



Noisy label relabeling by nonparallel support vector machine

A. Sahleh and M. Salahi*, 

Abstract

In machine learning, models are derived from labeled training data where labels signify classes and features define sample attributes. However, noise from data collection can impair the algorithm's performance. Blanco, Japón, and Puerto proposed mixed-integer programming (MIP) models within support vector machines (SVM) to handle label noise in training datasets. Nonetheless, it is imperative to underscore that their models demonstrate an observable escalation in the number of variables as sample size increases. The nonparallel support vector machine (NPSVM) is a binary classification method that merges the strengths of both SVM and twin SVM. It accomplishes this by determining two nonparallel hyperplanes by solving two optimization problems. Each hyperplane is strategically positioned to be closer to one of the classes while maximizing its distance

*Corresponding author

Received 30 August 2023; revised 4 December 2023; accepted 31 December 2023

Ali Sahleh

Department of Applied Mathematics, Faculty of Mathematical Sciences, Rasht, Iran.
e-mail: alisahleh@gmail.com

Maziar Salahi

Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran. e-mail: salahim@guilan.ac.ir

How to cite this article

Sahleh, A. and Salahi, M., Noisy label relabeling by nonparallel support vector machine. *Iran. J. Numer. Anal. Optim.*, 2024; 14(1): 265-290.

<https://doi.org/10.22067/ijnao.2023.84198.1307>

from the other class. In this paper, to take advantage of NPSVM's features, NPSVM-based relabeling (RENPSVM) MIP models are developed to deal with the label noises in the dataset. The proposed model adjusts observation labels and seeks optimal solutions while minimizing computational costs by selectively focusing on class-relevant observations within an ϵ -intensive tube. Instances exhibiting similarities to the other class are excluded from this ϵ -intensive tube. Experiments on 10 UCI datasets show that the proposed NPSVM-based MIP models outperform their counterparts in accuracy and learning time on the majority of datasets.

AMS subject classifications (2020): Primary 6BT09; Secondary 90C11.

Keywords: Label noise; SVM; Mixed-integer program; Nonparallel SVM.

1 Introduction

Support Vector Machine (SVM) [8, 34, 35] is a renowned technique employed for binary classification in diverse domains, such as abnormal recognition [22], stock market prediction [1], and pose estimation [36]. Despite its proficient performance, SVM encounters substantial computational demands when solving the Quadratic Programming Problem (QPP) for large datasets. In response to this challenge, Jayadeva, Khemchandani, and Chandra [19] introduced the Twin SVM (TWSVM), a method that utilizes two nonparallel hyperplanes. These hyperplanes are positioned closer to each of the two classes while maintaining a minimum unit distance from samples of the other class. In contrast to SVM, TWSVM tackles two smaller QPPs, thereby mitigating the training time complexity. The TWSVM framework has been extended through various adaptations, including the Wavelet TWSVM by Ding et al. [11, 12] with glowworm swarm optimization, an enhanced K-nearest neighbor TWSVM by Nasiri and Mir [26] addressing noise and outliers, and an automatic TWSVM by Jimenez-Castano, Alvarez-Meza, and Orozco-Gutierrez [20] for imbalanced datasets using kernel representation. Although TWSVM offers valuable attributes, it encounters a challenge in computing the inverses of specific matrices as part of its model training process. This task becomes impractical or even infeasible for siz-

able datasets when utilizing conventional methods. Conversely, the standard SVM can efficiently solve large-scale problems through algorithms like Sequential Minimal Optimization (SMO). To address this concern, Nonparallel SVM (NPSVM) [33] is introduced, which integrates SVM's benefits into TWSVM. This integration incorporates the utilization of the SMO algorithm [21, 28] and the concept of semi-sparseness [33], collectively enhancing the overall performance of the model.

The existence of label noise within datasets can have a substantial impact on the accuracy and generalizability of supervised learning algorithms. Instances with incorrect labels may originate from diverse origins, such as human errors, label switching, or intentional introduction of noise [25, 23, 24, 27]. It has been the subject of several research studies. Anguin and Laird [2] introduced a noise model that establishes a sample for learning in noisy environments. They also suggested computationally feasible learning algorithms for noisy domains and explored extending these concepts to broader contexts. However, a drawback of their model is the question of whether there are domains where approximately correct identification is computationally feasible without noise. However, it becomes computationally infeasible even with moderate levels of noise. Another study by Xiao et al. [38] devised an optimal attack strategy and used heuristic methods for practical computation. Biggio, Nelson, and Laskov [4] introduced an algorithmic strategy that effectively manages adversarial alterations of labels. This technique involves the adjustment of the kernel matrix when labels are independently flipped with equal probabilities. Another alternative, as presented in a prior work [15], entails the process of detecting and eliminating inaccurately labeled instances. This involves the selection of samples considered dubious and necessitating additional scrutiny. Obtaining labels with reduced levels of noise might entail increased time and expenses. Nevertheless, this endeavor holds the potential to significantly augment classification accuracy. To address this challenge, Duan and Wu [13] proposed a novel learning approach that leverages both noisy and less noisy labels extracted from a limited portion of the training dataset. This methodology involves the estimation of noise rate parameters and the inference of precise labels by utilizing a noise model built upon flipping probabilities and a logistic regres-

sion classifier. While the methods presented in [4, 15] effectively tackle label noise in data, they do so through a two-phase process. The aspect related to label noise is addressed before the training process, and the models do not handle label noise simultaneously with the primary task. Thulasidasan et al. [31] introduced an innovative approach to mitigate label noise within the context of Deep Neural Network (DNN) classification. Their methodology involves the introduction of a new loss function, enabling the DNN to decide to abstain from classifying certain samples, thereby avoiding confusion. At the same time, this approach improves the classification performance of samples that are not abstained from. The proposed method holds substantial promise for considerably augmenting the accuracy and robustness of DNN classifiers in real-world practical applications. In [7], the authors presented a technique for estimating the level of label noise and showed that implementing importance reweighting can enhance classification accuracy when dealing with label noise and evaluates the reliability of two classification approaches: Convolutional neural networks and convolutional neural networks with importance reweighting. Despite the merits of models in [31, 7], they suffer from the explosion of parameters as the number of layers increases for some tasks, such as natural language processing and computer vision, which results in demanding high computation resources. Blanco, Japón, and Puerto [5] introduced a unique approach for constructing optimal classification trees that consider the presence of noisy labels in the training data. Their method combines margin-based classifiers with outlier detection techniques to improve performance. It utilizes two main components: (1) the splitting rules of the tree are designed to maximize class separation margins, following SVM principles, and (2) during tree construction, some training sample labels can be adjusted to identify and address label noise. These elements are integrated to create the final optimal classification tree. Bertsimas et al. [3] introduced a robust optimization approach for addressing label noise by introducing a new variable representing the probability of mislabeling for each training point. They also imposed a constraint to limit the total number of mislabeled points below a specified threshold, considering worst-case scenarios. However, a limitation of their method is its primary focus on constructing a classifier robustly, concentrating on worst-case scenarios, and

controlling label noise with a budget hyperparameter. Also, their approach prioritizes worst-case scenarios and does not explore all possible parameter vectors (w, b) for different scenarios.

