AIMS Mathematics, 11(4): 10100–10132.
DOI: 10.3934/math.2026417
Received: 24 December 2025
Revised: 27 February 2026
Accepted: 28 February 2026
Published: 14 April 2026

https://www.aimspress.com/journal/Math

Research article

Robust average-weighted twin extreme learning machine for pattern classification

Yanrong Ma[1], **Jun Ma**[2,*] **and Bao Ma**[2]

[1] School of Preparatory Education, North Minzu University, Yinchuan 750021, China
[2] School of Mathematics and Information Sciences, North Minzu University, Yinchuan Ningxia 750021, PR China

* **Correspondence:** Email: jun_ma1990@nmu.edu.cn.

Abstract: This paper proposes a novel robust twin extreme learning machine (RTELM) for binary classification. To enhance its performance and robustness, we introduce two key techniques: (1) An average weight technique that assigns larger weights to data points near the class center and smaller weights to those near the boundary, leveraging the intra-class distribution; and (2) an improved pre-selection point technique that selects only the top-u sorted data points to mitigate the impact of noise and outliers. Furthermore, we extend RTELM by incorporating a manifold regularization term, resulting in the Lap-RTELM framework, which enhances data discriminability. Extensive experiments validate that both RTELM and Lap-RTELM achieve superior classification performance, robustness, and stability compared to traditional methods.

Keywords: average weight; pre-selection; classification; robust; manifold regularization
Mathematics Subject Classification: 68T10, 91C20

1. Introduction

As a classical and efficient machine learning algorithm, the extreme learning machine (ELM) is widely used in many fields, such as traffic flow forecasting [1], brain tumor detection [2], and wheat type discrimination [3]. ELM was first proposed by Huang et al. [4] based on the traditional gradient descent algorithm, for training single hidden layer feedforward neural networks (SLFNs). ELM can achieve fast learning speed and good generalization performance by randomly selecting the weights and biases between the input layer and the hidden layer. In addition, due to the randomness of parameter selection, ELM is more likely to achieve the global optimal solution. Later, Huang et al. [5] constructed a unified learning framework with a wide range of feature mappings using ELM. This framework can be directly applied to regression and classification tasks. In recent years, many

variants based on ELM have emerged. Zong et al. [6] proposed the weighted ELM (WELM), which is designed to address the issue of class imbalanced data and can be directly applied to classification tasks. Wan et al. proposed the twin ELM (TELM) [8] based on the twin support vector machine (TSVM) [7] algorithm, which is designed to address binary classification problems. TELM classifies data by learning two non-parallel hyperplanes. For each hyperplane, TELM minimizes the distance to the data points of one class while maximizing the distance to the other class. The experimental results show that TELM has obvious advantages when dealing with multi-class classification problems and large datasets. Rastogi et al. [9] introduced the least-squares method to solve the weight matrix between the hidden layer and the output layer and proposed the least-squares ELM (LS-TELM) version. LS-TELM directly solves linear equations by transforming inequality constraints into equality constraints, which greatly reduces the computational cost.

Semi-supervised learning (SSL) is a technique proposed by Hady et al. [10] that aims to develop more reliable models by utilizing a small amount of labeled data and a large amount of unlabeled data. One graph-based SSL method is manifold regularization, which establishes edges connecting labeled and unlabeled data points and creates a graph representing the similarity between samples from these edges [11]. To better capture the distribution of data within the manifold space, Belkin et al. [12] proposed a regularization method that utilizes the graph Laplacian operator. This method enhances the discriminability of data points by constructing the manifold structure of the data [13].

For the SSL problem in ELM, the randomness in the selection of weights and biases between the input layer and the hidden layer may lead to a full column rank of the hidden layer output matrix, impacting the efficacy of ELM. [14, 15]. To address this issue, Li et al. [16] introduced a regularized classification method for ELM by redefining the objective function to include loss, smoothness, intra-class, and inter-class regularization terms. However, this method introduces a high number of parameters and is challenging to adjust. Liu et al. [17] proposed a semi-supervised ELM based on manifold regularization, but neither of these methods penalizes the weight norm of the hidden layer output matrix, leading to high algorithm complexity. Based on this, Liu et al. [18] extended the manifold regularization framework based on ELM, providing a unified solution for different practical applications while retaining the advantages of manifold regularization. Subsequently, Li et al. proposed Lap-TELM [19], which simultaneously trains two related and paired semi-supervised ELMs with two nonparallel separating planes for the final classification, showcasing that SSL and supervised learning can form a unified learning framework.

Robust learning is also an important research direction in ELM as the presence of noise and outliers can directly impact model performance. Researchers have primarily focused on the following two aspects to enhance the robustness of ELM. First, the randomness in weight and bias selection between the input and hidden layers of ELM may lead to instability [20]. To solve this problem, Incremental ELM (IELM), improved IELM (IIELM), and an enhanced version of IIELM have been proposed by researchers [21–23], involving the addition of hidden nodes, recalculating of output weights of hidden nodes, and iterative updating of output weights. Second, the utilization of L_2-norm distance measurement and the hinge loss function in ELM may influence classification results [24–26]. These methods are recognized as insensitive to noise and outliers. To solve this problem, references [27–29] replaced the hinge loss function with ξ-insensitive pinball loss, capped L_1-norm, and adaptive capped $L_\theta(\varepsilon)$-loss, respectively. References [30–32] replaced the L_2-norm distance measurement with L_1-norm, capped L_1-norm, and capped $L_{2,p}$-norm, respectively. Recent

advances in robust loss functions have shown promising results for handling mixed noise environments. In particular, Sheng et al. [33, 34] proposed novel hybrid loss functions that combine the advantages of different robust estimators, achieving superior performance when dealing with both Gaussian noise and impulsive noise in matrix completion tasks. These functions leverage explicit regularizers to maintain convexity while providing strong outlier rejection capabilities.

A key limitation of traditional convex loss functions, such as L2, L1, and Huber, is their unbounded nature. While robust estimators like L1 and Huber can be implemented via iteratively reweighted least squares (IRLS), where the weight assigned to a sample diminishes with the residual magnitude, the underlying loss value itself continues to increase as the magnitude of the outlier grows. For L1 and Huber, this increase is linear, meaning that extreme outliers still exert a proportionally large influence on the overall cost being minimized. This sensitivity to large-magnitude outliers is a primary motivation for employing non-convex, redescending loss functions [35, 36]. These functions are designed so that beyond a certain threshold, the loss value stabilizes or even decreases, thereby "redescending." This property effectively caps the influence of severely corrupted measurements, allowing the model to discard their impact entirely. Our proposed framework builds upon this principle to achieve greater robustness in matrix completion tasks. Recently, Xu et al. [37] proposed the Twin Depth SVM (TDSVM), which introduces the average depth technique in TSVM to strengthen the center and weaken the margin. In addition, Yan et al. [38] proposed a novel robust SVM classifier with feature mapping (FMSVM), utilizing a novel pre-selection technique to achieve robustness. Both methods make full use of the data distribution within the dataset to construct robust classifiers.

Inspired by the above reference, we propose a novel and robust TELM framework based on average weight. Our first framework is called RTELM. In this framework, we introduce the concept of average weight based on TDSVM and apply it to the proposed RTELM. Specifically, we first sort the data points in the dataset in ascending and descending order, then calculate the weight and average weight of each data point. We assign larger average weight values to the class center data points that have a positive effect on constructing the hyperplane. Correspondingly, we assign smaller average weight values to the class margin data points that have a smaller effect on constructing the hyperplane. In addition, we propose an improved pre-selection technique for robust point selection based on FMSVM. This technique aims to eliminate or reduce the impact of noise and outliers on the model. After obtaining the descending order of the data points, we only select the top-u data points with larger average weight values. This can eliminate or weaken the influence of noise and outliers on the model. If the outliers are located at the extremes of the distribution, their ascending order values, weight values, and average weight values will be small after sorting the data points. Then, when we select the top-u data points with larger average weight values, the outliers will be eliminated. If the outliers are at the center of the class, by performing the same operation as above, we will obtain a larger average weight value for that point. Then, when we select the top-u data points with larger average weight values, the influence of the outlier on constructing the hyperplane will be weakened. Therefore, we can fully utilize the data distribution to construct a classifier that has good classification performance and robustness. Our second framework is called Lap-RTELM. In order to enhance the discriminability of data points, we incorporate manifold regularization terms into RTELM and obtain Lap-RTELM. While retaining the advantages of RTELM, Lap-RTELM can more accurately represent the data distribution in the manifold space and achieve the objective of constructing a classifier with excellent performance using

a limited number of labeled data points and a substantial amount of unlabeled data points. The main contributions of this paper can be summarized as follows.

(1) We propose an average weight method based on ELM, which assigns different weight values to different data points in the dataset according to their distribution. We assign larger weights to data points that are closer to the class center and smaller weights to data points that are closer to the class edge. This approach aims to construct a more accurate classifier.

(2) To further enhance the robustness of the proposed model, we proposed an improved pre-selection point technique. This technique utilizes the distribution of data points in the dataset and selects only the top-u data points with larger sequence values after sorting them in descending order. This technique is embedded in both of our proposed frameworks.

(3) To fully utilize the distribution of the data, we extend the proposed model to SSL by using manifold regularization.

(4) We conducted numerical experiments on various artificial datasets and UCI benchmark datasets. The experimental results show that our two proposed have satisfactory classification accuracy and are highly competitive compared to some well-known algorithms.

The remainder of this paper is organized as follows. In the next section, we provide a brief review of the TELM and Lap-TELM models. Section 3 introduces the improved pre-selection technique, the two average weight-based frameworks, and their solutions. In Sections 4–6, we present our numerical experiments and summarize our work, respectively.

2. Related works

2.1. TELM

For binary classification problems, we consider a training set $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$, where $\mathbf{x}_i \in \mathbb{R}^n$ is the input vector and $y_i \in \{+1, -1\}$ is the corresponding class label. Let m_1 and m_2 denote the number of positive class samples ($y_i = +1$) and negative class samples ($y_i = -1$), respectively, such that $m = m_1 + m_2$. The hidden layer output matrices for positive and negative class samples are denoted as $\mathbf{H}_1 \in \mathbb{R}^{m_1 \times L}$ and $\mathbf{H}_2 \in \mathbb{R}^{m_2 \times L}$, where L is the number of hidden nodes.

The classification principle of TELM [8] is to construct two non-parallel hyperplanes:

$$f_1(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta}_1 = 0 \quad \text{and} \quad f_2(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta}_2 = 0, \tag{2.1}$$

where $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_L(\mathbf{x})] \in \mathbb{R}^{1 \times L}$ represents the row vector of hidden layer outputs for input $\mathbf{x}$, with $h_i(\mathbf{x}) = G(\boldsymbol{\omega}_i, b_i, \mathbf{x})$ being the activation function, $\boldsymbol{\omega}_i \in \mathbb{R}^n$ the input weight vector, and $b_i \in \mathbb{R}$ the bias for the i-th hidden node. The vectors $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \in \mathbb{R}^{L \times 1}$ are the output weight vectors connecting the hidden layer to the output layer.

TELM aims to make each hyperplane close to samples of one class while being far from samples of the other class. This leads to the following optimization problems:

$$\begin{aligned} &\min_{\boldsymbol{\beta}_1, \boldsymbol{\xi}_1} \frac{1}{2}\|\mathbf{H}_1\boldsymbol{\beta}_1\|_2^2 + c_1\mathbf{e}_2^T\boldsymbol{\xi}_1 \\ &\text{s.t.} \quad -\mathbf{H}_2\boldsymbol{\beta}_1 + \boldsymbol{\xi}_1 \geq \mathbf{e}_2, \quad \boldsymbol{\xi}_1 \geq \mathbf{0} \end{aligned} \tag{2.2}$$

and

$$\min_{\boldsymbol{\beta}_2, \boldsymbol{\xi}_2} \frac{1}{2}\|\mathbf{H}_2\boldsymbol{\beta}_2\|_2^2 + c_2\mathbf{e}_1^T\boldsymbol{\xi}_2$$

$$\text{s.t.} \quad -\mathbf{H}_1\boldsymbol{\beta}_2 + \boldsymbol{\xi}_2 \geq \mathbf{e}_1, \quad \boldsymbol{\xi}_2 \geq \mathbf{0}, \tag{2.3}$$

where $\mathbf{e}_1 \in \mathbb{R}^{m_1}$ and $\mathbf{e}_2 \in \mathbb{R}^{m_2}$ are vectors of all ones, $\boldsymbol{\xi}_1 \in \mathbb{R}^{m_2}$ and $\boldsymbol{\xi}_2 \in \mathbb{R}^{m_1}$ are slack vectors, and $c_1, c_2 > 0$ are trade-off parameters.

Using Lagrange multipliers and the Karush-Kuhn-Tucker (KKT) conditions, the dual form can be expressed as:

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{H}_2(\mathbf{H}_1^T\mathbf{H}_1 + \epsilon\mathbf{I})^{-1}\mathbf{H}_2^T\boldsymbol{\alpha} - \mathbf{e}_2^T\boldsymbol{\alpha} \\ &\text{s.t.} \quad 0 \leq \alpha_i \leq c_1, \quad i = 1, \ldots, m_2 \end{aligned} \tag{2.4}$$

and

$$\begin{aligned} &\min_{\boldsymbol{\gamma}} \frac{1}{2}\boldsymbol{\gamma}^T\mathbf{H}_1(\mathbf{H}_2^T\mathbf{H}_2 + \epsilon\mathbf{I})^{-1}\mathbf{H}_1^T\boldsymbol{\gamma} - \mathbf{e}_1^T\boldsymbol{\gamma} \\ &\text{s.t.} \quad 0 \leq \gamma_i \leq c_2, \quad i = 1, \ldots, m_1, \end{aligned} \tag{2.5}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{m_2}$ and $\boldsymbol{\gamma} \in \mathbb{R}^{m_1}$ are Lagrange multiplier vectors, $\epsilon > 0$ is a small regularization parameter, and $\mathbf{I} \in \mathbb{R}^{L\times L}$ is an identity matrix. The solutions are:

$$\boldsymbol{\beta}_1 = -(\mathbf{H}_1^T\mathbf{H}_1 + \epsilon\mathbf{I})^{-1}\mathbf{H}_2^T\boldsymbol{\alpha}, \quad \boldsymbol{\beta}_2 = -(\mathbf{H}_2^T\mathbf{H}_2 + \epsilon\mathbf{I})^{-1}\mathbf{H}_1^T\boldsymbol{\gamma}. \tag{2.6}$$

For a new sample $\mathbf{x}^*$, the decision function is:

$$f(\mathbf{x}^*) = \arg\min_{r\in\{1,2\}} d_r(\mathbf{x}^*) = \arg\min_{r\in\{1,2\}} |\mathbf{h}(\mathbf{x}^*)\boldsymbol{\beta}_r|, \tag{2.7}$$

where $|\cdot|$ represents the perpendicular distance from $\mathbf{x}^*$ to the hyperplane $\boldsymbol{\beta}_r$.

