Annals of Mathematics and Artificial Intelligence
https://doi.org/10.1007/s10472-023-09877-8

An improved multi-task least squares twin support vector machine

Hossein Moosaei[1,2] · **Fatemeh Bazikar**[3] · **Panos M. Pardalos**[4]

Accepted: 1 June 2023

Abstract

In recent years, multi-task learning (MTL) has become a popular field in machine learning and has a key role in various domains. Sharing knowledge across tasks in MTL can improve the performance of learning algorithms and enhance their generalization capability. A new approach called the multi-task least squares twin support vector machine (MTLS-TSVM) was recently proposed as a least squares variant of the direct multi-task twin support vector machine (DMTSVM). Unlike DMTSVM, which solves two quadratic programming problems, MTLS-TSVM solves two linear systems of equations, resulting in a reduced computational time. In this paper, we propose an enhanced version of MTLS-TSVM called the improved multi-task least squares twin support vector machine (IMTLS-TSVM). IMTLS-TSVM offers a significant advantage over MTLS-TSVM by operating based on the empirical risk minimization principle, which allows for better generalization performance. The model achieves this by including regularization terms in its objective function, which helps control the model's complexity and prevent overfitting. We demonstrate the effectiveness of IMTLS-TSVM by comparing it to several single-task and multi-task learning algorithms on various real-world data sets. Our results highlight the superior performance of IMTLS-TSVM in addressing multi-task learning problems.

Missing Open Access funding information has been added in the Funding Note.

✉ Hossein Moosaei
hmoosaei@gmail.com

Fatemeh Bazikar
f.bazikar@gmail.com; fatemeh_bazikar@phd.guilan.ac.ir

Panos M. Pardalos
pardalos@ise.ufl.edu

1 Jan Evangelista Purkyně University in Ústí nad Labem, Department of Informatics, Faculty of Science, Ústí nad Labem, Czech Republic

2 Department of Mathematics, University of Bojnord, Bojnord, Iran

3 Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran

4 Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA

Published online: 27 July 2023

Keywords Least squares · Multi-task learning · Twin support vector machine · Multi-task twin support vector machine · Quadratic programming problems

Mathematics Subject Classification (2010) 90C20 · 90C25 · 68T10

1 Introduction

Artificial intelligence includes various subfields, one of which is machine learning (ML) which focuses on the development of algorithms and models that can learn patterns and insights from data without being explicitly programmed to do so. It has numerous real-world applications in areas such as letter and number recognition [20, 32], heart disease diagnosis [2], face recognition [34], wireless sensor networks [17], and medical sciences [4, 16, 21, 24, 40]. By leveraging large amounts of data and computational power, machine learning algorithms can make predictions, identify correlations, and automate decision-making processes that humans previously performed.

Multi-task learning (MTL) is a machine learning technique that trains a single model to perform multiple tasks simultaneously. The main aim of MTL is to leverage the shared information among related tasks to improve the overall performance compared to training separate models for each task. This is done by forcing the model to learn task-invariant representations that can be shared across tasks, which reduces the risk of overfitting and enhances generalization. As a result of this approach, numerous multi-task learning methods have been proposed, including multi-task logistic regression [10], multi-task linear discriminant analysis (MT-LDA) [54], multi-task Gaussian process (MTGP) [57], multi-task Bayesian methods [3], among others. MTL has been successful in a variety of applications, such as natural language processing [6], speech recognition [7], drug discovery [39], and computer vision [12].

MT has been explored in various frameworks by different researchers. However, many algorithms are complex and difficult to implement due to their mathematical intricacies, making them challenging for non-experts to understand. As a result, simpler algorithms that are more accessible are in high demand. Support vector machines (SVMs) have been one of the most successful single-task learning classification algorithms [5]. Researchers have also studied multi-task SVMs, with works including [13, 43, 44, 52, 55]. The first approach to multi-task support vector machines (MTL-SVM) is regularized multi-task learning (RMTL) [9]. Subsequently, various MTL-SVM approaches have been proposed, including multi-task one-class support vector machines (MTOC-SVM) [13, 52, 55]. The least squares version of SVM classifiers (LS-SVM) was proposed in [45]. Fifteen years later, Xu et al. introduced multi-task least squares support vector machine (MTLS-SVM) based on LS-SVM [51]. Building on the proximal support vector machine (PSVM) [11], Li et al. proposed the multi-task proximal support vector machine (MTPSVM) [27]. Huang et al. proposed a different approach, using an asymmetric squared loss function called asymmetric least squares SVM (aLS-SVM) [15]. Finally, utilizing the asymmetric squared loss function and multi-task learning, Lu et al. introduced two new methods, MTL-aLS-SVM I and MTL-aLS-SVM II [28]. Many researchers have studied the effectiveness of multi-task learning on support vector machines, as evidenced by several studies [18, 19, 47, 48, 58].

The twin support vector machine (TSVM) for binary classification was suggested by Jeyadeva et al. [22] in 2007 on the basis of the primary concept of generalized eigenvalue proximal SVM (GEPSVM) [29]. Different from SVM's single large QPP, TSVM solves two smaller QPPs to separate the positive and negative samples, resulting in two non-parallel hyperplanes. Because of the amazing performance of TSVMs in single-task learning, they have been discussed in many works, including improvements on twin support vector machines (TBSVM) [42], ν-twin support vector machine (ν-TSVM) [37], twin support vector regression(TSVR) [38], least squares twin support vector machine (LS-TSVM) [25, 26], least squares twin bounded support vector machine (LS-TBSVM) [53], wavelet transform-based weighted ν-twin support vector regression (WTWTSVR) [46], least squares recursive projection twin support vector machine (LSRP-TSVM) [41, 56], and generalized twin support vector machines [33].

Drawing on multi-task learning and TSVM, Xie and Sun suggested a directed multi-task twin support vector machine(DMTSVM) [49], which adopted the mean-regularized method as well. DMTSVM assumes all tasks to have two mean hyperplanes in common. It is different from multi-task SVMs. The DMTSVM algorithm achieves remarkable results when handling correlated tasks. Although the DMTSVM algorithm has its strengths in managing correlated tasks, it has a limitation in that it solely relies on the empirical risk minimization principle. This can lead to over-fitting and, consequently, a decrease in prediction accuracy. To enhance the generalization ability of the classifier, An et al. [1] presented a multi-task twin bounded support vector machine (MT-TBSVM), in which a regularization term is introduced into the objective function. In recent years, other research on multi-task learning and TSVM has been conducted. Based on LS-TSVM, Mei and Xu proposed the multi-task least squares twin support vector machine (MTLS-TSVM) [30], which solves two linear equations instead of two QPP problems. Additionally, to address the problem of outlier samples in each individual task, multi-task centroid twin support vector machines (MCTSVM) [50] were introduced. Moreover, Mei and Xu proposed two innovative multi-task ν-twin support vector machines (MTν-TSVMs) [31] to utilize the regularized multi-task learning and ν-TSVM optimally. Pang et al. [35] proposed a method based on the Hierarchical Bayes theory, named MTTHSVM. For a more accurate assessment of the similarity or difference between tasks, MTTHSVM divides the centers of the hyperspheres into task-specific and task-common regions.

The MTLS-TSVM model is formulated based on the empirical risk minimization principle, which only considers the training error of the two-class samples in its objective function for each task. However, this approach may not result in optimal generalization performance. In this study, we introduce an improved version of MTLS-TSVM, known as IMTLS-TSVM, which is inspired by research on RMTL that uses the empirical risk minimization principle instead of the structural risk minimization principle. Our proposed model considers not only the training errors but also the generalization ability of the classifiers.

To demonstrate the efficacy of IMTLS-TSVM for multi-task learning, we compared it with four single-task learning algorithms (SVM, LS-SVM, TBSVM, LS-TSVM) and two other multi-task learning algorithms (DMTSVM and MTLS-TSVM) using different real-world and medical data sets. The experimental results showed that IMTLS-TSVM outperformed these methods, highlighting its effectiveness in addressing multi-task learning problems.

The structure of the remainder of the paper is as follows: Section 2 provides a concise overview of DMTSVM and MTLS-TSVM. In Section 3, the IMTLS-TSVM model is introduced in detail. Section 4 presents the numerical results, and finally, Section 5 concludes the paper.