Recently, in [6], the authors have formulated SVM-based mixed-integer programming (MIP) models to effectively handle classification tasks in the presence of noisy labels. In contrast to the existing techniques, their methodology involves a simultaneous process of constructing an SVM-based classifier while adjusting the labels of observations to achieve an optimal solution. A significant advantage of their approach lies in its capability to derive separating hyperplanes that conventional SVM methods cannot achieve. However, it is important to highlight that the Relabel SVM-based (RESVM) model introduced in [6] exhibits a noteworthy increase in the number of variables as the number of samples rises. To tackle this issue, they further proposed a clustering-based relabeling (CRESVM) model by employing clustering and classification in the SVM framework.

In this paper, the relabeling idea is employed within NPSVM to take benefit of its features. Each proposed MIP model considers instances within the class represented by the ϵ -intensive tube. If any of these instances share similarities with the other class, the samples belonging to the original class are excluded from the ϵ -insensitive tube. In most datasets, RENPSVM exhibits fewer linear constraints and variables in comparison to RESVM. Moreover, while the CRESVM model introduced in [6] possesses fewer linear constraints and variables than both RESVM and RENPSVM, its demerit is the utilization of nonlinear constraints. Besides the above, the structure of NPSVM allows parallel implementation of the proposed MIP models, leading to faster learning times on most datasets. The main contributions of this paper are summarized as follows:

- (1) Expanding the NPSVM models into MIP models to address label noise in a manner that not only adjusts the labels of observations but also achieves an optimal solution simultaneously.
- (2) Minimizing the computational cost by avoiding the consideration of all observations as potential candidates for the label changes in the model.

Instead, we focus on instances related to the class that the model aims to represent within an epsilon (ϵ)-intensive tube.

- (3) Parallelization of the proposed MIP models, which results in faster learning times on the majority of datasets.
- (4) Computational experiments conducted on 10 UCI datasets reveal that RENPSVM outperforms RESVM and CRESVM in terms of classification accuracy while demonstrating similar learning times to RESVM and CRESVM.
- (5) The outcomes of evaluating our algorithms on diverse real-world datasets demonstrate that our suggestions exhibit greater resilience against attacks than the recent relabel models approach mentioned in [6].

The rest of this paper is organized as follows. Section 2 briefly reviews TWSVM and NPSVM. In Section 3, we delve into our proposed model and provide a comparison of the number of constraints and variables with RESVM and CRESVM models in [6]. Moving on to Section 4, computational experiments are conducted on 10 UCI datasets to illustrate the efficiency of the proposed models in comparison to those outlined in [6]. Also, this section encompasses two statistical tests aimed at highlighting differences between the proposed model and those from [6]. Ultimately, Section 5 presents concluding remarks.

2 Background

Consider a classification problem with the dataset $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$, where $x_i \in R^n$ and $y_i \in \{+1, -1\}$ for $i = 1, \dots, l$ denote samples and labels of samples, respectively. We further symbolize the sets of indices associated with positive and negative classes as I^+ and I^- , respectively. This is defined as

$$I^+ = \{i \mid y_i = +1\}, \quad I^- = \{i \mid y_i = -1\}.$$

In this section, we present a concise overview of the TWSVM and NPSVM models.

2.1 TWSVM

Consider $A \in \mathbb{R}^{m_1 \times n}$ and $B \in \mathbb{R}^{m_2 \times n}$ as the data matrices containing points belonging to I^+ and I^- , respectively. The TWSVM functions as a binary classifier, establishing two nonparallel hyperplanes via solving two smaller QPPs compared to a large one in SVM as follows:

$$\begin{aligned} \min_{w_1, b_1, \xi_2} \quad & \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + c_1 e_1^T \xi_2 \\ \text{s.t.} \quad & -(Bw_1 + e_2 b_1) + \xi_2 \geq e_2, \\ & \xi_2 \geq 0, \end{aligned} \quad (1)$$

and

$$\begin{aligned} \min_{w_2, b_2, \xi_1} \quad & \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + c_2 e_2^T \xi_1 \\ \text{s.t.} \quad & (Aw_2 + e_1 b_2) + \xi_1 \geq e_1, \\ & \xi_1 \geq 0, \end{aligned} \quad (2)$$

where c_1 and c_2 are predetermined trade-off factors between the error variable vectors ξ_1 and ξ_2 . Also, e_1 and e_2 are vectors of ones with appropriate dimensions. The first term in the objective function of (1) (or (2)) aims to maintain the hyperplane in proximity to the points of one class (I^+), while the constraints work to ensure that the hyperplane remains at a unit distance from the points of the other class (I^-). The Wolfe dual forms of (1) and (2) are given by

$$\begin{aligned} \max_{\alpha} \quad & e_2^T \alpha - \frac{1}{2} \alpha^T G (H^T H)^{-1} G^T \alpha \\ \text{s.t.} \quad & 0 \leq \alpha \leq c_1 e_2 \end{aligned} \quad (3)$$

and

$$\begin{aligned} \max_{\beta} \quad & e_1^T \beta - \frac{1}{2} \beta^T P (Q^T Q)^{-1} P^T \beta \\ \text{s.t.} \quad & 0 \leq \beta \leq c_2 e_1, \end{aligned} \quad (4)$$

where $G = [B; e_2]$, $H = [A; e_1]$, $P = [A; e_1]$, and $Q = [B; e_2]$. As we see, dual models involve using the inverse of $G^T G$ and $H^T H$, which are multiplied by Lagrangian multipliers $\alpha \in \mathbb{R}^{m_1}$ and $\beta \in \mathbb{R}^{m_2}$. Finally, the nonparallel hyperplanes are obtained from the solutions α and β of (3) and (4), respectively, through

$$z_1 = -(H^T H)^{-1} G^T \alpha, \quad \text{where } z_1 = [w_1^T \ b_1] \quad (5)$$

and

$$z_1 = -(Q^T Q)^{-1} P^T \beta, \quad \text{where } z_2 = [w_2^T \ b_2]. \quad (6)$$

While TWSVM handles smaller QPPs compared to SVM, it is not without drawbacks [32]. Firstly, TWSVM, in addressing its primal problems, exclusively minimizes empirical risk, neglecting the essential aspect of minimizing structural risk present in conventional SVMs. Secondly, to handle singularity concerns, TWSVM employs approximations by substituting inverse matrices, leading to solutions that are only approximative. Thirdly, the computational complexity of TWSVM is impeded by the necessity to compute inverse matrices, rendering it impractical for extensive datasets. Moreover, TWSVM is confined to linear classification and lacks a straightforward extension to non-linear scenarios. The demand for swift solvers, such as the SMO algorithm used for standard SVMs, adds another layer of complexity. Lastly, TWSVM compromises sparsity by employing a quadratic loss function, resulting in a situation where the majority of points in a class exert substantial influence on each decision function, consequently forfeiting the advantages associated with sparsity.