2.2. Lap-TELM

We assume that the training set is $\mathcal{T} = \mathcal{T}_l \cup \mathcal{T}_u = \{\mathbf{x}_i, y_i\}_{i=1}^{l} \cup \{\mathbf{x}_i, y_i\}_{i=l+1}^{l+u}$, where $l+u = n$, $\mathcal{T}_l = \{\mathbf{x}_i\}_{i=1}^{l} \in \mathbb{R}^{l\times n}$ represents the labeled data with corresponding labels $\mathcal{Y}_l = \{y_i\}_{i=1}^{l} \in \{-1, 1\}$, and $\mathcal{T}_u = \{\mathbf{x}_i\}_{i=1}^{u} \in \mathbb{R}^{u\times n}$ represents the unlabeled data with corresponding labels $\mathcal{Y}_u = 0$. By fully utilizing the geometric structural features of unlabeled samples, the mathematical formulation of the primary optimization problem for Lap-TELM [19] can be written as:

$$\begin{aligned} &\min_{\beta_1,\xi_1} \frac{1}{2}\|\mathbf{H}_1\beta_1\|_2^2 + c_1\mathbf{e}_2^T\xi_1 + c_3\beta_1^T\mathbf{H}^T\mathbf{L}\mathbf{H}\beta_1 \\ &\text{s.t.} \quad -\mathbf{H}_2\beta_1 + \xi_1 \geq \mathbf{e}_2, \xi_1 \geq 0 \end{aligned} \tag{2.8}$$

and

$$\begin{aligned} &\min_{\beta_2,\xi_2} \frac{1}{2}\|\mathbf{H}_2\beta_2\|_2^2 + c_2\mathbf{e}_1^T\xi_2 + c_4\beta_2^T\mathbf{H}^T\mathbf{L}\mathbf{H}\beta_2 \\ &\text{s.t.} \quad -\mathbf{H}_1\beta_2 + \xi_2 \geq \mathbf{e}_1, \xi_2 \geq 0, \end{aligned} \tag{2.9}$$

where c_3 and c_4 are both positive trade-off parameters, and $\mathbf{H} \in \mathbb{R}^{n\times L}$ represents all samples. Here, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian matrix, where $\mathbf{D} = diag(\sum_{j=1}^{n} W_{1j}, \sum_{j=1}^{n} W_{2j}, ..., \sum_{j=1}^{n} W_{nj})$, and $\mathbf{W}$

represents the adjacency matrix of W_i and W_j. Besides, the other symbols have the same meanings as the formulas (2.2) and (2.3) in TELM. Furthermore, the dual problem of Lap-TELM can be expressed as follows:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^T \mathbf{H}_2(\mathbf{H}_1^T\mathbf{H}_1 + \epsilon\mathbf{I} + c_3\mathbf{H}^T\mathbf{L}\mathbf{H})^{-1}\mathbf{H}_2^T\alpha - \mathbf{e}_2^T\alpha \\ \text{s.t.} \quad & 0 \le \alpha_i \le c_1,\ i = 1, \cdots, m_2 \end{aligned} \tag{2.10}$$

and

$$\begin{aligned} \min_{\gamma} \quad & \frac{1}{2}\gamma^T \mathbf{H}_1(\mathbf{H}_2^T\mathbf{H}_2 + \epsilon\mathbf{I} + c_4\mathbf{H}^T\mathbf{L}\mathbf{H})^{-1}\mathbf{H}_1^T\gamma - \mathbf{e}_1^T\gamma \\ \text{s.t.} \quad & 0 \le \gamma_i \le c_2,\ i = 1, \cdots, m_1. \end{aligned} \tag{2.11}$$

By solving the above equation, we can obtain $\beta_1 = -(\mathbf{H}_1^T\mathbf{H}_1 + \epsilon\mathbf{I} + c_3\mathbf{H}^T\mathbf{L}\mathbf{H})^{-1}\mathbf{H}_2^T\alpha$ and $\beta_2 = -(\mathbf{H}_2^T\mathbf{H}_2 + \epsilon\mathbf{I} + c_4\mathbf{H}^T\mathbf{L}\mathbf{H})^{-1}\mathbf{H}_1^T\gamma$. For a new sample, its classification decision function is similar to (2.7).

3. Main contributions

3.1. Motivation

In some classical TELM and its variants, the performance of the model is primarily improved by modifying the loss function and misclassification terms. However, this approach does not take into account the distribution of data in the dataset. In other words, it treats all data points as equally important. This may lead to the problem that not all data points in the dataset contribute positively to the classification hyperplane, which can degrade the classification performance or the model's robustness. Therefore, inspired by references such as [8, 29, 37, 38], we propose a novel supervised and semi-supervised framework based on TELM. This framework utilizes the average weight technique and an improved pre-selection technique in order to fully exploit the distribution information of the data. The goal is to enhance the data center while weakening the data edges. In the following, we will introduce these two techniques one by one.

Firstly, to enhance the classification performance of the model, we will assign higher weights to the data points that are in the center of their respective classes. Conversely, we will assign lower weights to data points that are farther away from the hyperplane. Specifically, for a set of data $\mathbf{x} = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$, we first sort it in ascending order to obtain $x(1), x(2), \cdots, x(n)$, and use $r_{x_i|x}$ to represent its rank in ascending order. Correspondingly, $n + 1 - r_{x_i|x}$ is used to represent its rank in descending order. For an n-dimensional dataset with m data points, the weight and average weight can be expressed as follows:

$$w_{x_i|x} = \min\{r_{x_i|x}, n + 1 - r_{x_i|x}\}, i = 1, 2, \cdots, n \tag{3.1}$$

and

$$\bar{w}_{x_i|x} = \frac{1}{n}\sum_{j=1}^{n} w_{x_{ij}|x_j}, i = 1, 2, \cdots, m. \tag{3.2}$$

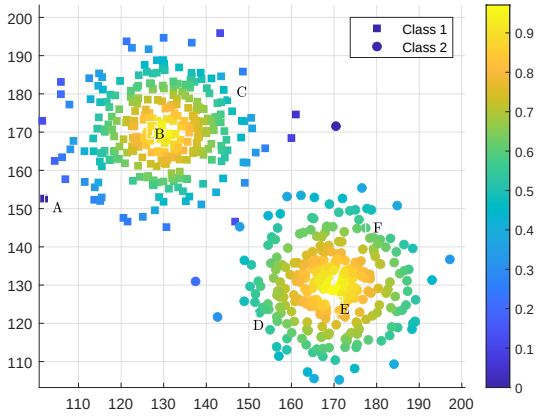


Figure 1. Average weight.

Figure 1 shows an illustration of the average weight of two classes with randomly generated data points, each with a size of 300. Table 1 shows an example of the method used to calculate the average weight. Of these, Table 1(a) contains the data points for the parts; Table 1(b) displays the results in ascending order, and Table 1(c) shows the weight and average weight calculated using Eqs (3.1) and (3.2). For point A, the ascending order is $r_{x_1|x_1} = 2$, and we can calculate its weight with respect to X_1 using Eq (3.1), where $n = 600$, and $w_{x_1|x_1} = \min\{2, 599\} = 2$. Similarly, we can determine its weight with respect to X_2. Then, using Eq (3.2), we can calculate the average weight for point A as $\bar{w}_{x_1|x} = \frac{2+295}{2} = 148.5$.

Table 1. Examples used to explain average weight.

(a) Data points.

	X_1	X_2
A	101.9	152.6
B	129.2	170.5
C	144.1	181.1
D	152.5	122.8
E	171.7	127.6
F	180.9	141.9

(b) Ascending order.

	$r_{x_1\|x_1}$	$r_{x_2\|x_2}$
A	2	306
B	162	409
C	276	557
D	293	39
E	412	155
F	554	271

(c) Weight.

	$w_{x_1\|x_1}$	$w_{x_2\|x_2}$	$\bar{w}_{x_i\|x}$
A	2	295	148.5
B	162	192	177
C	268	44	156
D	293	39	166
E	189	155	172
F	47	271	159

Thus, the following conclusion can be drawn from Table 1(c) and Figure 1 above. If the average weight of a data point is larger, its distribution is closer to the center of its class. Conversely, if the average weight of a data point is smaller, its distribution is farther away from the class center. Therefore, we can construct more robust classifiers based on the average weight technique.

Firstly, the formula for the pre-selection point technique is provided in [38] as follows:

$$d_i = 1 - \frac{|d_{i+} - d_{i-}|}{\max(|d_{i+} - d_{i-}|)}, i = 1, 2, ..., m. \tag{3.3}$$

where d_{i+} and d_{i-} represent the distances from $\mathbf{x}_i$ to the centers of the positive and negative classes, respectively. When $|d_{i+} - d_{i-}| = \max(|d_{i+} - d_{i-}|)$ for a class center data point, according to (3.3) we have

$d_i = 0$. This point may be considered an outlier, so the data points selected by this method may not be the most reasonable, and their robustness may be limited. Therefore, to enhance the overall robustness of our proposed model, we propose an improved pre-selection technique that selects the top-u data points with higher average weight values. Specifically, after obtaining the average weight values of the data points, we sort them in descending order and select the top-u data points with larger average weight values.

By utilizing the above-mentioned average weight technique and improved pre-selection technique, we can fully utilize the distribution of data points in the dataset, directly enhancing the class center and weakening the class margin, while minimizing the influence of outliers on the classification results.

3.2. Robust weighted twin extreme learning machine

3.2.1. RTELM

Our proposed RTELM framework can be written as:

$$\begin{aligned} \min_{\beta_1,\xi_{1,i}} \quad & \frac{1}{2}\sum_{i=1}^{u_1}(\omega_1 * \mathbf{h}_i\beta_1)^2 + c_1\sum_{i=1}^{m_2}\xi_{1,i} \\ \text{s.t.} \quad & -\sum_{i=1}^{u_2}\mathbf{h}_i\beta_1 + \sum_{i=1}^{m_2}\xi_i \geq 1, \xi_i \geq 0, \\ & \omega_1 = \begin{cases} \dfrac{1}{2}, & \bar{w}_{1_i|1} > \dfrac{u_1}{c_3}, \\ \dfrac{\bar{w}_{1_i|1}}{u_1}, & \bar{w}_{1_i|1} \leq \dfrac{u_1}{c_3}. \end{cases} \end{aligned} \tag{3.4}$$

and

$$\begin{aligned} \min_{\beta_2,\xi_{2,i}} \quad & \frac{1}{2}\sum_{i=1}^{u_2}(\omega_2 * \mathbf{h}_i\beta_2)^2 + c_2\sum_{i=1}^{m_1}\xi_{2,i} \\ \text{s.t.} \quad & -\sum_{i=1}^{u_1}\mathbf{h}_i\beta_2 + \sum_{i=1}^{m_1}\xi_i \geq 1, \xi_i \geq 0, \\ & \omega_2 = \begin{cases} \dfrac{1}{2}, & \bar{w}_{2_i|2} > \dfrac{u_2}{c_4}, \\ \dfrac{\bar{w}_{2_i|2}}{u_2}, & \bar{w}_{2_i|2} \leq \dfrac{u_2}{c_4}. \end{cases} \end{aligned} \tag{3.5}$$

where $\omega_1 = 2(w_{1_1|1}, w_{1_2|1}, \cdots, w_{1_{u_1}|1})^T$ and $\omega_2 = 2(w_{2_1|2}, w_{2_2|2}, \cdots, w_{2_{u_2}|2})^T$ represent the average weight of two classes of samples. h_i represents the feature mapping of x_i in the hidden layer, where $i = 1, 2, \cdots, u_1$, $u_1 < m_1$ in formula (3.4), and $i = 1, 2, \cdots, u_2$, $u_2 < m_2$ in formula (3.5). Here, u_1 and u_2 represent the top-u data points with a larger average weight for each class, where $u = u_1 + u_2$ represents the total number of selected data points from the two classes, and m_1 and m_2 represent the total number of positive and negative class samples, respectively; β_1 and β_2 represent two nonparallel hyperplanes, while c_1 and c_2 are positive trade-off parameters, c_3 and c_4 are positive constants. Finally, $\xi_{1,i}$ ($i = 1, 2, \cdots, m_2$) and $\xi_{2,i}$ ($i = 1, 2, \cdots, m_1$) represent the slack variables. Besides, for each hyperplane, the first term represents the minimization of the within-class margin. The second term represents the

regularization term, which is used to prevent overfitting. The constraint ensures that the within-class margin is greater than 1.

By introducing average weights (ω_1 and ω_2), we take into account the distribution of data points in the dataset. The larger the average weight, the closer the data points are to their class centers and the more significant role they play in constructing the hyperplane. Conversely, the smaller the average weight, the farther the data points are from their class centers, and the less significant their role in constructing the hyperplane. In addition, data points with a smaller average weight may also be considered as noise or outliers. To ensure the classification performance and robustness of our model, we only select the top-u ($u < m$) data points.