2 Preliminaries

In this section, we briefly discuss the direct multi-task twin support vector machine (DMTSVM) and multi-task least squares twin support vector machine (MTLS-TSVM). We assume that the training data set for each task is as follows:

$$\Omega_t = \{(x_1^t, y_1^t), \ldots, (x_{n_t}^t, y_{n_t}^t)\},$$

with $x_i^t \in \mathbb{R}^n$, $y_i^t \in \{\pm 1\}$, $i = 1, \ldots, n_t$, where t indexes task, and $t = 1, \ldots, T$, and also let $n = \sum_{i=1}^{T} n_t$.

We assume that X_p represents the set of positive samples and X_{pt} represents the positive samples in the t-th task. Similarly, X_n represents the set of negative samples and X_{nt} represents the negative samples in the t-th task. This means that the positive class can be represented as $X_p^T = [X_{p1}, X_{p2}, \ldots X_{pT}]$, while the negative class can be represented as $X_n^T = [X_{n1}, X_{n2}, \ldots X_{nT}]$.

Now, we define the following matrices:

$$A = [X_p \;\; e_1],\;\; A_t = [X_{pt} \;\; e_{1t}],\;\; B = [X_n \;\; e_2],\;\; B_t = [X_{nt} \;\; e_{2t}],$$

where each task $t \in 1, \ldots, T$, and e_1, e_2, e_{1t}, and e_{2t} are vectors of ones with appropriate dimensions. We assume that the two mean hyperplanes for all tasks are $u_0 = [w_1, b_1]^T$ and $v_0 = [w_2, b_2]^T$. In the t-th task, the hyperplanes for the positive and negative classes are $(u_0 + u_t) = [w_{1t}, b_{1t}]^T$ and $(v_0 + v_t) = [w_{2t}, b_{2t}]^T$, respectively. The vectors u_t and v_t indicate the bias between task t and the common mean vectors u_0 and v_0, respectively.

2.1 Multi-task twin support vector machine

In 2012, Xie et al. proposed the directed multi-task twin support vector machine (DMTSVM) [49]. They aimed to integrate the concept of regularized multi-task learning (RMTL) [9] into TSVM.

The DMTSVM mathematical programming problems are stated below:

$$\begin{aligned}
\min_{u_0, u_t, q_t} \;& \frac{1}{2}(Au_0)^T(Au_0) + \frac{\lambda_1}{2T}\sum_{t=1}^{T}(A_t u_t)^T(A_t u_t) + c_1 \sum_{t=1}^{T} e_{2t}^T q_t, \\
\text{s.t.} \;& -B_t(u_0 + u_t) + q_t \geq e_{2t}, \\
& q_t \geq 0,
\end{aligned} \tag{1}$$

and

$$\begin{aligned}
\min_{v_0, v_t, p_t} \;& \frac{1}{2}(Bv_0)^T(Bv_0) + \frac{\lambda_2}{2T}\sum_{t=1}^{T}(B_t v_t)^T(B_t v_t) + c_2 \sum_{t=1}^{T} e_{1t}^T p_t, \\
\text{s.t.} \;& A_t(v_0 + v_t) + p_t \geq e_{1t}, \\
& p_t \geq 0,
\end{aligned} \tag{2}$$

where c_1, c_2, λ_1 and λ_2 are positive trade-off parameters. e_{1t} and e_{2t} are one vectors of appropriate dimensions for $t \in \{1, \ldots, T\}$. Both q_t and p_t are slack variables. The parameters λ_1 and λ_2 may be used to change the relationships of all tasks. If λ_1 and λ_2 denote small penalty on vectors u_t and v_t, then u_t and v_t are inclined to be larger. Consequently, the models provide less similarity. When $\lambda_1 \to \infty$ and $\lambda_2 \to \infty$, u_t and v_t tend to be smaller and make the T models similar [30].

By defining

$$Q = B(A^T A)^{-1} B^T, \quad G_t = B_t (A_t^T A_t)^{-1} B_t^T,$$
$$\alpha_1 = [\alpha_{11}^T, \ldots, \alpha_{1T}^T]^T, \quad G = blkdiag(G_1, \ldots, G_T),$$

where the symbol $blkdiag(G_1, \ldots, G_T)$ represents the block-diagonal matrix produced by $G_1, \ldots, G_T$ matrices, the dual problem of the problem (1) is formulated as follows:

$$\begin{aligned} \max_{\alpha_1} \ & -\frac{1}{2}\alpha_1^T (Q + \tfrac{T}{\lambda_1} G)\alpha_1 + e_2^T \alpha_1 \\ \text{s.t.} \ & 0 \le \alpha_1 \le c_1 e_2. \end{aligned} \tag{3}$$

The solution to the aforementioned dual problem may lead us to:

$$u_0 = -(A^T A)^{-1} B^T \alpha_1,$$
$$u_t = -\frac{T}{\mu_1}(A_t^T A_t)^{-1} B_t^T \alpha_{1t}.$$

Likewise, the dual problem of the problem (2) can be derived as follows:

$$\begin{aligned} \max_{\alpha_2} \ & -\frac{1}{2}\alpha_2^T (R + \tfrac{T}{\lambda_2} S)\alpha_2 + e_1^T \alpha_2 \\ \text{s.t.} \ & 0 \le \alpha_2 \le c_2 e_1, \end{aligned} \tag{4}$$

where $\alpha_2 = [\alpha_{21}^T, \ldots, \alpha_{2T}^T]^T$, $R = A(B^T B)^{-1} A^T$, and $S_t = A_t (B_t^T B_t)^{-1} A_t^T$ and $S = blkdiag(S_1, \ldots, S_T)$. Hyperplanes for each task for positive class $[w_{1t}, b_{1t}]^T = (u_0 + u_t)$ and for negative class $[w_{2t}, b_{2t}]^T = (v_0 + v_t)$ may be determined by solving problems (3) and (4). The following decision function is used to assign a new data point x in the t-th task to class $i \in \{+1, -1\}$.

$$Class\ i = \arg\min_{k=1,2} |x^T w_{kt} + b_{kt}|. \tag{5}$$

2.2 Multi-task least squares twin support vector machine

Inspired by the ideas of DMTSVM and least squares twin support vector machine (LS-TSVM) [25], Mei et al. [30] introduced multi-task least squares twin support vector machine, known as MTLS-TSVM.

The following models are the formulation of the MTLS-TSVM problems:

$$\begin{aligned} \min_{u_0, u_t, q_t} \ & \frac{1}{2}(Au_0)^T (Au_0) + \frac{\lambda_1}{2T}\sum_{t=1}^{T}(A_t u_t)^T (A_t u_t) + \frac{c_1}{2}\sum_{t=1}^{T} q_t^T q_t, \\ \text{s.t.} \ & - B_t(u_0 + u_t) + q_t = e_{2t}, \end{aligned} \tag{6}$$

and

$$\begin{aligned} \min_{v_0, v_t, p_t} \ & \frac{1}{2}(Bv_0)^T (Bv_0) + \frac{\lambda_2}{2T}\sum_{t=1}^{T}(B_t v_t)^T (B_t v_t) + \frac{c_2}{2}\sum_{t=1}^{T} p_t^T p_t, \\ \text{s.t.} \ & A_t(v_0 + v_t) + p_t = e_{1t}, \end{aligned} \tag{7}$$

where $\lambda_1, c_1, \lambda_2$ and c_2 are positive parameters. For the mathematical model (6), the corresponding Lagrangian function is given by

$$L_1 = \frac{1}{2}\|Au_0\|^2 + \frac{\lambda_1}{2T}\sum_{t=1}^{T}\| A_t u_t \|^2 + \frac{c_1}{2}\sum_{t=1}^{T}\|q_t \|^2 - \sum_{t=1}^{T}\alpha_{1t}^T\big(- B_t(u_0 + u_t) + q_t - e_{2t}\big). \tag{8}$$

Here, $\alpha_1 = [\alpha_{11}^T, \ldots, \alpha_{1T}^T]^T$ represents the Lagrangian multipliers.

