quadratic sub-problems using concave-convex procedure (CCCP) [30] technique, which can then be optimized using Karush–Kuhn–Tucker (KKT) conditions [31].

$$\begin{aligned} \min_{\omega_i^{j+1}, b_i^{j+1}, \xi_i^{j+1}} \ & \frac{1}{2}\left\|X_i\omega_i^{j+1} + b_i^{j+1}e\right\|^2 + ce^T\xi_i^{j+1} \\ s.t. \quad & T(|\hat{X}_i\omega_i^{j+1} + b_i^{j+1}e|) \geq e - \xi_i^{j+1}, \ \xi_i^{j+1} \geq 0, \end{aligned} \tag{2}$$

where $j = 0, 1, 2, ...$ is the index of successive sub-problems and $c$ is the same penalty parameter. $T(\cdot)$ denotes the first order Taylor expansion. The sub-gradient of $|\hat{X}_i\omega_i^{j+1} + b_i^{j+1}e|$ can be calculated as $diag(sign(\hat{X}_i\omega_i^j + b_i^j e))$, resulting in the following:

$$T(|\hat{X}_i\omega_i^{j+1} + b_i^{j+1}e|) = Z_i(\hat{X}_i\omega_i^{j+1} + b_i^{j+1}e), \tag{3}$$

$$Z_i = diag(sign(\hat{X}_i\omega_i^j + b_i^j e)). \tag{4}$$

The solution to (2) can be obtained by solving its dual form. Using Karush–Kuhn–Tucker (KKT) conditions [31], we get the dual problem as:

$$\begin{aligned} \min_{\alpha} \ & \frac{1}{2}\alpha^T G(H^T H)^{-1}G^T\alpha - e^T\alpha \\ s.t. \quad & 0 \leq \alpha \leq ce, \end{aligned} \tag{5}$$

where $G = Z_i[\hat{X}_i \ e]$, $H = [X_i \ e]$, and $\alpha \in R^{m-m_i}$ is the the Lagrangian multiplier vector. After the solution of (5) is obtained, we obtain cluster plane parameters $[\omega_i^{j+1}; b_i^{j+1}] = (H^T H)^{-1}G^T\alpha$.

### B. Loss function

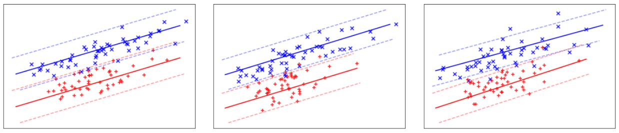
Pinball loss function [18] is defined as:

$$\mathcal{L}_\tau(u) = \begin{cases} u & u \geq 0, \\ -\tau u & u < 0, \end{cases}$$

here $\tau \in [0, 1]$. When $u < 0$, we still obtain a small penalty as opposed to zero penalty in hinge loss. Therefore, there is a smaller penalty for correctly classified points as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2021.3059910, IEEE Journal of Biomedical and Health Informatics





(a) $\sigma = 0$ (b) $\sigma = 0.05$ (c) $\sigma = 0.1$

Fig. 1: Plot showing different formation of clusters by linear SPTSVC for a two-dimensional dataset corrupted with Gaussian noise having zero-mean and $\sigma$ standard deviation.

well. This places an upper bound on the number of errors at decision boundaries, thus providing insensitivity to feature noise. However, since sub-gradient of pinball loss function is non-zero everywhere except origin, it leads to losing sparsity of the solution. To counter this limitation, $\epsilon$-insensitive pinball loss function [18] can be used:

$$\mathcal{L}_{\tau}^{\epsilon}(u) = \begin{cases} u-\epsilon & u>\epsilon, \\ 0 & -\frac{\epsilon}{\tau} \le u \le \epsilon \\ -\tau(u+\frac{\epsilon}{\tau}) & u< -\frac{\epsilon}{\tau}, \end{cases}$$

In range $[-\frac{\epsilon}{\tau}, \epsilon]$, the sub-gradient of the $\epsilon$-insensitive pinball loss function is $0$, thereby leading to sparsity. In this $\epsilon$-insensitive zone of width $\epsilon(1+\frac{1}{\tau})$, no penalty is given to any data point. This allows us to achieve both noise-insensitivity and sparsity at the same time.

## III. Proposed Sparse Pinball loss Twin Support Vector Clustering (SPTSVC)

We utilize the sparse pinball loss function to incorporate the property of noise-insensitivity and sparsity in the TWSVC formulation. In this section, we propose a Sparse Pinball loss Twin Support Vector Clustering (SPTSVC) algorithm for linear and non-linear cases.

### A. Linear SPTSVC

The proposed linear SPTSVC seeks $k$ cluster-center planes $[\omega_i, b_i]$, $i = 1,...,k$ by obtaining solution of the following optimization problem:

$$\begin{aligned} \min_{\omega_i, b_i, \xi_i} \; & \frac{1}{2}\|X_i\omega_i + b_i e\|^2 + ce^T\xi_i \\ s.t. \; & |\hat{X}_i\omega_i + b_i e| \ge e - \xi_i - e\epsilon, \\ & |\hat{X}_i\omega_i + b_i e| \le e + \frac{\xi_i}{\tau} + e\frac{\epsilon}{\tau} \\ & \xi_i \ge 0, \end{aligned} \tag{6}$$

here $\xi_i$ is the slack variable, $\tau \in [0,1]$ and $\epsilon \in [0,1]$ are the parameters of sparse pinball loss function. The first term of objective function seeks to minimize the sum of squared distance of points in $X_i$ from hyperplane $[\omega_i; b_i]$. The second term $\xi_i$ is the slack variable. The first and second constraints seeks to minimize the penalty arising from points in $\hat{X}_i$ within $1-\epsilon$ unit distance from hyperplane $[\omega_i; b_i]$, and also the points which are at-least $1+\frac{\epsilon}{\tau}$ distance away from hyperplane $[\omega_i; b_i]$. The third constraint makes sure that the slack variable remains positive. Therefore, a tube of width $\epsilon(1+\frac{1}{\tau})$ exists where we do not penalise the points at all. For illustrating the robustness of SPTSVC to noise, we have shown the clusters formed for a two-dimensional sample dataset consisting of two classes '+' and '$\times$' in Fig. 1.