2.2 The NPSVM

The NPSVM, which is a generalized version of TWSVM, provides a more comprehensive formulation than TWSVM and determines two nonparallel hyperplanes using a similar approach. The key difference is that NPSVM represents each class within ϵ -insensitive tubes (Figure 1) and inherits the advantages of SVM that TWSVM lacks, such as utilizing the SMO algorithm and avoiding the computation of matrix inverses during its model training process. The NPSVM solves the following two QPPs:

$$\begin{aligned} \min_{w_1, b_1, \eta_1, \eta_2, \xi_1} \quad & \frac{1}{2} \|w_1\|^2 + c_1 e_1^T (\eta_1 + \eta_2) + c_2 e_2^T \xi_1 \\ \text{s.t.} \quad & (w_1^T x_i + b_1) \leq \epsilon_1 + \eta_{1i}, \quad i \in I^+, \\ & -(w_1^T x_i + b_1) \leq \epsilon_1 + \eta_{2i}, \quad i \in I^+, \\ & w_1^T x_i + b_1 \leq -1 + \xi_{1i} \quad i \in I^-, \\ & \xi_1, \eta_1, \eta_2 \geq 0, \end{aligned} \quad (7)$$

and

$$\begin{aligned}
 \min_{w_2, b_2, \eta'_1, \eta'_2, \xi_2} \quad & \frac{1}{2} \|w_2\|^2 + c_3 e_2^T (\eta'_1 + \eta'_2) + c_4 e_1^T \xi_1 \\
 \text{s.t.} \quad & w_2^T x_i + b_2 \leq \epsilon_2 + \eta'_{1i}, & i \in I^-, \\
 & -(w_2^T x_i + b_2) \leq \epsilon_2 + \eta'_{2i}, & i \in I^-, \\
 & w_2^T x_i + b_2 \geq 1 - \xi_{2i} & i \in I^+, \\
 & \xi_2, \eta'_1, \eta'_2 \geq 0,
 \end{aligned} \tag{8}$$

where $c_i > 0$ ($i = 1, \dots, 4$) are trade-off factors for error variables η_i and ξ_i ($i = 1, 2$). The aim of (7) is to maximize the margin between the hyperplanes of ϵ -intensive tube, which can be mathematically expressed as $\frac{2\epsilon}{\|w_1\|}$. The first and second set of constraints ensure that the positive class is largely concentrated within the ϵ -band situated between the hyperplanes $w_1^T x + b_1 = \epsilon$ and $(w_1^T x) + b_1 = -\epsilon$. The third set of constraints push away negative class from the hyperplane $w_1^T x + b_1 = -1$ as far as possible. Similar description holds for problem (8).

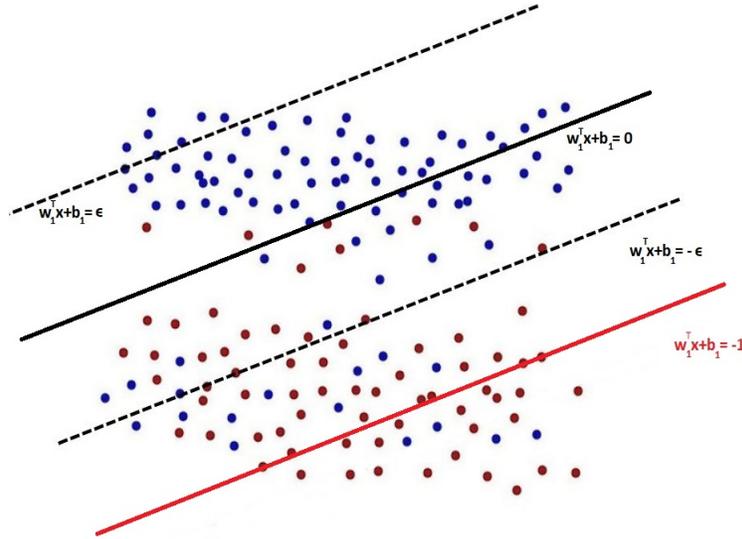


Figure 1: Illustration of NPSVM.

3 Relabel NPSVM

In contrast to TWSVM, NPSVM eliminates the need for specific matrix inversions during model training, making it particularly advantageous when dealing with substantial datasets, where traditional methods may become extremely challenging or unfeasible. However, akin to the standard SVM, NPSVM retains efficiency in addressing large-scale problems by utilizing the SMO algorithm. In NPSVM, the introduction of an ϵ -insensitive loss function naturally incorporates a regularization term. This characteristic distinguishes it from the initial TWSVM or improved TBSVM, with these latter models being special cases of the more general NPSVM. Notably, NPSVM reverts to the initial TWSVM or TBSVM when the corresponding parameters are appropriately chosen. Additionally, the transition from semi-sparseness to complete sparseness is promoted within the NPSVM framework [33]. In this section, we explore the implementation of the relabeling approach within NPSVM, aiming to bolster its robustness against label noise in datasets.