After setting the partial derivatives of the Lagrangian function (8) with respect to u_0, u_t, q_t, and α_{1t} equal to zero, we obtain an expression for α_1 as follows:

$$\alpha_1 = \left(Q + \frac{T}{\lambda_1}G + \frac{1}{c_1}I\right)^{-1} e_2,$$

where $Q = B(A^T A)^{-1}B^T$, $G_t = B_t(A_t^T A_t)^{-1}B_t^T$, and $G = blkdiag(G_1, \ldots, G_T)$.

Next, we will be able to calculate the solution to problem (6):

$$u_0 = -(A^T A)^{-1}B^T\alpha_1, \quad u_t = -\frac{T}{\mu_1}(A_t^T A_t)^{-1}B_t^T\alpha_{1t}.$$

Similarly, the solution to (7) may be found by using the following relations:

$$\alpha_2 = \left(R + \frac{T}{\lambda_2}S + \frac{1}{c_2}I\right)^{-1} e_1,$$

where $R = A(B^T B)^{-1}A^T$, $S_t = A_t(B_t^T B_t)^{-1}A_t^T$, $S = blkdiag(S_1, \ldots, S_T)$ and $\alpha_2 = [\alpha_{21}^T, \ldots, \alpha_{2T}^T]^T$. Therefore, the parameters of classifiers, i.e. u_0, u_t, v_0 and v_t of the t-th task are specified. The decision function, given by formula (5), is used to assign a class label $i \in \{+1, -1\}$ to a new data point x in the t-th task.

3 An improved multi-task least squares twin support vector machine

In this section, we propose an enhanced version of MTLS-TSVM called an improved multi-task least squares twin support vector machine (IMTLS-TSVM). Indeed, our IMTLS-TSVM is an improvement over MTLS-TSVM in that it operates on the empirical risk minimization principle by introducing regularization terms in each task and using quadratic loss functions. Instead of solving quadratic optimization problems, we must solve systems of linear equations as well as MTLS-TSVM. Similar to MTLS-TSVM, the proposed IMTLS-TSVM tries to find nonparallel hyperplanes for each task, i.e., $(u_0 + u_t)$ and $(v_0 + v_t)$.

3.1 Linear IMTLS-TSVM

The linear IMTLS-TSVM can be formulated by the quadratic optimization problems below,

$$\begin{aligned} \min_{u_0,u_t,q_t}\ & \frac{1}{2}(Au_0)^T(Au_0) + \frac{\lambda_1}{2T}\sum_{t=1}^{T}(A_t u_t)^T(A_t u_t) + \frac{c_1}{2}q_t^T q_t + \frac{c_2}{2}u_0^T u_0 + \frac{c_3}{2}\sum_{t=1}^{T}u_t^T u_t \\ \text{s.t.}\ & - B_t(u_0 + u_t) + q_t = e_{2t}, \end{aligned} \tag{9}$$

and

$$\min_{v_0,v_t,p_t} \frac{1}{2}(Bv_0)^T(Bv_0) + \frac{\lambda_2}{2T}\sum_{t=1}^{T}(B_tv_t)^T(B_tv_t) + \frac{c_4}{2}p_t^Tp_t + \frac{c_5}{2}v_0^Tv_0 + \frac{c_6}{2}\sum_{t=1}^{T}v_t^Tv_t$$
$$\text{s.t.}\;\; A_t(v_0+v_t)+p_t = e_{1t}, \tag{10}$$

where c_i for $i = 1, ..., 6$, λ_1 and λ_2 are positive trade-off parameters. In the following, due to the similarities of the optimization problems (9) and (10), we only focus on the problem (9). Now, we introduce the Lagrangian function of problem (9) by

$$L_1(u_0,u_t,q_t,\alpha_1) = \frac{1}{2}\|Au_0\|^2 + \frac{\lambda_1}{2T}\sum_{t=1}^{T}\|A_tu_t\|^2 + \frac{c_1}{2}\|q_t\|^2 + \frac{c_2}{2}\|u_0\|^2 + \frac{c_3}{2}\sum_{t=1}^{T}\|u_t\|^2 - \sum_{t=1}^{T}\alpha_{1t}^T(-B_t(u_0+u_t)+q_t-e_{2t}), \tag{11}$$

where $\alpha_1 = [\alpha_{11}^T, \ldots, \alpha_{1T}^T]^T$ is Lagrangian multiplier. Differentiating L_1 with respect to u_0, u_t, q_t and α_1 yields the following KKT conditions,

$$\frac{\partial L_1}{\partial u_0} = A^TAu_0 + c_2Iu_0 + B^T\alpha_1 = 0, \tag{12}$$
$$\frac{\partial L_1}{\partial u_t} = \frac{\lambda_1}{T}A_t^TA_tu_t + c_3I_tu_t + B_t^T\alpha_{1t} = 0, \tag{13}$$
$$\frac{\partial L_1}{\partial q_t} = c_1q_t - \alpha_{1t} = 0, \tag{14}$$
$$\frac{\partial L_1}{\partial \alpha_{1t}} = B_t(u_0+u_t) - q_t + e_{2t} = 0. \tag{15}$$

Equations (12), (13), and (14) yield the following results:

$$u_0 = -(A^TA + c_2I)^{-1}B^T\alpha_1, \tag{16}$$
$$u_t = -\frac{T}{\lambda_1}(A_t^TA_t + c_3I_t)^{-1}B_t^T\alpha_{1t}, \tag{17}$$
$$q_t = \frac{\alpha_{1t}}{c_1}. \tag{18}$$

Substituting the equations above into (15) yields:

$$B_t\left[-(A^TA+c_2I)^{-1}B^T\alpha_1 - \frac{T}{\lambda_1}(A_t^TA_t + c_3I_t)^{-1}B_t^T\alpha_{1t}\right] - \frac{\alpha_{1t}}{c_1} = -e_{2t},\;\; t = 1, ..., T. \tag{19}$$

By defining $Q = B(A^TA + c_2I)^{-1}B^T$, $G_t = B_t(A_t^TA_t + c_3I_t)^{-1}B_t^T$ and $G = blkdiag(G_1, \ldots, G_T)$, and identical matrix I, the (19) can be rewritten as follows:

$$Q\alpha_1 + \frac{T}{\lambda_1}G\alpha_1 + \frac{1}{c_1}I\alpha_1 = e_2. \tag{20}$$

As a results, $\alpha_1 = (Q + \frac{T}{\lambda_1}G + \frac{1}{c_1}I)^{-1}e_2$ can be obtained. The classifier parameters u_0 and u_t of the $t-$th task can be determined by putting the obtained α_1 into the (16) and (17).

Also, the Lagrangian function of the problem (10) may be expressed as:

$$L_2(v_0, v_t, p_t, \alpha_2) = \frac{1}{2}\|Bv_0\|^2 + \frac{\lambda_2}{2T}\sum_{t=1}^{T}\|B_t v_t\|^2 + \frac{c_4}{2}\|p_t\|^2 + \frac{c_5}{2}\|v_0\|^2 + \frac{c_6}{2}\sum_{t=1}^{T}\|v_t\|^2 - \sum_{t=1}^{T}\alpha_{2t}^T(A_t(v_0 + v_t) + p_t - e_{1t}), \tag{21}$$

where $\alpha_2 = [\alpha_{21}^T, \ldots, \alpha_{2T}^T]^T$ is the Lagrangian multiplier. By performing a similar process, the following equation is achieved:

$$R\alpha_2 + \frac{T}{\lambda_2}S\alpha_2 + \frac{1}{c_4}I\alpha_2 = e_1, \tag{22}$$

where $R = A(B^T B + c_5 I)^{-1} A^T$, $S_t = A_t(B_t^T B_t + c_6 I_t)^{-1} A_t^T$ and $S = blkdiag(S_1, \ldots, S_T)$. Then, we can obtain α_2 as follow:

$$\alpha_2 = (R + \frac{T}{\lambda_2}S + \frac{1}{c_4}I)^{-1}e_1. \tag{23}$$

Therefore, by using α_1 and α_2, the positive and negative hyperplanes can be determined. The label of a new sample $x \in \mathbb{R}^n$ in the $t-$th task can be determined using the decision function (5). As a result of the preceding discussion, we present the linear IMTLS-TSVM algorithm in Algorithm 1.