We now utilize the CCCP [30] technique to break the $i^{th}$ problem into smaller convex quadratic programming sub-problems.

$$\begin{aligned} \min_{\omega_i^{j+1}, b_i^{j+1}, \xi_i^{j+1}} \; & \frac{1}{2}\left\|X_i\omega_i^{j+1} + b_i^{j+1}e\right\|^2 + ce^T\xi_i^{j+1} \\ s.t. \; & T(|\hat{X}_i\omega_i^{j+1} + b_i^{j+1}e|) \ge e - \xi_i^{j+1} - e\epsilon, \\ & T(|\hat{X}_i\omega_i^{j+1} + b_i^{j+1}e|) \le e + \frac{\xi_i^{j+1}}{\tau} + e\frac{\epsilon}{\tau} \\ & \xi_i^{j+1} \ge 0. \end{aligned} \tag{7}$$

Using first order Taylor series expansion we get:

$$T(|\hat{X}_i\omega_i^{j+1} + b_i^{j+1}e|) = Z_i(\hat{X}_i\omega_i^{j+1} + b_i^{j+1}e), \tag{8}$$

where:

$$Z_i = diag(sign(\hat{X}_i\omega_i^j + b_i^j e)). \tag{9}$$

The above optimization problem is the primal formulation of SPTSVC. It can be observed that (7) is similar to TWSVC primal, with the difference in the constraints. This is due to the change in the loss function of the problem. We then convert the primal to its corresponding dual form. Now, consider the primal problem in the form of Lagrangian:

$$\begin{aligned} L = \; & \frac{1}{2}\left\|X_i\omega_i^{j+1} + b_i^{j+1}e\right\|^2 + ce^T\xi_i^{j+1} + \\ & \alpha^T\left(e - \xi_i^{j+1} - e\epsilon - Z_i(\hat{X}_i\omega_i^{j+1} + b_i^{j+1}e)\right) + \\ & \beta^T\left(Z_i(\hat{X}_i\omega_i^{j+1} + b_i^{j+1}e) - e - \frac{\xi_i^{j+1}}{\tau} - e\frac{\epsilon}{\tau}\right) - \gamma^T\left(\xi_i^{j+1}\right), \end{aligned} \tag{10}$$

here $\alpha, \beta, \gamma \ge 0$ are the Lagrangian multiplier vectors. Now applying the KKT conditions [31], we obtain:

$$u = -(H^TH)^{-1}G^T(\beta - \alpha), \tag{11}$$

where $G = Z_i[\hat{X}_i \;\; e]$, $H = [X_i \;\; e]$, and $u = \begin{bmatrix} \omega_i^{j+1} \\ b_i^{j+1} \end{bmatrix}$.

We add a small regularization term $\delta I, \delta > 0$ to $H^TH$, to prevent the ill-conditioning of $H^TH$. Here $I$ is the identity matrix of appropriate dimensions. We obtain the modified equation as:

$$u = -(H^TH + \delta I)^{-1}G^T(\beta - \alpha), \tag{12}$$

We continue to use (11) from here on wards with the understanding that if need arises (12) can be used in place of (11). Now, applying the KKT conditions, our Lagrangian is modified to give the following dual problem:

$$\min_{\alpha-\beta} \; \frac{1}{2}(\beta-\alpha)^T G(H^T H)^{-1} G^T(\beta-\alpha) - (\beta-\alpha)^T e + e\epsilon\left(\alpha^T + \frac{\beta^T}{\tau}\right)$$
$$s.t. \quad \alpha,\beta,\gamma \geq 0, ce = \alpha + \frac{\beta}{\tau} + \gamma \tag{13}$$

Making the substitution $\lambda = \alpha - \beta$. in (13) to get a simplified expression:

$$\min_{\lambda} \; \frac{1}{2}\lambda^T G(H^T H)^{-1} G^T \lambda - \lambda^T e\left(\frac{\epsilon}{\tau}+1\right) + \alpha^T e\left(\epsilon + \frac{\epsilon}{\tau}\right)$$
$$s.t. \quad \alpha \geq 0, \lambda \geq 0, \alpha\left(1+\frac{1}{\tau}\right) - \frac{\lambda}{\tau} \leq ce. \tag{14}$$

After the solution of (14) is obtained, we get $[\omega_i^j; b_i^j] = -(H^T H)^{-1} G^T(\beta-\alpha)$. For the $i^{th}$ cluster center plane, we stop after $\left\|[\omega_i^{j+1}; b_i^{j+1}] - [\omega_i^j; b_i^j]\right\| < \nu$, where $\nu$ is the error tolerance. One can also set maximum number of iterations incase the error stays above $\nu$. For a point $x_k$, we can predict the label $\hat{y}(x_k) = \arg\min_{i=1,2,...,N} \mid \omega_i^T x_k + b_i \mid$. Here $N$ denotes the number of clusters.

### B. Non-linear SPTSVC

SPTSVC can be easily extended to non-linear case by using the kernel trick. Lets consider $k$ kernel generated cluster center planes $K(x, X)z_i + b_i = 0$, $i = 1, ..., k$, where $K(\cdot,\cdot)$ is an arbitrary kernel function [32], selected on the basis of the problem. Similar to the linear case, the objective function for non-linear case can be formulated as:

$$\begin{aligned} \min_{\omega_i, b_i, \xi_i, X_i} \; & \frac{1}{2}\|K(X_i, X)z_i + b_i e\|^2 + ce^T \xi_i \\ s.t. \quad & |K(\hat{X}_i, X)z_i + b_i e| \geq e - \xi_i - e\epsilon, \\ & |K(\hat{X}_i, X)z_i + b_i e| \leq e + \frac{\xi_i}{\tau} + e\frac{\epsilon}{\tau} \\ & \xi_i \geq 0, \end{aligned} \tag{15}$$

Again, using CCCP, (15) can be reduced to smaller convex quadratic optimization problems. We obtain the following relations, which are almost identical to the linear case:

$$u = -(P^T P + \delta I)^{-1} Q^T(\beta - \alpha), \tag{16}$$

where $\delta > 0$, $Z_i = diag(sign(K(\hat{X}_i, X)z_i^j + b_i^j e))$, $P = [K(X_i, X) \;\; e]$, $Q = Z_i[K(\hat{X}_i, X) \;\; e]$, and $u = \begin{bmatrix} z_i^{j+1} \\ b_i^{j+1} \end{bmatrix}$.