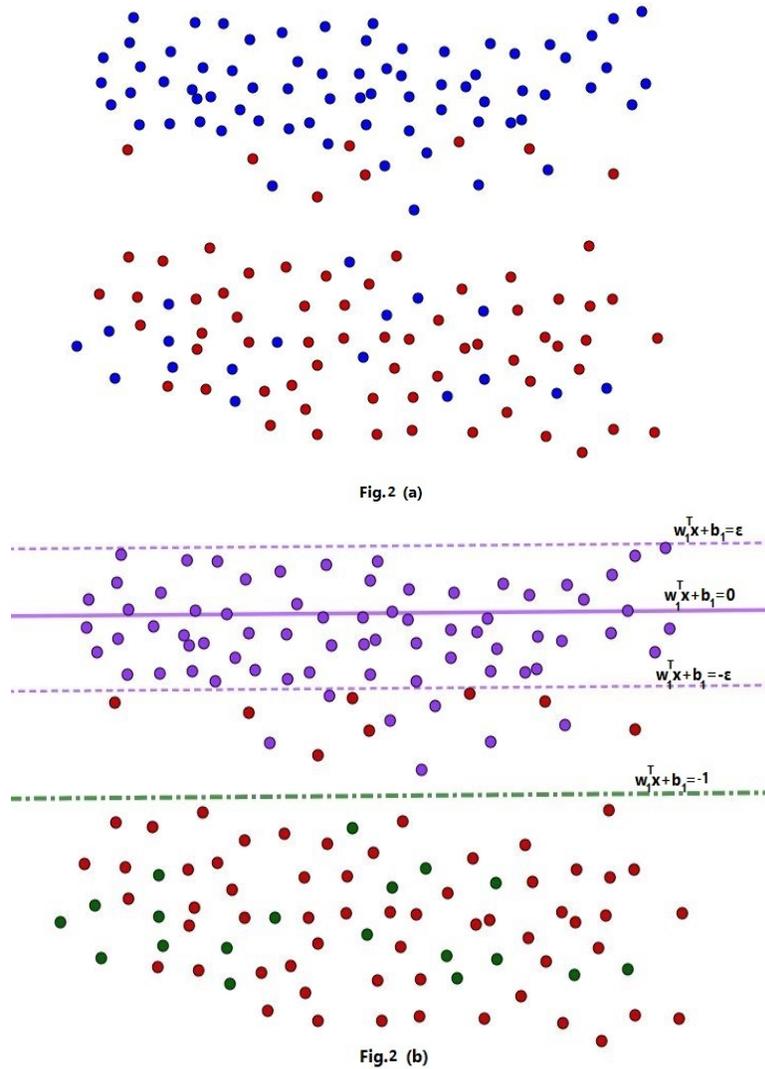


Figure 2: Original data dataset Figure 2 (a). Optimal separating hyperplanes with (9) Figure 2 (b). Instances from the positive dataset that remain within the ϵ -intensive tube are colored purple, while instances that are excluded from the respective constraint are colored green.

To apply relabeling on NPSVM, initially, we aim for the positive class to be predominantly positioned within the ϵ -intensive tube while maximizing its distance from the other class. The gap between the hyperplanes of ϵ -

intensive tube is controlled via the loss function. This results in enhancing the alignment between the nonparallel hyperplanes and the classes they represent. Due to the presence of label noise, there are some instances that appear to belong to a specific class based on their labels, but they bear a resemblance to a different class (the blue color in Figure 2 (a)). To determine whether the samples of the positive class are to be included within the ϵ -insensitive tube or not, a binary variable vector is incorporated into the model. This vector determines which instances are removed from the ϵ -insensitive tube (green color in Figure 2 (b)) and which instances remain inside it (purple color in Figure 2 (b)). This binary variable vector is integrated into both the constraints and the objective function, effectively preventing the extensive relabeling of observations. The first relabeling NPSVM (RENPSVM) model is formulated as follows:

$$\begin{aligned}
 \min_{w_1, b_1, \eta_1, \eta_2, \xi_1, \theta_1} \quad & \frac{1}{2} \|w_1\|^2 + c_1 e_1^T (\eta_1 + \eta_2) + c_2 e_3^T \xi_1 + c_3 e_1^T \theta_1 \\
 \text{s.t.} \quad & (w_1^T x_i + b_1) \theta_{1i} \leq \epsilon_1 + \eta_{1i}, & i \in I^+, \\
 & -(w_1^T x_i + b_1) \theta_{1i} \leq \epsilon_1 + \eta_{2i}, & i \in I^+, \\
 & (w_1^T x_i + b_1)(1 - \theta_{1i}) \leq -1 + \xi_{1i} + M_1(1 - \theta_{1i}), & i \in I^+, \\
 & w_1^T x_i + b_1 \leq -1 + \xi_i & i \in I^-, \\
 & \theta_{1i} \in \{0, 1\}, & i \in I^+, \\
 & w_1 \in \mathcal{R}^n, b_1 \in \mathcal{R}, \\
 & \eta_1, \eta_2, \xi_1 \geq 0,
 \end{aligned} \tag{9}$$

where $c_i > 0$ ($i = 1, 2, 3$) are trade-off parameters and M_1 is large positive constant that is chosen such that its associated constraint becomes redundant when $\theta = 0$. The second term in objective function $\eta_1 + \eta_2$ controls the error for the gap between the hyperplanes $w_1^T x + b_1 = \epsilon$ and $w_1^T x + b_1 = -\epsilon$. In the third set of constraints, we strive to distance the negative class from the hyperplane $w_1^T x + b_1 = -1$. The error vector ξ_1 is assessed using the soft margin loss function. The binary variable vector θ_1 determines whether the samples of the positive class are to be included within the ϵ -insensitive tube or not. Typically, when $\theta_{1i} = 1$, it signifies that the i th instance belongs to the positive class. This is represented by the first three sets of constraints in (9). The final term in the objective function serves to avoid extensive reassignment

of labels to observations, a situation that might result in producing ineffectual classifiers. The second RENPSVM model corresponding to other classes is as follows:

$$\begin{aligned}
& \min_{w_2, b_2, \eta'_1, \eta'_2, \xi_2, \theta_2} \frac{1}{2} \|w_2\|^2 + c_4 e_2^T (\eta'_1 + \eta'_2) + c_5 e_3^T \xi_2 + c_6 e_2^T \theta_2 \\
& \text{s.t. } (w_2^T x_i + b_2) \theta_{2i} \leq \epsilon_2 + \eta'_1, & i \in I^-, \\
& -(w_2^T x_i + b_2) \theta_{2i} \leq \epsilon_2 + \eta'_2, & i \in I^-, \\
& (w_2^T x_i + b_2)(1 - \theta_{2i}) \geq 1 - \xi_{2i} - M_2(1 - \theta_{2i}), & i \in I^-, \\
& w_2^T x_i + b_2 \geq 1 - \xi_{2i}, & i \in I^+, \\
& \theta_{2i} \in \{0, 1\}, & i \in I^-, \\
& w_2 \in \mathcal{R}^n, b_2 \in \mathcal{R}, \\
& \eta'_1, \eta'_2, \xi_2 \geq 0.
\end{aligned} \tag{10}$$

