Soft Computing
https://doi.org/10.1007/s00500-024-09917-3

DATA ANALYTICS AND MACHINE LEARNING

A novel fuzzy twin support vector machine based on centered kernel alignment

Jialiang Xie[1] · Jianxiang Qiu[1] · Dongxiao Zhang[1] · Ruping Zhang[2]

Accepted: 22 May 2024

Abstract
Twin Support Vector Machine (TSVM) transforms a single large quadratic programming problem (QPP) in support vector machine (SVM) into two smaller QPPs by finding two non-parallel classification hyperplanes, so that its computational time is reduced to a quarter of what the traditional SVM takes. However, TSVM ignores the data distribution of class, which makes TSVM sensitive to noise. In this paper, a fuzzy twin support vector machine based on centered kernel alignment (FTSVM-CKA) is proposed to solve the problem that TSVM is sensitive to noise. Firstly, a feature-weighted kernel function is constructed by using the information gain, and it is applied to the calculation of the centered kernel alignment (CKA). This assigns greater weight to strongly correlated features, emphasizing their classification importance over weakly correlated features. Secondly, the CKA method is utilized to derive a heuristic function for calculating the dependency between samples and their corresponding labels, which assigns fuzzy membership to different samples. Based on this, a fuzzy membership assignment strategy is proposed that can effectively address the sensitivity of TSVM to noise. Thirdly, this strategy is combined with TSVM to propose the FTSVM-CKA model. Moreover, this study employs a coordinate descent strategy with shrinking by active set to tackle the computational complexity arising from high-dimensional inputs. This can effectively accelerate the training speed of the model while ensuring classification performance. In order to evaluate the performance of FTSVM-CKA, this study conducts experiments designed on artificial and UCI datasets. The results demonstrate that FTSVM-CKA can efficiently and quickly solve binary classification problems with noise.

Keywords Classification · Twin support vector machine · Membership function · Centered kernel alignment · Feature weighting

1 Introduction

Support Vector Machine (SVM) (Cortes and Vapnik 1995) utilizes the principle of structural risk minimization and solves a convex quadratic programming problem (QPP) to find the optimal hyperplane, making it an effective machine learning algorithm for solving pattern recognition problems. Due to its theoretical advantages and excellent generalization performance, SVM is widely used in many fields. However, traditional SVM has high computational complexity and it is difficult to rapidly process huge and complex data. To solve this problem, Khemchandani and Chandra (2007) proposed the Twin Support Vector Machine (TSVM). Unlike conventional SVM, TSVM aims to find two non-parallel classification hyperplanes and makes each plane move closer to one class and stay as far as possible from the other. Furthermore, the single large QPP in SVM is transformed into two smaller QPPs, so that the computational time of TSVM is reduced to a quarter of that of traditional SVM. When dealing with large-scale classification problems (Xie et al. 2023a, b), TSVM exhibits shorter training times and lower training costs, which overcome the shortcomings of existing SVMs. Moreover, TSVM is also superior to some existing models in terms of classification performance (Tanveer et al.

✉ Jianxiang Qiu
895957339@qq.com

Jialiang Xie
xiejialiang@jmu.edu.cn

Dongxiao Zhang
dx.z@foxmail.com

Ruping Zhang
MAT2109476@xmu.edu.my

[1] School of Science, Jimei University, Xiamen 361021, China

[2] School of Mathematics and Physics, Xiamen University Malaysia, 43900 Sepang, Selangor, Malaysia

Published online: 24 July 2024

2022b). Therefore, TSVM has been widely used in many fields, such as Alzheimer's disease prediction (Ganaie et al. 2023; Sharma et al. 2022), EEG signal classification (Ganaie et al. 2022a; Hazarika et al. 2023), and text recognition (Francis and Sreenath 2022), etc.

It is worth noting that TSVM and SVM ignore the data distribution of class, which makes them sensitive to noise (Liang and Zhang 2022). To address this issue, researchers have combined fuzzy sets theory with them. Different fuzzy membership assignment strategies are proposed to describe the influence of different samples on the construction of the optimal hyperplane. Then, the negative impact of noise is reduced, and the classification performance is improved (Ganaie et al. 2022, 2020). For example, Yu et al. (2019) utilized a K-nearest neighbors-based probability density estimation alike strategy to calculate the relative density of each training instance, and thus proposed a fuzzy support vector machine with relative density information. Borah and Gupta (2022) incorporated fuzzy membership values, computed using transformed class probability and class affinity, into the objective function of least squares support vector machine type formulation, and then propose an affinity and transformed class probability-based fuzzy least squares support vector machine. Kung and Hao (2023) proposed a fuzzy least squares support vector machine with fuzzy hyperplane. The two key characteristics of the proposed model are that it assigns fuzzy membership degrees to every data vector according to the importance and the parameters for the hyperplane, such as the elements of normal vector and the bias term, are fuzzified variables.

In order to reduce the impact of outliers, Richhariya and Tanveer (2018) proposed a new fuzzy membership function that takes into account both the importance of samples and the data imbalance ratio. This function is combined with the least squares twin support vector machine to effectively address the class imbalance problems. Chen and Wu (2018) employed some available fuzzy membership functions from fuzzy neural networks to weigh the margin of each training sample. By design, the impact of the samples with high uncertainty can be mitigated, which improves the generalization ability of the model. Gupta et al. (2019) proposed a fuzzy membership assignment strategy based on information entropy and combined with TSVM for class imbalance learning. Hao et al. (2021) evaluated which fuzzy hyperplanes each sample lies closest to by defining the fuzzy partial ordering relation and then developed a novel fuzzy TSVM to merge the large volume of information from online news, using this to predict stock price trends. Ganaie et al. (2021) proposed a novel fuzzy least squares projection twin support vector machine, which seeks projections such that the samples of each class are clustered around its corresponding mean and assigns fuzzy weights to each sample to reduce the effect of outliers. Motivated by the idea of angle-based algorithms, Richhariya et al. (2021) proposed an efficient angle-based universum least squares twin support vector machine (AULSTSVM). It is capable of handling heteroscedastic noise in large-scale datasets. Richhariya et al. (2021a) proposed a fuzzy universum least squares twin support vector machine, which assigns fuzzy membership to the universum data, aiming to provide appropriate data distribution information to the classifier. This approach was applied to Alzheimer's disease and breast cancer detection.

Recently, Rezvani et al. (2019) combined intuitionistic fuzzy sets with TSVM to address the issue of sensitivity to noise, and then resulted in an extension of FTSVM known as intuitionistic fuzzy twin support vector machine (IFTSVM). In IFTSVMs, each training sample is assigned a membership degree and a non-membership degree to construct a scoring function that characterizes the sample's importance so as to reduce the impact of noise. On this basis, Rezvani and Wang (2021, 2022) respectively used fuzzy Adaptive Resonance Theory and the weighting strategy in conjunction with IFTSVM to tackle the problem of class imbalance learning, explicitly addressing the challenges posed by large-scale class imbalance problems containing noise. Tanveer et al. (2022a) proposed a novel intuitionistic fuzzy weighted least squares twin support vector machine which uses local neighborhood information among the data points and also uses both membership and non-membership weights to reduce the effect of noise and outliers. It was applied to the diagnosis of schizophrenia disease. Ju et al. (2021) combined interval-valued fuzzy sets with TSVMs to address multi-class problems. In this method, interval-valued fuzzy membership is assigned to each sample. Then, an interval-valued fuzzy twin support vector machine is proposed, which effectively reduces the influence of noise and improves the classification performance. In conclusion, constructing an appropriate fuzzy membership assignment strategy is a crucial method to effectively solve the sensitivity to noise in TSVM.

Centered kernel alignment (CKA) is a method that can measure the degree of similarity between two kernels (or kernel matrices). It has been applied to improve the performance of machine learning algorithms due to its effectiveness and low computational complexity. For example, Lu et al. (2014) employed CKA to unify the two tasks of clustering and multiple kernel learning into a single optimization framework, and then a novel multiple kernel clustering method was proposed. In Cárdenas et al. (2016) utilized CKA to assess the affinity between the resonance imaging data kernel matrix and the label target matrix, and then an improved artificial neural network algorithm was proposed to solve the diagnosis problem of Alzheimer's disease. Wang et al. (2020) combined CKA with SVM to propose a classification algorithm robust to noise where CKA is employed to calculate the dependence between a data point and its associated label.

Therefore, it is worth investigating the use of CKA to address the sensitivity to noise.

In this paper, a fuzzy twin support vector machine based on CKA is proposed to address the problem that TSVM is sensitive to noise. This method uses a heuristic function derived from the CKA to calculate the dependence between a data point and its corresponding label and then assigns fuzzy membership to different sample points. Furthermore, a fuzzy membership assignment strategy that can effectively solve the sensitivity of TSVM to noise is proposed. In order to mitigate the dominance of weakly correlated or irrelevant features in the calculation process, a feature-weighted kernel function is constructed by using the information gain, and it is applied to the calculation of the centered kernel alignment. This gives more weight to the strongly correlated features than to the weakly correlated features in order to describe the classification importance of different features. The strategy is combined with TSVM, and then a new fuzzy twin support vector machine (FTSVM-CKA) is proposed. Moreover, to speed up the training of the model, we employ a coordinate descent strategy with shrinking by active set to reduce computational complexity. This can effectively improve the training speed of the model while maintaining the classification performance. Experiments were conducted on an artificial data set and 15 UCI data sets to validate the performance of FTSVM-CKA. The results show that FTSVM-CKA can efficiently and rapidly solve binary classification problems with noise.

In summary, the main contributions of this paper are as follows:

(1) The idea of feature weighting is integrated into the centered kernel alignment method. This paper constructs a feature weighting kernel function and applies it to the calculation of the centered kernel alignment, thus avoiding being dominated by weakly correlated or uncorrelated features in the calculation process.

(2) A fuzzy membership assignment strategy based on the centered kernel alignment method is given. This strategy can significantly reduce the negative impact of noise.

(3) Combining the fuzzy membership assignment strategy based on centered kernel alignment with TSVM, this paper proposed a FTSVM-CKA model, which could effectively solve the classification problem with noise.

(4) The computational complexity brought by the high-dimensional input is addressed by the coordinate descent strategy with shrinking by active set, which then effectively improves the classification speed of the model.

(5) For nonlinear case, kernel trick is applied directly and hence, the exact formulation is solved.

(6) Experimental results on the benchmark dataset demonstrate the ability of the proposed FTSVM-CKA to reduce the negative impact of noise.