In a similar manner to the linear case, the dual formulation for (15) is obtained as:

$$\min_{\alpha-\beta} \; \frac{1}{2}(\beta-\alpha)^T Q(P^T P)^{-1} Q^T(\beta-\alpha) - (\beta-\alpha)^T e + e\epsilon(\alpha^T + \frac{\beta^T}{\tau}) \tag{17}$$
$$s.t. \quad \alpha,\beta,\gamma \geq 0, ce = \alpha + \frac{\beta}{\tau} + \gamma.$$

Making the substitution $\lambda = \alpha - \beta$. in (17), to get a simplified expression:

$$\min_{\lambda} \; \frac{1}{2}\lambda^T Q(P^T P)^{-1} Q^T \lambda - \lambda^T e\left(\frac{\epsilon}{\tau}+1\right) + \alpha^T e\left(\epsilon + \frac{\epsilon}{\tau}\right)$$
$$s.t. \quad \alpha \geq 0, \lambda \geq 0, \alpha\left(1+\frac{1}{\tau}\right) - \frac{\lambda}{\tau} \leq ce. \tag{18}$$

## IV. Discussion

In this section, we first provide rigorous analysis of noise-insensitive and sparsity properties of the proposed SPTSVC. Then, we discuss the time complexity of SPTSVC.

### A. Noise insensitivity and Sparsity

The main advantage of the proposed SPTSVC model is incorporating the property of noise-insensitivity, while at the same time maintaining the sparsity of the model. The $\epsilon$-insensitive pinball loss function $\mathcal{L}_\tau^\epsilon$ is non-differentiable at points $u = \epsilon$ and $u = -\frac{\epsilon}{\tau}$. Hence, for solving the QPP at each CCCP [30] iteration $j$, the sub-gradient of $\mathcal{L}_\tau^\epsilon$ is needed [33]. Sub-gradient of $\mathcal{L}_\tau^\epsilon(u)$, denoted by $g_\tau^\epsilon(u)$ is given as:

$$g_\tau^\epsilon(u) = \begin{cases} 1 & u > \epsilon, \\ [0,1] & u = \epsilon, \\ 0 & -\frac{\epsilon}{\tau} < u < \epsilon \\ [0,1] & u = -\frac{\epsilon}{\tau}, \\ -\tau & u < -\frac{\epsilon}{\tau}, \end{cases}$$

Further, we also partition the data-point indices $j$ belonging to $cluster_i$ having parameters $[\omega_i; b_i]$, into 5 sets as follows:

$$\begin{aligned} S_1 &= \{j : 1 - |\omega_i^T x_j + b_i| > \epsilon\}, \\ S_2 &= \{j : 1 - |\omega_i^T x_j + b_i| = \epsilon\}, \\ S_3 &= \{j : -\frac{\epsilon}{\tau} < 1 - |\omega_i^T x_j + b_i| < \epsilon\}, \\ S_4 &= \{j : 1 - |\omega_i^T x_j + b_i| = -\frac{\epsilon}{\tau}\}, \\ S_5 &= \{j : 1 - |\omega_i^T x_j + b_i| < -\frac{\epsilon}{\tau}\}. \end{aligned}$$

Set $S_5$ denotes the points which are greater than $1 + \frac{\epsilon}{\tau}$ distance away from the hyper-plane $[\omega_i; b_i]$. As $\tau$ increases, the number of points in set $S_5$ increases. These points then contribute to the final solution, as the sub-gradient $g_\tau^\epsilon(u)$ for the same interval is non-zero. For larger values of $\tau$, there are many points in all sets, which makes the model less sensitive to noise around the decision boundary.

Further, it can be observed that $g_\tau^\epsilon(u)$ is 0 in the interval $(-\frac{\epsilon}{\tau}, \epsilon)$. As a result, the points belonging to the set $S_2$ which correspond to the same interval $(-\frac{\epsilon}{\tau}, \epsilon)$, do not contribute to the final solution. This $\epsilon$-insensitive zone leads to sparsity in the solution. This sparsity is due to the novel approach proposed in this work, which improves the drawbacks in the previous twin SVM based clustering methods.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2021.3059910, IEEE Journal of Biomedical and Health Informatics



### B. Time Complexity

The solution of TWSVC requires the solutions of QPPs. For $p$ clusters, $p$ number of QPPs are solved for the clustering of data. Moreover, the time complexity of solving a QPP with $k$ number of constraints is $O(k^3)$ [24]. Therefore, the total time complexity of the TWSVC algorithm is $O(pk^3)$.

On the other hand, proposed SPTSVC involves the solution of QPPs with twice the number of constraints as that in TWSVC. This is due to the $\epsilon$-insensitive loss function in SPTSVC. Therefore, the time complexity of SPTSVC is $O(p(8k^3))$. Clearly, the computational time of proposed SPTSVC is eight times that of TWSVC. However, this can be seen as a tradeoff for the noise insensitivity of proposed SPTSVC, in comparison to existing algorithms. The computation time of calculating the inverses in SPTSVC is same as in TWSVC.

## V. Numerical Experiments

This section presents the results of experiments on various clustering datasets. The proposed SPTSVC is compared with FCM [34], TWSVC [24], and TBSVC [26]. In order to test the noise-insensitivity of the proposed SPTSVC, we added zero-mean Gaussian noise to datasets with different standard deviation $\sigma$. For all the algorithms, Gaussian kernel function $K(x,y) = \exp(-\|x-y\|^2/(2\mu^2))$ is used. All the experiments are performed using MATLAB R2017a, installed on a Windows 10 PC with 128 GB RAM, Intel(R) Xeon(R) CPU E5-2697 v4 @ 2.30GHz CPU. The codes of the proposed algorithm will be available on the author's Github page: https://github.com/mtanveer1.