Algorithm 1 Linear IMTLS-TSVM

Input:
- The training set;
- Determine the total number of tasks in the data collection and give this amount to T;
- Select a classification task Ω_t from the training data set, where $t = 1, \ldots, T$;
- Select suitable parameters $c_1, c_2, c_3, c_4, c_5, c_6, \lambda_1$ and λ_2.

Outputs:
- $u_0,\ u_t,\ v_0$, and v_t.

Process:

1: Get α_1 and α_2 by solving the two systems of linear equations given in (20) and (23).
2: Determine the values of $u_0,\ u_t,\ v_0$, and v_t.
3: Apply the decision function (5), classify a new point x as belonging to either class $+1$ or -1 in the t-th task.

3.2 Nonlinear IMTLS-TSVM

The goal of this subsection is to expand the linear IMTLS-TSVM method to the nonlinear case by applying the kernel trick. By utilizing a nonlinear kernel function, denoted as $K(.,.)$, the input data is mapped into a higher-dimensional feature space. By utilizing this kernel function and defining

$$C = \left[\mathfrak{A}_1^T, \mathfrak{B}_1^T, \mathfrak{A}_2^T, \mathfrak{B}_2^T, \ldots, \mathfrak{A}_T^T, \mathfrak{B}_T^T\right]^T,$$
$$\mathfrak{A} = \left[K(\mathfrak{A}, C^T), e_1\right], \quad \mathfrak{A}_t = \left[K(\mathfrak{A}_t, C^T), e_{1t}\right],$$
$$\mathfrak{B} = \left[K(\mathfrak{B}, C^T), e_2\right], \quad \mathfrak{B}_t = \left[K(\mathfrak{B}_t, C^T), e_{2t}\right],$$

the nonlinear version of the optimization problems (9) and (10) can be formulated as follows:

$$\min_{u_0,u_t,q_t} \frac{1}{2}(\mathfrak{A}u_0)^T(\mathfrak{A}u_0) + \frac{\lambda_1}{2T}\sum_{t=1}^{T}(\mathfrak{A}_t u_t)^T(\mathfrak{A}_t u_t) + \frac{c_1}{2}q_t^T q_t + \frac{c_2}{2}u_0^T u_0 + \frac{c_3}{2}\sum_{t=1}^{T} u_t^T u_t$$
$$\text{s.t.} \quad -\mathfrak{B}_t(u_0 + u_t) + q_t = e_{2t}, \tag{24}$$

and

$$\min_{v_0,v_t,p_t} \frac{1}{2}(\mathfrak{B}v_0)^T(\mathfrak{B}v_0) + \frac{\lambda_2}{2T}\sum_{t=1}^{T}(\mathfrak{B}_t v_t)^T(\mathfrak{B}_t v_t) + \frac{c_4}{2}p_t^T p_t + \frac{c_5}{2}v_0^T v_0 + \frac{c_6}{2}\sum_{t=1}^{T} v_t^T v_t$$
$$\text{s.t.} \quad \mathfrak{A}_t(v_0 + v_t) + p_t = e_{1t}, \tag{25}$$

where c_i for $i = 1, ..., 6$, λ_1 and λ_2 are positive trade-off parameters. Similar to the linear case, we can write the Lagrangian functions of the problems (24) and (25) and apply the KKT optimality conditions. The Lagrangian function of the problem (24) can be shown as follows:

$$L_1(u_0, u_t, q_t, \alpha_1) = \frac{1}{2}\|\mathfrak{A}u_0\|^2 + \frac{\lambda_1}{2T}\sum_{t=1}^{T}\|\mathfrak{A}_t u_t\|^2 + \frac{c_1}{2}\|q_t\|^2 + \frac{c_2}{2}\|u_0\|^2 + \frac{c_3}{2}\sum_{t=1}^{T}\|u_t\|^2$$
$$- \sum_{t=1}^{T}\alpha_{1t}^T(-\mathfrak{B}_t(u_0 + u_t) + q_t - e_{2t}), \tag{26}$$

where α_1 is Lagrangian multiplier that yields the gradients of (26) based on u_0, u_t, q_t, and α_1 and sets them to zero. Hence, the KKT conditions are as follows:

$$\frac{\partial L_1}{\partial u_0} = \mathfrak{A}^T\mathfrak{A}u_0 + c_2 I u_0 + \mathfrak{B}^T\alpha_1 = 0, \tag{27}$$
$$\frac{\partial L_1}{\partial u_t} = \frac{\lambda_1}{T}\mathfrak{A}_t^T\mathfrak{A}_t u_t + c_3 I_t u_t + \mathfrak{B}_t^T\alpha_{1t} = 0, \tag{28}$$
$$\frac{\partial L_1}{\partial q_t} = c_1 q_t - \alpha_{1t} = 0, \tag{29}$$
$$\frac{\partial L_1}{\partial \alpha_{1t}} = \mathfrak{B}_t(u_0 + u_t) - q_t + e_{2t} = 0. \tag{30}$$

Next, we have

$$u_0 = -(\mathfrak{A}^T\mathfrak{A} + c_2 I)^{-1}\mathfrak{B}^T\alpha_1, \tag{31}$$
$$u_t = -\frac{T}{\lambda_1}(\mathfrak{A}_t^T\mathfrak{A}_t + c_3 I_t)^{-1}\mathfrak{B}_t^T\alpha_{1t}, \tag{32}$$
$$q_t = \frac{\alpha_{1t}}{c_1}. \tag{33}$$

By replacing u_0, u_t, and q_t into (30), we have

$$\mathfrak{B}_t\left[-(\mathfrak{A}^T\mathfrak{A}+c_2I)^{-1}\mathfrak{B}^T\alpha_1-\frac{T}{\lambda_1}(\mathfrak{A}_t^T\mathfrak{A}_t+c_3I_t)^{-1}\mathfrak{B}_t^T\alpha_{1t}\right]-\frac{\alpha_{1t}}{c_1}=-e_{2t},\quad t=1,...,T. \tag{34}$$

By defining $Q=\mathfrak{B}(\mathfrak{A}^T\mathfrak{A}+c_2I)^{-1}\mathfrak{B}^T$, $G_t=\mathfrak{B}_t(\mathfrak{A}_t^T\mathfrak{A}_t+c_3I_t)^{-1}\mathfrak{B}_t^T$, $G=blkdiag(G_1,\ldots,G_T)$, and identical matrix I, the (34) can be rewritten as follows:

$$Q\alpha_1+\frac{T}{\lambda_1}G\alpha_1+\frac{1}{c_1}I\alpha_1=e_2, \tag{35}$$

Therefore, the optimal solution to problems (24)is taken from

$$\alpha_1=(Q+\frac{T}{\lambda_1}G+\frac{1}{c_1}I)^{-1}e_2. \tag{36}$$

Likewise, the optimal solution to problem (25) can be obtained by:

$$\alpha_2=(R+\frac{T}{\lambda_2}S+\frac{1}{c_4}I)^{-1}e_1, \tag{37}$$

where $R=\mathfrak{A}(\mathfrak{B}^T\mathfrak{B}+c_5I)^{-1}\mathfrak{A}^T$, $S_t=\mathfrak{A}_t(\mathfrak{B}_t^T\mathfrak{B}_t+c_6I_t)^{-1}\mathfrak{A}_t^T$, and $S=blkdiag(S_1,\ldots,S_T)$.

Then, the decision function for the t-th task may be calculated using nonlinear version of (5). The nonlinear procedure is outlined in Algorithm 2.

Algorithm 2 Nonlinear IMTLS-TSVM

Input:

– The training set;

– Determine the total number of tasks in the data collection and give this amount to T;

– Select a classification task Ω_t from the training data set, where $t=1,\ldots,T$;

– Choose appropriate parameters $c_1, c_2, c_3, c_4, c_5, c_6, \lambda_1$ and λ_2.

– Choose a user-defined kernel function and its corresponding kernel parameter.

Outputs:

$-u_0,\ u_t,\ v_0$, and v_t.