Both models (9) and (10) exhibit nonlinearity in the constraints. To linearize those constraints, we introduce variables $\alpha, \alpha_0, \beta, \beta_0$ as

$$\beta_i = w_1 \theta_{1i}, \quad \beta_{0i} = b_1 \theta_{1i}, \tag{11}$$

$$\alpha_i = w_2 \theta_{2i}, \quad \alpha_{0i} = b_2 \theta_{2i}. \tag{12}$$

Now by adding the following constraints to (9) and (10)

$$\begin{aligned}
w_1 - M_3 \theta_i &\leq \beta_i \leq w_1 + M_3 \theta_i, & i \in I^+, \\
-M_3(1 - \theta_{1i}) &\leq \beta_i \leq M_3(1 - \theta_{1i}), & i \in I^+, \\
b_1 - M_3 \theta_{1i} &\leq \beta_{0i} \leq b_1 + M_3 \theta_{1i}, & i \in I^+, \\
-M_3(1 - \theta_{1i}) &\leq \beta_{0i} \leq M_3(1 - \theta_{1i}), & i \in I^+,
\end{aligned}$$

and

$$\begin{aligned}
w_2 - M_4(\theta_{2i}) &\leq \alpha_i \leq w_2 + M_4(\theta_{2i}), & i \in I^-, \\
-M_4(1 - \theta_{2i}) &\leq \alpha_i \leq M_4(1 - \theta_{2i}), & i \in I^-, \\
b_2 - M_4(\theta_{2i}) &\leq \alpha_{0i} \leq b_2 + M_4(\theta_{2i}), & i \in I^-, \\
-M_4(1 - \theta_{2i}) &\leq \alpha_{0i} \leq M_4(1 - \theta_{2i}), & i \in I^-.
\end{aligned}$$

We obtain the following problems:

$$\begin{aligned}
& \min_{w_1, b_1, \eta_1, \eta_2, \beta, \beta_0, \xi_1, \theta_1} \frac{1}{2} \|w_1\|^2 + c_1 e_1^T (\eta_1 + \eta_2) + c_2 e_3^T \xi_1 + c_3 e_1^T \theta_1 \\
& \text{s.t. } \beta_i^T x_i + \beta_{0i} \leq \epsilon_1 + \eta_{1i}, & i \in I^+, \\
& \quad -(\beta_i^T x_i + \beta_{0i}) \leq \epsilon_1 + \eta_{2i}, & i \in I^+, \\
& \quad w_1^T x_i + b_1 - (\beta_i^T x_i + \beta_{0i}) \leq -1 + \xi_{1i} + M_1(1 - \theta_{1i}), & i \in I^+, \\
& \quad w_1^T x_i + b_1 \leq -1 + \xi_i & i \in I^-, \\
& \quad w_1 - M_3 \theta_i \leq \beta_i \leq w_1 + M_3 \theta_i, & i \in I^+, \\
& \quad -M_3(1 - \theta_{1i}) \leq \beta_i \leq M_3(1 - \theta_{1i}), & i \in I^+, \\
& \quad b_1 - M_3 \theta_{1i} \leq \beta_{0i} \leq b_1 + M_3 \theta_{1i}, & i \in I^+, \\
& \quad -M_3(1 - \theta_{1i}) \leq \beta_{0i} \leq M_3(1 - \theta_{1i}), & i \in I^+, \\
& \quad \beta_i \in \mathcal{R}^n, \beta_{0i} \in \mathcal{R}, & i \in I^+, \\
& \quad \theta_{1i} \in \{0, 1\}, & i \in I^+, \\
& \quad w_1 \in \mathcal{R}^n, b_1 \in \mathcal{R}, \\
& \quad \eta_1, \eta_2, \xi_1 \geq 0,
\end{aligned}$$

and

$$\begin{aligned}
& \min_{w_2, b_2, \eta'_1, \eta'_2, \alpha, \alpha_0, \xi_2, \theta_2} \frac{1}{2} \|w_2\|^2 + c_4 e_2^T (\eta'_1 + \eta'_2) + c_5 e_3^T \xi_2 + c_6 e_2^T \theta_2 \\
& \text{s.t. } \alpha_i^T x_i + \alpha_{0i} \leq \epsilon_2 + \eta'_1, & i \in I^-, \\
& \quad -(\alpha_i^T x_i + \alpha_{0i}) \leq \epsilon_2 + \eta'_2, & i \in I^-, \\
& \quad w_2^T x_i + b_2 - (\alpha_i^T x_i + \alpha_{0i}) \geq 1 - \xi_{2i} - M_2(1 - \theta_{2i}), & i \in I^-, \\
& \quad w_2^T x_i + b_2 \geq 1 - \xi_{2i}, & i \in I^+, \\
& \quad w_2 - M_4(\theta_{2i}) \leq \alpha_i \leq w_2 + M_4(\theta_{2i}), & i \in I^-, \\
& \quad -M_4(1 - \theta_{2i}) \leq \alpha_i \leq M_4(1 - \theta_{2i}), & i \in I^-, \\
& \quad b_2 - M_4(\theta_{2i}) \leq \alpha_{0i} \leq b_2 + M_4(\theta_{2i}), & i \in I^-, \\
& \quad -M_4(1 - \theta_{2i}) \leq \alpha_{0i} \leq M_4(1 - \theta_{2i}), & i \in I^-, \\
& \quad \alpha_{2i} \in \mathcal{R}^n, \alpha_{0i} \in \mathcal{R}, & i \in I^-, \\
& \quad \theta_2 \in \{0, 1\}, & i \in I^-, \\
& \quad w_2 \in \mathcal{R}^n, b_2 \in \mathcal{R}, \\
& \quad \eta'_1, \eta'_2, \xi_2 \geq 0,
\end{aligned}$$

where M_i , with $i = 1, \dots, 4$, represent significant positive constants. As known, MIP models are NP-hard problems, and solving large-scale MIP problems can be computationally challenging and often requires sophisticated optimization algorithms and heuristics to find near-optimal solutions within reasonable time frames [37]. To compare the proposed MIP mod-

els with those in [6], their number of variables and constraints of models are provided in Table 1. According to this table, it becomes evident that the number of variables for each model of RENPSVM is lower than that of RESVM when $\frac{m_i}{l} \leq \frac{n+2}{n+4}$ (for $i = 1, 2$), which is the case for datasets where the number of features exceeds seven. Also, each RENPSVM MIP model has less linear constraints than RESVM when $\frac{m_i}{l} \leq \frac{n+1}{n+2}$ (for $i = 1, 2$), which is the case for datasets where $n \geq 7$. It should also be noted that despite the fact that CRESVM has less linear constraints compared to both RESVM and RENPSVM, its demerit is that it has nonlinear constraints.