Furthermore, formulas (3.4) and (3.5) can be written as:

$$\begin{aligned} &\min_{\beta_1,\xi_1} \quad \frac{1}{2}\|\omega_1\tilde{\mathbf{H}}_{\mathbf{1}}\beta_1\|_2^2 + c_1\mathbf{e}_2^T\xi_1 \\ &\text{s.t.} \quad -\tilde{\mathbf{H}}_{\mathbf{2}}\beta_1 + \xi_1 \geq \mathbf{e}_2, \xi_1 \geq 0. \\ &\omega_1 = \begin{cases} \dfrac{1}{2}, & \bar{w}_{1_i|1} > \dfrac{u_1}{c_3}, \\ \dfrac{\bar{w}_{1_i|1}}{u_1}, & \bar{w}_{1_i|1} \leq \dfrac{u_1}{c_3}. \end{cases} \end{aligned} \tag{3.6}$$

and

$$\begin{aligned} &\min_{\beta_2,\xi_2} \quad \frac{1}{2}\|\omega_2\tilde{\mathbf{H}}_{\mathbf{2}}\beta_2\|_2^2 + c_2\mathbf{e}_1^T\xi_2 \\ &\text{s.t.} \quad -\tilde{\mathbf{H}}_{\mathbf{1}}\beta_2 + \xi_2 \geq \mathbf{e}_1, \xi_2 \geq 0. \\ &\omega_2 = \begin{cases} \dfrac{1}{2}, & \bar{w}_{2_i|2} > \dfrac{u_2}{c_4}, \\ \dfrac{\bar{w}_{2_i|2}}{u_2}, & \bar{w}_{2_i|2} \leq \dfrac{u_2}{c_4}. \end{cases} \end{aligned} \tag{3.7}$$

where

$$\tilde{\mathbf{H}}_{\mathbf{1}} = \begin{bmatrix} h_1(\mathbf{x}_1) & h_2(\mathbf{x}_1) & \cdots & h_L(\mathbf{x}_1) \\ h_1(\mathbf{x}_2) & h_2(\mathbf{x}_2) & \cdots & h_L(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_{u_1}) & h_2(\mathbf{x}_{u_1}) & \cdots & h_L(\mathbf{x}_{u_1}) \end{bmatrix} \in \mathbb{R}^{u_1\times L} \tag{3.8}$$

and

$$\tilde{\mathbf{H}}_{\mathbf{2}} = \begin{bmatrix} h_1(\mathbf{x}_1) & h_2(\mathbf{x}_1) & \cdots & h_L(\mathbf{x}_1) \\ h_1(\mathbf{x}_2) & h_2(\mathbf{x}_2) & \cdots & h_L(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_{u_2}) & h_2(\mathbf{x}_{u_2}) & \cdots & h_L(\mathbf{x}_{u_2}) \end{bmatrix} \in \mathbb{R}^{u_2\times L} \tag{3.9}$$

represent the hidden layer output matrices of the two classes of data; $\mathbf{e}_1 \in \mathbb{R}^{u_1}$ and $\mathbf{e}_2 \in \mathbb{R}^{u_2}$ are vectors of ones, and ξ_1 and ξ_2 are slack vectors.

3.2.2. Optimization

We give a detailed explanation and proof of the proposed average weight technique. First, we propose a useful lemma, as follows:

Lemma 1. *The average weight we mentioned is always* $\leq \frac{1}{2}$ *, that is,*

$$\lim_{u\to+\infty} \max_{1\leq i\leq u}\{\frac{\bar{w}_{x_i|x}}{u}\} \leq \frac{1}{2}, i = 1, 2, \cdots, u. \tag{3.10}$$

Proof. For $i = 1, 2, \cdots, u$, from (3.1) and (3.2), we have

$$\begin{aligned}
&\max_{1\leq i\leq u}\{\frac{\bar{w}_{x_i|x}}{u}\} \\
&= \frac{\max_{1\leq i\leq u}\{\bar{w}_{x_i|x}\}}{u} \\
&= \frac{\max_{1\leq i\leq u}\{\frac{1}{n}\sum_{j=1}^{n} w_{x_{ij}|x_j}\}}{u} \\
&\leq \frac{\frac{1}{n}\sum_{j=1}^{n}\max_{1\leq i\leq u}\{w_{x_{ij}|x_j}\}}{u} \\
&= \frac{\sum_{j=1}^{n}\max_{1\leq i\leq u}\{\min(r_{x_i|x}, n+1-r_{x_i|x})\}}{un},
\end{aligned} \tag{3.11}$$

If the average weights of the previous u data points are all equal, that is

$$\max_{1\leq i\leq u}\{w_{x_{i1}|x_1}\} = \max_{1\leq i\leq u}\{w_{x_{i2}|x_2}\} = \cdots = \max_{1\leq i\leq u}\{w_{x_{in}|x_n}\}, \tag{3.12}$$

we can obtain

$$\max_{1\leq i\leq u}\{\frac{1}{n}\sum_{j=1}^{n} w_{x_{ij}|j}\} = \frac{1}{n}\sum_{j=1}^{n}\max_{1\leq i\leq u}\{w_{x_{ij}|j}\}. \tag{3.13}$$

If $u = 2k$, we can obtain the following equation:

$$\max_{1\leq i\leq u}\{\min(r_{x_i|x}, n+1-r_{x_i|x})\} = k, \tag{3.14}$$

$$\frac{\frac{1}{n}\sum_{j=1}^{n}\max_{1\leq i\leq u}\{w_{x_{ij}|x_j}\}}{u} = \frac{\sum_{j=1}^{n} k}{2kn} = \frac{1}{2}; \tag{3.15}$$

If $u = 2k + 1$, we can obtain

$$\max_{1\leq i\leq u}\{\min(r_{x_i|x}, n+1-r_{x_i|x})\} = k+1, \tag{3.16}$$

$$\frac{\frac{1}{n}\sum_{j=1}^{n}\max_{1\leq i\leq u}\{w_{x_{ij}|x_j}\}}{u} = \frac{\sum_{j=1}^{n} k+1}{2kn} = \frac{k+1}{2k}. \tag{3.17}$$

Thus, we can obtain

$$\frac{\frac{1}{n}\sum_{j=1}^{n}\max_{1\leq i\leq u}\{w_{x_{ij}|x_j}\}}{u} = \begin{cases} \frac{1}{2}, & m = 2k, \\ \frac{k+1}{2k+1}, & m = 2k+1. \end{cases} \tag{3.18}$$

If $u \to +\infty$, $k \to +\infty$, $\frac{k+1}{2k+1} \to +\infty$; therefore, we can prove that the proposed average weight is always $\leq \frac{1}{2}$. Hence, the inequality $\lim_{u\to+\infty} \max_{1\leq i\leq u}\{\frac{\bar{w}_{x_i|x}}{u}\} \leq \frac{1}{2}$ holds true for all cases. □

From 1, it can be seen that in RTELM, the average weights ω_1 and ω_2 are at most $\frac{1}{2}$. If $\bar{w}_{1_i|1} > \frac{u_1}{c_3}$, then all data points in the dataset have the same average weight and play an equally important role. Conversely, $\bar{w}_{1_i|1} \leq \frac{u_1}{c_3}$. The data points that are closer to their class centers have a higher average weight and play a more significant role in constructing the hyperplane. However, in general, the number of data points u in the dataset is large. Thus, in the proposed model, the average weights are generally less than $\frac{1}{2}$.

The Lagrange function corresponding to the optimization problem (3.6) can be written as follows:

$$\begin{aligned} L(\beta_1, \xi_1, \alpha, \gamma) = \frac{1}{2}\|\omega_1 \tilde{\mathbf{H}}_\mathbf{1}\beta_1\|_2^2 + c_1 \mathbf{e}_2^T \xi_1 \\ -\alpha^T(-\tilde{\mathbf{H}}_\mathbf{2}\beta_1 + \xi_1 - \mathbf{e}_2) - \gamma^T \xi_1, \end{aligned} \tag{3.19}$$

where $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_{u_2})^T$ and $\gamma = (\gamma_1, \gamma_2, \cdots, \gamma_{u_2})^T$ are the Lagrange multipliers, we can derive the Lagrange function concerning β_1 and ξ_1, we can obtain the following KKT conditions.

$$\begin{cases} \dfrac{\partial L}{\partial \beta_1} = (\omega_1 \tilde{\mathbf{H}}_\mathbf{1})^T \omega_1 \tilde{\mathbf{H}}_\mathbf{1}\beta_1 + \alpha^T \tilde{\mathbf{H}}_\mathbf{2} = 0, \\ \dfrac{\partial L}{\partial \xi_1} = c_1 \mathbf{e}_2^T - \alpha^T - \gamma^T = 0, \\ \dfrac{\partial L}{\partial \alpha} = -\tilde{\mathbf{H}}_\mathbf{2}\beta_1 + \xi_1 - \mathbf{e}_2 = 0, \\ \dfrac{\partial L}{\partial \gamma} = \xi_1 = 0, \\ \alpha \geq 0, \gamma \geq 0. \end{cases} \tag{3.20}$$

From formula (3.20), we can obtain:

$$\beta_1 = -[(\omega_1 \tilde{\mathbf{H}}_\mathbf{1})^T (\omega_1 \tilde{\mathbf{H}}_\mathbf{1})]^{-1} \tilde{\mathbf{H}}_\mathbf{2}^T \alpha \tag{3.21}$$

and

$$0 \leq \alpha \leq c_1. \tag{3.22}$$

Since $\omega_1 \leq \frac{1}{2}$, to prevent a singular solution, we introduce a small regularization term $\epsilon\mathbf{I}$ into the Eq (3.21), where ϵ is a small positive number and $\mathbf{I}$ represents an identity matrix. Thus, formula (3.21) can be rewritten as follows:

$$\beta_1 = -[(\omega_1 \tilde{\mathbf{H}}_\mathbf{1})^T (\omega_1 \tilde{\mathbf{H}}_\mathbf{1}) + \epsilon\mathbf{I}]^{-1} \tilde{\mathbf{H}}_\mathbf{2}^T \alpha. \tag{3.23}$$

Similar to the solution for β_1, we can obtain

$$\beta_2 = -[(\omega_2 \tilde{\mathbf{H}}_\mathbf{2})^T (\omega_2 \tilde{\mathbf{H}}_\mathbf{2}) + \epsilon\mathbf{I}]^{-1} \tilde{\mathbf{H}}_\mathbf{1}^T \gamma. \tag{3.24}$$

By substituting β_1 and β_2 into Eq (3.19) and utilizing Eq (3.20), we can derive the dual problem for the primary problems (3.4) and (3.5) in the context of RTELM.

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^T \omega_2 \tilde{\mathbf{H}}_\mathbf{2}[(\omega_1 \tilde{\mathbf{H}}_\mathbf{1})^T (\omega_1 \tilde{\mathbf{H}}_\mathbf{1}) + \epsilon\mathbf{I}]^{-1} \tilde{\mathbf{H}}_\mathbf{2}^T \alpha - \mathbf{e}_2^T \alpha \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq c_1, \ i = 1, \cdots, u_2 \end{aligned} \tag{3.25}$$

and

$$\begin{aligned} \min_{\gamma} \quad & \frac{1}{2}\gamma^T \omega_1 \tilde{\mathbf{H}}_1[(\omega_2\tilde{\mathbf{H}}_2)^T(\omega_2\tilde{\mathbf{H}}_2) + \epsilon\mathbf{I}]^{-1}\tilde{\mathbf{H}}_1^T\gamma - \mathbf{e}_1^T\gamma \\ \text{s.t.} \quad & 0 \le \gamma_i \le c_2,\ i = 1, \cdots, u_1, \end{aligned} \tag{3.26}$$

where ω_1 and ω_2 represent the average weights of two sample classes, and u_1 and u_2 represent the top u_1 and u_2 data points with higher average weights for the two classes, respectively. Therefore, a new sample point $\mathbf{x}^*$ is classified using the following decision function:

$$f(\mathbf{x}^*) = \arg\min_{r=1,2} d_r(\mathbf{x}^*) = \arg\min_{r=1,2} |\beta_r^T h(\mathbf{x}^*)|,$$

where $|.|$ represents the perpendicular distance of a new sample point $\mathbf{x}^*$ from the hyperplane β_r.

The pseudocode of RTELM is presented in Algorithm 1.

Algorithm 1 RTELM-algorithm.

Input: Training set $\mathcal{T}_l = \{x_i, y_i\}_{i=1}^l$, activation function $G(.)$, parameters c_1 and c_2.
Output: β_1 and β_2 and the decision function $f(\mathbf{x}^*)$ of RTELM.
Step:
1: To begin, sort the dataset in ascending and descending order. Then, calculate the weighted sum and average weight using formulas (3.1) and (3.2), respectively. Finally, obtain the average weights ω_1 and ω_2.
2: Initiate an ELM network with L hidden nodes using the random input weight ω_i and $\mathbf{b}_i$.
3: Calculate the hidden layer output matrixes $\tilde{\mathbf{H}}_1$ and $\tilde{\mathbf{H}}_2$ by (3.8) and by (3.9), respectively.
4: Calculate α_1 and γ_1 by (3.25) and (3.26), respectively.
5: Calculate β_1 and β_2 by
$\beta_1 = -[(\omega_1\tilde{\mathbf{H}}_1)^T(\omega_1\tilde{\mathbf{H}}_1) + \epsilon\mathbf{I}]^{-1}\tilde{\mathbf{H}}_2^T\alpha$.
and
$\beta_2 = -[(\omega_2\tilde{\mathbf{H}}_2)^T(\omega_2\tilde{\mathbf{H}}_2) + \epsilon\mathbf{I}]^{-1}\tilde{\mathbf{H}}_1^T\gamma$.
Return: The decision functions: $f(\mathbf{x}^*) = \arg\min_{r=1,2} d_r(\mathbf{x}^*) = \arg\min_{r=1,2} |\beta_r^T h(\mathbf{x}^*)|$.