The remaining part of this paper is organized as follows: Section 2 reviews some preliminaries. Section 3 describes the structure of the proposed FTSVM-CKA model in detail. The experimental results are reported in Sect. 4. Finally, conclusions and further work are presented in Sect. 5.

2 Related works

In this section, the model structure of TSVM is introduced, and then the concepts of centered kernel alignment and information gain are elaborated. Let $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}$ be the training sample set, where l is the total number of training samples, $x_i \in R^d$ and $y_i \in \{-1, +1\}$, $i = 1, 2, \ldots, l$ denote the ith training sample and its corresponding target class, respectively. d is the feature dimension of the sample.

2.1 Twin support vector machine

Different from the conventional SVM, TSVM aims to generate two non-parallel planes $w_1^T x + b_1 = 0$ and $w_2^T x + b_2 = 0$. Each plane is closer to one of the two classes and as far away from the other as possible. The optimization problem for TSVM can be modeled as the following two smaller scale QPPs:

$$\begin{aligned} &\min_{w_1, b_1, \xi_2} \frac{1}{2}(Aw_1 + e_1 b_1)^T (Aw_1 + e_1 b_1) + C_1 {e_2}^T \xi_2 \\ &s.t. -(Bw_1 + e_2 b_1) + \xi_2 \geq e_2, \xi_2 \geq 0 \end{aligned} \tag{1}$$

and

$$\begin{aligned} &\min_{w_2, b_2, \xi_1} \frac{1}{2}(Bw_2 + e_2 b_2)^T (Bw_2 + e_2 b_2) + C_2 {e_1}^T \xi_1 \\ &s.t. (Aw_2 + e_1 b_2) + \xi_1 \geq e_1, \xi_1 \geq 0 \end{aligned} \tag{2}$$

where, A and B denote all samples belonging to the positive and negative classes, respectively. ξ_1 and ξ_2 are slack variables, e_1 and e_2 are the vector of ones with adequate length, C_1 and C_2 are penalty parameters.

By solving the dual problems of Eq. (1) and Eq. (2), two optimal hyperplanes can be obtained. For any input sample x^*, its classification decision function is as follows:

$$y^* = \arg\min_{i \in 1,2} \frac{\left| w_i^T x^* + b_i \right|}{\| w_i \|} \tag{3}$$

2.2 Centered kernel alignment

Centered kernel alignment (CKA) (Cortes et al. 2012) measures the degree of similarity between two kernels (or kernel matrices) and has been widely used for kernel learning and

selection due to its effectiveness and low computational complexity.

For data set $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}$, The kernel matrix K derived from kernel functions k is given by $K_{i,j} = k\left(x_i, x_j\right)$. Given two kernel functions k_1 and k_2, their corresponding kernel matrices are K_1 and K_2, respectively. The Frobenius inner product between matrices K_1 and K_2 is expressed as follows:

$$\langle K_1, K_2\rangle_F = \sum_{i=1}^{l}\sum_{j=1}^{l} k_1(x_i, x_j)k_2(x_i, x_j) \tag{4}$$

Let $e = (1, 1, \ldots, 1)^T \in R^l$ and $I \in R^{l\times l}$ is the identity matrix, then the centering matrix H and the centered kernel matrix $\overline{K}$ are calculated as follows:

$$H = I - \frac{ee^T}{l} \in R^{l\times l} \tag{5}$$

$$\overline{K} = HKH \tag{6}$$

The CKA of k_1 and k_2 on data set S is defined as

$$CKA(K_1, K_2) = \frac{\left\langle \overline{K_1}, \overline{K_2}\right\rangle_F}{\sqrt{\left\langle \overline{K_1}, \overline{K_1}\right\rangle_F \left\langle \overline{K_2}, \overline{K_2}\right\rangle_F}} \tag{7}$$

2.3 Information gain

Information gain (Han et al. 2022) is often used for feature correlation analysis.

Suppose the sample set S has m category labels $C_i, i = 1, 2, \ldots, m$, S_i denotes the set of all samples in S with label C_i, then the information entropy of S is defined as follows:

$$Info(S) = -\sum_{i=1}^{m} p_i \log_2(p_i) \tag{8}$$

where, $p_i = \frac{|S_i|}{|S|}$ denotes the proportion of samples with label C_i in the sample set S, $|\cdot|$ denotes the cardinality.

For a certain feature F, suppose it has different values $f_i, i = 1, 2, \ldots, v$ and the sample set S is correspondingly split into $S_i, i = 1, 2, \ldots, v$, where S_i contains all the samples in S whose feature F take the value f_i. Then the information gain $IG(S, F)$ is defined as follows:

$$IG(S, F) = Info(S) - \sum_{i=1}^{v} \frac{|S_i|}{|S|} \cdot Info(S_i) \tag{9}$$

3 A novel fuzzy twin support vector machine based on centered kernel alignment

In this section, we first propose a fuzzy membership assignment strategy based on centered kernel alignment. Then we elaborate the model structure of FTSVM-CKA in the linear and nonlinear cases. Finally, a coordinate descent strategy with shrinking by active set is introduced.

3.1 A fuzzy membership assignment strategy based on centered kernel alignment

Firstly, a feature-weighted kernel function is constructed by using the information gain, and it is applied to the calculation of the centered kernel alignment. This gives more weight to the strongly correlated features than to the weakly correlated features, in order to describe the classification importance of different features. Secondly, the centered kernel alignment method is employed to derive a heuristic function that calculates the dependency between sample points and their corresponding labels. This function assigns fuzzy membership degrees to different sample points, effectively mitigating the detrimental effects of noise.

Let a feature-weighted matrix P derived from the information gain be represented as follows:

$$P = \begin{bmatrix} w_1 & & & \\ & w_2 & & \\ & & \ddots & \\ & & & w_d \end{bmatrix} \tag{10}$$

where, $w_i, i = 1, 2, \ldots, d$ denotes the weight of the ith feature calculated by the information gain. Then the feature-weighted kernel function can be defined as $k_p\left(x_i, x_j\right) = k\left({x_i}^T P, {x_j}^T P\right)$. Here are three typical kernels with feature weights:

(1) Linear kernel:

$$k_p\left(x_i, x_j\right) = \left(x_i P\right)\cdot\left(x_j P\right) = x_i P P^T {x_j}^T \tag{11}$$

(2) Polynomial kernel:

$$\begin{aligned} k_p\left(x_i, x_j\right) &= \left[\gamma\left(x_i P\right)\cdot\left(x_j P\right) + r\right]^d \\ &= \left(\gamma x_i P P^T {x_j}^T + r\right)^d, \gamma > 0 \end{aligned} \tag{12}$$

(3) Gaussian kernel:

$$\begin{aligned} k_p(x_i, x_j) &= \exp\left(-\gamma\left\|x_i^T P - x_j^T P\right\|^2\right) \\ &= \exp(-\gamma((x_i - x_j)^T P P^T (x_i - x_j))) \end{aligned} \tag{13}$$

For a binary classification problem, let $K, G \in R^{l\times l}$ be kernel matrices defined as $K_{i,j} = k(x_i, x_j)$ and $G_{i,j} =$

$g(y_i, y_j)$, respectively. The $g(y_i, y_j)$ is defined as follows:

$$g(y_i, y_j) = \begin{cases} +1, y_i = y_j \\ -1, y_i \neq y_j \end{cases} \tag{14}$$

where, the similarities from the same class are set to $+1$, and those from different classes are -1. This definition reveals the ideal pairwise similarities between samples. Let $y = (y_1, y_2, \ldots, y_l)^T$, then

$$\begin{aligned} &CKA(K, G) \\ &= \frac{\langle \overline{K}, \overline{G} \rangle_F}{\sqrt{\langle \overline{K}, \overline{K} \rangle_F \langle \overline{G}, \overline{G} \rangle_F}} = \frac{\langle \overline{K}, G \rangle_F}{\sqrt{\langle \overline{K}, K \rangle_F \langle \overline{G}, G \rangle_F}} \\ &= \frac{\langle \overline{K}, yy^T \rangle_F}{\sqrt{\langle \overline{K}, K \rangle_F \langle \overline{G}, G \rangle_F}} = \frac{\sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \overline{k}(x_i, x_j)}{\sqrt{\langle \overline{K}, K \rangle_F \langle \overline{G}, G \rangle_F}} \\ &= \frac{1}{\sqrt{\langle \overline{K}, K \rangle_F \langle \overline{G}, G \rangle_F}} \left[\sum_{y_i = y_j} \overline{k}(x_i, x_j) - \sum_{y_i \neq y_j} \overline{k}(x_i, x_j) \right] \end{aligned} \tag{15}$$

where, $\overline{k}(x_i, x_j) = \overline{K}_{i,j}$ is the centered kernel function.

For given data set $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}$, kernel functions k and g,we get

$$\overline{K} = HKH, \overline{G} = HGH \tag{16}$$

where $K_{i,j} = k(x_i, x_j)$ and $G_{i,j} = g(y_i, y_j)$, $H = I - \frac{ee^T}{l} \in R^{l \times l}$. Then,

$$\langle \overline{K}, K \rangle_F = \sum_{i=1}^{l} \sum_{j=1}^{l} \overline{k}(x_i, x_j) k(x_i, x_j) \tag{17}$$

and

$$\langle \overline{G}, G \rangle_F = \sum_{i=1}^{l} \sum_{j=1}^{l} \overline{g}(x_i, x_j) g(x_i, x_j) \tag{18}$$

are obtained, where $\overline{k}$ and $\overline{g}$ are the centered kernel functions corresponding to $\overline{K}$ and $\overline{G}$, respectively. Thus, $\frac{1}{\sqrt{\langle \overline{K}, K \rangle_F \langle \overline{G}, G \rangle_F}}$ is a constant, CKA of x_t can be expressed as follows:

$$d_t = CKA(K, G, x_t) = \sum_{y_t = y_i} \overline{k}(x_t, x_i) - \sum_{y_t \neq y_i} \overline{k}(x_t, x_i) \tag{19}$$

Since $\overline{k}(x_t, x_j)$ is the centered kernel function that measures the similarity between points x_t and x_j, it is worth noting that the larger the similarity represented by the kernel for input patterns of the same class and the smaller the similarity for patterns from different classes, the larger the d_t. In other words, a sample with a larger d_t value contributes more to the construction of the optimal classification hyperplane, and a sample with a smaller d_t value is more likely to be noise. Thus the fuzzy membership function based on the CKA to measure the importance of each sample point to the classification can be expressed as follows:

$$s_t = \frac{d_t - d_{\min}}{d_{\max} - d_{\min}} \tag{20}$$

where $d_{\max}$ and $d_{\min}$ denote the largest and smallest CKA value among all sample points, respectively. Therefore, the larger the value of s_t, the greater the contribution of the sample x_t to the construction of the optimal classification hyperplane, and conversely, the sample x_t is likely to be noise. Different from the existing fuzzy membership function based on distance, relative density and entropy, the proposed strategy utilizes the CKA method to derive a heuristic function for calculating the dependency between samples and their corresponding labels, which assigns fuzzy membership to different samples. In addition, the proposed strategy incorporates the idea of feature weighting, which effectively reduces the influence of weakly correlated features. The corresponding input dataset S is thus modified as $S = \{(x_1, y_1, s_1), (x_2, y_2, s_2), \ldots, (x_l, y_l, s_l)\}$.