### A. Parameter selection

For the parameter selection, we used 5-fold cross-validation using the grid search method. For TWSVC, TBSVC and SPTSVC, the value of penalty parameter $c$ is selected from the set $\{2^i | i = -5, -3, \ldots, 3, 5\}$. For $\epsilon$-insensitive pinball loss, the value of the parameter $\tau$ is selected from the set $\{0.25, 0.50, 1\}$ and $\epsilon$ from the set $\{0.1, 0.3, 0.5\}$. The parameter $\delta$ is taken as $10^{-4}$. The value of the Gaussian kernel parameter $\mu$ is selected from the range $\{2^i | i = -5, -4, -3, \ldots, 3, 4, 5\}$ for all the methods. In case of FCM, the parameter $m$ is selected from the set $\{1, 1.25, 1.5, 1.75, 2, 2.25, 2.5\}$ [35].

### B. Benchmark datasets

Experiments are performed on various benchmark datasets. The datasets include synthetic as well as real world datasets. The synthetic datasets are taken from the website: https://github.com/deric/clustering-benchmark/tree/master/src/main/resources/datasets/artificial. The real-world datasets are taken from the UCI Machine Learning Repository [36]. The datasets are further included with different feature noise by adding zero mean Gaussian noise with standard deviation $\sigma$ set as 0 (noise-free), 0.05 and 0.1.

TABLE I: Comparison of clustering accuracy obtained on UCI and synthetic datasets with different noise levels.

| **Datasets** (Samples× Features ×Clusters) | **Noise** $\sigma$ | **FCM** [34] Accuracy(%) Time (s) | **TWSVC** [24] Accuracy(%) Time (s) | **TBSVC** [26] Accuracy(%) Time (s) | **SPTSVC** Accuracy(%) Time (s) |
|---|---|---|---|---|---|
| ds4c2sc8 | 0 | 26.4562 | **95.1804** | 94.7466 | 94.6478 |
| ($485 \times 2 \times 8$) | | 0.0716 | 0.7969 | 0.5560 | 9.5016 |
| | 0.05 | 69.0206 | 76.7526 | 76.8127 | **77.2466** |
| | | 0.1342 | 0.6770 | 0.5992 | 9.0470 |
| | 0.1 | 71.2027 | 76.2801 | 77.0103 | **77.1735** |
| | | 0.1410 | 0.6357 | 0.5588 | 8.8263 |
| hayes-roth | 0 | 60.3943 | 77.0462 | 79.1499 | **79.8746** |
| ($132 \times 4 \times 3$) | | 0.0119 | 0.0390 | 0.0309 | 0.1827 |
| | 0.05 | 60.9778 | **68.939** | 66.0057 | 68.4034 |
| | | 0.0122 | 0.0414 | 0.0325 | 0.1710 |
| | 0.1 | 59.8336 | 63.7607 | **63.7835** | 62.6279 |
| | | 0.0025 | 0.0374 | 0.0351 | 0.1695 |
| jain | 0 | 52.3068 | **100** | **100** | **100** |
| ($373 \times 2 \times 2$) | | 0.0069 | 0.1281 | 0.1177 | 0.6755 |
| | 0.05 | 51.8732 | 96.8152 | **98.9189** | 97.8739 |
| | | 0.0036 | 0.1479 | 0.1198 | 0.6580 |
| | 0.1 | 51.5449 | 97.9027 | 96.2811 | **97.3265** |
| | | 0.0065 | 0.1256 | 0.1242 | 0.6551 |
| lense | 0 | 50 | 68 | **82** | 78 |
| ($24 \times 4 \times 3$) | | 0.0037 | 0.0210 | 0.0166 | 0.0127 |
| | 0.05 | 46.6667 | 65.3333 | 67.3333 | **74** |
| | | 0.0091 | 0.0166 | 0.0142 | 0.0375 |
| | 0.1 | 60 | 68 | 66 | **72** |
| | | 0.0012 | 0.0119 | 0.0130 | 0.0135 |
| libras | 0 | 67.9734 | 91.4554 | 80.1565 | **91.4867** |
| ($360 \times 90 \times 15$) | | 0.6308 | 0.5774 | 0.5810 | 9.4616 |
| | 0.05 | 56.4319 | **88.3177** | 88.0047 | 87.9186 |
| | | 0.9540 | 0.5194 | 0.5049 | 10.4479 |
| | 0.1 | 41.1189 | **87.9186** | 87.3944 | 87.5352 |