3.3. Robust weighted twin extreme learning machine with manifold regularization

3.3.1. Lap-RTELM

SSL utilizes manifold regularization to leverage the geometric distribution information of data points in the dataset. Manifold regularization utilizes regularization terms to represent the distribution of unlabeled data points in the manifold space. Our proposed Lap-RTELM framework can be written as follows:

$$\begin{aligned}
&\min_{\beta_1,\xi_i} \quad \frac{1}{2}\sum_{i=1}^{u_1}(\omega_1 * \mathbf{h}_i\beta_1)^2 + c_1\sum_{i=1}^{m_2}\xi_{1,i} + c_5\|\mathbf{f}_1\|_M^2 \\
&\text{s.t.} \quad -\sum_{i=1}^{u_2}\mathbf{h}_i\beta_1 + \sum_{i=1}^{m_2}\xi_i \geq 1, \xi_i \geq 0, \\
&\omega_1 = \begin{cases} \dfrac{1}{2}, & \bar{w}_{1_i|1} > \dfrac{u_1}{c_3}, \\ \dfrac{\bar{w}_{1_i|1}}{u_1}, & \bar{w}_{1_i|1} \leq \dfrac{u_1}{c_3}. \end{cases}
\end{aligned} \tag{3.27}$$

and

$$\begin{aligned}
&\min_{\beta_2,\xi_i} \quad \frac{1}{2}\sum_{i=1}^{u_2}(\omega_2 * \mathbf{h}_i\beta_2)^2 + c_2\sum_{i=1}^{m_1}\xi_{2,i} + c_6\|\mathbf{f}_2\|_M^2 \\
&\text{s.t.} \quad -\sum_{i=1}^{u_1}\mathbf{h}_i\beta_2 + \sum_{i=1}^{m_1}\xi_i \geq 1, \xi_i \geq 0, \\
&\omega_2 = \begin{cases} \dfrac{1}{2}, & \bar{w}_{2_i|2} > \dfrac{u_2}{c_4}, \\ \dfrac{\bar{w}_{2_i|2}}{u_2}, & \bar{w}_{2_i|2} \leq \dfrac{u_2}{c_4}. \end{cases}
\end{aligned} \tag{3.28}$$

where c_5 and c_6 are positive regularization parameters, $\|\mathbf{f}_1\|_M^2$ and $\|\mathbf{f}_2\|_M^2$ represents manifold regularization terms, where

$$\|\mathbf{f}_1\|_M^2 = \frac{1}{(l+u)^2}\sum_{i,j=1}^{l+u} W_{ij}(\mathbf{f}_1(\mathbf{x}_i) - \mathbf{f}_1(\mathbf{x}_j))^2 = \mathbf{f}_1^T\mathbf{L}\mathbf{f}_1$$

and

$$\|\mathbf{f}_2\|_M^2 = \frac{1}{(l+u)^2}\sum_{i,j=1}^{l+u} W_{ij}(\mathbf{f}_2(\mathbf{x}_i) - \mathbf{f}_2(\mathbf{x}_j))^2 = \mathbf{f}_2^T\mathbf{L}\mathbf{f}_2,$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian matrix, where $\mathbf{D} = diag(\sum_{j=1}^n W_{1j}, \sum_{j=1}^n W_{2j}, ..., \sum_{j=1}^n W_{nj})$, and $\mathbf{W}$ represents the adjacency matrix of W_i and W_j. The vectors $\mathbf{f}_1 = [\mathbf{f}_1(\mathbf{x}_1), \mathbf{f}_1(\mathbf{x}_2), \cdots, \mathbf{f}_1(\mathbf{x}_{l+u})]^T = \mathbf{H}\beta_1$, and $\mathbf{f}_2 = [\mathbf{f}_2(\mathbf{x}_1), \mathbf{f}_2(\mathbf{x}_2), \cdots, \mathbf{f}_2(\mathbf{x}_{l+u})]^T = \mathbf{H}\beta_2$, where $\mathbf{H} \in \mathbb{R}^{n\times L}$ represents all samples. Thus, the formulas (3.27) and (3.28) can be rewritten as:

$$\begin{aligned}
&\min_{\beta_1,\xi_1} \quad \frac{1}{2}\|\omega_1\tilde{\mathbf{H}}_1\beta_1\|_2^2 + c_1\mathbf{e}_2^T\xi_1 + c_5\beta_1^T\mathbf{H}^T\mathbf{L}\mathbf{H}\beta_1 \\
&\text{s.t.} \quad -\tilde{\mathbf{H}}_2\beta_1 + \xi_1 \geq \mathbf{e}_2, \xi_1 \geq 0. \\
&\omega_1 = \begin{cases} \dfrac{1}{2}, & \bar{w}_{1_i|1} > \dfrac{u_1}{c_3}, \\ \dfrac{\bar{w}_{1_i|1}}{u_1}, & \bar{w}_{1_i|1} \leq \dfrac{u_1}{c_3}, \end{cases}
\end{aligned} \tag{3.29}$$

and

$$\begin{aligned} &\min_{\beta_2,\xi_2} \quad \frac{1}{2}\|\omega_2\tilde{\mathbf{H}}_{\mathbf{2}}\beta_2\|_2^2 + c_2\mathbf{e}_1^T\xi_2 + c_6\beta_2^T\mathbf{H}^T\mathbf{L}\mathbf{H}\beta_2 \\ &\text{s.t.} \quad -\tilde{\mathbf{H}}_{\mathbf{1}}\beta_2 + \xi_2 \geq \mathbf{e}_1, \xi_2 \geq 0. \\ &\omega_2 = \begin{cases} \dfrac{1}{2}, & \bar{w}_{2_i|2} > \dfrac{u_2}{c_4}, \\ \dfrac{\bar{w}_{2_i|2}}{u_2}, & \bar{w}_{2_i|2} \leq \dfrac{u_2}{c_4}. \end{cases} \end{aligned} \tag{3.30}$$

3.3.2. Optimization

The Lagrange function corresponding to the optimization problem (3.29) can be written as follows:

$$\begin{aligned} L(\beta_1, \xi_1, \alpha, \gamma) = \frac{1}{2}\|\omega_1\tilde{\mathbf{H}}_{\mathbf{1}}\beta_1\|_2^2 + c_1\mathbf{e}_2^T\xi_1 + c_3\beta_1^T\mathbf{H}^T\mathbf{L}\mathbf{H}\beta_1 \\ -\alpha^T(-\tilde{\mathbf{H}}_{\mathbf{2}}\beta_1 + \xi_1 - \mathbf{e}_2) - \gamma^T\xi_1, \end{aligned} \tag{3.31}$$

where $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_{u_2})^T$ and $\gamma = (\gamma_1, \gamma_2, \cdots, \gamma_{u_2})^T$ are the Lagrange multipliers, we can derive the Lagrange function concerning β_1 and ξ_1, we can obtain the following KKT conditions.

$$\begin{cases} \dfrac{\partial L}{\partial \beta_1} = (\omega_1\tilde{\mathbf{H}}_{\mathbf{1}})^T\omega_1\tilde{\mathbf{H}}_{\mathbf{1}}\beta_1 + c_3\mathbf{H}^T\mathbf{L}\mathbf{H}\beta_1 + \alpha^T\tilde{\mathbf{H}}_{\mathbf{2}} = 0, \\ \dfrac{\partial L}{\partial \xi_1} = c_1\mathbf{e}_2^T - \alpha^T - \gamma^T = 0, \\ \dfrac{\partial L}{\partial \alpha} = -\tilde{\mathbf{H}}_{\mathbf{2}}\beta_1 + \xi_1 - \mathbf{e}_2 = 0, \\ \dfrac{\partial L}{\partial \gamma} = \xi_1 = 0, \\ \alpha \geq 0, \gamma \geq 0. \end{cases} \tag{3.32}$$

From formula (3.32), we can obtain:

$$\beta_1 = -[(\omega_1\tilde{\mathbf{H}}_{\mathbf{1}})^T(\omega_1\tilde{\mathbf{H}}_{\mathbf{1}}) + c_5\mathbf{H}^T\mathbf{L}\mathbf{H}]^{-1}\tilde{\mathbf{H}}_{\mathbf{2}}^T\alpha \tag{3.33}$$

and

$$0 \leq \alpha \leq c_1\mathbf{e}_2^T. \tag{3.34}$$

From 1, we obtain $\omega_1 \leq \frac{1}{2}$, in order to prevent a singular solution, we introduce a small regularization term $\epsilon\mathbf{I}$ into the formula (3.33), where ϵ is a small positive number and $\mathbf{I}$ is an identity matrix. Thus, formula (3.33) can be rewritten as

$$\beta_1 = -[(\omega_1\tilde{\mathbf{H}}_{\mathbf{1}})^T(\omega_1\tilde{\mathbf{H}}_{\mathbf{1}}) + c_5\mathbf{H}^T\mathbf{L}\mathbf{H} + \epsilon\mathbf{I}]^{-1}\tilde{\mathbf{H}}_{\mathbf{2}}^T\alpha. \tag{3.35}$$

Similar to the solution for β_1, we can obtain

$$\beta_2 = -[(\omega_2\tilde{\mathbf{H}}_{\mathbf{2}})^T(\omega_2\tilde{\mathbf{H}}_{\mathbf{2}}) + c_6\mathbf{H}^T\mathbf{L}\mathbf{H} + \epsilon\mathbf{I}]^{-1}\tilde{\mathbf{H}}_{\mathbf{1}}^T\gamma. \tag{3.36}$$

By substituting β_1 and β_2 into Eq (3.35) and utilizing Eq (3.36), we can derive the dual problem for Lap-RTELM from the primary problems (3.27) and (3.28).

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^T \omega_2 \tilde{\mathbf{H}}_2[(\omega_1 \tilde{\mathbf{H}}_1)^T(\omega_1 \tilde{\mathbf{H}}_1) + c_5 \mathbf{H}^T \mathbf{L}\mathbf{H} + \epsilon \mathbf{I}]^{-1} \tilde{\mathbf{H}}_2^T \alpha - \mathbf{e}_2^T \alpha \\ \text{s.t.} \quad & 0 \le \alpha_i \le c_1, \ i = 1, \cdots, u_2 \end{aligned} \tag{3.37}$$

and

$$\begin{aligned} \min_{\gamma} \quad & \frac{1}{2}\gamma^T \omega_1 \tilde{\mathbf{H}}_1[(\omega_2 \tilde{\mathbf{H}}_2)^T(\omega_2 \tilde{\mathbf{H}}_2) + c_6 \mathbf{H}^T \mathbf{L}\mathbf{H} + \epsilon \mathbf{I}]^{-1} \tilde{\mathbf{H}}_1^T \gamma - \mathbf{e}_1^T \gamma \\ \text{s.t.} \quad & 0 \le \gamma_i \le c_2, \ i = 1, \cdots, u_1. \end{aligned} \tag{3.38}$$

Therefore, a new sample point $\mathbf{x}^*$ is classified by the following decision function:

$$f(\mathbf{x}^*) = \arg\min_{r=1,2} d_r(\mathbf{x}^*) = \arg\min_{r=1,2} |\beta_r^T h(\mathbf{x}^*)|.$$

where $|.|$ represents the perpendicular distance of a new sample point $\mathbf{x}^*$ from the hyperplane β_r.

The pseudocode of Lap-RTELM is presented in Algorithm 2.

Algorithm 2 Lap-RTELM-algorithm.

Input: Training set $\mathcal{T} = \mathcal{T}_l \cup \mathcal{T}_u$, activation function $G(.)$, parameters c_1, c_2, c_3 and c_4.
Output: β_1 and β_2 and the decision function $f(\mathbf{x}^*)$ of Lap-RTELM.
Step:
1: Sort the dataset in ascending and descending order, calculate the weight sum and average weight using formulas (3.1) and (3.2), and obtain the average weights ω_1 and ω_2.
2: Initiate an ELM network with L hidden nodes using the random input weight ω_i and b_i.
3: Construct graph Laplacian matrix $\mathbf{L}$, and then get $\mathbf{H}$.
4: Calculate the corresponding hidden layer output matrixes $\tilde{\mathbf{H}}_1$ and $\tilde{\mathbf{H}}_2$ by (3.8) and (3.9), respectively.
5: Calculate α_1 and γ_1 by (3.37) and (3.38), respectively.
6: Calculate β_1 and β_2 by
$\beta_1 = -[(\omega_1 \tilde{\mathbf{H}}_1)^T(\omega_1 \tilde{\mathbf{H}}_1) + c_5 \mathbf{H}^T \mathbf{L}\mathbf{H} + \epsilon \mathbf{I}]^{-1} \tilde{\mathbf{H}}_2^T \alpha$.
$\beta_2 = -[(\omega_2 \tilde{\mathbf{H}}_2)^T(\omega_2 \tilde{\mathbf{H}}_2) + c_6 \mathbf{H}^T \mathbf{L}\mathbf{H} + \epsilon \mathbf{I}]^{-1} \tilde{\mathbf{H}}_1^T \gamma$.
Return: The decision functions: $f(\mathbf{x}^*) = \arg\min_{r=1,2} d_r(\mathbf{x}^*) = \arg\min_{r=1,2} |\beta_r^T h(\mathbf{x}^*)|$.

3.4. Complexity analysis

In our work, the dimensions of the vectors related to algorithm complexity are $\omega_1 \in \mathbb{R}^{1\times u_1}$, $\tilde{\mathbf{H}}_1 \in \mathbb{R}^{u_1\times L}$, $\mathbf{I} \in \mathbb{R}^{L\times L}$, $\tilde{\mathbf{H}}_2 \in \mathbb{R}^{u_2\times L}$, $\mathbf{H} \in \mathbb{R}^{n\times L}$, $\mathbf{L} \in \mathbb{R}^{n\times n}$, $\alpha \in \mathbb{R}^{u_2\times 1}$, and $\gamma \in \mathbb{R}^{u_2\times 1}$. Here, u_1 and u_2 represent the number of positive and negative samples selected, respectively. $u = u_1 + u_2$ represents the total number of samples, and L represents the number of hidden nodes. Additionally, since our algorithm does not require iteration, the complexity mainly arises from matrix multiplication and matrix inversion operations. In Algorithm 1, the complexity of calculating output matrices $\tilde{\mathbf{H}}_1$ and $\tilde{\mathbf{H}}_2$ in the third step is $O(Lu)$. The complexity of calculating α_1 and γ_1 in the fourth step is $O(u)$, and the complexity of

calculating β_1 and β_2 in the fifth step is $O(L^3 + Lu^2)$. Therefore, the total complexity of Algorithm 1 is approximately $O(L^3)$. In Algorithm 2, compared to Algorithm 1, there is an additional of the manifold regularization term, which has a complexity of $O(mn^2)$. Therefore, the total complexity of Algorithm 2 is approximately $O(L^3 + mn^2)$.

4. Numerical experiments

In this section, we further investigate the effectiveness, robustness, and classification performance of the proposed RTELM and Lap-RTELM algorithms through numerical experiments. We first introduce the experimental setup, then present the experimental results on synthetic datasets, UCI datasets, and image classification, and analyze the impact of parameters in the proposed RTELM and Lap-RTELM. Finally, we compare the performance differences between our method and six other algorithms through statistical test analysis.

4.1. Experimental setup

We compared our proposed RTELM and Lap-RTELM with the following algorithms. Table 2 presents the eight algorithms and their corresponding parameters. All of the above algorithms are implemented in MATLAB (R2021a), and the experimental equipment was a PC with a 2.11 GHz CPU and 16GB of RAM. References [8] and [19] suggest that TELM is insensitive to parameters, and the number of hidden layer nodes L increasing will consume more time, but the prediction accuracy remains stable. Therefore, we set the parameters of all TELM and its variants as $c = 1$, and $L = 150$. In FMSVM, $c_p = 1$. In Lap-RMTELM, $\epsilon_i = 10^{-4}$ and $\theta \in \{0, ..., +\infty\}$. In addition, the remaining parameters c_i were selected from the range of $\{2^{-7}, ..., 2^7\}$. In SSL, the number of nearest neighbors k is selected from the set $\{10, 15, 20\}$, and the kernel parameter σ is selected from the range $\{2^{-7}, ..., 2^7\}$.

Table 2. The eight algorithms and their corresponding parameters.