3.2 Linear FTSVM-CKA

In linear case, the FTSVM-CKA finds the optimal classifier by solving the following two QPPs:

$$\begin{aligned} &\min_{w_1, b_1, \xi_2} \tfrac{1}{2} \|Aw_1 + e_1 b_1\|^2 + \tfrac{1}{2} C_1 \|w_1\|^2 + C_2 S_2{}^T \xi_2 \\ &s.t. -(Bw_1 + e_2 b_1) + \xi_2 \geq e_2, \xi_2 \geq 0 \end{aligned} \tag{21}$$

and

$$\begin{aligned} &\min_{w_2, b_2, \xi_1} \tfrac{1}{2} \|Bw_2 + e_2 b_2\|^2 + \tfrac{1}{2} C_3 \|w_2\|^2 + C_4 S_1{}^T \xi_1 \\ &s.t. (Aw_2 + e_1 b_2) + \xi_1 \geq e_1, \xi_1 \geq 0 \end{aligned} \tag{22}$$

where, C_1, C_2, C_3 and C_4 are penalty parameters, ξ_1 and ξ_2 are slack variables, e_1 and e_1 are the vector of ones with adequate length. $S_1 \in R^{l+}$ and $S_2 \in R^{l-}$ denote the corresponding fuzzy membership of positive and negative class samples, respectively.

This paper takes the process of solving problem (21) as an example. The Lagrangian of problem (21) is written as

$$\begin{aligned} &L(w_1, b_1, \xi_2, \alpha, \beta) \\ &= \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + \frac{1}{2} C_1 \|w_1\|^2 \end{aligned}$$

$$+ C_2 {S_2}^T \xi_2 + \alpha[(Bw_1 + e_2 b_1) - \xi_2 + e_2] - \beta\xi_2 \quad (23)$$

where α and β are Lagrange multipliers. Applying Karush-Kuhn-Tucker (KKT) conditions, we get

$$\frac{\partial L}{\partial w_1} = A^T(Aw_1 + e_1 b_1) + C_1 w_1 + \alpha B = 0 \quad (24)$$

$$\frac{\partial L}{\partial b_1} = e_1^T(Aw_1 + e_1 b_1) + \alpha e_2 = 0 \quad (25)$$

$$\frac{\partial L}{\partial \xi_2} = C_2 S_2^T - \alpha - \beta = 0 \quad (26)$$

According to Eq. (24) and Eq. (25),

$$\begin{pmatrix} A^T \\ e_1^T \end{pmatrix} \begin{pmatrix} A & e_1 \end{pmatrix} \begin{pmatrix} w_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} B \\ e_2 \end{pmatrix} \alpha = 0 \quad (27)$$

can be obtained. Let $H_1 = \begin{pmatrix} A & e_1 \end{pmatrix}$, $G_2 = \begin{pmatrix} B & e_2 \end{pmatrix}$, $u_1 = \begin{pmatrix} w_1 \\ b_1 \end{pmatrix}$, $u_2 = \begin{pmatrix} w_2 \\ b_2 \end{pmatrix}$, then, $H_1^T H_1 u_1 + G_2^T \alpha = 0$. Further, we can get

$$u_1 = -(H_1^T H_1)^{-1} G_2^T \alpha \quad (28)$$

Since $(H_1^T H_1)^{-1}$ is difficult to calculate, $(H_1^T H_1 + C_1 I)^{-1}$ is used instead of it in Eq. (28), where I is the identity matrix with the appropriate dimension. Thus,

$$u_1 = -(H_1^T H_1 + C_1 I)^{-1} G_2^T \alpha \quad (29)$$

Similarly,

$$u_2 = (G_2^T G_2 + C_3 I)^{-1} H_1^T \beta \quad (30)$$

According to the KKT conditions, the dual problems of Eq. (21) and Eq. (22) are as follows:

$$\begin{aligned} &\max_{\alpha} e_2{}^T \alpha - \tfrac{1}{2}\alpha^T G_2 (H_1^T H_1 + C_1 I)^{-1} G_2^T \alpha \\ &s.t. 0 \le \alpha \le C_2 S_2 \end{aligned} \quad (31)$$

and

$$\begin{aligned} &\max_{\beta} e_1{}^T \beta - \tfrac{1}{2}\beta^T H_1 (G_2^T G_2 + C_3 I)^{-1} H_1^T \beta \\ &s.t. 0 \le \beta \le C_4 S_1 \end{aligned} \quad (32)$$

We get the optimal $u_1^* = \begin{pmatrix} w_1^* \\ b_1^* \end{pmatrix}$ and $u_2^* = \begin{pmatrix} w_2^* \\ b_2^* \end{pmatrix}$ by solving the two corresponding dual problems. For any input sample x^*, its class label y^* can be determined as follows:

$$y^* = \arg\min_{i \in 1,2} \left\{ \frac{\left| w_i^{*T} x^* + b_i^* \right|}{\left\| w_i^* \right\|} \right\} \quad (33)$$

3.3 Nonlinear FTSVM-CKA

In nonlinear case, the kernel function $k(x_1, x_2) = (\phi(x_1), \phi(x_2))$ is introduced, where ϕ is the Hilbert space transformation. Thus, the classification hyperplanes in the nonlinear case can be represented as $k(x, X^T)w_1 + b_1 = 0$, $k(x, X^T)w_2 + b_2 = 0$, where $X = [A; B]$. The nonlinear FTSVM-CKA is formulated in the primal form as

$$\begin{aligned} &\min_{w_1, b_1, \xi_2} \tfrac{1}{2}\left\| k(A, X^T)w_1 + e_1 b_1 \right\|^2 + \tfrac{1}{2} C_1 \|w_1\|^2 + C_2 {S_2}^T \xi_2 \\ &s.t. -(k(B, X^T)w_1 + e_2 b_1) + \xi_2 \ge e_2, \xi_2 \ge 0 \end{aligned} \quad (34)$$

and

$$\begin{aligned} &\min_{w_2, b_2, \xi_1} \tfrac{1}{2}\left\| k(B, X^T)w_2 + e_2 b_2 \right\|^2 + \tfrac{1}{2} C_3 \|w_2\|^2 + C_4 {S_1}^T \xi_1 \\ &s.t. (K(A, X^T)w_2 + e_1 b_2) + \xi_1 \ge e_1, \xi_1 \ge 0 \end{aligned} \quad (35)$$

The Lagrangian function of the Eq. (34) is written as

$$\begin{aligned} &L(w_1, b_1, \xi_2, \alpha, \beta) = \\ &\tfrac{1}{2}\left\| k(A, X^T)w_1 + e_1 b_1 \right\|^2 + \tfrac{1}{2} C_1 \|w_1\|^2 + C_2 {S_2}^T \xi_2 \\ &+\alpha[(k(B, X^T)w_1 + e_2 b_1) - \xi_2 + e_2] - \beta\xi_2 \end{aligned} \quad (36)$$

Following the same procedure as in the linear case, we get

$$u_1 = -(H_1^{*T} H_1^* + C_1 I)^{-1} G_2^{*T} \alpha. \quad (37)$$

and

$$u_2 = (G_2^{*T} G_2^* + C_3 I)^{-1} H_1^{*T} \beta. \quad (38)$$

where $H_1^* = \left(k(A, X^T)\ e_1\right)$, $G_2^* = \left(k(B, X^T)\ e_2\right)$, $u_1 = \begin{pmatrix} w_1 \\ b_1 \end{pmatrix}$, $u_2 = \begin{pmatrix} w_2 \\ b_2 \end{pmatrix}$. Then, the dual problems of Eq. (34) and Eq. (35) are as follows:

$$\begin{aligned} &\max_{\alpha} e_2{}^T \alpha - \tfrac{1}{2}\alpha^T G_2^* (H_1^{*T} H_1^* + C_1 I)^{-1} G_2^{*T} \alpha \\ &s.t. 0 \le \alpha \le C_2 S_2 \end{aligned} \quad (39)$$

and

$$\begin{aligned} &\max_{\beta} e_1{}^T \beta - \tfrac{1}{2}\beta^T H_1^* (G_2^{*T} G_2^* + C_3 I)^{-1} H_1^{*T} \beta \\ &s.t. 0 \le \beta \le C_4 S_1 \end{aligned} \quad (40)$$

We get the optimal $u_1^* = \begin{pmatrix} w_1^* \\ b_1^* \end{pmatrix}$ and $u_2^* = \begin{pmatrix} w_2^* \\ b_2^* \end{pmatrix}$ by solving the dual problems. For any input sample x^*, its classification decision function is as follows:

$$y^* = \arg\min_{i\in 1,2}\left\{\frac{\left|w_1^{*T}k(x^*, X^T)+b_1^*\right|}{\sqrt{w_1^{*T}k(A, X^T)w_1^*}}, \frac{\left|w_2^{*T}k(x^*, X^T)+b_2^*\right|}{\sqrt{w_2^{*T}k(B, X^T)w_2^*}}\right\}. \tag{41}$$

3.4 The coordinate descent strategy with active set shrinking

To speed up the training, the FTSVM-CKA employs a coordinate descent strategy with shrinking by active set which handles the computational complexity brought by high-dimensional inputs (Gao et al. 2015). Since the dual problems involved in FTSVM-CKA can be solved similarly, we take Eq. (31) as an example. Let $R = (H_1^T H_1 + C_1 I)^{-1} G_2^T$, $\widetilde{R} = G_2 R$, then Eq. (31) can be reduced to the following problem:

$$\begin{aligned} &\min_{\alpha} \mathrm{g}(\alpha) = -e_2{}^T\alpha + \tfrac{1}{2}\alpha^T\widetilde{R}\alpha \\ &s.t. 0 \le \alpha \le C_2 S_2 \end{aligned} \tag{42}$$

A coordinate descent strategy with shrinking by active set is adopted to solve Eq. (42). Its pseudo code is shown in Algorithm 1. $g_{\nabla i}(\alpha)$ is a projection gradient, as follows:

$$g_{\nabla i}(\alpha) = \begin{cases} \min(0, g_{\partial i}(\alpha)), if \alpha_i = 0 \\ g_{\partial i}(\alpha), if 0 < \alpha_i < C_2 \mathrm{s}_i \\ \max(0, g_{\partial i}(\alpha)), if \alpha_i = C_2 \mathrm{s}_i \end{cases} \tag{43}$$

where $g_{\partial i}$ is the ith component of gradient g_∂. Reference to Chang and Lin (2011), Chang et al. (2008), and Shao and Deng (2012) for some details.