| | | 0.0317 | 0.5592 | 0.5212 | 10.7237 |
| lsun | 0 | 70.3987 | **100** | **100** | 99.2595 |
| ($400 \times 2 \times 3$) | | 0.0062 | 0.1806 | 0.1553 | 1.3830 |
| | 0.05 | 70.1519 | 97.4241 | 97.5696 | **98.2911** |
| | | 0.0225 | 0.1920 | 0.1526 | 1.4916 |
| | 0.1 | 68.7215 | **91.7152** | 91.6962 | 89.5759 |
| | | 0.0218 | 0.1816 | 0.1466 | 1.4028 |
| new-thyroid | 0 | 53.4219 | 86.4673 | 79.6013 | **89.0587** |
| ($215 \times 5 \times 3$) | | 0.0038 | 0.0608 | 0.0593 | 0.5206 |
| | 0.05 | 53.4219 | **89.3245** | 80.7752 | 87.309 |
| | | 0.0031 | 0.0566 | 0.0554 | 0.5024 |
| | 0.1 | 53.4219 | **87.4419** | 81.9491 | 85.4707 |
| | | 0.0037 | 0.0674 | 0.0549 | 0.4823 |
| seeds | 0 | 33.3566 | 85.5517 | 81.417 | **85.8537** |
| ($210 \times 7 \times 3$) | | 0.0336 | 0.0643 | 0.0493 | 0.3950 |
| | 0.05 | 33.3566 | 81.8351 | **85.0871** | 82.9501 |
| | | 0.0248 | 0.0511 | 0.0516 | 0.3951 |
| | 0.1 | 33.3566 | 73.6353 | **81.9512** | 76.3995 |
| | | 0.0172 | 0.0539 | 0.0479 | 0.3911 |
| spherical_4_3 | 0 | 95.0633 | 99.7658 | **100** | 99.4747 |
| ($400 \times 3 \times 4$) | | 0.0098 | 0.2157 | 0.1911 | 2.1070 |
| | 0.05 | 94.3165 | **100** | **100** | 98.6962 |
| | | 0.0033 | 0.2136 | 0.2038 | 2.2116 |
| | 0.1 | 94.5759 | 98.8671 | 98.9241 | **100** |
| | | 0.0120 | 0.2585 | 0.1945 | 2.1557 |
| spherical_5_2 | 0 | 78.498 | **96.4735** | 93.6816 | 94.9224 |
| ($250 \times 2 \times 5$) | | 0.0223 | 0.1289 | 0.1049 | 1.2002 |
| | 0.05 | 75.8041 | 90.6449 | 89.5673 | **90.9224** |
| | | 0.0088 | 0.1006 | 0.1023 | 1.1128 |
| | 0.1 | 74.7918 | 85.9265 | 85.7796 | **88.2286** |
| | | 0.0152 | 0.1221 | 0.0973 | 1.1067 |
| tae | 0 | 32.092 | **62.7586** | 61.6092 | 61.5632 |
| ($150 \times 5 \times 3$) | | 0.0298 | 0.0301 | 0.0335 | 0.1960 |
| | 0.05 | 32.092 | **57.1034** | 56 | 57.0115 |
| | | 0.0178 | 0.0337 | 0.0329 | 0.1963 |
| | 0.1 | 32.092 | 55.8621 | 56.5517 | **56.6437** |
| | | 0.0284 | 0.0300 | 0.0310 | 0.1944 |
| teachingeval | 0 | 33.8969 | 60.4212 | 61.9726 | **62.9692** |
| ($151 \times 5 \times 3$) | | 0.0028 | 0.0323 | 0.0321 | 0.2278 |
| | 0.05 | 33.8969 | 58.1001 | **58.3908** | 57.2621 |
| | | 0.0189 | 0.0335 | 0.0348 | 0.2086 |
| | 0.1 | 33.8969 | **57.3615** | 56.0623 | 55.4127 |
| | | 0.0012 | 0.0311 | 0.0334 | 0.1900 |
| tetra | 0 | 73.0506 | **100** | 94.8038 | **100** |
| ($400 \times 3 \times 4$) | | 0.0220 | 0.1932 | 0.1823 | 2.1000 |
| | 0.05 | 72.2152 | 98.9367 | **99.2025** | **99.2025** |
| | | 0.0194 | 0.1922 | 0.1761 | 2.3028 |
| | 0.1 | 71.4241 | 98.3038 | **98.5506** | **98.5506** |
| | | 0.0203 | 0.1855 | 0.1713 | 2.1862 |
| zelnik1 | 0 | 58 | 99.6949 | **100** | 98.9815 |
| ($299 \times 2 \times 3$) | | 0.0420 | 0.1006 | 0.0937 | 0.8230 |
| | 0.05 | 56.247 | 56.0374 | 54.9764 | **57.6295** |
| | | 0.0175 | 0.1077 | 0.0915 | 0.5694 |
| | 0.1 | 55.4358 | 52.7878 | **55.7565** | 54.9784 |
| | | 0.0266 | 0.0942 | 0.0920 | 0.7501 |
| Average Accuracy | | 56.9714 | 82.0034 | 81.5589 | **82.6841** |
| Average Rank | | 3.9048 | 2.0833 | 2.1191 | **1.8929** |
| Average Training Time | | 0.0590 | 0.1749 | 0.1547 | 2.0806 |