Algorithms	References	Parameters	Algorithms	References	Parameters
RTELM		$c_i(i = 1, 2, 3, 4)$	TELM	(2017) [8]	$c_i(i = 1, 2)$
TDSVM	(2023) [37]	$c_i(i = 1, 2, 3, 4, 5, 6)$	FMSVM	(2022) [38]	$c_i(i = l, p)$,σ
Lap-RTELM		$c_i(i = 1, 2, 3, 4, 5, 6)$,$k$,$\sigma$	Lap-TELM	(2018) [19]	$c_i(i = 1, 2)$,k,σ
Lap-TSVM	(2012) [39]	$c_i(i = 1)$,k,σ	Lap-RMTELM	(2021) [29]	$c_i(i = 1, 2, 3, 4)$,k,σ,$\epsilon_i(i = 1, 2, 3, 4)$,$\theta$

In our experiments, we work with the pre-selection ratio $r = \frac{u_c}{m_c} \times 100\%$, where u_c is the number of top-ranked points retained from a class with m_c samples. This ratio is more interpretable and dataset-independent than absolute counts. We perform grid search over:

$$r \in \{10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%, 100\%\}$$

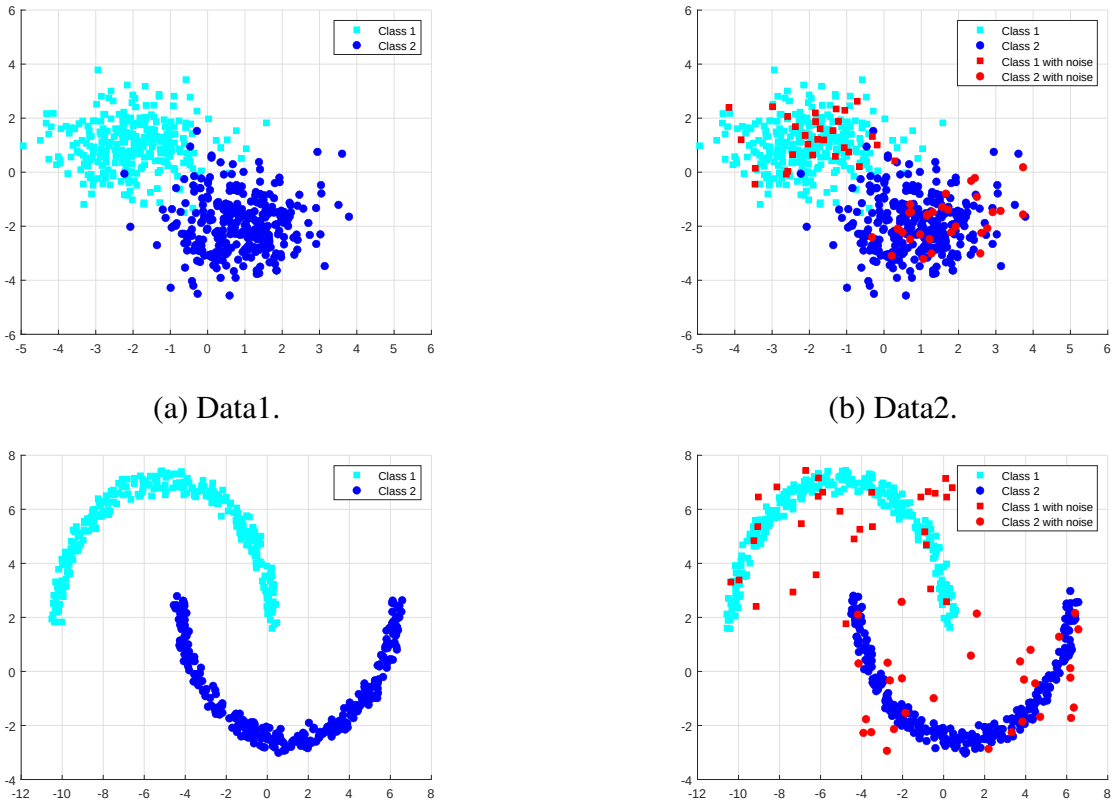
where $r = 100\%$ corresponds to the baseline case without pre-selection. The absolute number of selected points is $u_c = \lceil r \cdot m_c \rceil$, with a minimum floor of $u_c \geq 5$ to ensure statistical reliability for very small classes (e.g., the negative class in Hepat with $m_2 = 32$ has effective lower bound $r \approx 15.6\%$). This range was chosen based on preliminary sensitivity analysis (Section 5.5, Figure 4a) showing that optimal performance typically occurs within 30%–60%, with diminishing returns beyond 70%. For each benchmark dataset, we evaluate all parameter combinations using 5-fold cross-validation,

selecting the combination that maximizes average validation accuracy. We have updated Table 2 to explicitly include the pre-selection ratio in the parameter list for RTELM and Lap-RTELM, ensuring full reproducibility of our experiments.

To fairly evaluate the classification performance of the eight models mentioned above, we measured their performance using the classical metrics of accuracy (ACC) and F_1-score (F_1) [40]. The formulas for calculating ACC and F_1 are as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \quad F_1 = \frac{2TP}{2TP + FN + FP}, \tag{4.1}$$

where TP and TN represent true positives and true negatives, respectively, while FP and FN represent false positives and false negatives, respectively. Higher values of ACC and F_1 indicate better performance.



(a) Data1. (b) Data2.

(c) Data3. (d) Data4.

Figure 2. Images of four synthetic datasets.

4.1.1. Dataset description

Synthetic Datasets: We utilized two types of synthetic datasets, as shown in Figure 2. The first type is a dataset consisting of a mixture of Gaussian distributions, Data 1 and Data 2, each with 300 samples. For Data 1: the mean of the class 1 is $\mu_1 = (-2, 1)^T$ with a covariance matrix $\Sigma_1 = diag(1, 1)$; the mean of the class 2 is $\mu_2 = (1, -2)^T$ with a covariance matrix $\Sigma_2 = diag(1, 1)$. The second type is the classic Two Moons datasets, consisting of Data 3 and Data 4, each with 300 samples. Data 2 and

Data 4 are obtained by adding 30 noise samples to each class of Data 1 and Data 3, respectively. In addition, we used a cyan "□" to represent the samples of class 1, use blue "∘" to represent the samples of class 2. The noise of the two types of data is represented in red. Then, we evenly divide each dataset into training and testing sets, with a sample ratio of 7:3.

UCI datasets: We selected the following ten datasets from the UCI benchmark database* The terms include: Balance, Pima, Australian, Hepat, Mushroom, German, QSAR, Spam, Sonar. The specific information can be found in Table 3. In Table 3, "n" represents the dimension of each dataset, while "m_1" and "m_2" represent the number of positive and negative samples in each dataset, respectively. First, we normalized the downloaded raw data to keep the data within the range [0,1], in order to reduce the differences between different sample features. Then, we randomly divided each dataset into training and testing sets, with a sample ratio of 7:3. Finally, to reduce the impact of random experimental results, we performed 10-fold cross-validation on each dataset. In addition, to verify the robustness of the proposed model, we conducted experiments in environments with varying levels of Gaussian noise. The Gaussian noise follows a normal distribution $N(\mu, \sigma^2)$, where μ represents the mean of the Gaussian noise distribution, and σ^2 represents the variance of the Gaussian noise distribution. The experiment was conducted in the following three Gaussian environments in sequence: $N(0, 0^2)$, $N(0, 0.1^2)$, and $N(0, 0.2^2)$.

Table 3. Description of the UCI datasets.

Datasets	n	m_1 vs. m_2	Datasets	n	m_1 vs. m_2
Balance	4	288 vs. 288	German	24	700 vs. 300
Pima	8	500 vs. 268	QSAR	41	699 vs. 356
Australian	14	307 vs. 383	Spam	57	2788 vs. 1813
Hepat	19	123 vs. 32	Sonar	60	111 vs. 97
Mushroom	22	4208 vs. 3916			

Table 4. Experimental results on synthetic datasets.

	RTELM	TELM	TDSVM	FMSVM	Lap-RTELM	Lap-TELM	Lap-TSVM	Lap-RMTELM
	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC
	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1
Data1	97.67	97.08	97.50	97.76	**98.07**	95.97	97.50	97.15
	97.52	96.67	97.44	**98.55**	97.11	96.29	95.63	97.11
Data2	96.83	97.08	97.17	95.44	97.09	96.82	95.97	**97.44**
	96.67	**97.52**	84.77	97.00	97.00	96.35	95.90	96.95
Data3	**100.00**	**100.00**	96.00	98.91	**100.00**	99.70	96.50	98.79
	99.85	99.92	97.23	99.00	**100.00**	99.04	96.93	96.86
Data4	**100.00**	99.92	95.88	96.40	**100.00**	99.50	96.50	98.04
	99.24	97.50	96.00	96.41	**100.00**	98.39	96.23	97.00

MNIST Datasets: The MNIST dataset is a collection of handwritten numeric images with 10 classes

*https://archive.ics.uci.edu/ml/index.php/.

ranging from "0" to "9". It consists of 70,000 samples, with 60,000 in the training set and 10,000 in the test set. Each image is 28x28 pixels in grayscale. The MNIST20 dataset is a subset of MNIST, containing 3,500 randomly chosen samples for binary classification tasks. The training set includes 3,000 samples, and the test set has 500 samples. This subset focuses on classifying the numbers "2" and the original images were resized to 16x16 pixels.

COIL-20 Datasets: The COIL-20 dataset comprises grayscale images of 20 objects. We selected 72 images from each object and resized the original 1440 images to 32×32 pixels. For binary classification tasks, the first ten objects are paired with the last ten objects.

5. Experimental results

5.1. Results of synthetic datasets

We first conducted experiments on the two types of synthetic datasets mentioned above, and the experimental results are shown in Table 4. Table 4 shows the best ACC and F_1 achieved by eight algorithms on four datasets.

From Table 4, we can deduce the following: For datasets Data 1 and Data 2, both adhere to a mixed normal distribution, each comprising 300 samples from two classes. The only difference is that in Data 2, each class has 30 noise samples. The overall ACC and F_1 of the two models proposed by us are relatively high. For datasets Data 3 and Data 4, neither follows a mixed normal distribution. Each dataset consists of 300 samples from two classes. The only difference is that in Data 4, each class has 30 noisy samples. The ACC and F_1 of the Lap-RTELM proposed by us are both 100 on these datasets, and the ACC of RTELM is also 100, which indicates that the two algorithms proposed by us are effective and have good classification performance. Besides, compared with the other six algorithms, the two algorithms proposed by us are more robust. We will further verify the effectiveness, performance, and robustness of the proposed Lap-RTELM and RTELM in UCI datasets and image recognition experiments.

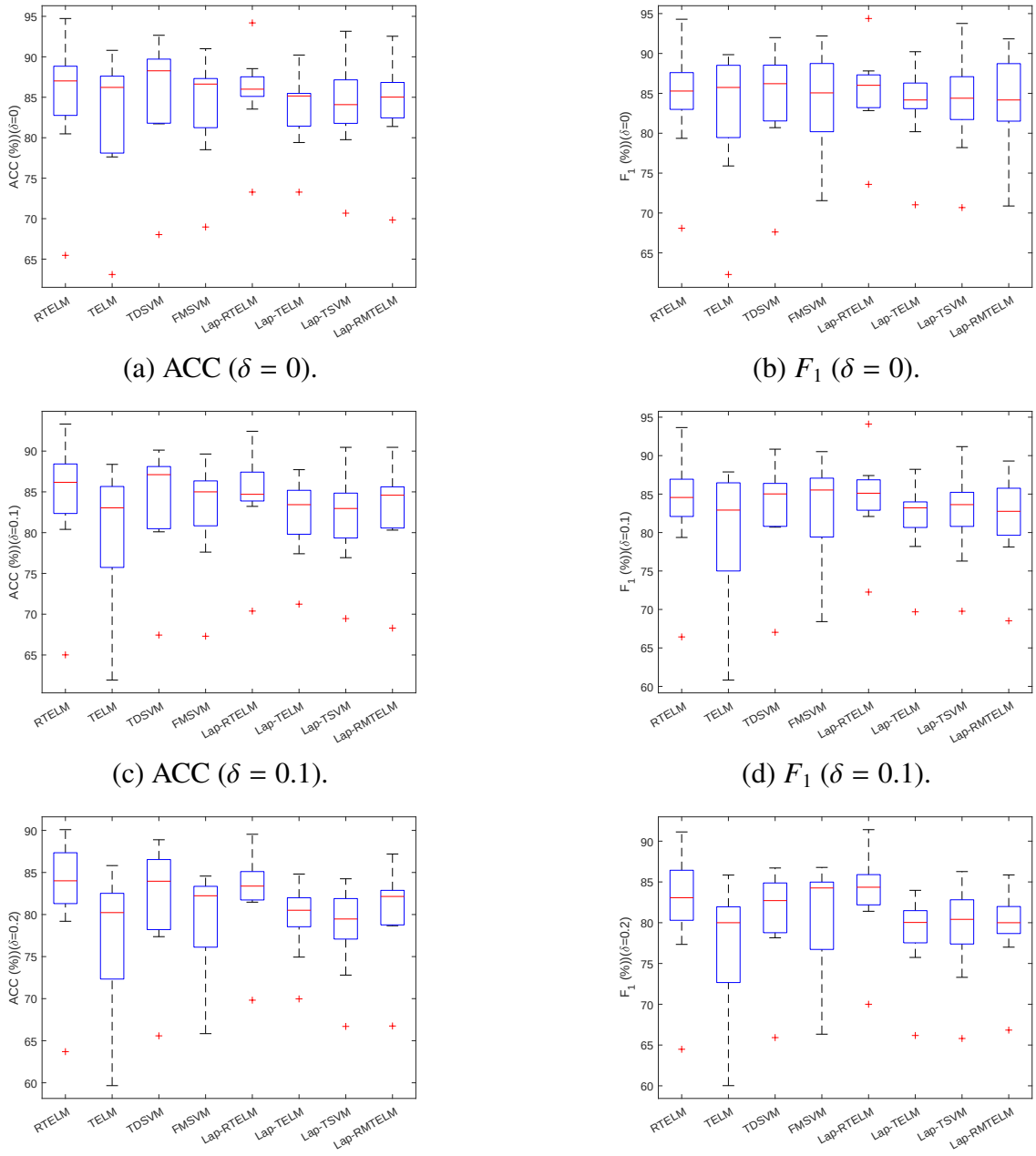
To further validate the effectiveness and robustness of the proposed RTELM and Lap-RTELM, we conducted extensive experiments on the UCI dataset [†]. The UCI datasets used in our experiments are presented in Table 3. Table 5, Table 6, and Table 7 represent the experimental results under Gaussian noise ratios of $\delta = 0$, $\delta = 0.1$, and $\delta = 0.2$, respectively. In addition, Avg. ACC, Avg.F_1, Avg. Time, and Avg. rank, respectively, represent the average ACC, average F_1, average time, and average rank of the same algorithm on different UCI datasets. Furthermore, the visual box plots of Table 5, Table 6, and Table 7 are shown in Figure 3.