Table 1: Number of variables and constraints of RESVM, CRESVM, and RENPSVM

	RESVM	CRESVM	RENPSVM (13)	RENPSVM (13)
Variables	$ln + 3l + n + 1$	$4l + 3n + 1$	$m_1n + 4m_1 + l + n + 1$	$m_2n + 4m_2 + l + n + 1$
Linear constraints	$6l + 4ln$	$5l$	$4m_1n + 9m_1 + m_2 + l$	$4m_2n + 9m_2 + m_1 + l$
Nonlinear constraints	0	$2l$	0	0

4 Computational experiments

To demonstrate the effectiveness of RENPSVM, we conducted experiments using a set of 10 UCI datasets, as detailed in Table 2. To evaluate the models' resilience to label noise, we executed three distinct experiments for each dataset. For the Vertebral dataset, two scenarios are considered. First (Vertebral1), distinguishing patients as either Normal (100) or those with Disk Hernia (60); and second (Vertebral2), categorizing patients as either Normal (100) or Abnormal, with Abnormal encompassing individuals with Disk Hernia (60) or Spondylolisthesis (150). These experiments encompassed the original datasets, along with two scenarios involving the introduction of random label flips in the training data at percentages of 20% and 50%. The implementation of all models is carried out in MATLAB 2020 (64-bit) on a computer equipped with an Intel Core i5 processor and 4 GB of RAM.

Also, RESVM, CRESVM, and RENPSVM models are solved using the CVX-Mosek [16]. For all models, the hyperparameters c_i ($i = 1, \dots, 4$) are chosen from the set $\{2^i, i = -8, \dots, 8\}$, taking into consideration their impact on the models' performance. To mitigate issues like overfitting and bias across all datasets, we employed a 10-fold cross-validation methodology. This technique partitions the dataset into ten equally sized subsets, as recommended by [14]. Subsequently, the models are trained on nine of these subsets, while the remaining subset is utilized to compute the prediction error of the models. This process is repeated for each of the ten subsets. Finally, the average classification accuracy is computed using the following formula:

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

where TP, TN, FP, and FN denote the number of true positive, true negative, false positive and false negative, respectively. Computational results are summarized in Table 3.

Table 2: Characteristics of datasets

Datasets	Samples	Positive	Negative	Features	Classes
Car	1594	1210	384	7	2
Haberman	306	225	81	3	2
Cancer	699	458	241	9	2
Vertebral	310	60	100	7	3
Hayes-Ruth	102	51	51	5	2
Diabetes	768	500	268	8	2
Ionosphere	351	225	126	34	2
Votes	435	267	168	16	2
Heart	270	260	120	13	2

Table 3: Performance comparison of all models

Datasets	Method	Flip percentage		
		0.0	0.2	0.5
		Accuracy(Time)		
Car	RESVM	100(53.40)	79.98(51.38)	51.38(56.04)
	CRESVM	100(68.61)	78.41(82.67)	52.01 (86.69)
	RENPSVM	100(161.89)	79.99 (99.37)	51.25(73.93)
Haberman	RESVM	71.29(449.52)	63.36(405.02)	48.93(116.95)
	CRESVM	75.438(13.79)	59.57(290.36)	49.18(188.52)
	RENPSVM	72.87(104.42)	66.025(30.28)	55.89 (28.32)
Cancer	RESVM	94.85(40.15)	58.50(125.75)	48.93(116.95)
	CRESVM	96.28(311.55)	69.07(311.35)	48.074 (309.69)
	RENPSVM	97 (45.92)	77.97(42.02)	53.06(36.01)
Vertebral1	RESVM	100(23.055)	77.07(108.57)	55.01 (108.75)
	CRESVM	100(114.80)	76.87(9.23)	55.01 (21.57)
	RENPSVM	100(16.12)	80 (16.21)	56.26 (30.89)

Vertebral2	RESVM	81.29 (347.71)	68.81(366.69)	55.005(108.76)
	CRESVM	76.7742(311.88)	64.39(25.41)	51.62(96.93)
	RENPSVM	79.67(18.47)	69.032(18.06)	55.447(18.82)
Hayes-roth	RESVM	51.09(111.36)	50.12(107.37)	52.90(108.51)
	CRESVM	53.51(308.04)	53.92(51.91)	49.22(47.81)
	RENPSVM	62.54(32.57)	62.30(14.33)	54.63(13.99)
Diabet	RESVM	65.10(115.84)	55.73(114.52)	51.53(115.80)
	CRESVM	69.52(309.61)	55.75(281.17)	50.51 (305.35)
	RENPSVM	75.65(43.79)	56.51(34.75)	51.69(36.04)
Ionosphere	RESVM	84.88(316.98)	56.13(318.21)	51.85(316.65)
	CRESVM	82.9(312.50)	69.22(40.40)	45.48(27.55)
	RENPSVM	85.18(68.81)	68.39(65.44)	53.85(65.21)
Votes	RESVM	95.88 (310.98)	76.99(310.41)	50.33 (319.23)
	CRESVM	94.49(299.53)	74.41(308.84)	44.17(311.22)
	RENPSVM	95.62(35.25)	78.88(32.73)	54.28 (27.02)

Heart	RESVM	78.89(09.23)	54.81(309.49)	53.387 (359.49)
	CRESVM	83.70(28.73)	74.41(308.84)	52.96 (72.06)
	RENPSVM	85.18(21.81)	72.59(326.27)	55.93 (319.05)

According to Table 3, for the original datasets, RENPSVM exhibits superior accuracy compared to the other models across all datasets except for Votes and Vertebral2. Additionally, it demonstrates equivalent accuracy to the other models for datasets such as Car and Vertebral1. In the aspect of learning time, RENPSVM outperforms CRESVM and RESVM, except for Car. Additionally, RENPSVM secures the second-best position in terms of learning time for Haberman and Cancer datasets. When considering a label flip scenario of 20%, it becomes evident that the accuracy of RENPSVM surpasses that of the other models for all datasets except Heart and Ionosphere. However, in terms of learning time, RENPSVM generally outperforms other models, except for Car, Vertebral1, Ionosphere, and Heart. Among these, Vertebral and Ionosphere are notable as being the second-best in terms of learning time. When dealing with a label flip rate of 50%, among all the datasets, only Car does not exhibit the highest accuracy with the proposed model. Turning to learning time, the proposed model demonstrates superiority over all other models, except for Car, Vertebral1, Ionosphere, and Heart, which secure the position of being the second-best performers. By analyzing the comprehensive results presented in Table 3, it becomes evident that as the percentage of flipped labels increases, the proposed model exhibits superior accuracy compared to the referenced models and demonstrates enhanced robustness.

Next, the modified Friedman test is initially conducted to assess whether distinctions exist among the three models. Following this, the Nemenyi post-hoc test is utilized to enable the comparison of multiple methods, offering pairwise assessments between them. This post-hoc analysis assists in deter-

mining the presence of significant differences between the considered methods.