Algorithm 1 The coordinate descent strategy with active set shrinking

Let $R = (H_1^T H_1 + C_1 I)^{-1} G_2^T$, $\widetilde{R}_{ii} = G_{2i} R_i$ and $D_1 = \{1, 2, \ldots, l_1\}$
Given τ and initialized $\alpha \leftarrow 0$, $u_1 \leftarrow 0$, $\widetilde{K} \leftarrow \infty$ and $\widetilde{k} \leftarrow -\infty$
while do
  Initialize $K \leftarrow -\infty$ and $k \leftarrow \infty$
  for all $i \in D_1$ **do**
    Let $g_{\partial i}(\alpha) = -G_{2i} u_1 - 1$
    Let $g_{\nabla i}(\alpha) \leftarrow 0$
    if $\alpha_i = 0$ **then**
      if $g_{\nabla i}(\alpha) < 0$, then $g_{\nabla i}(\alpha) \leftarrow g_{\partial i}(\alpha)$ end if
      if $g_{\nabla i}(\alpha) > \widetilde{K}$, then $D_1 = D_1 \setminus \{i\}$ end if
    else if $\alpha_i = C_2 s_i$ **then**
      if $g_{\nabla i}(\alpha) > 0$, then $g_{\nabla i}(\alpha) \leftarrow g_{\partial i}(\alpha)$ end if
      if $g_{\nabla i}(\alpha) < \widetilde{k}$, then $D_1 = D_1 \setminus \{i\}$ end if
    else
      $g_{\nabla i}(\alpha) \leftarrow g_{\partial i}(\alpha)$
    end if
    $K \leftarrow \max(K, g_{\nabla i}(\alpha))$, $k \leftarrow \min(k, g_{\nabla i}(\alpha))$
    if $g_{\nabla i}(\alpha) \neq 0$ **then**
      $\widetilde{\alpha}_i \leftarrow \alpha_i$
      $\alpha_i \leftarrow \min(\max(\alpha_i - g_{\partial i}(\alpha)/\widetilde{R}_{ii}, 0), C_2 s_i)$
      $u_{1i} \leftarrow u_{1i} - R_i(\alpha_i - \widetilde{\alpha}_i)$
    end if
  end for
  if $K - k < \tau$ **then**
    if $D_1 = \{1, 2, \ldots, l_1\}$, break
  else
    $D_1 \leftarrow \{1, 2, \ldots, l_1\}$, $\widetilde{K} \leftarrow \infty$, $\widetilde{k} \leftarrow -\infty$
    if $K \ge 0$, then $\widetilde{k} \leftarrow -\infty$, else $\widetilde{k} \leftarrow k$, end if
    if $K \le 0$, then $\widetilde{K} \leftarrow \infty$, else $\widetilde{K} \leftarrow K$, end if
  end if
end while

4 Experimental results

In this paper, different experiments are designed on an artificial dataset, i.e., Ripleys (Ripley 2007) and 15 real-world data sets from UCI machine learning repository (Dua et al. 2017), to evaluate the performance of FTSVM-CKA. TSVM (Khemchandani and Chandra 2007), CDFTSVM (Gao et al. 2015), IFTSVM (Rezvani et al. 2019), AULSTSVM (Richhariya et al. 2021), CatBoost (Prokhorenkova et al. 2018), LightGBM (Ke et al. 2017), XGBoost (Chen et al. 2015), SVM (Cortes and Vapnik 1995), RandomForest (RF) (Breiman 2001) are used as comparison algorithms. For parameters $C_i, i = 1, 2, 3, 4$ in FTSVMs, as for CDFTSVM and IFTSVM, we set $C_1 = C_3, C_2 = C_4$ for FTSVM-CKA. They are correctly explored in $\{10^i | i = -5, -4, \ldots, 4, 5\}$. While the parameters are set as $c_1 = c_2$, $c_3 = c_5 = c_1 \cdot c_4$, $c_4 = c_6$ for AULSTSVM (Richhariya et al. 2021). In addition, Gaussian kernel function, i.e., $k(x_1, x_2) = \exp(\frac{-\|x_1 - x_2\|^2}{\sigma^2})$ is used in this paper, and σ is explored in $\{2^i | i = -5, -4, \ldots, 4, 5\}$. The 10-fold cross-validation is performed for all the algorithms. All samples are normalized. To simulate label noise, we randomly select a given proportion of samples and flip their corresponding labels, and this proportion is called the noise rate. The experimental environment is listed as follows: Inter Core i5-11500 CPU, 8 G, Windows10, MATLAB2018b.

4.1 Parameter effect

In this subsection, the effect of different C_i and σ are considered in the Horse dataset to identify the optimal parameters, i.e., C_i for linear case and C_i and σ for nonlinear case, that produce the best performance. First, C_i, which varies in $\{1, 2, \ldots, 10\}$, is optimized for the linear case. FTSVM-CKA generates better outcomes when $C_1 = 5$ and $C_2 = 2$.

Similarly, for nonlinear case, C_i and σ are optimized and can differ in $\{i \cdot \frac{1}{2} | i = 1, 2, \ldots, 10\}$. FTSVM-CKA with $C_1 = 5, C_2 = 2$ and $\sigma = 1.5$ produces better outcomes. After obtaining the optimal parameter settings, the performance of the model was evaluated on the remaining testing parts.

4.2 Artificial data sets

The Ripleys data set is a mixture of two Gaussian distributions. It comprises two categories, with each sample

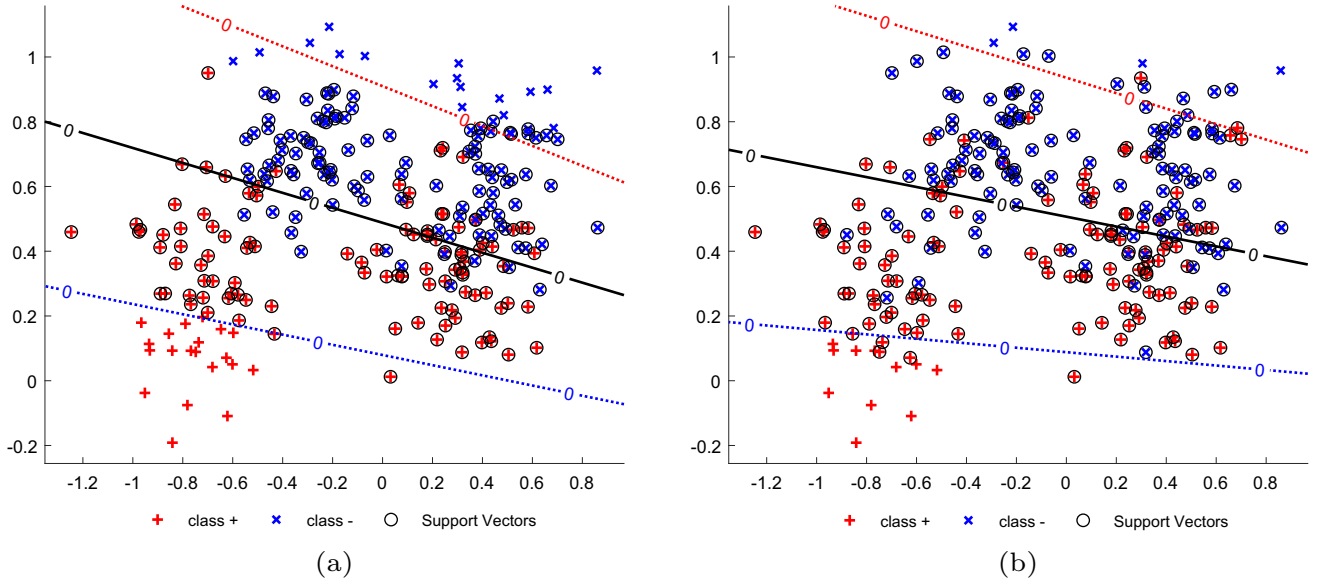

Fig. 1 The hyperplanes by TSVM at 0% noise rate (**a**) and 10% noise rate (**b**)

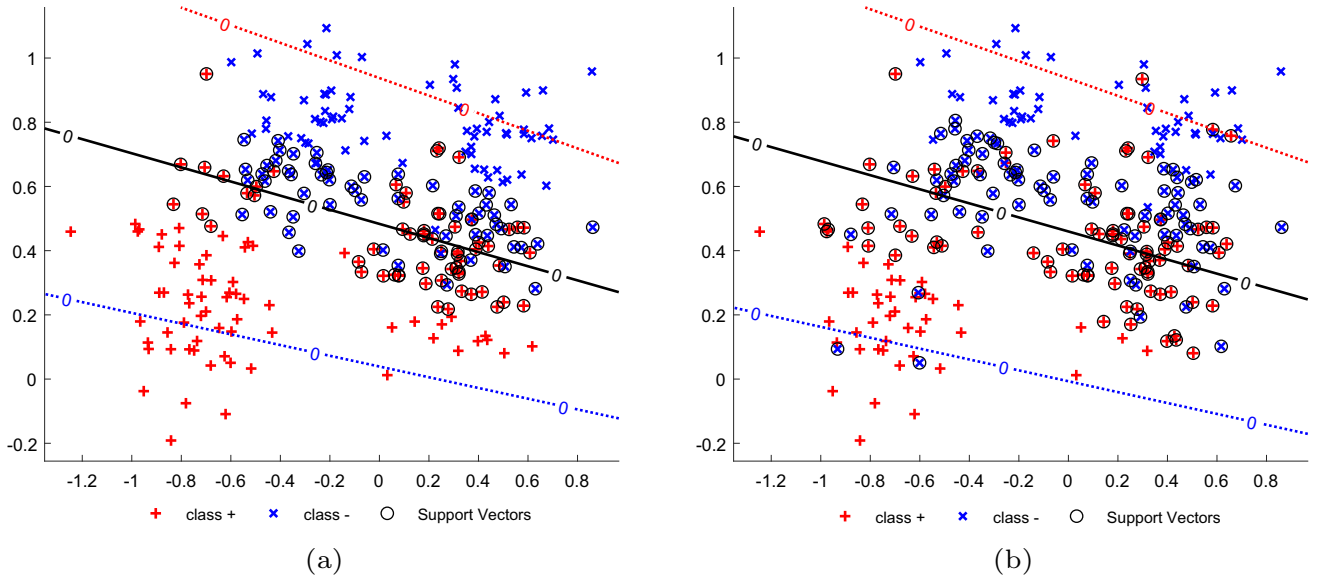

Fig. 2 The hyperplanes by FTSVM-CKA at 0% noise rate (**a**) and 10% noise rate (**b**)

consisting of two features. Figures 1 and 2 show the linear separating hyperplanes generated by TSVM and FTSVM-CKA at a noise rate of 0 and 10%, respectively. Figure 1 reveals that the hyperplanes generated by TSVM exhibit noticeable variations across different noise rates. From Fig. 2, one can observe that the disparity between the hyperplanes generated by FTSVM-CKA at various noise rates is significantly smaller in comparison to TSVM.