Process:

1: Get α_1 and α_2 by solving the two systems of linear equations given in (36) and (37).

2: Determine the values of $u_0,\ u_t,\ v_0$, and v_t.

3: Apply the nonlinear version of the decision function (5), and classify a new point x as belonging to either class $+1$ or -1 in the t-th task.

4 Numerical experiments

To evaluate the effectiveness of our proposed IMTLS-TSVM algorithm, we compared its performance with six single-task and multi-task learning algorithms, including SVM, LS-SVM, TBSVM, LS-TSVM, DMTSVM, and MTLS-TSVM. The experiments were conducted on seven data sets: Monk, Heart, Immunotherapy, Caesarean, Breast Cancer Coimbra, Ljubljana Breast Cancer, and Landmine. These data sets varied in the number of instances, features, and tasks, as shown in Table 1. We divided each data set into a number of tasks based on the task

Table 1 The description of data sets

Data set	# Instances	# Features	# Tasks
Monk	432	6	3
Heart	270	13	2
Immunotherapy	90	8	3
Caesarean	80	5	2
Breast Cancer Coimbra	116	9	3
Ljubljana Breast Cancer	286	9	5
Landmine	9674	9	4

variable before multi-task learning. We used five-fold cross-validation to assess the accuracy of the classification and performance of the algorithms. The performance was measured by averaging the results of five iterations, where each iteration involved using one subset as a test set and the rest as training data. All numerical experiments were carried out on a personal computer with Matlab 2018b, an Intel Core(TM) i7 CPU @2.20 GHz, 4 GB of memory, and Microsoft Windows 7(64-bit).

4.1 Parameters selection

Various machine learning algorithms have tuning parameters that need to be optimized for the best performance. For instance, SVM and LS-SVM have c_1, TBSVM and LS-TSVM have c_1 and c_2, and DMTSVM and MTLS-TSVM have c_1, c_2, λ_1 and λ_2 as tuning parameters. The proposed IMTLS-TSVM algorithm also has tuning parameters, namely c_1, c_2, c_3, c_4, c_5, c_6, λ_1 and λ_2.

Here, we aim to investigate the impact of changing parameters such as c_1, c_2, c_3, λ_1 and λ_2 on the behavior of IMTLS-TSVM using the Caesarian data set in a linear state. All other parameters are kept constant during the experiment. From Fig. 1 ((*a*) and (*b*)), we can observe that different parameter values have a noticeable impact on the accuracy of the model.

Thus, selecting the appropriate parameter values is crucial for achieving optimal performance. Grid search is a widely used method for tuning the parameters of machine learning algorithms. It involves evaluating a model's performance for different combinations of hyper-

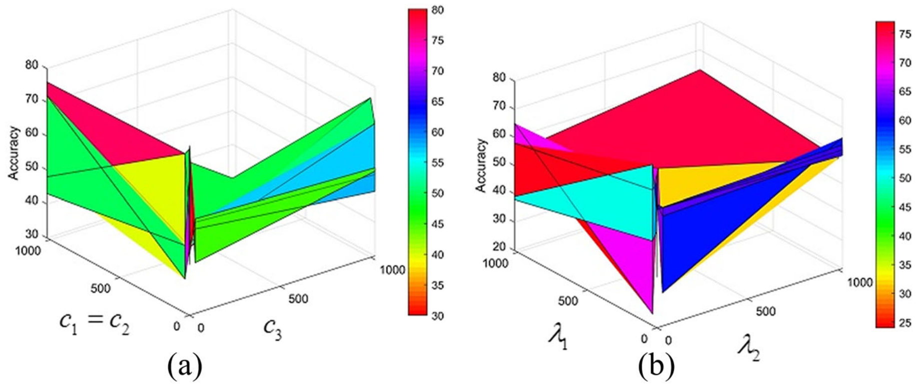

Fig. 1 The effect of different parameter values on the Caesarian data set

Table 2 The linear performance of classifiers on Monk data set

Classifier	Accuracy	Standard deviation	Time
SVM	67.30	±0.05	47.92
LS-SVM	67.48	±0.02	0.4
TBSVM	67.65	±0.02	2.17
LS-TSVM	69.10	±0.04	**0.06**
DMTSVM	71.67	±0.03	2.83
MTLS-TSVM	72.45	±0.04	2.01
IMTLS-TSVM	**73.56**	±0.05	1.99

parameters to identify the optimal set of parameters that result in the best performance [14, 21]. For our experiments, we utilized grid search to identify the optimal parameter values.

We tuned the parameters $c_1, c_2, c_3, c_4, c_5, c_6, \lambda_1$ and λ_2 from the set $\{2^i \mid i = -10, \ldots, 10\}$. In our experiments, we used the Gaussian kernel function, i.e., $K(x, y) = \exp(-\gamma|x - y|^2)$, where $\gamma > 0$. This kernel function provides better results for inseparable data sets. To set the kernel parameter γ, we selected the optimal value from the range $\{2^i \mid i = -10, \ldots, 10\}$.

4.2 Results and discussion on data sets

In this subsection, we compare the single-task and multi-task learning algorithms mentioned earlier with the proposed IMTLS-TSVM method in both linear and nonlinear cases.

4.2.1 The results of the monk data set

The Monk data set is renowned for being the first international comparison of learning algorithms. In July 1991, the 2nd European Summer School on Machine Learning presented a challenge to the monks of Corsendonk Priory, who had spent a week studying various learning algorithms but were still unsure which one to choose. They decided to create a simple task that could be used to compare all the algorithms, resulting in the three Monk's problems. The data set contains 432 records with six features, classified into two classes. In this study, we evaluated the performance of our proposed method along with six other methods on the Monk data set. In Tables 2 and 3, the best values for each performance measure are bolded to indicate the best results achieved by each method. Our results indicate that our method performs well in both linear and nonlinear cases, achieving satisfactory accuracy.

Table 3 The nonlinear performance of classifiers on Monk data set

Classifier	Accuracy	Standard deviation	Time
SVM	79.15	±0.02	287.55
LS-SVM	79.52	±0.22	3.36
TBSVM	79.65	±0.01	4.98
LS-TSVM	80.22	±0.03	**3.12**
DMTSVM	92.76	±0.02	27.85
MTLS-TSVM	94.80	±0.01	25.46
IMTLS-TSVM	**95.17**	±0.03	24.11

Table 4 The linear performance of classifiers on Heart data set

Classifier	Accuracy	Standard deviation	Time
SVM	84	±0.02	1.94
LS-SVM	85.18	±0.03	0.06
TBSVM	84.07	±0.04	1.40
LS-TSVM	84.87	±0.00	**0.04**
DMTSVM	86.42	±0.09	1.48
MTLS-TSVM	86.43	±0.02	0.19
IMTLS-TSVM	**87.79**	±0.03	0.17

4.2.2 The results of the heart data set

Nowadays, one of the most prevalent illnesses is heart disease (HD), and early detection of the disease is a crucial responsibility for many medical professionals who aim to protect their patients from HD and save their lives. The Heart data set contains 270 instances, each with 13 features. The data set aims to predict the presence or absence of heart disease using these 13 features, which include age, sex, resting blood pressure, chest pain type, fasting blood sugar levels, serum cholesterol levels, maximum heart rate achieved, resting electrocardiographic results, exercise-induced angina, old peak (ST depression induced by exercise proportionate to rest), the number of major vessels, Thalassemia, and the slope of the peak exercise ST segment. We used the ”sex” feature to categorize the data into male and female groups. In this part, we compare the efficiency of our proposed method with six other algorithms on the Heart data set, as reported in Tables 4 and 5. The tables display the best performance values (Accuracy) obtained by the IMTLS-TSVM algorithm in both linear and nonlinear cases.