To determine whether the results of the three models in Table 3 differ or not, the modified Friedman test is conducted. The Friedman test, being a nonparametric statistical test, does not rely on assumptions about the underlying data distribution [9]. For each dataset, individual ranks are assigned to all algorithms, with the top-performing algorithm receiving rank 1, the second-best algorithm receiving rank 2, and so forth. In cases of ties, average ranks are employed. Let r_{ij} denote the rank of the j th algorithm on the i th dataset. The test examines the average rank for each algorithm, denoted as $\bar{r}_j = \sum_{i=1}^n r_{ij}$. To account for the potential conservatism of the Friedman test, a modified version of it is calculated as outlined by [18].

$$F_f = \frac{(N-1)\mathcal{X}_F^2}{N(k-1) - \mathcal{X}_F^2}, \quad (13)$$

where \mathcal{X}_F^2 is equal to $\frac{12N}{k(k+1)}(\sum_{j=1}^k \bar{r}_j^2 - \frac{k(k+1)^2}{4})$, N represents the number of datasets, and k denotes the number of methods. Furthermore, F_f is distributed according to the F-distribution with degrees of freedom $(k-1, (k-1)(N-1))$. The average accuracy ranks corresponding to Table 3 are presented in a tabular format as shown in Table 4. The critical value at a significance level of $\alpha = 0.1$ for $F_f(2, 18)$ is determined to be 3.63. Considering the average ranks (Table 4), the \mathcal{X}_F^2 values for scenarios with label flip percentages of 0%, 20%, and 50% are 3.8, 9.8, and 10.05, respectively. The corresponding F_f values are 2.1111, 8.6471, and 9.0905. Given that the F_f values for the 20% and 50% scenarios exceed the critical value of $F_f(3, 16) = 3.63$, and considering that the rank of RENPSVM is lower than that of RESVM and CRESVM, it can be inferred that there exists a significant distinction between RENPSVM and the models introduced in [6] for these particular scenarios.

Table 4: Average accuracy rank of all models

	Method	Flip percentage		
		0.0	0.2	0.5
Average rank	RESVM	2.3	2.5	2.25
	CRESVM	2.2	2.3	2.55
	RENPSVM	1.5	1.2	1.2

The Nemenyi post-hoc test serves as a statistical technique utilized for the comparison of multiple methods, offering pairwise comparisons to ascertain the presence of significant distinctions. To execute this test for pairwise comparisons, we compute a parameter known as the critical difference (CD). The CD is determined by considering the number of datasets, the number of methods, the chosen significance level, and the average rank associated with each model from Table 4. When the difference in average ranks between the two methods exceeds the CD value, it can be inferred that a noteworthy and statistically significant difference exists between those two methods. The CD value is calculated as follows:

$$CD = q_{\alpha=0.1} \sqrt{\frac{k(k+1)}{6N}},$$

where the parameter q represents the critical value, while k signifies the number of models, and N denotes the number of datasets. For a significance level of 0.1 and considering four methods, the critical value extracted from the Nemenyi distribution table amounts to $q = 2.3122$. Substituting these values into the above equation yields a computed CD value of 1.0340. The difference between the average ranks of the two models is represented as Ξ . In the scenario where the label flip percentage is 0%, we encounter the following conditions:

$$\Xi(\text{RENPSVM} - \text{RESVM}) = |1.5 - 2.3| = 0.8 < CD(1.0340),$$

$$\Xi(\text{RENPSVM} - \text{CRESVM}) = |1.5 - 2.2| = 0.7 < CD(1.0340),$$

When the flip percentage of labels is 20% we have:

$$\Xi(\text{RENPSVM} - \text{RESVM}) = |1.2 - 2.5| = 1.3 > CD(1.0340),$$

$$\Xi(\text{RENPSVM} - \text{CRESVM}) = |1.2 - 2.3| = 1.1 > CD(1.0340).$$

Finally, in the case when the flip percentage of labels is 50%, we have

$$\Xi(\text{RENPSVM} - \text{RESVM}) = |1.2 - 2.25| = 1.05 > CD(1.0340),$$

$$\Xi(\text{RENPSVM} - \text{CRESVM}) = |1.2 - 2.55| = 1.35 > CD(1.0340).$$

Based on the preceding results, it is evident that a significant difference exists between RENPSVM and the other models, except for the original dataset.

5 Conclusions

In this paper, we have introduced MIP models based on NPSVM for the purpose of relabeling noisy data. Our approach effectively refines observation labels while simultaneously achieving an optimal solution. We achieve significant reductions in computational costs by strategically avoiding the consideration of all observations as potential candidates for label adjustments in the model. Instead, we concentrate on instances associated with the class that the model aims to represent within an ϵ -intensive tube. The inherent structure of NPSVM allows for parallel execution of the proposed MIP models, resulting in accelerated learning times across the majority of datasets. Our findings indicate that, for datasets with a number of features exceeding seven, each RENPSVM MIP model has fewer linear constraints and variables compared to RESVM, subject to specific conditions. This holds true for the majority of datasets. Additionally, the CRESVM model also exhibits fewer linear constraints and variables compared to both RESVM and RENPSVM, although it introduces the trade-off of incorporating nonlinear constraints. The effectiveness of our proposed models is evaluated through experiments conducted on 10 UCI datasets. The outcomes showcased that RENPSVM models exhibit better performance in terms of classification accuracy and

learning, akin to RESVM and CRESVM, respectively, for most datasets and as the percentage of flipped labels increases, the proposed RENPSVM model demonstrates superior accuracy compared to the referenced models and showcases enhanced robustness. Moreover, we employed the modified Friedman test and Nemenyi post-hoc test to assess the influence of label noise on our model's performance relative to other methods. The tests revealed that a notable distinction between RENPSVM and other models exists, except for the original datasets. For future work, one may consider extending the proposed model to multi-class classification, either through a one-vs-one-vs-rest approach [29] or by adapting it into a regression model. This adaptation can be particularly useful for handling label noise in target values, a common challenge in regression tasks [30]. Also, when dealing with datasets containing a large number of features, the computational cost can become prohibitively high. In such cases, it is efficient to derive hyperplane classifiers using the dual problem formulation [10]. Therefore, studying label noise using dual models might be another interesting future research direction.

6 Acknowledgements

The authors would like to thank the editor and reviewers for their useful comments and suggestions.