TABLE II: Number of non-zero dual variables for different algorithms.

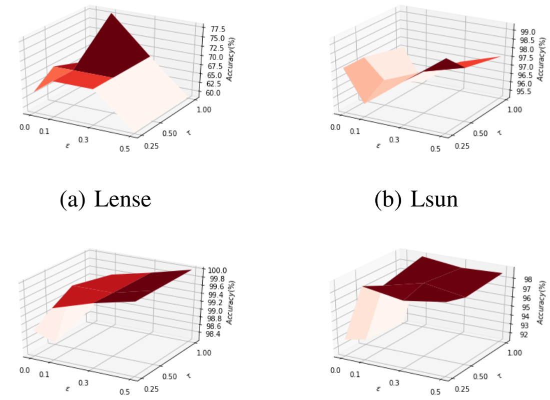
| Dataset | $\epsilon$ | TWSVC | TBSVC | SPTSVC | | |
|---|---|---|---|---|---|---|
| | | | | $\tau = 0.25$ | $\tau = 0.5$ | $\tau = 1$ |
| ds4c2sc8 | 0 | 17.4 | 54.375 | 339.45 | 339.425 | 339.5 |
| | 0.1 | | | 138.625 | 149.45 | 144.875 |
| | 0.3 | | | 73.9 | 90.1 | 88.525 |
| | 0.5 | | | 48.825 | 60.575 | 70.05 |
| hayes-roth | 0 | 28.8 | 40.467 | 70.4 | 70.4 | 70.4 |
| | 0.1 | | | 56.933 | 56.2 | 58.133 |
| | 0.3 | | | 30.4 | 42.2 | 45.6 |
| | 0.5 | | | 29.333 | 28.733 | 33.733 |
| jain | 0 | 14.9 | 25 | 149.2 | 149.2 | 149.2 |
| | 0.1 | | | 70.2 | 76 | 79.2 |
| | 0.3 | | | 39.4 | 47.7 | 52.3 |
| | 0.5 | | | 24.6 | 32.8 | 39.4 |
| lense | 0 | 5.8 | 5.733 | 12.8 | 12.8 | 12.8 |
| | 0.1 | | | 9.467 | 9.867 | 9.933 |
| | 0.3 | | | 8.533 | 8.533 | 8.6 |
| | 0.5 | | | 8.067 | 7.867 | 7.933 |
| libras | 0 | 139.32 | 158.347 | 256.48 | 256.547 | 256.84 |
| | 0.1 | | | 177.067 | 193.573 | 207.027 |
| | 0.3 | | | 142.707 | 150.947 | 160.187 |
| | 0.5 | | | 128.227 | 131.267 | 136.053 |
| lsun | 0 | 8.933 | 18.733 | 213.333 | 213.333 | 213.333 |
| | 0.1 | | | 87.867 | 100.133 | 109.733 |
| | 0.3 | | | 50.667 | 62.267 | 74.4 |
| | 0.5 | | | 29.533 | 39.267 | 49.467 |
| new-thyroid | 0 | 22.333 | 25.533 | 114.667 | 114.667 | 114.6 |
| | 0.1 | | | 58.133 | 53.4 | 61.867 |
| | 0.3 | | | 27.933 | 30.667 | 36.467 |
| | 0.5 | | | 30.533 | 24.933 | 27.067 |
| seeds | 0 | 17.067 | 22.933 | 112 | 112 | 112 |
| | 0.1 | | | 57 | 68.667 | 79.667 |
| | 0.3 | | | 31.667 | 36.467 | 43.667 |
| | 0.5 | | | 26.4 | 28.667 | 30.867 |
| spherical_4_3 | 0 | 22.45 | 20.7 | 240 | 239.9 | 239.85 |
| | 0.1 | | | 48.75 | 55.55 | 59.15 |
| | 0.3 | | | 24.7 | 29.75 | 33.15 |
| | 0.5 | | | 15.5 | 19.8 | 24.45 |
| spherical_5_2 | 0 | 11.04 | 49.88 | 159.96 | 160 | 159.96 |
| | 0.1 | | | 68.84 | 73.4 | 78.2 |
| | 0.3 | | | 36.08 | 47.28 | 54.48 |
| | 0.5 | | | 25.4 | 30.8 | 37.08 |
| tae | 0 | 27.4 | 70.533 | 74.333 | 74.4 | 74.4 |
| | 0.1 | | | 62.7333 | 64.4 | 66.2 |
| | 0.3 | | | 55.4 | 59 | 60.333 |
| | 0.5 | | | 51.6 | 51.933 | 52.067 |
| teachingeval | 0 | 56.467 | 68.6 | 79.733 | 79.533 | 79.4 |
| | 0.1 | | | 64.333 | 70.267 | 72.267 |
| | 0.3 | | | 48.133 | 54.4 | 59.333 |
| | 0.5 | | | 38.067 | 43.133 | 48.2 |
| tetra | 0 | 72.25 | 177.9 | 240 | 240 | 240 |
| | 0.1 | | | 129.7 | 164.65 | 187.05 |
| | 0.3 | | | 58.5 | 63.1 | 93.9 |
| | 0.5 | | | 47.55 | 47.55 | 47.55 |
| zelnik1 | 0 | 20.4 | 72.4 | 158.733 | 158.733 | 158.667 |
| | 0.1 | | | 52.4 | 59.7333 | 67.533 |
| | 0.3 | | | 31.333 | 34.733 | 40.067 |
| | 0.5 | | | 26.667 | 28.467 | 30.733 |

### C. Discussion on experimental results

The experimental results of the different algorithms are shown in Table I. It is observable from Table I that the proposed SPTSVC is performing better than existing algorithms in terms of clustering of the different datasets. The proposed SPTSVC achieves an average accuracy $82.68\%$ and average rank $1.89$. However, in terms of training time, the proposed SPTSVC is taking more time than the other algorithms. This is due to the $\epsilon$-insensitive constraint in the formulation of the proposed SPTSVC, as discussed in time complexity analysis.

### D. Statistical analysis

To verify the significant difference between the methods, we apply the Friedman test [37] and Nemenyi post-hoc test [37]. The Friedman test is calculated on the basis of $\chi^2$, using the rank of each algorithm given in Table I. Here, first we assume that all methods are having similar performance as null hypothesis. The $\chi^2$ value is calculated to be 67.0639.



(a) Lense (b) Lsun

(c) Spherical_4_3 (d) Zelnik1

Fig. 2: $3D$ surface showing accuracy sensitivity of the proposed SPTSVC with respect to model parameters $\epsilon$ and $\tau$.

$$F_F = \frac{(42-1)\times 67.0639}{42\times 3 - 67.0639} \approx 46.6539. \tag{19}$$

Now, degrees of freedom for F-distribution is calculated as $(l-1, (l-1)(N-1))$. For $l = 4$ and $N = 42$, the F-distribution has (3,123) degrees of freedom. For (3,123) degrees of freedom and $\alpha = 0.05$ level of significance, the critical value is 2.6783. As $F_F = 46.6539 > 2.6783$, null hypothesis does not hold true.
Further, to determine pair-wise statistical difference between algorithms, we utilize the Nemenyi post-hoc test [37].

$$CD = 2.569 \times \sqrt{\frac{4\times 5}{6\times 42}} \approx 0.7237. \tag{20}$$

Here, $CD = 0.7237$ denotes critical difference at $alpha = 0.05$. Therefore, there is significant difference between the methods if difference between average ranks of the algorithms is atleast $CD$ by the Nemenyi test [37]. So, there is significant difference between the proposed SPTSVC and FCM algorithm, but no statistical difference with TWSVC [24], and TBSVC [26].

### E. Parameter Sensitivity

The $3D$ surface plots in Fig. 2 show the variation in accuracy with changes in parameters $\epsilon$ and $\tau$. It can be inferred from these plots that the accuracy obtained is quite sensitive to parameters chosen. Therefore, the choice of parameters depends on the distribution of points in a particular dataset. In absence of a proper-mechanism to select optimal parameters, we have grid-searched over wide range of values for all the parameters.