4.2.3 The results of the immunotherapy data set

The Immunotherapy data set is publicly available in the UCI database [8]. This data set comprises 90 patients from the dermatology clinic at Ghaem Hospital in Mashhad, Iran [23]. It contains eight features such as sex, age, types and the number of warts, induration diameter, area, and the treatment outcome. The first seven features provide patient details, and the last feature indicates the response to treatment. A ”YES” response indicates a reduction in the size of the largest wart by more than 75%, while a ”NO” response indicates otherwise. In this study, the data was divided into three tasks based on the wart type: task 1 (type ”1” = Common, 47 instances), task 2 (type ”2” = Plantar, 22 instances), and task 3 (type ”3” = Both,

Table 5 The nonlinear performance of classifiers on Heart data set

Classifier	Accuracy	Standard deviation	Time
SVM	70	±0.09	2.76
LS-SVM	71.11	±0.08	**0.09**
TBSVM	72.33	±0.22	1.86
LS-TSVM	73.33	±0.05	1.26
DMTSVM	71.89	±0.21	1.68
MTLS-TSVM	70.18	±0.01	0.51
IMTLS-TSVM	**73.49**	±0.02	0.49

Table 6 The linear performance of classifiers on Immunotherapy data set

Classifier	Accuracy	Standard deviation	Time
SVM	79.97	±0.01	1.62
LS-SVM	80.04	±0.02	0.05
TBSVM	77.81	±0.00	1.63
LS-TSVM	80.22	±0.10	**0.04**
DMTSVM	84.63	±0.00	1.49
MTLS-TSVM	86.11	±0.02	0.22
IMTLS-TSVM	**87.37**	±0.01	0.19

21 instances). The wart type was also included in the model. The results presented in Tables 6 and 7 show that the proposed method outperforms other methods on the Immunotherapy data set. The IMTLS-TSVM classifier achieved 87.37% and 82.48% accuracy and 0.19s and 0.24s running time for linear and nonlinear cases, respectively. In comparison, MTLS-TSVM achieved 86.11% and 80.22% accuracy for linear and nonlinear cases, respectively, which is lower than the accuracy achieved by the proposed method.

4.2.4 The results of the caesarean data set

Nowadays, despite the various complications, Caesarean delivery is preferred over natural birth, and this trend is growing fast throughout the world. Therefore, a Caesarean should only be carried out if it is deemed necessary for the mother and fetus. To avoid performing Caesarean delivery unnecessarily, researchers have designed various machine learning-based clinical decision support systems to predict Caesarean delivery using electronic health records of pregnant women and collected the Caesarean data set. This data collection contains information on whether the delivery is intended to be by cesarean section or natural birth and is derived from 80 pregnant women who claimed to have experienced the most severe delivery difficulties in the medical profession. The Caesarean data set consists of 80 samples, each with five characteristics: age, delivery number, blood pressure, delivery time, and the presence or absence of a heart problem. We use the feature ”heart problem” to divide the data into two distinct sections: ”task 1” indicates that the patient suffers from a heart problem, whereas ”task 2” indicates that there is no heart problem (Tables 8 and 9).

The calculations on the Caesarean data set show that our proposed method has the best accuracy among seven classifiers in the nonlinear case, with an accuracy of 78.30% and

Table 7 The nonlinear performance of classifiers on Immunotherapy data set

Classifier	Accuracy	Standard deviation	Time
SVM	78.27	±0.01	1.60
LS-SVM	78.92	±0.01	0.6
TBSVM	78.88	±0.02	1.50
LS-TSVM	78.88	±0.21	1.01
DMTSVM	78.88	±0.79	1.58
MTLS-TSVM	80.22	±0.00	0.27
IMTLS-TSVM	**82.48**	±0.01	**0.24**

Table 8 The linear performance of classifiers on Caesarean data set

Classifier	Accuracy	Standard deviation	Time
SVM	63.70	±0.05	1.94
LS-SVM	68.78	±0.07	0.04
TBSVM	69.09	±0.00	1.42
LS-TSVM	69.88	±0.21	**0.03**
DMTSVM	63.86	±0.01	1.48
MTLS-TSVM	**76.95**	±0.03	0.16
IMTLS-TSVM	**76.95**	±0.00	0.16

an acceptable speed of learning. Additionally, in the linear case, our proposed method and MTLS-TSVM achieved the same accuracy with an acceptable speed of learning

4.2.5 The results of the breast cancer coimbra data set

Breast cancer is known to be the most common invasive cancer in women and is also the second leading cause of cancer-related deaths among them. A variety of biological factors influence the recurrence of breast cancer. In recent years, research and prevention of breast cancer have been a major focus among researchers. Data mining methods have emerged as effective strategies to extract useful information from databases for classification purposes. The Breast Cancer Coimbra data set consists of 116 instances with ten attributes for each case, as reported on March 6th, 2018 [36]. These attributes include age, BMI, insulin, glucose, HOMA, leptin, resistin, adiponectin, and MCP-1, all of which are quantitative characteristics obtained from anthropometric data and regular blood analysis. These characteristics are likely to be used as biomarkers for breast cancer. The data set is divided into three tasks based on body mass index analysis. Body mass index (BMI) is a basic rule of thumb that classifies a person as underweight (less than 18.5 kg/m2), normal weight (18.5 kg/m2 to 24.9 kg/m2), overweight (25 kg/m2 to 29.9 kg/m2), or obese (30 kg/m2 or more) based on their tissue mass (muscle, fat, and bone) and height. The first task is for underweight individuals, the second for those who are normal weight, and the third for people who are overweight or obese.

The results presented in Tables 10 and 11 demonstrate that the accuracy of IMTLS-TSVM is superior to that of six other methods. The best results are highlighted in bold within the tables.

Table 9 The nonlinear performance of classifiers on Caesarean data set

Classifier	Accuracy	Standard deviation	Time
SVM	72.42	±0.14	1.58
LS-SVM	73	±0.12	**0.06**
TBSVM	71.35	±0.09	1.47
LS-TSVM	72	±0.07	1.08
DMTSVM	72.56	±0.13	1.56
MTLS-TSVM	73.15	±0.01	0.24
IMTLS-TSVM	**78.30**	±0.00	0.21

Table 10 The linear performance of classifiers on Breast Cancer Coimbra data set

Classifier	Accuracy	Standard deviation	Time
SVM	72.40	±0.11	1.65
LS-SVM	74.89	±0.09	0.05
TBSVM	72.36	±0.00	1.40
LS-TSVM	78.31	±0.03	**0.03**
DMTSVM	82.24	±0.01	1.50
MTLS-TSVM	83.39	±0.01	0.19
IMTLS-TSVM	**84.63**	±0.02	0.19

4.2.6 The results of the ljubljana breast cancer data set

The Ljubljana Breast Cancer data set was collected in July 1988 from 286 patients at the University Medical Center, Institute of Oncology, Ljubljana, Yugoslavia [59]. All patients had undergone surgery to remove the cancer-affected tissue. The data set contains nine predictor variables and one response variable that indicates whether a patient had any recurrence event(s) within five years of undergoing the operation. Nine incomplete entries, each of which missed a single predictor value, were removed, resulting in a final data set of 277 complete instances. The data set's nine attributes include age, PostMeno (whether the patient is pre- or post-menopausal at the time of diagnosis), Tumor Size (the greatest diameter (mm) of the removed tumor), Inv Nodes (the number of axillary lymph nodes with visible metastatic breast cancer at the time of diagnosis), Node Caps (whether cancer metastasized to a lymph node or not), Deg-Maligi (histological grade (range 1-3) of the removed tumor), Breast (the left or right breast where the tumor occurred), Quadrant (the location of the tumor within the breast (upper left, upper right, central, lower left, or lower right)), and Radiation (whether the patient underwent radiation therapy or not). Using the tumor size variable, the data was divided into five distinct tasks: task 1 ($0 \leq tumor\ size \leq 19$), task 2 ($20 \leq tumor\ size \leq 24$), task 3 ($25 \leq tumor\ size \leq 29$), task 4 ($30 \leq tumor\ size \leq 34$), and task 5 ($35 \leq tumor\ size \leq 54$). The results of the proposed method in this study were compared to those of six other methods on this data set, as shown in Tables 12 and 13. The tables demonstrate that IMTLS-TSVM achieved higher accuracy than the other methods.