Figure 3 illustrates the accuracy of FTSVM-CKA and TSVM with varying noise rates in linear and nonlinear cases. It can be observed that the accuracy of FTSVM-CKA is better than that of TSVM, and the classification performance of both algorithms demonstrate a decreasing trend. From Fig. 3, we can find that the accuracy of TSVM fluctuates significantly with increasing noise rate, suggesting its sensitivity to noise. It is worth noting that compared with TSVM, the accuracy of FTSVM-CKA exhibits lower susceptibility to noise and changes more gently. This indicates that FTSVM-CKA can effectively mitigate the sensitivity of TSVM to noise. In conclusion, the experimental results show that the proposed FTSVM-CKA can suppress the adverse effects of noise because we introduce a fuzzy membership assignment strategy based on CKA.

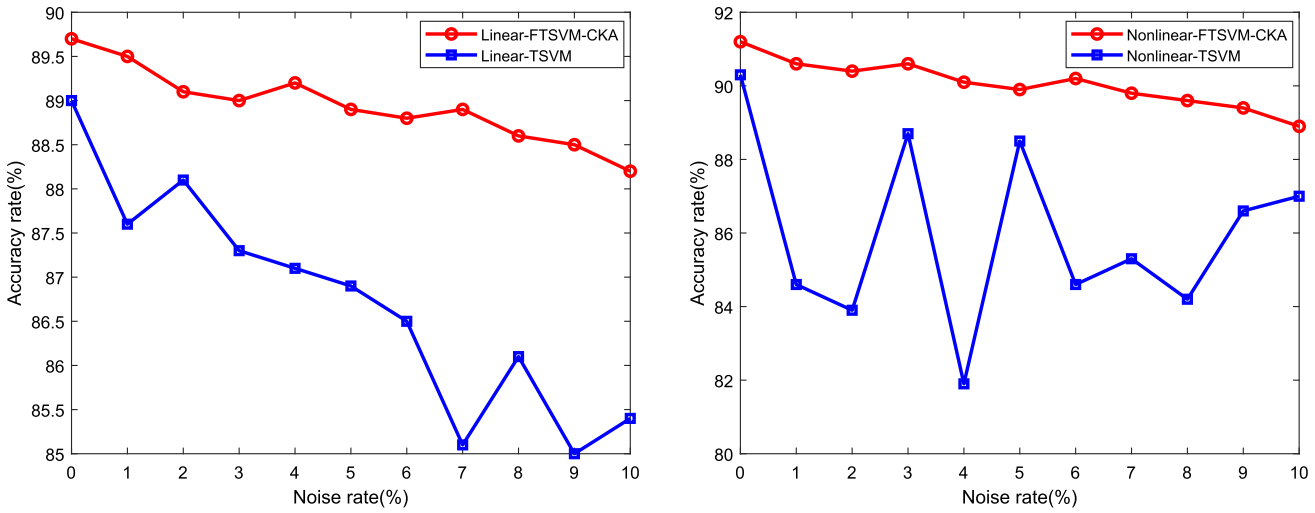

Fig. 3 Accuracy of FTSVM-CKA and TSVM in linear (**a**) and nonlinear (**b**) cases

Table 1 Details of UCI data sets

Data set	Number of samples	Number of features	Number of classes	Original data set
Breast	699	10	2	Breast Cancer Wisconsin (Original)
Credit	690	15	2	Credit Approval
Diabetes	768	8	2	Diabetes (part)
Echocardiogram	132	12	2	Echocardiogram
Heart	294	14	2	Heart Disease (hungarian)
Horse	368	27	2	Horse Colic
Musk	476	168	2	Musk (Version 1)
Parkinsons	197	23	2	Parkinsons
Statlog_1	690	14	2	Statlog(Australian Credit Approval)
Statlog_2	270	13	2	Statlog(Heart)
South	1000	21	2	South German Credit
Cardiotocography	1204	22	2	Cardiotocography1vs6-9
Steel	1941	28	2	Steel Plates Faults0-4vs5-6
Abalone	2730	9	2	Abalone0vs1
Statlog_3	6435	37	2	Statlog (Landsat Satellite)0-2vs3-5

4.3 UCI data sets

Table 1 shows the details of the 15 UCI datasets selected in this paper. In the experiments, the noise rate is set to 0, 5, and 10%, respectively. The average accuracy, along with the standard deviations (SD) and computational time, are calculated to evaluate the experimental results.

We implement TSVM and TSVM-related methods including CDFTSVM and IFTSVM. Tables 2, 3 and 4 present the experimental results of FTSVM-CKA and TSVM, IFTSVM, CDFTSVM in the linear case, with noise rates set at 0, 5, and 10% in sequence. The bold in all tables means the value obtained is the best. The results from Tables 2, 3 and 4 demonstrate that, out of the 13 UCI datasets mentioned earlier, the proposed FTSVM-CKA achieves the highest classification accuracy on 11, 10, and 10 datasets, respectively. The average ranks of accuracy of FTSVM-CKA under different noise rates is 1.15, 1.31 and 1.46, respectively, which are superior to the existing algorithms. This indicates that FTSVM-CKA outperforms the other three algorithms in terms of classification performance in the linear case. FTSVM-CKA utilizes the fuzzy membership assignment strategy based on CKA to mitigate the adverse impact of noise in the classification process, thereby significantly enhancing the classification performance. In addition, it can be found that the calculation time of FTSVM-CKA and CDFTSVM is very close, and both

Table 2 Experimental results on UCI data sets at 0% noise rate in the linear case

Data set	FTSVM-CKA		TSVM		IFTSVM		CDFTSVM		AULSTSVM	
	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time
Breast	**96.1366** (2.1251)	0.000241	95.7017 (2.7977)	0.231902	95.7058 (2.7151)	0.328277	95.8447 (2.9083)	0.000126	95.8630 (2.5322)	0.000458
Credit	85.5290 (3.3835)	0.000272	**85.9535** (3.6054)	0.194665	85.5166 (3.9935)	0.260817	85.4980 (4.4071)	0.000193	85.5185 (3.7181)	0.000333
Diabetes	**75.9279** (4.6666)	0.000320	65.1059 (0.3418)	0.214414	65.7553 (1.1526)	0.357727	74.2242 (4.8936)	0.000279	75.9005 (4.5432)	0.000357
Echocardiogram	**85.7601** (6.8258)	0.000153	67.2161 (2.7215)	0.040214	85.5403 (7.8676)	0.042370	85.4853 (9.9943)	0.000155	85.3388 (6.1167)	0.000173
Heart	**83.6864** (6.3677)	0.000232	81.9450 (5.2620)	0.052636	81.3005 (5.7895)	0.073763	81.2644 (6.3877)	0.000170	83.0197 (3.9179)	0.000499
Horse	**85.3477** (3.9986)	0.000193	81.2624 (7.6862)	0.131241	85.3398 (5.2792)	0.151350	82.6521 (4.1831)	0.000346	84.1718 (4.6643)	0.000432
Musk	**83.8165** (3.5693)	0.000267	80.6472 (6.4600)	0.238643	82.1232 (4.7860)	0.284921	79.8034 (7.3605)	0.000301	83.3910 (6.0426)	0.001128
Parkinsons	**86.2251** (5.4496)	0.000168	71.1053 (13.9457)	0.043082	83.6813 (4.8445)	0.057841	80.9854 (7.6444)	0.000164	86.1316 (3.3121)	0.000226
Statlog_1	**67.8275** (0.3875)	0.000278	67.0391 (0.4996)	0.164965	67.5047 (0.7231)	0.236405	65.5037 (3.4479)	0.000267	67.2499 (0.7717)	0.000264
Statlog_2	85.1852 (6.8293)	0.000182	80.3704 (9.0797)	0.048802	81.8519 (11.2887)	0.064218	83.7037 (7.0662)	0.000162	**85.5556** (7.7000)	0.000357
South	**75.3000** (4.4283)	0.000355	70.0000 (0.0000)	0.371352	74.9000 (3.0806)	0.605975	73.5000 (6.4070)	0.000479	75.0000 (4.3128)	0.000425
Cardiotocography	**98.3388** (1.1746)	0.000253	96.8423 (2.1673)	0.603864	97.3416 (1.2750)	0.800481	97.5888 (1.1453)	0.000341	98.0069 (1.3498)	0.000479
Steel	**80.3637** (2.7761)	0.000280	72.3135 (7.8047)	0.728583	79.5776 (3.0186)	0.937729	77.6803 (2.6071)	0.000347	80.2050 (2.5746)	0.000525
Average rank	**1.15**		4.38		3.23		3.92		2.31	

Table 3 Experimental results on UCI data sets at 5% noise rate in the linear case