### F. Sparsity analysis

The sparsity is illustrated in terms of total number of non-zero dual variables in Table II. The sparseness of the proposed algorithm is shown for different values of $\epsilon$ and $\tau$. The number of non-zero dual variables are averaged for the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2021.3059910, IEEE Journal of Biomedical and Health Informatics

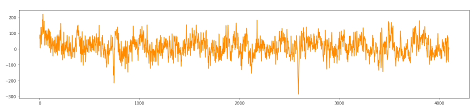


different classes. The proposed SPTSVC is able to generate the clustering hyperplanes with sparsity similar to TWSVC and TBSVC. Further, It can be observed from Table II that with increase in the value of $\epsilon$, the sparsity increases and with increase in value of $\tau$, sparsity decreases. This is due to widening and narrowing of the $\epsilon$-insensitive zone respectively. One more observation is more number of non-zero dual variables of TBSVC in comparison to TWSVC in Table II. This is due to the use of regularization in TBSVC algorithm.

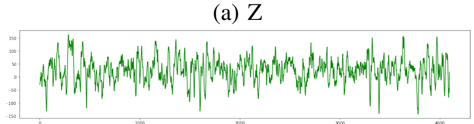
## VI. Biomedical Applications

In this section, we illustrate the applicability of the proposed SPTSVC clustering method on biomedical time-series and image data.

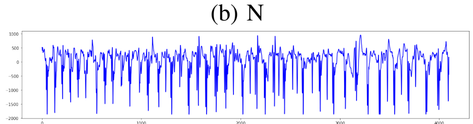
### A. EEG signal clustering



(a) Z



(b) N



(c) S

Fig. 3: Samples from EEG dataset [38].

TABLE III: EEG clustering results.

| Dataset | FCM | TWSVC | TBSVC | SPTSVC |
|---|---|---|---|---|
| EEG | 47.7253 | 68.4283 | 69.5232 | **71.1192** |

We show the clustering performance of the different algorithms on the single channel voltage recordings of EEG signals. The EEG dataset [38] utilized for this application contains five classes - Z, O, N, F and S. The classes includes one class of seizure data, and 2 classes each of healthy and seizure free signals. Some of the sample signals of the dataset are shown in Fig. 3. We applied independent component analysis (ICA) [6] to extract the prominent features for clustering of the time-series data. The results of the clustering performance of the different algorithms are shown in Table III.

### B. Breast cancer clustering

TABLE IV: Breast cancer clustering results.

| Dataset | FCM [34] | TWSVC [24] | TBSVC [26] | SPTSVC |
|---|---|---|---|---|
| BreaKHis | 26.4367 | 77.9114 | 78.2278 | **78.6772** |

For application on breast cancer data, we have taken the images from the BreaKHis dataset [39], [7]. The dataset

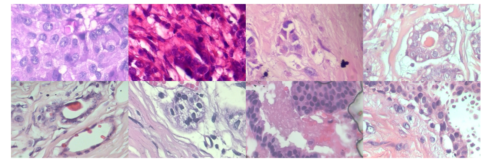


Fig. 4: Sample images from BreaKHis dataset [39].

consists of 4 benign and 4 malignant types of samples. The different types of benign breast tumors are as: phyllodes tumor, fibroadenoma, adenosis, and tubular adenona. The malignant tumors are lobular, papillary, mucinous, and ductal carcinoma. The different images from each class in the dataset are shown in Fig. 4.

We selected 50 images from each of the different classes. The features from the histopathological images are extracted using principal component analysis (PCA) method. This resulted into a dataset size of $400 \times 100$. The comparative performance of the different algorithms on clustering of the breast cancer dataset is shown in Table IV.

### C. Discussion of results on biomedical data

For EEG dataset, from the Table III, one can clearly observe the superior performance of the proposed SPTSVC model. This suggests that the proposed SPTSVC can be potentially used for the detection of EEG signals. Further, the results obtained also hint at the potential use of the proposed SPTSVC model for automated diagnosis of diseases involving EEG signals such as epilepsy. Moreover, the effectiveness of proposed model for EEG implies its usage for several applications, such as brain computer interface (BCI), detecting brain dysfunction etc.

The results obtained for breast-cancer clustering dataset has been shown in Table IV. We can infer that the proposed SPTSVC performed slightly better than the existing plane-based clustering algorithms for clustering of breast tumour images. This shows the possible applicability of proposed SPTSVC on clustering of various image datasets for biomedical applications involving MRI images, X-ray, and other multimodal imaging data.

## VII. Conclusion and Future Directions

In this paper, we presented a novel clustering algorithm termed as sparse pinball loss twin support vector clustering (SPTSVC). The proposed SPTSVC includes an $\epsilon$-insensitive pinball loss function, in comparison to the traditional hinge loss. The novelty and main advantage of the proposed model is accommodating noise-insensitivity property while at the same time providing good sparsity, which leads to lesser running time. Pinball loss function also provides re-sampling stability, which is useful for large scale datasets. Thus, the proposed algorithm can be useful for various noise corrupted datasets. Further, the proposed SPTSVC gave better performance on biomedical time-series and image data.

In future, the proposed algorithm can be used in various applications involving noisy data. The proposed algorithm can be useful for applications involving less time for prediction of labels, as testing time of proposed algorithm is lesser due to sparsity. In future, models can be developed to decrease the training time of the proposed model. In order to avoid grid-searching over wide range of parameters, there is a need to develop an efficient parameter selection methodology. For further experiments, the concept of $\epsilon$-insensitive zone can be used with other loss functions in order to increase the sparsity while still maintaining the desired properties of the loss function in hand.

## Acknowledgments

This work was supported by Science & Engineering Research Board (SERB) Government of INDIA under Ramanujan fellowship grant no. SB/S2/RJN-001/2016 and Early Career Research Award (ECRA) grant no. ECR/2017/000053. It is also supported by Council of Scientific & Industrial Research (CSIR), New Delhi, INDIA under Extra Mural Research (EMR) Scheme grant no. 22(0751)/17/EMR-II. We gratefully acknowledge the Indian Institute of Technology Indore for providing facilities and support. A sincere thanks to to the Indian Institute of Technology Indore for providing Institute fellowship to Mr. Bharat Richhariya.