Table 11 The nonlinear performance of classifiers on Breast Cancer Coimbra data set

Classifier	Accuracy	Standard deviation	Time
SVM	60.26	±0.11	1.66
LS-SVM	63.74	±0.08	**0.09**
TBSVM	60.39	±0.07	1.49
LS-TSVM	63.33	±0.01	1.20
DMTSVM	63.74	±0.09	1.67
MTLS-TSVM	65.33	±0.01	0.35
IMTLS-TSVM	**69.98**	±0.02	0.30

Table 12 The linear performance of classifiers on Ljubljana Breast Cancer data set

Classifier	Accuracy	Standard deviation	Time
SVM	72.94	±0.03	2.24
LS-SVM	74.06	±0.01	0.05
TBSVM	75.13	±0.00	1.49
LS-TSVM	74.01	±0.00	**0.03**
DMTSVM	73.26	±0.01	1.60
MTLS-TSVM	75.09	±0.01	0.28
IMTLS-TSVM	**75.36**	±0.02	0.25

4.2.7 The results of the landmine data set

The Landmine data set is used to detect the presence of landmines in an area based on radar images. This data set is a binary classification problem with samples labeled as either 1 for landmines or 0 for clusters, indicating positive and negative classifications, respectively. The data is collected from 29 landmine fields, each of which corresponds to a separate geographical region, resulting in a total of 9674 samples. The data set consists of nine features and the total number of nodes is 29, with the first 15 areas being foliated regions and the latter 14 belonging to bare ground or uninhabited places.

We followed specific procedures for the Landmine data set to ensure fair and better experimental outcomes. Since the number of negative samples is higher than the positive samples, we balanced the data set by removing some negative samples. We divided the data set into four tasks, selecting four densely foliated regions as positive data and four areas from bare ground or desert regions as negative data. Specifically, we used sites 1, 2, 3, and 4 from foliated regions and identified areas 16, 17, 18, and 19 from bare earth regions to construct our experimental data set.

We evaluated our proposed IMTLS-TSVM algorithm on this data set and compared its performance with other single-task and multi-task methods. As shown in Tables 14 and 15, our method outperformed the other methods in terms of accuracy on this large-scale data set. Furthermore, the IMTLS-TSVM algorithm showed acceptable prediction time when dealing with the Landmine data set.

Table 13 The nonlinear performance of classifiers on Ljubljana Breast Cancer data set

Classifier	Accuracy	Standard deviation	Time
SVM	74	±0.04	5.98
LS-SVM	74.73	±0.02	**0.1**
TBSVM	72.54	±0.05	1.49
LS-TSVM	73.72	±0.32	1.21
DMTSVM	73.71	±0.09	2.41
MTLS-TSVM	75.32	±0.00	0.75
IMTLS-TSVM	**75.63**	±0.05	0.71

Table 14 The linear performance of classifiers on Landmine data set

Classifier	Accuracy	Standard deviation	Time
SVM	94.04	±0.01	117.13
LS-SVM	94	±0.02	0.33
TBSVM	95.51	±0.03	5.94
LS-TSVM	96.52	±0.01	**0.14**
DMTSVM	96.80	±0.02	4.99
MTLS-TSVM	97.05	±0.01	1.21
IMTLS-TSVM	**97.36**	±0.02	1.18

4.2.8 Comparison of experiments results

The results of our experiments demonstrate the superiority of our proposed IMTLS-TSVM algorithm over six other popular algorithms (SVM, LS-SVM, TBSVM, LS-TSVM, DMTSVM, and MTLS-TSVM) in handling multi-task learning scenarios. The comparisons were performed on seven multi-task data sets, and Tables 2 to 15 show that IMTLS-TSVM consistently outperformed all other algorithms in linear and nonlinear situations.

Furthermore, our experiments demonstrate that the proposed IMTLS-TSVM algorithm achieves comparable or superior prediction accuracy to other multi-task algorithms, including DMTSVM and MTLS-TSVM. Additionally, in terms of computation time, IMTLS-TSVM exhibits similar performance to MTLS-TSVM while being significantly faster than DMTSVM. These findings suggest that IMTLS-TSVM is a promising algorithm for solving multi-task learning problems, particularly in situations where computational efficiency is a critical factor.

5 Conclusion

In this study, we proposed the IMTLS-TSVM algorithm, an improved version of the MTLS-TSVM algorithm for multi-task learning. Our approach incorporates empirical risk minimization and achieved superior performance compared to several other single-task and multi-task learning algorithms on seven traditional multi-task data sets. The results demonstrate the effectiveness and feasibility of the IMTLS-TSVM algorithm for addressing multi-task learning problems.

Table 15 The nonlinear performance of classifiers on Landmine data set

Classifier	Accuracy	Standard deviation	Time
SVM	94	±0.08	312.81
LS-SVM	94.09	±0.01	9.30
TBSVM	94.10	±0.02	11.53
LS-TSVM	94.38	±0.4	**6.26**
DMTSVM	94.40	±0.01	46.60
MTLS-TSVM	94.45	±0.05	36.69
IMTLS-TSVM	**96.76**	±0.01	33.60

However, further research is needed to explore the scalability and robustness of the IMTLS-TSVM algorithm on larger and more complex multi-task data sets. Investigating its applicability to other machine learning tasks beyond classification, such as regression and clustering, could broaden its scope and enable its use in a wider range of applications. Additionally, exploring the incorporation of other regularization methods into the IMTLS-TSVM framework could enhance its performance even more.

In conclusion, the IMTLS-TSVM algorithm shows promising potential for improving the accuracy and efficiency of multi-task learning. Future research could further enhance its capabilities and expand its applicability.

Funding Open access publishing supported by the National Technical Library in Prague.

Data availability statement The data that support the findings of this study are available from the UCI machine learning repository, associated with the following link: https://archive.ics.uci.edu/ml/index.php.

Declarations

Conflicts of interest The authors state that they do not have any conflicts of interest.

References

1. An, R., Xu, Y., Liu, X.: Multi-task twin bounded support vector machine and its safe screening rule. Appl. Soft Comput. **138**, 110–188 (2023)
2. Arabasadi, Z., Alizadehsani, R., Roshanzamir, M., Moosaei, H., Yarifard, A.A.: Computer aided decision making for heart disease detection using hybrid neural network-Genetic algorithm. Comput. Methods Prog. Biomed. **141**, 19–26 (2017)
3. Bakker, B., Heskes, T.: Task clustering and gating for bayesian multitask learning. J. Mach. Learn. Res. **4**, 83–99 (2003)
4. Bazikar, F., Ketabchi, S., Moosaei, H.: DC programming and DCA for parametric-margin ν-support vector machine. Appl. Intell. **50**(6), 1763–1774 (2020)
5. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Disc. **2**(2), 121–167 (1998)
6. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning, pp. 160–167 (2008)
7. Deng, L., Hinton, G., Kingsbury, B.: New types of deep neural network learning for speech recognition and related applications: An overview. In: 2013 IEEE International Conference On Acoustics, Speech And Signal Processing, pp. 8599–8603. IEEE (2013)
8. Dua, D., Graff, C.: Uci machine learning repository. University of California, School Of Information And Computer Science, Irvine, Ca (2019)
9. Evgeniou, T., Pontil, M.: Regularized multi–task learning. In: Proceedings of the tenth ACM SIGKDD international conference on Knowl. Discov Data Min. pp. 109–117 (2004)
10. Fiebig, K.H., Jayaram, V., Peters, J., Grosse-Wentrup, M.: Multi-task logistic regression in brain-computer interfaces. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 002,307–002,312. IEEE (2016)