We first conducted experiments without Gaussian noise to further validate the effectiveness of the proposed algorithms. From the experimental results in Table 5, it can be seen that our proposed RTELM and Lap-RTELM models exhibit the best overall classification performance. Moreover, RTELM and Lap-RTELM are significantly superior to the three supervised algorithms and three semi-supervised algorithms. For example, in the Mushroom dataset, RTELM achieves 3.93%, 2.07%, and 3.73% higher ACC than TELM, TDSVM, and FMSVM, respectively, while Lap-RTELM achieves 3.97%, 1.03%, and 1.65% higher ACC than Lap-TELM, Lap-TSVM, and Lap-RMTELM, respectively. Similar situations can be observed in other datasets. Furthermore, Lap-RTELM slightly outperforms RTELM in terms of classification performance and achieves the highest Avg.ACC,

[†]https://archive.ics.uci.edu/ml/index.php/.

Avg.F_1, and Avg.rank. Therefore, we can conclude that the average weight technique and pre-selection technique used in our two learning frameworks can improve classification performance. Thus, in a noise-free environment, the two proposed algorithms are effective.

5.2. *Results of UCI datasets*



(a) ACC ($\delta = 0$). (b) F_1 ($\delta = 0$).

(c) ACC ($\delta = 0.1$). (d) F_1 ($\delta = 0.1$).

(e) ACC ($\delta = 0.2$). (f) F_1 ($\delta = 0.2$).

Figure 3. Visualization of box plots for Table 5, Table 6, and Table 7 results.

Table 5. Experimental results on UCI datasets without Gaussian noise.

	RTELM	TELM	TDSVM	FMSVM	Lap-RTELM	Lap-TELM	Lap-TSVM	Lap-RMTELM
	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC
	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1
	Times	Times	Times	Times	Times	Times	Times	Times
Hepat	**89.25**	87.21	89.10	86.76	87.19	82.11	83.32	85.02
	86.98	**87.26**	86.21	86.11	86.01	84.17	84.39	85.17
	2.73	0.21	2.20	3.76	1.30	0.72	2.06	1.88
Sonar	83.52	81.29	82.47	82.53	**85.63**	83.75	84.09	83.16
	84.77	81.45	83.70	81.08	83.33	84.15	82.89	83.44
	5.45	3.00	6.01	6.48	3.25	5.03	5.03	1.74
Balance	80.48	78.26	81.72	82.15	**86.42**	85.23	82.45	81.39
	79.36	80.64	80.70	81.54	**86.27**	84.04	83.80	82.34
	1.28	0.03	0.85	2.11	1.77	0.03	1.46	2.44
Australian	**88.70**	87.35	88.27	87.21	86.00	85.16	87.04	86.75
	88.53	88.24	87.32	85.06	85.62	85.83	87.86	**88.55**
	0.89	0.07	1.62	1.38	2.47	0.40	0.53	2.48
Pima	65.47	63.10	68.03	68.97	**73.28**	**73.28**	70.68	69.84
	68.09	62.29	67.62	71.55	**73.59**	71.03	70.68	70.88
	0.05	0.11	0.29	0.11	0.99	0.65	0.65	3.16
German	**83.64**	77.63	81.82	78.52	83.54	79.41	79.76	82.81
	84.20	75.90	81.82	77.52	82.84	80.19	78.20	79.04
	1.21	1.21	3.55	3.84	2.31	1.37	1.22	1.58
QSAR	88.02	86.22	**89.40**	86.63	88.54	85.85	86.31	87.08
	85.29	85.74	87.65	**89.73**	87.81	87.68	86.39	89.25
	4.35	2.03	3.06	4.80	5.30	6.07	7.14	3.72
Spam	87.02	88.39	**90.66**	87.63	85.91	85.35	87.49	86.31
	87.29	89.85	**91.20**	88.41	87.15	85.58	86.83	84.17
	5.08	7.66	6.55	7.33	4.87	4.04	6.39	5.26
Mushroom	**94.73**	90.80	92.66	91.00	94.19	90.22	93.16	92.54
	94.29	89.35	91.99	92.20	**94.38**	90.22	93.76	91.83
	13.33	24.79	13.52	16.48	7.18	5.00	6.39	12.04
Avg.ACC	84.54	82.25	84.90	83.49	**85.63**	83.37	83.81	83.89
Avg.F_1	84.31	82.30	84.25	83.69	**85.22**	83.65	83.87	83.85
Avg.Time	3.82	4.35	4.18	5.14	3.27	**2.59**	3.43	3.81
Avg.rank(ACC)	3.33	6.00	3.56	5.00	**3.06**	5.83	4.22	5.00
Avg.rank(F_1)	3.78	5.67	4.11	4.67	**3.33**	4.87	5.11	4.44

Table 6. Experimental results on UCI datasets with 10% Gaussian noise.

	RTELM	TELM	TDSVM	FMSVM	Lap-RTELM	Lap-TELM	Lap-TSVM	Lap-RMTELM
	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC
	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1
	Times	Times	Times	Times	Times	Times	Times	Times
Hepat	**88.75**	85.46	88.05	85.29	87.19	80.60	82.96	85.02
	86.23	84.07	86.00	85.53	86.01	81.48	83.63	83.19
	2.77	0.33	3.03	3.42	1.66	0.45	2.27	1.61
Sonar	83.12	79.18	81.31	82.08	**84.70**	81.75	82.51	83.11
	83.12	80.02	82.59	80.20	83.17	**83.52**	82.30	81.48
	5.09	3.66	6.14	6.52	3.34	4.42	5.37	2.48
Balance	80.40	76.10	80.61	81.91	**84.19**	83.78	80.14	80.31
	79.36	75.37	80.70	81.54	**85.11**	82.27	82.37	80.15
	1.42	0.02	1.33	2.62	2.23	0.19	1.82	2.09
Australian	**88.31**	86.25	88.27	86.69	84.10	85.16	85.37	86.73
	86.91	**87.88**	86.25	85.55	84.94	83.60	84.24	85.15
	0.97	0.14	1.14	2.13	2.22	0.36	0.39	2.78
Pima	65.01	61.92	67.44	67.30	70.38	**71.23**	69.46	68.29
	66.43	60.82	67.03	68.42	**72.26**	69.69	69.77	68.52
	1.15	0.11	0.29	1.45	0.99	0.65	0.65	3.16
German	83.00	74.63	80.10	77.61	**83.22**	77.41	76.93	80.65
	83.00	73.90	80.84	77.06	82.09	78.19	76.29	78.13
	1.33	1.20	3.90	4.06	2.00	1.03	1.12	1.55
QSAR	86.17	83.04	87.58	85.00	**88.11**	85.28	84.67	85.24
	84.56	82.92	85.02	**87.70**	87.41	85.14	85.16	87.59
	4.28	2.26	3.55	5.49	4.88	3.16	5.22	3.77
Spam	86.36	85.30	**87.11**	86.23	85.50	83.43	84.17	84.59
	87.00	86.08	86.81	86.86	86.67	83.21	85.44	82.76
	7.73	5.60	6.55	8.57	6.18	4.83	5.00	6.69
Mushroom	**93.31**	88.37	90.12	89.64	92.43	87.72	90.46	90.46
	93.65	87.62	90.84	90.52	**94.10**	88.22	91.17	89.30
	14.16	24.11	15.81	18.23	10.18	7.17	7.48	11.29
Avg.ACC	83.83	80.03	83.40	82.42	**84.42**	81.82	81.85	82.71
Avg.F_1	83.36	79.85	82.90	82.60	**84.64**	81.70	82.26	81.81
Avg.Time	4.32	4.12	4.64	5.83	3.74	**2.47**	3.26	3.94
Avg.rank(ACC)	2.67	6.78	3.56	4.78	**2.56**	5.56	5.72	4.39
Avg.rank(F_1)	3.44	6.56	4.22	4.22	**2.44**	5.11	4.56	5.44

Table 7. Experimental results on UCI datasets with 20% Gaussian noise.

	RTELM	TELM	TDSVM	FMSVM	Lap-RTELM	Lap-TELM	Lap-TSVM	Lap-RMTELM
	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC
	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1
	Times	Times	Times	Times	Times	Times	Times	Times
Hepat	**87.75**	82.88	86.15	83.24	85.19	80.51	79.47	82.15
	85.14	81.40	84.74	84.78	**85.82**	79.49	79.16	81.23
	2.42	0.21	3.22	4.03	2.19	1.14	1.40	3.13
Sonar	82.00	77.18	80.31	80.08	**82.91**	79.75	78.51	81.11
	82.56	76.87	80.92	78.15	82.45	80.03	78.75	80.00
	5.14	2.64	6.27	6.74	3.19	4.20	4.50	3.34
Balance	79.20	72.65	77.37	76.32	**81.82**	81.11	79.18	78.80
	77.34	73.29	78.15	77.32	**83.00**	81.11	80.41	79.33
	1.77	0.50	2.09	3.40	2.35	0.44	1.88	3.80
Australian	87.21	85.82	**87.70**	84.57	83.38	81.82	80.88	83.40
	86.50	85.84	84.39	84.26	84.35	80.56	82.59	81.47
	1.29	0.37	1.74	2.00	2.66	0.34	1.57	2.60
Pima	63.70	59.66	65.58	65.84	69.82	**69.97**	66.70	66.75
	64.48	60.02	65.91	66.33	**70.00**	66.17	65.80	66.83
	1.82	0.04	1.37	1.55	1.89	0.38	0.73	3.64
German	**82.28**	71.37	78.49	75.52	81.47	74.94	72.80	78.67
	81.28	70.78	78.98	74.93	**81.40**	75.74	73.30	77.00
	2.07	1.20	3.58	3.96	2.85	1.55	1.72	1.67
QSAR	84.01	80.23	**85.70**	82.23	85.08	84.81	82.19	82.70
	83.06	81.25	82.71	84.60	**86.12**	83.97	83.47	83.52
	4.04	2.41	3.42	5.51	4.83	3.20	4.58	3.15
Spam	**84.31**	81.89	83.96	82.67	83.68	80.43	81.80	82.17
	86.41	80.00	85.27	**86.78**	85.39	78.13	80.86	79.23
	7.18	5.58	6.56	8.78	6.22	5.07	5.13	6.78
Mushroom	**90.09**	82.40	88.89	83.69	89.54	82.50	84.25	87.19
	91.13	83.60	86.71	85.58	**91.41**	82.59	86.27	85.86
	13.00	15.33	9.12	15.64	10.35	7.20	7.51	10.43
Avg.ACC	82.28	77.12	81.57	79.35	**82.54**	79.54	78.42	80.33
Avg.F_1	81.99	77.01	80.86	80.30	**83.33**	78.64	78.96	79.39
Avg.Time	4.30	3.14	4.15	5.73	4.06	**2.61**	3.22	4.28
Avg.rank(ACC)	**2.44**	6.89	3.22	5.11	**2.44**	5.22	6.33	4.33
Avg.rank(F_1)	3.22	6.67	4.11	4.44	**1.67**	5.44	5.56	4.89

To further validate the effectiveness and robustness of the two proposed algorithms, we conducted experiments in environments with 10% and 20% Gaussian noise, respectively. From the experimental results in Table 6, it can be seen that the ACC and F_1 of all eight algorithms decrease to varying degrees after introducing 10% Gaussian noise. However, RTELM and Lap-RTELM still have the highest ACC

and F_1. For example, on the German dataset, RTELM has the highest F_1, which is 6.1%, 2.16%, and 5.94% higher than TELM, TDSVM, and FMSVM, respectively, while Lap-RTELM has the highest ACC, which is 5.81%, 6.29%, and 2.57% higher than Lap-TELM, Lap-TSVM, and Lap-RMTELM, respectively. Similar situations can also be observed in other datasets. Furthermore, the average results show that Lap-RTELM slightly outperforms RTELM in terms of classification performance. Therefore, we can conclude that the average weight technique and pre-selection technique used in our two learning frameworks can enhance the robustness of the models. This is because both models utilize these two techniques to enhance the influence of central data points while weakening the influence of edge data points. In addition, noise points generally exist near the edges of the class. Therefore, in an environment with 10% Gaussian noise, the two proposed algorithms are effective and exhibit good robustness. Similar conclusions can be drawn from Table 7. In addition, by comparing the Avg.ACC and Avg.F_1 in Table 6 and Table 7, we find that after introducing 20% Gaussian noise, the Avg.ACC of all eight algorithms decreases by 2.26%, 5.13%, 3.33%, 4.14%, 3.09%, 3.83%, 5.39%, and 3.56%, respectively, and the Avg.F_1 decreases by 2.32%, 5.29%, 3.39%, 3.39%, 1.89%, 5.01%, 4.91%, and 4.46%, respectively. Therefore, our proposed RTELM and Lap-RTELM are the most stable.

In summary, our proposed RTELM and Lap-RTELM are effective and achieve the highest ACC and F_1, as well as demonstrate robustness and stability when compared to other algorithms on UCI datasets with different levels of Gaussian noise.

5.3. Robustness to label noise and impulsive noise

To comprehensively validate the "Robust" claim in our title, we extend our experiments beyond Gaussian feature noise to include two additional noise types commonly encountered in real-world applications: label noise and impulsive noise. For label noise, we randomly flip a percentage $\rho \in \{0\%, 10\%, 20\%, 30\%\}$ of training labels ($+1 \leftrightarrow -1$), simulating unreliable annotations or human labeling errors. For impulsive noise, we consider two variants: (i) *salt-and-pepper noise*, where randomly selected samples have their features replaced with values uniformly drawn from $[-10, 10]$, far outside the normalized feature range $[0, 1]$; and (ii) *contaminated Gaussian noise*, where features follow a mixture $(1 - \rho)\mathcal{N}(0, \sigma^2) + \rho\,\text{Cauchy}(0, 1)$, introducing heavy-tailed contamination with infinite variance. We also test combined noise scenarios with both 10% label noise and 10% impulsive noise simultaneously, representing the most challenging real-world conditions where multiple corruption types coexist.