Data set	FTSVM-CKA		TSVM		IFTSVM		CDFTSVM		AULSTSVM	
	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time
Breast	**95.5630**(0.7839)	0.000311	93.9833 (3.1287)	0.226737	95.0512(4.9025)	0.324607	95.5589 (2.6883)	0.000129	95.4224 (1.8799)	0.000672
Credit	**85.5205** (5.0289)	0.000203	80.6849 (9.3325)	0.199187	85.5166 (3.9935)	0.238613	85.4993 (3.7814)	0.000180	85.4961 (1.4649)	0.000327
Diabetes	74.0943 (2.4381)	0.000237	65.1059 (0.3418)	0.191610	66.4046 (2.6635)	0.347833	73.8260 (4.0980)	0.000130	**74.7437** (5.0710)	0.001180
Echocardiogram	**85.6960** (5.8271)	0.000253	72.4908 (4.9868)	0.033569	85.5403 (7.8676)	0.037351	85.5220 (6.8646)	0.000115	85.5211 (3.5071)	0.000279
Heart	**81.8530** (8.2736)	0.000235	68.9836 (6.6900)	0.053512	81.6338 (5.6062)	0.066287	81.2529 (5.2669)	0.000120	81.6207 (7.5325)	0.000192
Horse	**85.8966** (4.6364)	0.000198	66.8958 (7.8954)	0.073114	84.2587 (4.6397)	0.101150	80.1664 (4.7030)	0.000250	83.7478 (5.5683)	0.000472
Musk	82.3316 (3.3799)	0.000248	77.0833 (6.4015)	0.208932	**82.5576** (5.6431)	0.270178	78.1826 (5.6014)	0.000243	81.4805 (4.6225)	0.001102
Parkinsons	**85.7632** (6.1191)	0.000302	69.1404 (8.4938)	0.038576	77.4357 (2.3725)	0.050250	80.1053 (10.7645)	0.000151	85.7339 (5.7452)	0.000197
Statlog_1	**67.8268** (0.4931)	0.000301	65.6308 (0.4971)	0.157825	65.9447 (0.3844)	0.244421	64.4750 (5.5245)	0.000172	66.6674 (1.9208)	0.000391
Statlog_2	**84.8148** (9.2296)	0.000300	68.1481 (9.6864)	0.047033	82.2222 (13.1260)	0.053390	81.1111 (6.9191)	0.000210	84.4444 (2.9215)	0.000253
South	74.7000 (2.6926)	0.000289	70.0000 (0.0000)	0.370359	**75.1000** (2.8792)	0.556702	73.5000 (2.9069)	0.000300	75.0000 (4.3589)	0.000346
Cardiotocography	**98.0084** (1.9241)	0.000247	95.7652 (2.6055)	0.627751	98.0041 (1.1306)	0.781998	97.8437 (1.8537)	0.000372	97.6763 (1.5670)	0.000687
Steel	**80.2862** (2.0628)	0.000277	58.2808 (0.1794)	0.919613	79.5794 (2.8623)	0.991378	77.9953 (1.4113)	0.000246	79.4969 (4.3350)	0.000455
Average rank	**1.31**		4.92		2.46		3.54		2.77	

Table 4 Experimental results on UCI data sets at 10% noise rate in the linear case

Data set	FTSVM-CKA		TSVM		IFTSVM		CDFTSVM		AULSTSVM	
	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time
Breast	**95.2731** (2.5743)	0.000241	82.4176 (8.3387)	0.230535	95.1364 (2.8708)	0.318942	94.8568 (2.1254)	0.000183	95.2711 (1.5437)	0.000549
Credit	**85.5293** (4.1933)	0.000236	55.5077 (0.4839)	0.203459	85.5166 (3.9935)	0.230433	85.4870 (4.1410)	0.000634	85.2326 (4.9007)	0.000265
Diabetes	72.0130 (2.0183)	0.000256	65.1059 (0.3418)	0.188546	64.9761 (0.5594)	0.338643	68.3527 (3.2511)	0.000127	**73.5800** (5.2523)	0.000524
Echocardiogram	**85.5952** (5.0534)	0.000144	77.8480 (10.5656)	0.030397	85.5403 (7.8676)	0.035699	84.7802 (8.2894)	0.000189	85.3480 (9.0272)	0.000289
Heart	76.0805 (8.8638)	0.000205	63.9606 (0.8613)	0.049124	**82.3120** (5.5438)	0.067555	80.9631 (8.1006)	0.000146	82.3062 (4.5247)	0.000264
Horse	**83.6593** (4.3051)	0.000242	63.0520 (0.7740)	0.075821	83.4554 (6.2628)	0.130301	79.5914 (4.7059)	0.000224	83.1065 (6.2454)	0.000260
Musk	**80.4743** (3.5935)	0.000310	70.3812 (5.3258)	0.215265	80.2482 (4.0146)	0.264672	75.4282 (4.9484)	0.000264	79.4268 (3.1374)	0.001029
Parkinsons	80.4737 (5.6524)	0.000185	67.3363 (11.2583)	0.040913	76.4094 (3.2695)	0.050661	79.0497 (5.3247)	0.000125	**83.6579** (6.2023)	0.000257
Statlog_1	**67.8268** (0.4931)	0.000263	64.2041 (0.4994)	0.147291	64.7259 (0.8637)	0.213931	62.3179 (5.6133)	0.000239	67.1063 (1.1445)	0.000283
Statlog_2	**84.0741** (6.4257)	0.000316	60.7407 (8.3148)	0.042912	83.7037 (8.6384)	0.055653	80.3704 (7.4167)	0.000148	**84.0741** (4.9537)	0.000509
South	**74.6000** (2.7276)	0.000361	70.0000 (0.0000)	0.327578	73.0000 (2.3238)	0.561949	72.9000 (6.5184)	0.000414	**74.6000** (4.1521)	0.000389
Cardiotocography	**98.0176** (1.0552)	0.000297	80.2156 (11.2841)	0.649579	98.0055 (1.1280)	0.847871	97.5055 (0.9896)	0.000280	97.7562 (0.9892)	0.000409
Steel	**80.2043** (2.0180)	0.000261	58.2808 (0.1794)	0.886136	79.7350 (2.5174)	0.859876	77.1322 (3.6081)	0.000274	79.6488 (2.6224)	0.000502
Average rank	**1.46**		4.85		2.62		3.77		2.31	

Table 5 Experimental results on UCI data sets at 0% noise rate in the nonlinear case

Data set	FTSVM-CKA		TSVM		IFTSVM		CDFTSVM		AULSTSVM	
	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time
Breast	**97.0047** (1.9073)	0.000723	94.7038 (2.9359)	3.383631	96.5569 (2.1606)	2.734348	96.7059 (1.2970)	0.000681	95.8527 (2.6552)	0.067541
Credit	**85.5171** (3.2556)	0.000855	80.7169 (3.8612)	3.392196	85.2267 (4.4799)	2.772590	85.4997 (2.7202)	0.000811	85.5018 (3.6955)	0.073305
Diabetes	**75.7912** (4.5840)	0.000745	72.6623 (4.4292)	3.956387	75.4286 (4.1904)	3.350103	73.7047 (4.5288)	0.000704	74.7420 (3.0595)	0.087088
Echocardiogram	**86.5293** (7.8734)	0.000291	80.9799 (8.5119)	0.170087	85.5403 (7.8676)	0.121090	85.4670 (4.1668)	0.000204	85.5952 (7.3941)	0.003555
Heart	**84.7332** (8.5680)	0.000379	71.5640 (11.3853)	0.549028	81.6453 (6.1495)	0.441622	81.6207 (5.9443)	0.000371	83.3768 (4.5093)	0.012216
Horse	**85.0494** (6.7604)	0.000515	74.7301 (6.8950)	0.917470	83.4183 (5.3935)	0.776540	81.2775 (6.2493)	0.000561	83.4049 (4.6737)	0.019668
Musk	**89.5074** (4.5532)	0.000859	84.8449 (7.8697)	1.934506	83.8032 (4.0632)	1.472778	85.9071 (4.5354)	0.000798	82.3159 (4.2954)	0.069200
Parkinsons	84.6579 (8.5354)	0.000373	**87.7573** (7.2148)	0.272302	85.7895 (6.7144)	0.245093	80.4854 (7.1884)	0.000277	84.0789 (4.3133)	0.005885
Statlog_1	**67.8268** (0.4931)	0.000705	67.5356 (1.8455)	3.453156	66.6617 (1.9942)	2.680865	62.9074 (3.9559)	0.000946	**67.8268** (0.4931)	0.069843
Statlog_2	**84.8148** (5.6047)	0.000383	74.8148 (10.7088)	0.493400	82.2222 (9.3404)	0.392790	82.5926 (7.9522)	0.000322	**84.8148** (6.4031)	0.011583
South	**77.5000** (2.4187)	0.001048	71.3000 (1.3454)	6.552742	75.7000 (3.5791)	6.796388	74.7000 (5.7454)	0.001012	75.4000 (3.5270)	0.172600
Cardiotocography	**98.6721** (0.7620)	0.001731	97.5069 (1.1160)	10.954997	98.6715 (0.9256)	11.275718	98.3409 (1.1114)	0.001845	98.3395 (1.4398)	0.300394
Steel	**89.9056** (2.3025)	0.001767	80.3743 (10.5769)	12.523425	88.0952 (2.4379)	13.198699	88.4083 (3.6050)	0.001873	87.8521 (1.9234)	0.330912
Average rank	**1.23**		4.38		2.92		3.46		3.00	

Table 6 Experimental results on UCI data sets at 5% noise rate in the nonlinear case