## References

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep 1995. [Online]. Available: https://doi.org/10.1007/BF00994018

[2] E. Pellegrini et al., "Machine learning of neuroimaging for assisted diagnosis of cognitive impairment and dementia: A systematic review," *Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring*, vol. 10, pp. 519–535, 2018.

[3] B. Richhariya, M. Tanveer, A. H. Rashid, and Alzheimer's Disease Neuroimaging Initiative, "Diagnosis of Alzheimer's disease using universum support vector machine based recursive feature elimination (USVM-RFE)," *Biomedical Signal Processing and Control*, vol. 59, p. 101903, 2020.

[4] R. Khemchandani and P. Saigal, "Color image classification and retrieval through ternary decision structure based multi-category TWSVM," *Neurocomputing*, vol. 165, pp. 444–455, 2015.

[5] W. Long, Y.-r. Tang, and Y.-j. Tian, "Investor sentiment identification based on the universum svm," *Neural Computing and Applications*, vol. 30, no. 2, pp. 661–670, 2018.

[6] B. Richhariya and M. Tanveer, "EEG signal classification using universum support vector machine," *Expert Systems with Applications*, vol. 106, pp. 169–182, 2018.

[7] C. Gautam, P. K. Mishra, A. Tiwari, B. Richhariya, H. M. Pandey, S. Wang, M. Tanveer, and A. D. N. Initiative, "Minimum variance-embedded deep kernel regularized least squares method for one-class classification and its applications to biomedical data," *Neural Networks*, vol. 123, pp. 191–216, 2020.

[8] Jayadeva, R. Khemchandani, and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 905–910, 2007.

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

[10] K. Li, G. Zhou, Y. Yang, F. Li, and Z. Jiao, "A novel prediction method for favorable reservoir of oil field based on grey wolf optimizer and twin support vector machine," *Journal of Petroleum Science and Engineering*, p. 106952, 2020.

[11] M. Tanveer, "Application of smoothing techniques for linear programming twin support vector machines," *Knowledge and Information Systems*, vol. 45, no. 1, pp. 191–214, Oct 2015.

[12] ——, "Newton method for implicit lagrangian twin support vector machines," *International Journal of Machine Learning and Cybernetics*, vol. 6, no. 6, pp. 1029–1040, 2015.

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

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

[15] M. Tanveer, C. Gautam, and P. Suganthan, "Comprehensive evaluation of twin svm based classifiers on uci datasets," *Applied Soft Computing*, vol. 83, p. 105617, 2019.

[16] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 464–471, 2002.

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

[18] X. Huang, L. Shi, and J. A. Suykens, "Support vector machine classifier with pinball loss," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 984–997, 2013.

[19] Y. Xu, Z. Yang, and X. Pan, "A novel twin support-vector machine with pinball loss," *IEEE Transactions on Neural networks and learning systems*, vol. 28, no. 2, pp. 359–370, 2016.

[20] M. Tanveer, A. Tiwari, R. Choudhary, and S. Jalan, "Sparse pinball twin support vector machines," *Applied Soft Computing*, vol. 78, pp. 164–175, 2019.

[21] M. Tanveer, T. Rajani, and M. Ganaie, "Improved sparse pinball twin svm," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019, pp. 3287–3291.

[22] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[23] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," *Journal of machine learning research*, vol. 2, no. Dec, pp. 125–137, 2001.

[24] Z. Wang, Y.-H. Shao, L. Bai, and N.-Y. Deng, "Twin support vector machine for clustering," *IEEE Transactions on Neural networks and learning systems*, vol. 26, no. 10, pp. 2583–2588, 2015.

[25] R. Khemchandani, A. Pal, and S. Chandra, "Fuzzy least squares twin support vector clustering," *Neural Computing and Applications*, vol. 29, no. 2, pp. 553–563, 2018.

[26] L. Bai, Y.-H. Shao, Z. Wang, and C.-N. Li, "Clustering by twin support vector machine and least square twin support vector classifier with uniform output coding," *Knowledge-Based Systems*, vol. 163, pp. 227–240, 2019.

[27] Z. Wang, X. Chen, Y.-H. Shao, and C.-N. Li, "Ramp-based twin support vector clustering," *Neural Computing and Applications*, pp. 1–12, 2019.

[28] B. Richhariya and M. Tanveer, "Least squares projection twin support vector clustering (LSPTSVC)," *Information Sciences*, vol. 533, pp. 1–23, 2020.

[29] M. Tanveer, T. Gupta, and M. Shah, "Pinball loss twin support vector clustering," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. to appear, 2021.

[30] A. L. Yuille and A. Rangarajan, "The concave-convex procedure (cccp)," in *Advances in Neural information processing systems*, 2002, pp. 1033–1040.

[31] R. Fletcher, *Practical methods of optimization*. John Wiley & Sons, 2013.

[32] B. Schölkopf, A. J. Smola, *et al.*, "Learning with kernels: Support vector machines, regularization," *Optimization, and Beyond. MIT press*, vol. 1, no. 2, 2002.

[33] M. Held, P. Wolfe, and H. P. Crowder, "Validation of subgradient optimization," *Mathematical programming*, vol. 6, no. 1, pp. 62–88, 1974.

[34] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[35] Gao Xinbo, Li Jie, and Xie Weixin, "Parameter optimization in fcm clustering algorithms," in *WCC 2000 - ICSP 2000. 2000 5th International Conference on Signal Processing Proceedings. 16th World Computer Congress 2000*, vol. 3, Aug 2000, pp. 1457–1461 vol.3.

[36] C. L. Blake and C. Merz, "Uci repository of machine learning databases. university of california, irvine, dept. of information and computer sciences," 1998.

[37] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[38] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "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*, vol. 64, no. 6, p. 061907, 2001.

[39] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "A dataset for breast cancer histopathological image classification," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 7, pp. 1455–1462, 2015.