Table 8. Classification accuracy (%) under symmetric label noise.

Dataset	Noise	RTELM	TELM	TDSVM	FMSVM	Lap-RTELM	Lap-TELM	Lap-TSVM	Lap-RMTELM
Australian	0%	88.70	87.35	88.27	87.21	88.53	85.16	85.62	87.86
	10%	87.25	83.41	84.62	83.95	**87.68**	81.33	81.97	84.21
	20%	85.36	79.28	81.03	80.17	**86.12**	77.84	78.25	80.93
	30%	82.94	74.65	76.89	75.42	**83.75**	73.21	73.89	76.54
German	0%	83.64	77.63	81.82	78.52	84.20	75.90	81.82	79.04
	10%	81.90	73.45	77.21	74.83	**82.35**	71.62	76.48	74.91
	20%	79.47	69.28	73.05	70.96	**80.18**	67.34	71.92	70.85
	30%	76.23	64.71	68.44	66.57	**77.42**	62.95	67.13	66.28
Pima	0%	73.28	73.28	70.68	69.84	73.59	71.03	70.68	70.88
	10%	71.45	68.92	66.73	65.81	**71.98**	66.47	66.21	66.95
	20%	69.12	64.37	62.58	61.43	**69.84**	62.18	61.95	62.73
	30%	66.38	59.84	58.21	57.06	**67.21**	57.93	57.48	58.39

As shown in Table 8, all algorithms experience performance degradation as label noise increases. However, RTELM and Lap-RTELM consistently maintain higher accuracy across all noise levels and

datasets. At 30% label noise on the Australian dataset, RTELM achieves 82.94% accuracy compared to 74.65% for TELMła relative improvement of 11.1%. Lap-RTELM performs even better at 83.75%, benefiting from manifold regularization that leverages unlabeled data structure to mitigate label errors.

The superior performance under label noise stems from our average weight and pre-selection techniques: mislabeled samples typically appear as outliers in their assigned class, receiving low average weight values. The pre-selection mechanism then excludes or down-weights these points, preventing them from distorting the hyperplanes. This is fundamentally different from methods that treat all labeled points equally regardless of label reliability.

Table 9. Classification accuracy (%) under salt-and-pepper noise (20% contamination).

Dataset	RTELM	TELM	TDSVM	FMSVM	Lap-RTELM	Lap-TELM	Lap-TSVM	Lap-RMTELM
Australian	86.94	81.23	83.47	82.15	**87.32**	79.84	80.61	83.58
German	81.27	72.18	76.34	73.89	**82.06**	70.45	75.28	74.12
Pima	70.83	66.71	65.28	64.53	**71.45**	64.82	64.17	65.39
Sonar	81.46	74.92	76.85	75.18	**82.73**	74.21	75.46	76.82
Mushroom	91.84	84.36	87.21	85.94	**92.58**	83.97	86.43	87.15

Table 10. Classification accuracy (%) under contaminated Gaussian noise (20% Cauchy).

Dataset	RTELM	TELM	TDSVM	FMSVM	Lap-RTELM	Lap-TELM	Lap-TSVM	Lap-RMTELM
Australian	87.23	82.45	84.18	83.02	**87.91**	80.63	81.47	84.36
German	82.15	73.84	77.52	75.13	**82.89**	71.92	76.35	75.84
Pima	71.46	67.83	66.41	65.78	**72.18**	65.93	65.26	66.57
Sonar	82.37	75.68	77.43	76.05	**83.24**	75.12	76.18	77.43
Mushroom	92.56	85.91	88.34	87.12	**93.41**	85.26	87.58	88.36

Tables 9 and 10 present results under salt-and-pepper noise and contaminated Gaussian noise. Impulsive noise causes more severe degradation than Gaussian noise of the same magnitude, as extreme values can disproportionately influence unrobust estimators. Nevertheless, RTELM and Lap-RTELM maintain their advantage, with Lap-RTELM achieving over 92% accuracy on Mushroom even under 20% Cauchy contamination.

The rank-based weighting mechanism proves particularly effective for impulsive noise. Unlike distance-based methods that can be fooled by extreme values (which inflate scale estimates and can make outliers appear less extreme), ranks are unaffected by the magnitude of outliersłan outlier with value 1000 and one with value 10000 receive the same extreme rank and thus the same low weight. This explains why our methods maintain performance even under heavy-tailed distributions.

Table 11. Classification accuracy (%) Under combined noise (10% label noise + 10% impulsive noise).

Dataset	RTELM	TELM	TDSVM	FMSVM	Lap-RTELM	Lap-TELM	Lap-TSVM	Lap-RMTELM
Australian	85.62	78.94	81.23	79.86	**86.18**	77.32	78.95	81.47
German	79.84	69.73	73.85	71.24	**80.53**	67.98	72.46	72.18
Pima	68.95	63.47	62.18	61.52	**69.73**	61.84	60.95	62.83

Table 11 presents the most challenging scenario: simultaneous label noise and impulsive noise. Under these conditions, the performance gap between our methods and competitors widens further. On the German dataset, Lap-RTELM achieves 80.53% accuracy compared to 67.98% for Lap-TELMła 12.55 percentage point advantage. This demonstrates that our methods are robust not just to individual noise types but to complex, real-world corruptions where multiple noise sources coexist.

Table 12. Experimental results on real datasets.

	RTELM	TELM	TDSVM	FMSVM	Lap-RTELM	Lap-TELM	Lap-TSVM	Lap-RMTELM	CNN
	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC
	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1
MNIST20($\delta = 0$)	99.87	99.39	98.28	99.09	**100.00**	99.87	99.45	98.06	**100.00**
	99.22	98.40	99.27	**100.00**	99.69	99.45	98.38	98.92	99.23
COIL-20($\delta = 0$)	87.07	82.25	85.53	83.28	89.19	84.08	85.74	86.57	**91.25**
	89.14	82.43	84.47	84.18	88.00	83.57	84.30	86.35	88.49
MNIST20($\delta = 0.15$)	99.67	99.01	98.28	99.56	**100.00**	99.71	99.18	98.79	99.94
	100.00	98.11	98.28	99.82	99.77	99.04	97.93	99.00	99.23
COIL-20($\delta = 0.15$)	86.34	80.46	81.38	78.40	**88.72**	78.20	81.18	82.94	88.02
	87.63	79.02	79.81	80.41	86.32	78.39	80.23	83.84	87.33
MNIST20($\delta = 0.3$)	99.20	98.39	98.00	98.91	**99.53**	98.67	98.84	98.00	98.39
	99.44	98.81	98.00	99.18	**99.77**	99.00	98.07	97.86	99.20
COIL-20($\delta = 0.3$)	86.34	80.46	81.38	78.40	**88.72**	78.20	81.18	82.94	87.49
	87.63	79.02	79.81	80.41	86.32	78.39	80.23	83.84	87.49

5.4. Results of image classification

In this subsection, we apply the proposed RTELM and Lap-RTELM to image recognition. Two well-known and publicly available datasets corresponding to typical image classification problems, handwritten digits (MNIST)[‡] and objects (COIL-20)[§], are used to evaluate RTELM and Lap-RTELM with TELM, TDSVM, FMSVM, Lap-TELM, Lap-TSVM, Lap-RMTELM and CNN.

To further validate the robustness of our proposed models, we conducted additional experimental analyses on the MNIST20 and COIL-20 datasets in environments without Gaussian noise, with 15% Gaussian noise, and with 30% Gaussian noise. The experimental results are shown in Table 12. From the table, it can be seen that on the MNIST20 dataset, all algorithms achieve good classification performance, while Lap-RMTELM achieves the highest ACC. With the increase of Gaussian noise, ACC and F_1 of all algorithms decrease to varying degrees, while RTELM and Lap-RMTELM decrease the least. Therefore, we can draw the same conclusion as the previous experiments: our proposed RTELM and Lap-RTELM algorithms have better classification performance, robustness, and stability than the other six algorithms.

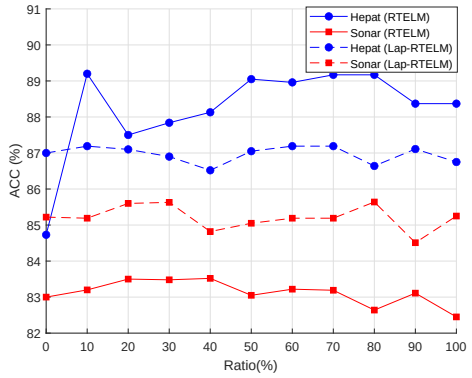
5.5. Parameters analysis

Firstly, we analyzed the impact of parameters c_5 and c_6 on the ACC of the proposed Lap-RTELM model. We fixed the remaining parameters and set $c_5 = c_6$. Then, we conducted experiments on Date1 and Data3 datasets under optimal parameter conditions, without Gaussian noise. The experimental results are shown in Table 13. From the table, it can be observed that as the value of c_5 increases, the ACC of Lap-RTELM also increases gradually. Moreover, the increase in ACC is more pronounced in the Two Moons dataset. This indicates that the manifold regularization term can better utilize the data distribution to improve the model's ACC.

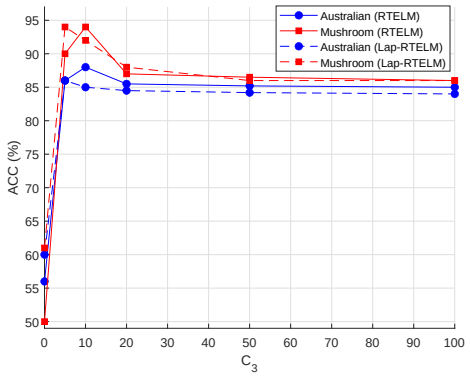
[‡]http://yann.lecun.com/exdb/mnist/.
[§]https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php/.

Table 13. Effect of parameter c_5.

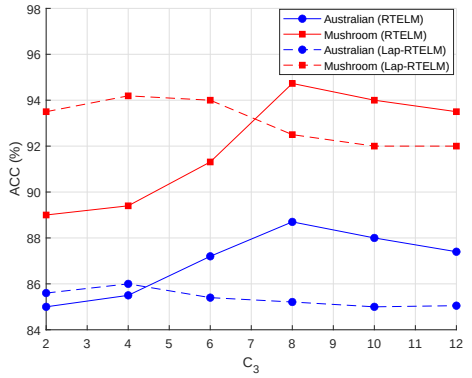
c_5	0.01	0.05	0.1	0.5	1	5
Data1	96.90	97.95	97.21	98.51	98.07	98.07
Data3	98.66	98.01	99.78	99.75	100.00	99.99



(a) Effect of the pre-selection ratios.



(b) Effect of parameter c_3.



(c) Effect of parameter c_3.

Figure 4. Effect of pre-selection technique and average weight technique.

Then, we analyzed the impact of the pre-selection ratio of the improved pre-selection technique on the classification ACC of the proposed model. After sorting the data in the dataset in ascending or descending order, we utilized the formula (3.3) to pre-select the data. This was done to enhance the class centers and weaken the class edges, thereby improving the robustness of the model. We conducted experiments on the Hepat and Sonar datasets in the UCI database without Gaussian noise to determine the optimal pre-selection ratio. The experimental results are shown in a. In Figure 4 (a), the x-axis represents the pre-selection ratio, and the y-axis represents ACC. From the figure, it can be seen that in the proposed RTELM, the ACC generally decreases as the pre-selection ratio increases. When the pre-selection ratio is 100%, the model's ACC is not the highest. On the Hepat dataset, the model already achieves the highest ACC when the pre-selection ratio is 10%. On the Sonar dataset, the model achieves the highest ACC when the pre-selection ratio is 10%. Similarly, in the proposed Lap-RTELM, as the pre-selection ratio increases, the model's ACC also generally decreases. When the pre-selection

ratio is 100%, the model's ACC is not the highest. On the Hepat dataset, the model already achieves the highest ACC when the pre-selection ratio is 10%. On the Sonar dataset, the model achieves the highest ACC when the pre-selection ratio is 30%. Therefore, we believe that the appropriate pre-selection ratio should be controlled within 40%, to fully utilize the data distribution information in the dataset. This also confirms the effectiveness of the proposed pre-selection technique.

Finally, we analyzed the impact of parameters c_3 and c_4 in the proposed model on the ACC. We conducted experiments on the Australian and Mushroom datasets in the UCI database without Gaussian noise. As mentioned earlier, both c_3 and c_4 are positive constants. We first set $c_3 = c_4$, and fixed $c_1 = c_2 = 1$ for RTELM and $c_1 = c_2 = c_5 = c_6 = 1$ for Lap-RTELM. Then we determined the relationship between c_3 and ACC, and the experimental results are shown in Figure 4 (b). From the figure, it can be seen that as c_3 gradually increases, both RTELM and Lap-RTELM first reach their highest ACC and then stabilize almost completely. This indicates that the parameter related to our average weight is insensitive, and the optimal value can be obtained around $c_3 = 10$. Furthermore, we tested the optimal range of c_3. The experimental results are shown in Figure 4 (c). From the figure, it can be seen that for RTELM, the optimal value is achieved at $c_3 = 8$, and for Lap-RTELM, the optimal value is achieved at $c_3 = 4$. Therefore, this confirms the effectiveness of the proposed average weight technique.

5.6. Statistical test analysis

In order to further compare the performance differences between the two algorithms we proposed and the other six algorithms, we first utilized the Friedman test [41]. The null hypothesis of the Friedman test is that all algorithms have the same performance, and the results are shown in Table 14. Based on the experimental results from Table 5 to Table 7, which display the results of the eight algorithms on the UCI datasets in environments without Gaussian noise, 10% Gaussian noise, and 20% Gaussian noise, we utilized formula (5.1) to compute the Friedman parameters.

$$X_F^2 = \frac{12N}{k(k+1)}[\sum_j R_j^2 - \frac{k(k+1)^2}{4}], \quad F_F((k-1),(k-1)(N-1)) = \frac{(N-1)X_F^2}{N(k-1)-X_F^2}, \tag{5.1}$$

where $N = 9$ represents the number of UCI datasets, $k = 8$ represents the number of algorithms, and R_j represents the average rank of the jth method. In addition, $F_F((k-1),(k-1)(N-1))$ follows an F-distribution with $(k-1)$ and $(k-1)(N-1)$ degrees of freedom. After obtaining X_F^2 and F_F^2, when $\alpha = 0.1$, we obtained $F_{0.1}(7, 56) = 1.8269$. From Table 14, we found that $F_F > F_\alpha$ holds for all cases. Therefore, we reject the null hypothesis, indicating that there are significant performance differences among these eight algorithms.