Data set	FTSVM-CKA		TSVM		IFTSVM		CDFTSVM		AULSTSVM	
	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time
Breast	95.8487 (1.8643)	0.000794	90.4138 (4.6959)	3.363306	**96.5610** (1.7265)	2.747181	95.5548 (2.9048)	0.000682	96.1508 (3.9190)	0.069408
Credit	**85.5270** (4.7892)	0.000922	77.9798 (2.5817)	3.384324	85.5208 (4.9319)	2.780429	85.4887 (4.6770)	0.000879	85.5110 (3.9336)	0.075264
Diabetes	**74.8915** (5.4447)	0.001321	72.2659 (6.0070)	4.018853	74.7862 (2.9507)	3.319513	73.6928 (3.6900)	0.001051	74.6224 (4.7681)	0.082794
Echocardiogram	85.4670 (8.1963)	0.000322	81.0348 (9.6178)	0.135903	84.7711 (6.7067)	0.115251	84.6703 (8.4781)	0.000422	**85.5769** (7.1698)	0.003943
Heart	**83.2939** (4.2555)	0.000388	69.8399 (9.6885)	0.546859	81.6576 (5.8551)	0.430978	81.6092 (6.5263)	0.000384	82.9655 (6.3520)	0.010759
Horse	**83.9667** (4.7080)	0.000551	72.2740 (7.1017)	0.912759	82.6505 (6.0568)	0.707751	78.7925 (5.6948)	0.000496	82.1179 (5.9852)	0.019644
Musk	**88.6658** (5.8361)	0.000865	86.7642 (5.2497)	1.914219	83.1649 (4.5993)	1.439712	83.6126 (5.6292)	0.000758	82.7571 (5.2505)	0.072052
Parkinsons	**85.1579** (7.0644)	0.000278	82.7047 (6.2361)	0.265087	84.7632 (7.5334)	0.216296	80.5263 (6.6844)	0.000230	84.1813 (5.2514)	0.008053
Statlog_1	**67.8275** (0.3875)	0.000879	66.5169 (1.5823)	3.393569	64.3470 (2.3293)	2.748386	61.2716 (5.3114)	0.000721	67.8268 (0.4931)	0.073274
Statlog_2	**84.4444** (6.6769)	0.000338	75.5556 (8.9504)	0.460358	82.2222 (10.1835)	0.389271	81.8519 (8.3559)	0.000510	84.0741 (7.8178)	0.011704
South	**77.2000** (2.2716)	0.001036	71.3000 (1.4177)	6.772229	75.8000 (2.7857)	7.004796	74.5000 (3.5000)	0.001479	76.0000 (3.2863)	0.165918
Cardiotocography	**98.6701** (0.8492)	0.001850	91.3636 (2.9736)	10.959710	98.5048 (0.7246)	11.357362	98.0062 (1.4888)	0.001440	98.0082 (1.1820)	0.265927
Steel	**88.8733** (3.5245)	0.001866	73.9789 (5.8756)	12.969809	87.7034 (3.2534)	13.169697	87.6147 (2.8801)	0.001849	86.9098 (3.0075)	0.324171
Average rank	**1.23**		4.54		2.54		4.00		2.69	

Table 7 Experimental results on UCI data sets at 10% noise rate in the nonlinear case

Data set	FTSVM-CKA		TSVM		IFTSVM		CDFTSVM		AULSTSVM	
	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time	Accuracy (SD)	Time
Breast	**96.4263** (1.3141)	0.000783	89.4073 (5.3751)	3.326705	96.0018 (4.9367)	2.730439	95.7140 (2.7855)	0.000743	95.8550 (2.3736)	0.069136
Credit	**85.5334** (5.0756)	0.000883	75.3580 (3.2146)	3.331040	84.7877 (4.2297)	2.657230	85.5025 (1.6198)	0.000740	85.5224 (4.7341)	0.075140
Diabetes	73.8192 (3.2954)	0.000773	70.4477 (5.9845)	3.960804	**75.1350** (4.1036)	3.301821	73.4416 (3.6073)	0.000739	74.2139 (3.7999)	0.083741
Echocardiogram	**86.2271** (6.2056)	0.000259	78.6538 (6.4873)	0.132885	85.5403 (7.8676)	0.108059	84.8810 (5.4735)	0.000285	85.5952 (8.2354)	0.003345
Heart	**82.5567** (7.1730)	0.000362	63.9253 (6.5792)	0.535866	80.9319 (6.2129)	0.426975	80.4647 (8.6371)	0.000413	81.6141 (7.8908)	0.012508
Horse	**83.1279** (3.9253)	0.000483	70.0743 (5.8761)	0.867247	81.0064 (5.9868)	0.761977	78.5100 (3.2386)	0.000501	81.2221 (4.5279)	0.019152
Musk	**86.9947** (3.9187)	0.000818	80.4344 (5.7841)	1.869657	81.7287 (3.3540)	1.414734	81.1026 (6.7801)	0.001019	83.0221 (5.2259)	0.069324
Parkinsons	**85.7368** (7.3936)	0.000318	80.0146 (4.7676)	0.256111	83.7368 (8.7666)	0.217590	80.9883 (5.8028)	0.000488	83.5789 (5.1004)	0.008719
Statlog_1	**67.8275** (0.3875)	0.000796	63.9225 (3.3348)	3.366230	61.0090 (4.8769)	2.723839	60.2846 (5.3260)	0.000746	67.3920 (0.9235)	0.071853
Statlog_2	82.9630 (5.5060)	0.000476	71.4815 (9.3770)	0.459114	80.7407 (10.5799)	0.374241	81.8519 (8.0209)	0.000475	**84.0741** (7.2093)	0.011900
South	**77.5000** (3.0414)	0.000904	71.2000 (1.7205)	6.754338	75.2000 (3.7895)	7.023854	73.6000 (4.8826)	0.001154	75.4000 (4.9031)	0.166433
Cardiotocography	**98.4917** (4.0035)	0.001058	86.2927 (3.0356)	10.974690	98.4215 (1.0163)	11.699510	98.0069 (1.2939)	0.001072	98.0096 (1.0562)	0.259387
Steel	**89.1982** (1.6388)	0.001849	70.3375 (8.0582)	12.936773	87.0710 (3.1423)	13.425210	86.6754 (3.4107)	0.001916	86.5104 (2.2105)	0.323720
Average rank	**1.23**		4.85		2.77		3.85		2.31	

Table 8 Win-Tie-Loss accuracy comparison from Tables 2, 3 and 4

Noise rate	FTSVM-CKA vs. TSVM	FTSVM-CKA vs. IFTSVM	FTSVM-CKA vs. CDFTSVM	FTSVM-CKA vs. AULSTSVM
0%	12-0-1	13-0-0	13-0-0	12-0-1
5%	13-0-0	11-0-2	13-0-0	11-0-2
10%	13-0-0	12-0-1	12-0-1	8-2-3
Sum	38-0-1	36-0-3	38-0-1	31-2-6

Table 9 Win-Tie-Loss accuracy comparison from Tables 5, 6 and 7

Noise rate	FTSVM-CKA vs. TSVM	FTSVM-CKA vs. IFTSVM	FTSVM-CKA vs. CDFTSVM	FTSVM-CKA vs. AULSTSVM
0%	12-0-1	12-0-1	13-0-0	11-2-0
5%	13-0-0	12-0-1	13-0-0	11-0-2
10%	13-0-0	12-0-1	13-0-0	11-0-2
Sum	38-0-1	36-0-3	39-0-0	33-2-4

Table 10 The pairwise significant difference between the proposed FTSVM-CKA and existing algorithms in the linear case

Noise rate	FTSVM-CKA vs. TSVM	FTSVM-CKA vs. IFTSVM	FTSVM-CKA vs. CDFTSVM	FTSVM-CKA vs. AULSTSVM
0%	Yes	Yes	Yes	No
5%	Yes	No	Yes	Yes
10%	Yes	No	Yes	No

are significantly shorter than that of TSVM and IFTSVM. It shows that the FTSVM-CKA proposed in this paper has a faster training speed than other algorithms. This can be attributed to both FTSVM-CKA and CDFTSVM employ a coordinated descent strategy with shrinking by active set. However, the training time of FTSVM-CKA is slightly higher than that of CDFTSVM, mainly because CDFTSVM uses the simplest fuzzy membership calculation method.

Tables 5, 6 and 7 present the experimental results of FTSVM-CKA, TSVM, CDFTSVM, and IFTSVM under nonlinear conditions, and the noise rates are 0, 5, and 10% sequentially. In the case of different noise rates, the proposed model achieved the best results in 12, 11 and 12 of the 13 datasets, respectively. In the nonlinear case, the average ranks of accuracy of FTSVM-CKA at different noise rates are 1.23, which are better than the existing algorithms. It can be observed that the classification performance of FTSVM-CKA is better than that of other algorithms in nonlinear condition. Similar to the linear case, the computational time of FTSVM-CKA exhibits significantly shorter computational time compared to both TSVM and IFTSVM in the nonlinear case. The experimental results demonstrate that FTSVM-CKA outperforms other algorithms in terms of both classification performance and training speed.

In order to compare the proposed FTSVM-CKA with other algorithms in terms of the classification performance, we utilize the Win-Tie-Loss (Xu et al. 2016) statistical analysis to examine the datasets and record the number of datasets in which FTSVM-CKA outperforms, achieves equal performance, or performs worse than other algorithms in both linear and nonlinear cases. The corresponding results are presented in Tables 8 and 9. One can observe that FTSVM-CKA give better performance than other methods on the majority of datasets. Furthermore, it is evident that FTSVM-CKA exhibits a clear advantage in the presence of noise.

We perform the Friedman test with the post-hoc test to prove the statistical significance of the proposed FTSVM-CKA for generalization performance. The Friedman test uses

Table 11 The pairwise significant difference between the proposed FTSVM-CKA and existing algorithms in the nonlinear case

Noise rate	FTSVM-CKA vs. TSVM	FTSVM-CKA vs. IFTSVM	FTSVM-CKA vs. CDFTSVM	FTSVM-CKA vs. AULSTSVM
0%	Yes	Yes	Yes	Yes
5%	Yes	No	Yes	Yes
10%	Yes	Yes	Yes	No

Table 12 Experimental results with machine learning algorithms at 0% noise rate

Data set	CatBoost	LightGBM	RF	XGBoost	SVM	LinearFTSVM-CKA	NonlinearFTSVM-CKA
Heart	81.2298	80.5287	79.1724	77.0919	81.9655	83.6864	**84.7332**
Statlog_1	67.6364	65.6117	62.6939	64.8699	67.5362	**67.8275**	67.8268
Statlog_2	80.6695	80.6695	80.2706	81.0398	80.7407	**85.1852**	84.8148
Abalone	61.6920	64.4633	77.8754	76.2637	77.3993	**77.9862**	77.5090
Statlog_3	88.1946	86.8054	95.6487	95.8818	95.4469	93.5702	**95.9048**

Table 13 Experimental results with machine learning algorithms at 5% noise rate

Data set	CatBoost	LightGBM	RF	XGBoost	SVM	LinearFTSVM-CKA	NonlinearFTSVM-CKA
Heart	80.8850	79.1724	78.8045	78.1494	80.5862	81.8530	**83.2939**
Statlog_1	67.3402	63.8662	62.5575	63.7127	67.3913	67.8268	**67.8275**
Statlog_2	81.7664	79.9145	78.8034	78.8176	81.1111	**84.8148**	84.4444
Abalone	61.3671	64.9541	77.5457	74.6520	77.0329	**78.3156**	77.3636
Statlog_3	84.4600	83.7936	95.1095	95.3378	95.3536	93.8570	**95.7445**

Table 14 Experimental results with machine learning algorithms at 10% noise rate

Data set	CatBoost	LightGBM	RF	XGBoost	SVM	LinearFTSVM-CKA	NonlinearFTSVM-CKA
Heart	79.8735	78.8160	75.7356	73.3103	80.2874	73.8391	**82.5567**
Statlog_1	65.3112	61.6880	62.4062	61.1082	67.1014	67.8268	**67.8275**
Statlog_2	80.9972	77.6638	79.5726	77.6638	80.3704	**84.0741**	82.9630
Abalone	63.2421	66.5530	76.3369	74.7619	76.8864	**77.6546**	77.4738
Statlog_3	84.2856	83.5072	95.3177	95.3222	95.1982	93.4139	**95.5276**

the average ranks of the algorithms. The average ranks of all algorithms in different situations are recorded in Tables 2, 3, 4, 5, 6 and 7. Under the null hypothesis that all the algorithms have equal ranks, the Friedman statistics which are distributed with χ_F^2 with $\kappa - 1$ degree of freedom are calculated , where κ is the number of the algorithms. The performance of two algorithms is significantly different if their average ranks differ by at least the critical difference defined by $CD = q_\alpha \sqrt{\frac{\kappa(\kappa+1)}{6N}}$, where N is the number of datasets and q_α is computed by using the Studentized range statistic. The critical difference for our case at $\alpha = 0.10$ level of significance level is $CD = 2.241\sqrt{\frac{5(5+1)}{6\times 13}} \approx 1.39$. Tables 10 and 11 show the pairwise significant difference between the algorithms in linear and nonlinear cases, respectively. It can be found that the proposed FTSVM-CKA is significantly different from most of the algorithms.