11. Fung, G., Mangasarian, O.L.: Proximal support vector machine classifiers. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 77–86 (2001)
12. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference On Computer Vision, pp. 1440–1448 (2015)
13. He, X., Mourot, G., Maquin, D., Ragot, J., Beauseroy, P., Smolarz, A., Grall-Maës, E.: Multi-task learning with one-class svm. Neurocomputing **133**, 416–426 (2014)
14. Hsu, C.W., Chang, C.C., Lin, C.J., et al.: A practical guide to support vector classification (2003). https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf
15. Huang, X., Shi, L., Suykens, J.A.: Asymmetric least squares support vector machine classifiers. Comput. Stat. Data Anal. **70**, 395–405 (2014)
16. Ivanciuc, O.: Reviews in Computational Chemistry. Wiley (2007)
17. Javadi, S.H., Moosaei, H., Ciuonzo, D.: Learning wireless sensor networks for source localization. Sensors **19**(3), 635 (2019)
18. Ji, Y., Sun, S.: Multitask multiclass support vector machines: model and experiments. Pattern Recognit. **46**(3), 914–924 (2013)
19. Jia, X., Wang, S., Yang, Y.: Least-squares support vector machine for semi-supervised multi-tasking. In: 2018 IEEE 16th International Conference on Software Engineering Research, Management and Applications (SERA), pp. 79–86. IEEE (2018)
20. Joachims, T.: Text categorization with suport vector machines: Learning with many relevant features. In: Proceedings of the 10th European Conference on Machine Learning, ECML '98, pp. 137–142. Springer, London, UK (1998)
21. Ketabchi, S., Moosaei, H., Razzaghi, M., Pardalos, P.M.: An improvement on parametric ν-support vector algorithm for classification. Ann. Oper. Res. **276**(1), 155–168 (2019)
22. Khemchandani, R., Chandra, S., et al.: Twin support vector machines for pattern classification. IEEE Trans. Pattern Anal. Mach. Intell. **29**(5), 905–910 (2007)
23. Khozeimeh, F., Alizadehsani, R., Roshanzamir, M., Khosravi, A., Layegh, P., Nahavandi, S.: An expert system for selecting wart treatment method. Comput. Biol. Med. **81**, 167–175 (2017)
24. Khozeimeh, F., Sharifrazi, D., Izadi, N.H., Joloudari, J.H., Shoeibi, A., Alizadehsani, R., Gorriz, J.M., Hussain, S., Sani, Z.A., Moosaei, H., et al.: Combining a convolutional neural network with autoencoders to predict the survival chance of COVID-19 patients. Sci. Rep. **11**(1), 1–18 (2021)
25. Kumar, M.A., Gopal, M.: Least squares twin support vector machines for pattern classification. Expert Syst. Appl. **36**(4), 7535–7543 (2009)
26. Kumar, M.A., Gopal, M.: Least squares twin support vector machines for text categorization. In: 2015 39th National Systems Conference (NSC), pp. 1–5. IEEE (2015)
27. Li, Y., Tian, X., Song, M., Tao, D.: Multi-task proximal support vector machine. Pattern Recognit. **48**(10), 3249–3257 (2015)
28. Lu, L., Lin, Q., Pei, H., Zhong, P.: The als-svm based multi-task learning classifiers. Appl. Intell. **48**(8), 2393–2407 (2018)
29. Mangasarian, O.L., Wild, E.W.: Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE Trans. Pattern Anal. Mach. Intell. **28**(1), 69–74 (2005)
30. Mei, B., Xu, Y.: Multi-task least squares twin support vector machine for classification. Neurocomputing **338**, 26–33 (2019)
31. Mei, B., Xu, Y.: Multi-task ν-twin support vector machines. Neural Comput. & Applic. **32**(15), 11329–11342 (2020)
32. Moosaei, H., Bazikar, F., Ketabchi, S., Hladík, M.: Universum parametric-margin ν-support vector machine for classification using the difference of convex functions algorithm. Appl. Intell. pp. 1–21 (2021)
33. Moosaei, H., Ketabchi, S., Razzaghi, M., Tanveer, M.: Generalized twin support vector machines. Neural Process. Lett. **53**(2), 1545–1564 (2021)
34. Osuna, E., Freund, R., Girosit, F.: Training support vector machines: an application to face detection. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 130–136 (1997)
35. Pang, X., Zhang, Y., Xu, Y.: A novel multi-task twin-hypersphere support vector machine for classification. Inf. Sci. **598**, 37–56 (2022)
36. Patrício, M., Pereira, J., Crisóstomo, J., Matafome, P., Gomes, M., Seiça, R., Caramelo, F.: Using resistin, glucose, age and bmi to predict the presence of breast cancer. BMC Cancer **18**(1), 1–8 (2018)
37. Peng, X.: A ν-twin support vector machine (ν-tsvm) classifier and its geometric algorithms. Inf. Sci. **180**(20), 3863–3875 (2010)
38. Peng, X.: Tsvr: an efficient twin support vector machine for regression. Neural Netw. **23**(3), 365–372 (2010)

39. Ramsundar, B., Kearnes, S., Riley, P., Webster, D., Konerding, D., Pande, V.: Massively multitask networks for drug discovery. arXiv preprint arXiv:1502.02072 (2015)
40. Roshanzamir, M., Shamsi, A., Asgharnezhad, H., Alizadehsani, R., Hussain, S., Moosaei, H., Mohammadi, A., Acharya, U.R., Alinejad, H.: Quantifying uncertainty in automated detection of alzheimer's patients using deep neural network. arXiv preprint arXiv:1502.02072 (2023)
41. Shao, Y.H., Deng, N.Y., Yang, Z.M.: Least squares recursive projection twin support vector machine for classification. Pattern Recognit. **45**(6), 2299–2307 (2012)
42. Shao, Y.H., Zhang, C.H., Wang, X.B., Deng, N.Y.: Improvements on twin support vector machines. IEEE Trans. Neural Netw. **22**(6), 962–968 (2011)
43. Shiao, H.T., Cherkassky, V.: Implementation and comparison of SVM-based multi-task learning methods. In: The 2012 International Joint Conference on Neural Networks (IJCNN), pp. 1–7. IEEE (2012)
44. Song, Y., Zhu, W.: Multi-task support vector machine for data classification. Int. J. Signal Process. Image Process. Pattern Recognit. **9**(7), 341–350 (2016)
45. Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. Neural Process. Lett. **9**(3), 293–300 (1999)
46. Wang, L., Gao, C., Zhao, N., Chen, X.: Wavelet transform-based weighted ν-twin support vector regression. Int. J. Mach. Learn. Cybern. **11**(1), 95–110 (2020)
47. Wang, S., Chang, X., Li, X., Sheng, Q.Z., Chen, W.: Multi-task support vector machines for feature selection with shared knowledge discovery. Signal Process. **120**, 746–753 (2016)
48. Xiao, Y., Wen, J., Liu, B.: A new multi-task learning method with universum data. Appl. Intell. **51**(6), 3421–3434 (2021)
49. Xie, X., Sun, S.: Multitask twin support vector machines. In: International Conference on Neural Information Processing, pp. 341–348. Springer (2012)
50. Xie, X., Sun, S.: Multitask centroid twin support vector machines. Neurocomputing **149**, 1085–1091 (2015)
51. Xu, S., An, X., Qiao, X., Zhu, L.: Multi-task least-squares support vector machines. Multimedia Tools Appl. **71**(2), 699–715 (2014)
52. Xue, Y., Beauseroy, P.: Multi-task learning for one-class svm with additional new features. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 1571–1576. IEEE (2016)
53. Yan, H., Ye, Q., Zhang, T., Yu, D.J., Yuan, X., Xu, Y., Fu, L.: Least squares twin bounded support vector machines based on l1-norm distance metric for classification. Pattern Recognit. **74**, 434–447 (2018)
54. Yan, Y., Ricci, E., Subramanian, R., Liu, G., Sebe, N.: Multitask linear discriminant analysis for view invariant action recognition. IEEE Trans. Image Process. **23**(12), 5599–5611 (2014)
55. Yang, H., King, I., Lyu, M.R.: Multi-task learning for one-class classification. In: The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2010)
56. Yang, Z.M., Wu, H.J., Li, C.N., Shao, Y.H.: Least squares recursive projection twin support vector machine for multi-class classification. Int. J. Mach. Learn. Cybern. **7**(3), 411–426 (2016)
57. Yu, K., Tresp, V., Schwaighofer, A.: Learning gaussian processes from multiple tasks. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 1012–1019 (2005)
58. Zhang, X., Zhu, Z.: Sparse multi-task least-squares support vector machine. In: Zhang, H., Zhang, Z., Wu, Z., Hao, T. (eds.) Neural Computing for Advanced Applications, NCAA 2020, CCIS, vol. 1265, pp. 157–167. Springer, Singapore (2020)
59. Zwitter, M., Soklic, M.: Breast cancer data. University Medical Center, Ljubljana, Yugoslavia, Institute Of Oncology (1988)