Table 14. Friedman test.

	X_F^2	F_F	F_α	q_α	CD
Result in Table 5	13.3851	2.1582	1.8269	2.450	2.0004
Result in Table 6	24.1155	4.9615	1.8269	2.450	2.0004
Result in Table 7	29.8890	7.2215	1.8269	2.450	2.0004

To further compare the performance differences among these eight algorithms, we utilized the

Table 15. The differences in average rank for each algorithm.

RTELM vs.	TELM	TDSVM	FMSVM	Lap-RTELM	Lap-TELM	Lap-TSVM	Lap-RMTELM
Result in Table 5	**2.67**	0.23	1.67	0.27	**2.50**	0.89	1.67
Result in Table 6	**4.11**	0.89	**2.11**	0.11	**2.89**	**3.05**	1.72
Result in Table 7	**4.45**	0.78	**2.67**	0	**2.78**	**3.89**	1.89
Lap-RTELM vs.	TELM	TDSVM	FMSVM	RTELM	Lap-TELM	Lap-TSVM	Lap-RMTELM
Result in Table 5	**2.94**	0.50	1.94	0.27	**2.77**	1.16	1.94
Result in Table 6	**2.22**	1.00	**2.22**	0.11	**3.00**	**3.16**	1.83
Result in Table 7	**4.45**	0.78	**2.67**	0	**2.78**	**3.89**	1.89

Bonferroni-Dunn test [42], and the results of the post-hoc test are shown in Table 14. When $\alpha = 0.1$, we obtained $q_{0.1}(7, 56) = 2.450$ [41]. We calculated the critical difference (CD) using formula (5.2).

$$CD = q_{\alpha=0.1} \times \sqrt{\frac{k(k+1)}{6N}}. \tag{5.2}$$

Then, we compared the average ranks of the proposed RTELM and Lap-RTELM with the other six algorithms on the UCI datasets under different Gaussian noise environments. The results are shown in Table 15. If the difference in average rank between the two algorithms is greater than the CD value, it indicates a significant difference in performance between them. Note that we have bolded the numbers in the table that have a difference greater than the CD value. From Table 15, we found that when there is no Gaussian noise, our proposed RTELM algorithm shows significantly different performance compared to the TELM and Lap-TELM algorithms. As the level of Gaussian noise increases to 10% and 20%, significant performance differences can be observed between RTELM and TELM, FMSVM, Lap-TELM, and Lap-TSVM. As the noise increases, the overall difference in average rank between RTELM and other models also increases. This suggests that RTELM has better robustness and stability compared to other algorithms. In addition, Lap-RTELM has a similar situation to RTELM, and there is only a small performance difference between the proposed RTELM and Lap-RTELM. It is worth noting that when the Gaussian noise is 20%, the performance of the two proposed algorithms is the same because their average ranks are the same. Therefore, we can conclude from the results of the statistical analysis that the proposed RTELM and Lap-RTELM are not only effective but also distinct from the other six algorithms. This distinction is particularly evident when there is more Gaussian noise, as the difference becomes more significant. This finding also suggests that the proposed models have better robustness and stability.

6. Conclusions

In this paper, we propose two frameworks for binary classification, namely RTELM and Lap-RTELM. These frameworks include two novel techniques: the average weight technique and the improved pre-selection point technique. By utilizing these frameworks, we can effectively leverage the distribution of data points in the dataset to construct classifiers that demonstrate excellent performance. Specifically, we first sort the data points in ascending and descending order, and then calculate the weight and average weight for each data point. The data points that are closer to the

class centers, which have a positive impact on hyperplanes construction, are assigned larger average weight values. On the other hand, the data points that are closer to the class boundaries, which have less influence on hyperplanes construction, are assigned smaller average weight values. Furthermore, after obtaining the sorted data points, we use the pre-selection technique to select only the top-u data points with higher average weight values. This can eliminate or reduce the influence of noise and outliers on the model. In the experimental section, we initially performed numerous experiments on artificial datasets, UCI benchmark datasets, and real datasets under different levels of Gaussian environments, using ACC and F_1 as the criteria. Then, we also perform a detailed analysis of the parameters in the proposed RTELM and Lap-RTELM frameworks. Finally, we conduct statistical detection analysis. The experimental results demonstrate the effectiveness, robustness, and stability of our proposed methods.

In future work, we plan to embed the average weight technique and the improved pre-selection point technique into other types of classifiers, for handling multi-class and large-scale classification tasks. In addition, developing faster methods is also worth considering.

Author contributions

Yanrong Ma and Jun Ma : Writing-original draft, Writing-reviewing & editing, Methodology, Software, Data curation, Supervision, Validation, Project administration, Funding acquisition. Bao Ma: Writing-original draft, Writing-reviewing & editing, Methodology, Software, Data curation, Supervision. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China (No.62366001, No.12361062), in part by the Natural Science Foundation of Ningxia Provincial (No.2024AAC05055, No.2023AAC02053), in part by the Fundamental Research Funds for the Central Universities of North Minzu University (No.2023ZRLG01), and in part by the Postgraduate Innovation Project of North Minzu University (YCX24089).

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Informed consent

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

Conflict of interest

There are no conflicts of interest in this study.

References

1. J. Lou, Y. Jiang, Q. Shen, R. Wang, Z. Li, Probabilistic regularized extreme learning for robust modeling of traffic flow forecasting, *IEEE Trans. Neural Netw. Learn. Syst.*, **34** (2020), 1732–1741. http://doi.org/10.1109/TNNLS.2020.3027822
2. V. V. V. Reddy, S. K. Tiwari, Precise medical diagnosis for brain tumor detection and data sample imbalance analysis using enhanced kernel extreme learning machine model with deep belief network compared to extreme machine learning, *2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, 2022. http://doi.org/10.1109/MACS56771.2022.10022465
3. M. Dogan, I. A. Ozkan, Determination of wheat types using optimized extreme learning machine with metaheuristic algorithms, *Neural Comput. Appl.*, **35** (2023), 12565–12581. http://doi.org/10.1007/s00521-023-08518-9
4. G. B. Huang, Q. Y. Zhu, C. K. Siew, Extreme learning machine: Theory and applications, *Neurocomputing*, **70** (2006), 489–501. http://doi.org/10.1016/j.neucom.2005.12.126
5. G. B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, **42** (2012), 513–529. http://doi.org/10.1109/TSMCB.2011.2168604
6. W. Zong, G. B. Huang, Y. Chen, Weighted extreme learning machine for imbalance learning, *Neurocomputing*, **101** (2013), 229–242. http://doi.org/10.1016/j.neucom.2012.08.010
7. R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, *IEEE Trans. Pattern Anal. Mach. Intell.*, **29** (2007), 905–910. http://doi.org/10.1109/TPAMI.2007.1059
8. Y. Wan, S. Song, G. Huang, S. Li, Twin extreme learning machines for pattern classification, *Neurocomputing*, **260** (2017), 235–244. http://doi.org/10.1016/j.neucom.2017.05.028
9. R. Rastogi, A. Bharti, Least squares twin extreme learning machine for pattern classification, *IInnovations in Infrastructure: Proceedings of ICIIF 2018*, 2019. https://doi.org/10.1007/978-981-13-1966-2_50
10. M. F. A. Hady, F. Schwenker, Semi-supervised learning, *Handbook on Neural Information Processing*, Berlin, Heidelberg: Springer, 2013, 215–239. https://doi.org/10.1007/978-3-642-36657-4_7
11. B. Ma, J. Ma, G. Yu, A Novel Robust Metric Distance Optimization-Driven Manifold Learning Framework for Semi-Supervised Pattern Classification, *Axioms*, **12** (2023), 737. http://doi.org/10.3390/axioms12080737
12. M. Belkin, Problems of learning on manifolds, PhD thesis, University Chicago, 2004.
13. L. Rossi, A. Torsello, E. R. Hancock, Unfolding kernel embeddings of graphs: Enhancing class separation through manifold learning, *Pattern Recognit.*, **48** (2015), 3357–3370. http://doi.org/10.1016/j.patcog.2015.04.026
14. Y. Wang, F. Cao, Y. Yuan, A study on effectiveness of extreme learning machine, *Neurocomputing*, **74** (2011), 2483–2490. http://doi.org/10.1016/j.neucom.2010.11.022

15. F. Cao, B. Liu, D. S. Park, Image classification based on effective extreme learning machine, *Neurocomputing*, **102** (2013), 90–97. http://doi.org/10.1016/j.neucom.2012.02.054

16. L. Li, D. Liu, J. Ouyang, A new regularization classification method based on extreme learning machine in network data, *J. Inf. Comput. Sci.*, **9** (2012), 3351–3363.

17. J. Liu, Y. Chen, M. Liu, Z. Zhao, SELM: Semi-supervised ELM with application in sparse calibrated location estimation, *Neurocomputing*, **74** (2011), 2566–2572. http://doi.org/10.1016/j.neucom.2010.12.036

18. B. Liu, S. X. Xia, F. R. Meng, Y. Zhou, Manifold regularized extreme learning machine, *Neural Comput. Appl.*, **27** (2016), 255–269. http://doi.org/10.1007/s00521-015-1857-9

19. S. Li, S. Song, Y. Wan, Laplacian twin extreme learning machine for semi-supervised classification, *Neurocomputing*, **321** (2018), 17–27. http://doi.org/10.1016/j.neucom.2018.02.016

20. J. Wang, S. Lu, S. H. Wang, Y. D. Zhang, A review on extreme learning machine, *Multimedia Tools Appl.*, **81** (2022), 41611–41660. http://doi.org/10.1007/s11042-022-13166-7

21. G. B. Huang, L. Chen, C. K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, *IEEE Trans. Neural Netw.*, **17** (2006), 879–892. http://doi.org/10.1109/TNN.2006.875977

22. G. B. Huang, L. Chen, Convex incremental extreme learning machine, *Neurocomputing*, **70** (2007), 3056–3062. http://doi.org/10.1016/j.neucom.2007.02.009

23. Z. Xu, M. Yao, Z. Wu, W. Dai, Incremental regularized extreme learning machine and it's enhancement, *Neurocomputing*, **174** (2016), 134–142. http://doi.org/10.1016/j.neucom.2015.03.112

24. C. N. Li, Y. H. Shao, N. Y. Deng, Robust L_1-norm non-parallel proximal support vector machine, *Optimization*, **65** (2016), 169–183. http://doi.org/10.1080/02331934.2014.994625

25. N. Kwak, Principal component analysis based on L_1-norm maximization, *IEEE Trans. Pattern Anal. Mach. Intell.*, **30** (2008), 1672–1680. http://doi.org/10.1109/TPAMI.2008.78

26. C. Yuan, L. Yang, P. Sun, Correntropy-based metric for robust twin support vector machine, *Inf. Sci.*, **545** (2021), 82–101. http://doi.org/10.1016/j.ins.2020.10.029

27. J. Shen, J. Ma, Sparse twin extreme learning machine with epsilon-insensitive zone pinball loss, *IEEE Access*, **7** (2019), 112067–112078. http://doi.org/10.1109/ACCESS.2019.2933581

28. Y. Yang, Z. Xue, J. Ma, X. Chang, Robust projection twin extreme learning machines with capped L_1-norm distance metric, *Neurocomputing*, **517** (2023), 229–242. http://doi.org/10.1016/j.neucom.2022.10.003

29. J. Ma, L. Yang, Robust supervised and semi-supervised twin extreme learning machines for pattern classification, *Signal Proc*, **180** (2021), 107861. http://doi.org/10.1016/j.sigpro.2020.107861

30. K. Zhang, M. Luo, Outlier-robust extreme learning machine for regression problems, *Neurocomputing*, **151** (2015), 1519–1527. http://doi.org/10.1016/j.neucom.2014.07.075

31. J. Ma, Capped L_1-norm distance metric-based fast robust twin extreme learning machine, *Appl. Intell.*, **50** (2020), 3775–3787. http://doi.org/10.1007/s10489-020-01825-9

32. Y. Jiang, G. Yu, J. Ma, Distance Metric Optimization-Driven Neural Network Learning Framework for Pattern Classification, *Axioms*, **12** (2023), 765. http://doi.org/10.3390/axioms12080765

33. H. N. Sheng, Z. Y. Wang, H. C. So, Robust rank-one matrix completion via explicit regularizer, *IEEE Trans. Neural Netw. Learn. Syst.*, **36** (2025), 19092–19105. http://doi.org/10.1109/TNNLS.2025.3571594

34. H. N. Sheng, Z. Y. Wang, Z. Liu, H. C. So, Hybrid ordinary-Welsch function-based robust matrix completion for MIMO radar, *IEEE Trans. Aerosp. Electron. Syst.*, **61** (2025), 3950–3962. http://doi.org/10.1109/TAES.2024.3456789

35. Z. Wang, H. C. So, A. M. Zoubir, Robust Low-Rank Matrix recovery via hybrid Ordinary-Welsch function, *IEEE Trans. Signal Proc.*, **71** (2023), 2548–2563. http://doi.org/10.1109/TSP.2023.3298765

36. Z. Wang, X. P. Li, H. C. So, Robust matrix completion based on factorization and Truncated-Quadratic Loss function, *IEEE Trans. Circuits Syst. Video Technol.*, **33** (2023), 1521–1534. http://doi.org/10.1109/TCSVT.2022.3218765

37. J. Xu, H. Wang, L. Zhang, S. Wen, Robust twin depth support vector machine based on average depth, *Knowl.-Based Syst.*, **274** (2023), 110627. http://doi.org/10.1016/j.knosys.2023.110627

38. X. Yan, H. Zhu, A novel robust support vector machine classifier with feature mapping, *Knowl.-Based Syst.*, **257** (2022), 109928. http://doi.org/10.1016/j.knosys.2022.109928

39. Z. Qi, Y. Tian, Y. Shi, Laplacian twin support vector machine for semi-supervised classification, *Neural Netw.*, **35** (2012), 46–53. http://doi.org/10.1016/j.neunet.2012.07.011

40. T. Fawcett, An introduction to ROC analysis, *Pattern Recognit. Lett.*, **27**(8) (2006), 861–874. http://doi.org/10.1016/j.patrec.2005.10.010

41. J. Demšar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.*, **7** (2006), 1–30.

42. A. Benavoli, G. Corani, F. Mangili, Should we really use post-hoc tests based on mean-ranks? *J. Mach. Learn. Res.*, **17** (2016), 152–161. http://doi.org/10.5555/2946645.2946647

AIMS Press