References

- [1] Abdi, A., Nabi, R.M., Sardasht, M. and Mahmood R. *Multiclass classifiers for stock price prediction: A comparison study*, J. Harbin Inst. Technol. 54(3) (2022), 32–39.
- [2] Angluin, D. and Laird, P. *Learning from noisy examples*, Mach. Learn. 2 (1988), 343–370.
- [3] Bertsimas, D., Dunn, J., Pawlowski, C. and Zhuo, Y.D. *Robust classification*, INFORMS Journal on Optimization 1(1) (2019), 2–34.

- [4] Biggio, B., Nelson, B. and Laskov, P. *Support vector machines under adversarial label noise*, Asian conference on machine learning (2011), 97–112.
- [5] Blanco, V., Japón, A. and Puerto, J. *Robust optimal classification trees under noisy labels*, Adv. Data Anal. Classif. 16(1) (2022), 155–179.
- [6] Blanco, V., Japón, A. and Puerto, J. *A mathematical programming approach to SVM-based classification with label noise*, Comput. Ind. Eng. 172 (2022), 108611.
- [7] Chen, Z., Song, A., Wang, Y., Huang, X. and Kong, Y. *A noise rate estimation method for image classification with label noise*, J. Phys. Conf. Ser. IOP Publishing 2433(1) (2023), 012039.
- [8] Cortes, C. and Vapnik, V.N. *Support vector networks*, Mach. Learn. 20(3) (1995), 273–297.
- [9] Demsar, J. *Statistical comparisons of classifiers over multiple data sets*, J. Mach. Learn Res. 7 (2006), 1–30.
- [10] Deng, N., Tian, Y. and Zhang, C. *Support vector machines: Optimization based theory, algorithms, and extensions*, CRC press, 2012.
- [11] Ding, S., Zhang, N., Zhang, X. and Wu, F. *Twin support vector machine: Theory, algorithm and applications*, Neural Comput. Appl. 28(11) (2017), 3119–3130.
- [12] Ding, S., Zhao, X., Zhang, J., Zhang, X. and Xue, Y. *A review on multi-class TWSVM*, Artif. Intell. Rev. 52(2) (2019), 775–801.
- [13] Duan, Y. and Wu, O. *Learning with auxiliary less-noisy labels*, IEEE Trans. Neural Netw. Learn. Syst. 28(7) (2018), 1716–1721.
- [14] Duda, R.O., Hart, P.E. and Stork, D.G. *Pattern Classification*, John Wiley & Sons, 2012.
- [15] Ekambaram, R., Fefilatyeve, S., Shreve, M., Kramer, K., Hall, L.O. and Goldgof, D.B. *Active cleaning of label noise*, Pattern Recognit. 51 (2016), 463–480.

- [16] Grant, M., Boyd, S. and Ye, Y. *Cvx: Matlab software for disciplined convex programming*, version 2.0 beta, 2013.
- [17] Hassani, S.F., Eskandari, S. and Salahi, M. *CInf-FS: An efficient infinite feature selection method using K-means clustering to partition large feature spaces*, Pattern Anal. Appl. (2023), 1–9.
- [18] Iman, R.L. and Davenport, J.M. *Approximations of the critical region of the fbietkan statistic*, Commun. Stat. Theory Methods 9 (6) (1980), 571–595.
- [19] Jayadeva, Khemchandani, R. and Chandra, S. *Twin support vector machines for pattern classification*, Trans. Pattern Anal. Mach. Intell. 29(5) (2007), 905–910.
- [20] Jimenez-Castano, C., Alvarez-Meza, A. and Orozco-Gutierrez, A. *Enhanced automatic twin support vector machine for imbalanced data classification*, Pattern Recognit. 107 (2020), 107442.
- [21] Keerthi, S.S., Shevade, S.K., Bhattacharyya, C. and Murthy, K.R.K. *Improvements to platt's SMO algorithm for SVM classifier design*, Neural Comput. 13(3) (2001), 637–649.
- [22] Kshirsagar, A.P. and Shakkeera, L. *Recognizing Abnormal Activity Using MultiClass SVM Classification Approach in Tele-health Care*, IOT with Smart Systems: Proceedings of ICTIS 2021, Springer Singapore, 2 (2022), 739–750.
- [23] Lachenbruch, P.A. *Discriminant analysis when the initial samples are misclassified*, Technometrics 8(4) (1966), 657–662.
- [24] Lachenbruch, P.A. *Note on initial misclassification effects on the quadratic discriminant function*, Technometrics 21(1) (1979), 129–132.
- [25] McLachlan, G.J. *Asymptotic results for discriminant analysis when the initial samples are misclassified*, Technometrics 14(2) (1972), 415–422.
- [26] Nasiri, J.A. and Mir, A.M. *An enhanced KNN-based twin support vector machine with stable learning rules*, Neural Comput. Appl. 16 (2020), 12949–12969.

- [27] Okamoto, S. and Yugami, N. *An average-case analysis of the k-nearest neighbor classifier for noisy domains*, 15th International Joint Conference on Artificial Intelligence (IJCAI) (1997), 238–245.
- [28] Platt, J. *Fast Training of Support Vector Machines using Sequential Minimal Optimization*, MIT Press, 1998.
- [29] Sahleh, A., Salahi, M. and Eskandari, S. *Multi-class nonparallel support vector machine*, Prog. Artif. Intell. (2023), 1–15.
- [30] Tanveer, M., Rajani, T., Rastogi, R., Shao, Y.H. and Ganaie, M.A. *Comprehensive review on twin support vector machines*, Ann. Oper. Res. (2022), 1–46.
- [31] Thulasidasan, S., Bhattacharya, T., Bilmes, J., Chennupati, G., and Mohd-Yusof, J. *Combating label noise in deep learning using abstention*, arXiv preprint arXiv, (2019), 1905.10964.
- [32] Tian, Y. and Qi, Z. *Review on: twin support vector machines*, Ann. Data Sci. 1 (2014), 253–277.
- [33] Tian, Y., Qi, Z., Ju, X., Shi, Y. and Liu, X. *Nonparallel support vector machines for pattern classification*, IEEE Trans. Cybern. 44(7) (2014), 1067–1079.
- [34] Vapnik, V.N. *The Nature of Statistical Learning Theory*, Springer, New York, 1996.
- [35] Vapnik, V.N. *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.
- [36] Wittoonchart, P. and Chongstitvatana, P. *Application of structured support vector machine backpropagation to a convolutional neural network for human pose estimation*, Neural Networks 92 (2017), 39–46.
- [37] Wolsey, L.A. and Nemhauser, G.L. *Integer and combinatorial optimization*, John Wiley & Sons, 55, 1999.
- [38] Xiao, H., Biggio, B., Nelson, B., Xiao, H., Eckert, C. and Roli, F. *Support vector machines under adversarial label contamination*, Neurocomputing 160 (2015), 53–62.