In addition, FTSVM-CKA is compared with some existing machine learning algorithms whose effectiveness has been recognized, including CatBoost, LightGBM, XGBoost, SVM, RF. The experiment was performed on 5 UCI datasets with 10-fold cross-validation. These comparison algorithms are all set with default parameters. Tables 12, 13 and 14 show the classification accuracy results at 0, 5, and 10% noise rates, respectively. We can find that FTSVM-CKA achieves optimal classification results in all cases. This shows that the classification performance of FTSVM-CKA is significantly better than that of some existing classical and effective machine learning algorithms. It is worth noting that FTSVM-CKA is superior to traditional SVM in both generalization performance and training speed. One fact is that SVM cannot distinguish the importance of different samples for classification, which makes it sensitive to noise. The proposed FTSVM-CKA utilizes the CKA method to derive a heuristic function for calculating the dependency between samples and their corresponding labels, which assigns fuzzy membership to different samples. This effectively identifies noise and reduces the negative impact on classification.

5 Conclusion

To address the problem that traditional TSVM is sensitive to noise, we proposed a novel and efficient fuzzy twin support vector machine based on centered kernel alignment, termed as FTSVM-CKA. FTSVM-CKA utilizes the CKA method which incorporates the idea of feature weighting to assigns

fuzzy membership to different samples. We conducted experiments on an artificial dataset and 15 UCI datasets. Noise is added to the original data set to verify the noise robustness of the proposed FTSVM-CKA. The experimental results demonstrate that FTSVM-CKA outperforms several existing learning models and exhibits excellent classification performance. Statistical tests on experimental results confirm the significance of the proposed algorithm. Nevertheless, FTSVM-CKA does not take into account the class imbalance data set. Our future work is to extend FTSVM-CKA to class imbalance learning.

Acknowledgements This paper would like to thank the editors and the anonymous referees for their professional comments, which improved the quality of the manuscript.

Funding This work was supported in part by the National Natural Science Foundation of China (Nos. 12271211, 12071179), the National Natural Science Foundation of Fujian Province (Nos. 2021J01861, 2020J01710), the Youth Innovation Fund of Xiamen City (3502Z20206020), the Open Fund of Digital Fujian Big Data Modeling and Intelligent Computing Institute, Pre-Research Fund of Jimei University.

Data availibility This paper uses the UCI Machine Learning Repository, which is publicly available on the Internet. As follows: https://archive.ics.uci.edu/

Declarations

Conflict of interest All authors declare that they have no conflict of interest.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

Borah P, Gupta D (2022) Affinity and transformed class probability-based fuzzy least squares support vector machines. Fuzzy Sets Syst 443:203–235

Breiman L (2001) Random forests. Mach Learn 45:5–32

Cárdenas-Peña D, Collazos-Huertas D, Castellanos-Dominguez G et al (2016) Centered kernel alignment enhancing neural network pretraining for MRI-based dementia diagnosis. Comput Math Methods Med. https://doi.org/10.1155/2016/9523849

Chang CC, Lin CJ (2011) Libsvm: a library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2(3):1–27

Chang KW, Hsieh CJ, Lin CJ (2008) Coordinate descent method for large-scale l2-loss linear support vector machines. J Mach Learn Res 9(7):1369–1398

Chen SG, Wu XJ (2018) A new fuzzy twin support vector machine for pattern classification. Int J Mach Learn Cybern 9:1553–1564

Chen T, He T, Benesty M, et al (2015) Xgboost: extreme gradient boosting. R package version 04-2 1(4):1–4

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

Cortes C, Mohri M, Rostamizadeh A (2012) Algorithms for learning kernels based on centered alignment. J Mach Learn Res 13(1):795–828

Dua D, Graff C, et al (2017) Uci machine learning repository. http://archive.ics.uci.edu/ml 7(1)

Francis LM, Sreenath N (2022) Robust scene text recognition: using manifold regularized twin-support vector machine. J King Saud Univ Comput Inf Sci 34(3):589–604

Ganaie M, Tanveer M, Initiative ADN et al (2021) Fuzzy least squares projection twin support vector machines for class imbalance learning. Appl Soft Comput 113:107933

Ganaie M, Kumari A, Malik AK et al (2022) Eeg signal classification using improved intuitionistic fuzzy twin support vector machines. Neural Comput Appl 36:163–179

Ganaie M, Tanveer M, Lin CT (2022) Large-scale fuzzy least squares twin svms for class imbalance learning. IEEE Trans Fuzzy Syst 30(11):4815–4827

Ganaie M, Kumari A, Girard A et al (2023) Diagnosis of Alzheimer's disease via intuitionistic fuzzy least squares twin svm. Appl Soft Comput 149:110899

Ganaie M, Tanveer M, Suganthan PN (2020) Regularized robust fuzzy least squares twin support vector machine for class imbalance learning. In: 2020 international joint conference on neural networks (IJCNN), IEEE, pp 1–8

Gao BB, Wang JJ, Wang Y, et al (2015) Coordinate descent fuzzy twin support vector machine for classification. In: 2015 IEEE 14th international conference on machine learning and applications (ICMLA), IEEE, pp 7–12

Gupta D, Richhariya B, Borah P (2019) A fuzzy twin support vector machine based on information entropy for class imbalance learning. Neural Comput Appl 31(11):7153–7164

Han J, Pei J, Tong H (2022) Data mining: concepts and techniques. Morgan Kaufmann, San Francisco

Hao PY, Kung CF, Chang CY et al (2021) Predicting stock price trends based on financial news articles and using a novel twin support vector machine with fuzzy hyperplane. Appl Soft Comput 98:106806

Hazarika BB, Gupta D, Kumar B (2023) Eeg signal classification using a novel Universum-based twin parametric-margin support vector machine. Cogn Comput 16:2047–2062

Ju H, Qiang W, Jing L (2021) A novel interval-valued fuzzy multiple twin support vector machine. Iran J Fuzzy Syst 18(2):93–107

Ke G, Meng Q, Finley T, et al (2017) Lightgbm: A highly efficient gradient boosting decision tree. In: Advances in neural information processing systems 30

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

Kung CF, Hao PY (2023) Fuzzy least squares support vector machine with fuzzy hyperplane. Neural Process Lett 55(6):7415–7446

Liang Z, Zhang L (2022) Intuitionistic fuzzy twin support vector machines with the insensitive pinball loss. Appl Soft Comput 115:108231

Lu Y, Wang L, Lu J et al (2014) Multiple kernel clustering based on centered kernel alignment. Pattern Recogn 47(11):3656–3664

Prokhorenkova L, Gusev G, Vorobev A, et al (2018) Catboost: unbiased boosting with categorical features. In: Advances in neural information processing systems 31

Rezvani S, Wang X (2021) Class imbalance learning using fuzzy art and intuitionistic fuzzy twin support vector machines. Inf Sci 578:659–682

Rezvani S, Wang X (2022) Intuitionistic fuzzy twin support vector machines for imbalanced data. Neurocomputing 507:16–25

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

Richhariya B, Tanveer M (2018) A robust fuzzy least squares twin support vector machine for class imbalance learning. Appl Soft Comput 71:418–432

Richhariya B, Tanveer M, Initiative ADN (2021) A fuzzy Universum least squares twin support vector machine (fulstsvm). Neural Comput Appl 34:11411–11422

Richhariya B, Tanveer M, of Mathematics ADNID (2021) An efficient angle-based universum least squares twin support vector machine for classification. ACM Trans Internet Technol 21(3):1–24

Ripley BD (2007) Pattern recognition and neural networks. Cambridge University Press, Cambridge

Shao YH, Deng NY (2012) A coordinate descent margin based-twin support vector machine for classification. Neural Netw 25:114–121

Sharma R, Goel T, Tanveer M et al (2022) FDN-ADNET: Fuzzy LS-TWSVM based deep learning network for prognosis of the Alzheimer's disease using the sagittal plane of mri scans. Appl Soft Comput 115:108099

Tanveer M, Ganaie M, Bhattacharjee A et al (2022) Intuitionistic fuzzy weighted least squares twin svms. IEEE Trans Cybern 53:4400–4409

Tanveer M, Rajani T, Rastogi R et al (2022) Comprehensive review on twin support vector machines. Ann Oper Res. https://doi.org/10.1007/s10479-022-04575-w

Wang T, Qiu Y, Hua J (2020) Centered kernel alignment inspired fuzzy support vector machine. Fuzzy Sets Syst 394:110–123

Xie X, Li Y, Sun S (2023a) Deep multi-view multiclass twin support vector machines. Inf Fusion 91:80–92

Xie X, Sun F, Qian J et al (2023b) Laplacian lp norm least squares twin support vector machine. Pattern Recogn 136:109192

Xu Y, Yang Z, Pan X (2016) A novel twin support-vector machine with pinball loss. IEEE Trans Neural Netw Learn Syst 28(2):359–370

Yu H, Sun C, Yang X et al (2019) Fuzzy support vector machine with relative density information for classifying imbalanced data. IEEE Trans Fuzzy Syst 27(12):2353–2367